SHARED CONTROL OF HYDRAULIC MANIPULATORS TO DECREASE CYCLE TIME

A Thesis
Presented to
The Academic Faculty

by

Aaron R. Enes

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Mechanical Engineering

Georgia Institute of Technology
December 2010

SHARED CONTROL OF HYDRAULIC MANIPULATORS TO DECREASE CYCLE TIME

Approved by:

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

Dr. Ayanna Howard
School of Electrical and Computer
Engineering
Georgia Institute of Technology

Dr. Kok-Meng Lee
George W. Woodruff School of
Mechanical Engineering
Georgia Institute of Technology

Dr. Jun Ueda
George W. Woodruff School of
Mechanical Engineering
Georgia Institute of Technology

Dr. Magnus Egerstedt
School of Electrical and Computer
Engineering
Georgia Institute of Technology

Date Approved: 23 August 2010

For Carson

and Dorina,

and all our adventures.

ACKNOWLEDGEMENTS

It is with delight that I thank the many people who made this thesis possible.

It is difficult to overstate my gratitude to my advisor, Dr. Wayne Book. His energy, his enthusiasm, and his innate intuition for System Dynamics and Controls has been of tremendous value during graduate school. I am inspired by his passion and creativity.

I am also grateful for the guidance, insights, and feedback provided by my Thesis committee, Dr. Kok-Meng Lee, Dr. Magnus Egerstedt, Dr. Ayanna Howard, and Dr. Jun Ueda.

This thesis simply would not have even begun if not for the generous financial support of Mr. Agustin Ramirez, Chairman and CEO of HUSCO, Int'l, and Mr. Dwight Stephenson, Vice President of Engineering at HUSCO. Grateful acknowledgment of additional funding is given to the National Science Foundation Center for Compact and Efficient Fluid Power (Award EEC-0540834) for funding this research and to the ARCS Foundation, the Woodruff Fellowship fund, and the Presidential Fellowship fund for providing me additional support.

While shaping the ideas of this research, I have had the pleasure of calling upon several experts in mobile hydraulics, particularly Mr. Dwight Stephenson, Mr. Joe Pfaff, and Mr. Peter Maloney of HUSCO; and Dr. Matthew Kontz and Dr. Patrick Opdenbosch of Caterpillar. I especially thank all of the engineers and technicians at HUSCO for collecting and sharing a wealth of excavation data. Their time and insights are appreciated. As part of the Center for Compact and Efficiency Fluid Power, I have been privileged to enjoy innumerable interactions on a professional and social level with the most brilliant minds in the Fluid Power industry. I particularly acknowledge the Engineers, Managers, and Friends at HUSCO, Caterpillar, Poclain, Bobcat, Enfield Technologies, Parker Hannifan Corp., and John Deere.

A special word of thanks is owed to Mr. James Huggins, Research Engineer of the IMDL and a true Master of All Trades.

I am indebted to my friends and colleagues in the IMDL who knowingly or unknowingly served as sounding boards while I carved this research, particularly Longke Wang, Mark Elton, Ryder Winck, Brian Post, Heather Humphreys, Matt Kontz, and Patrick Opdenbosch. I also acknowledge the rewarding discussions with Joshua Schultz, Shaohui Foong, and Farhod Farzbod.

A truly special group of life-long friends helped make this journey at Georgia Tech meaningful, fun, and memorable for my family and me: Dr. Adam Cardi, Mr. Paul George, Dr. Stephanie George, Dr. Mario Romero, Dr. Natalia Landazuri, Dr. Allison Dennis, Dr. Patrick Opdenbosch, Mrs. Claudia Yepes, Dr. Jay Cho, and Mrs. Jenny Han. This group provided limitless emotional support, camaraderie, and entertainment.

I wish to particularly thank four exceptionally remarkable teachers and mentors who built the ground which shaped my early personal and academic growth as a youth: Ms. Paula Mikkelson, Mr. Edwin Fell, Ms. Shannon Hummer, and Dr. Michael Kipp. In a very formative period, these had a truly profound impact which cannot be overstated. I also owe gratitude to three key undergraduate professors, Dr. Dan Lawrence, Dr. Bill Packard, and Dr. Gary Ganske, who were equally relentless in their academic challenges and their passion to teaching.

To my parents, who raised me, supported me, taught me, and loved me: Thank you.

Finally, and most importantly, I thank Dorina and Carson—my beautiful wife and son—who have provided an unwavering emotional foundation, an enumerable quantity of joy, a limitless supply of love, and not a single dull moment. To them I dedicate this thesis.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

This thesis presents a technique termed Blended Shared Control, whereby a human operator's commands are merged with the commands of an electronic agent in real time to control a manipulator. A four degree-of-freedom hydraulic excavator is used as an application example, and two types of models are presented: a fully dynamic model incorporating the actuator and linkage systems suitable for high-fidelity user studies, and a reduced-order velocity-constrained kinematic model amenable for real-time optimization. Intended operator tasks are estimated with a recursive algorithm; the task is optimized in real time; and a command perturbation is computed which, when summed with the operator command, results in a lower task completion time. Experimental results compare Blended Shared Control to other types of controllers including manual control and haptic feedback. Trials indicate that Blended Shared Control decreases task completion time when compared to manual operation.

CHAPTER I

THESIS PREVIEW

Earthmoving machines have many interesting challenges germane to robotics, and their solutions have potential to positively impact a very large industry. Chapter 2 gives background into the application domain of this research, followed in Chapter 3 by a review of the general methods applicable to the control of excavators. The outcome of this research is a shared control architecture termed Blended Shared Control, which is introduced in the end of Chapter 3. The architecture allows two agents, in this case a human and a computer, to simultaneously influence the machine response.

The primary hypothesis is that certain tasks can be completed with lower cost when computer assistance is applied through shared control. The literature review in Chapter 4 places this architecture in the context of other shared control research. To demonstrate the broader applicability of the framework and to whet the reader's appetite, a straight-forward application example is given in Chapter 5.

The Blended Shared Control paradigm perturbs the operator input based on the estimated operator task, discussed in Chapter 7, and the optimal way of completing this tasks. This optimization is presented in Chapter 9. The computation of Blended Shared Control perturbations is the subject of Chapter 10.

The paradigm was tested in different situations for different operators, and performance was evaluated from analytical and subjective points of view. Experimental results indicated a measurable effect on the task time, and are detailed in Chapter 11. Finally, Chapter 12 will summarize the conclusions, contributions, and directions of future work which have grown from this research.

CHAPTER II

INTRODUCTION

The research objective of this work is to understand, calculate, and measure the increase in operator productivity enabled by a novel shared control algorithm applied to point to point motions of flow constrained hydraulic manipulators.

Despite the increased capabilities of autonomous control and with the exception of a few research prototypes, most machines that interact with the environment—from hydraulic excavators to mobile wheelchairs—are manually controlled by the human operator. With manual control, the operator assumes responsibility for providing control inputs that cause the machine to complete a given task at maximum utility, for example in minimum time or with the least energy consumption. However, the optimal control solutions of even very simple nonlinear systems may be non-intuitive or otherwise beyond the capabilities of a human operator responsible for providing the control input. Consequently, manually controlled machines often are operated suboptimally. In an effort to draw the operator nearer to some degree of optimality, an Electronic Agent may be given authority to share control with the operator.

Decreases to a task's time or energy cost are historically achieved through operator training or by modifying existing hardware components or subsystems. In contrast, the approach discussed here relies on few, if any, changes to a system's hardware. Machine performance will be improved by modifying only the operator commands. The work therefore complements conventional research in the areas of component design and control.

Within a shared control (SC) paradigm, both the human and machine have authority to influence end-effector motions. In contrast with autonomous systems, SC keeps the operator continually *in the loop*, meaning an operator input immediately elicits changes in actuator motion. The operator is continually in a position to supervise the system and has the authority to effect change in the manipulator response at any time. The operator is not

in full control of the actuator motion, as the Blended SC architecture is exemplified by a controller having authority to modify the original command, thus control is shared with the operator. The command perturbation is a function of estimates of the intended task and a calculated time-optimal input for completing the task. The operator *at all times* maintains a degree of authority in the control loop.

Blended SC requires three key components: First, the operator's intended task should be deduced based on the operator's original command; ideally this inference will be automatic, computed from already present signals, and will require a minimal amount of operator effort. Second, a closer-to-optimal trajectory for completing the intended task must be determined. This relies on suitable system models and derived necessary conditions for optimally completing the task. Third is to develop an understanding of the role of human-machine interaction in the context of the particular application, specifically to understand the desirable (and undesirable) performance attributes of the shared controller.

Through the literature in this thesis, it will become clear that current techniques and scientific understanding are inadequate for these three problems.

2.1 Context, Impact, and Relevance to the Earthmoving Industry

A hydraulic excavator is used as an application example of Blended Shared Control. Several forces motivate advancements in the control of earthmoving machines including safety; a shortage of skilled workers; and market demands for increased performance, lower operating cost, and higher quality work [1, 2]. Further, a fleet of excavators consumes a substantial amount of fuel annually; thus, large industry-wide savings can be realized even if a solution yields only modest increases in per-cycle efficiency. Unfortunately, the heavy equipment industry is risk adverse. There exists a strong and often unseen barrier against the adoption of otherwise promising technologies. Any technological advancement that requires a substantial departure from conventional excavator system architectures—particularly within the actuator subsystems—will rightfully face substantial scrutiny and resistance in adoption.

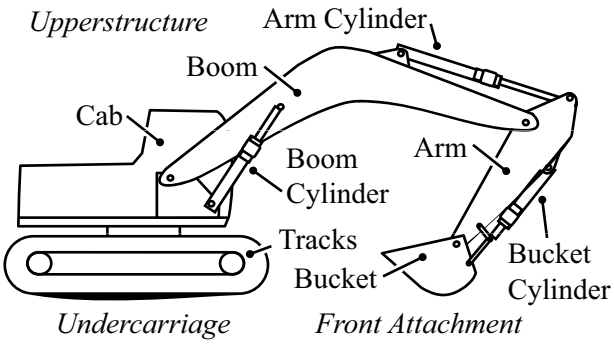


Figure 2.1: The major components of a tracked excavator

2.2 Target Application Domain

An excavator is a type of *earthmoving machine* commonly used in construction for digging, drilling, pushing, or hauling. Several books provide a general introduction to excavation topics, including overviews of the machines, methods, and history of earthmoving [3, 4]. The excavator is essentially a serial link manipulator with four degrees of freedom (DOFs):[1] *swing*, *boom*, *arm*, and *bucket* (the bucket may also be called the *end-effector*) as in Figure 2.1. In reality, the linkage and actuator systems are attached to a mobile, tracked base, but here the base is assumed locally fixed. Figure 2.2 shows machines that are kinematically similar to the generic 4-DOF excavator considered here including log feller-bunchers, tractor-mounted backhoes, and mini-excavators. Hydraulic industrial manipulators are also relevant to this work.

The linkage system is actuated by a hydraulic *actuator system* often comprised of cylinders, conduits, controlled orifices, pumps, accumulators, and a prime mover. Typically, a human operator interacts with a control system that drives the actuator system which in turn propels the linkage system. The operator sits within the excavator cab (Figure 2.3) and clenches two joysticks, each able to pivot in two directions; the joystick angular displacement (e.g., fore/aft or left/right) maps to the commanded velocity of the corresponding hydraulic actuator.

The task domain considered in this work includes trenching, bulk digging, or other

[1]The DOFs are often referred to as *functions*. Usage examples include "...the boom *function* is ...", or "...each of the excavator *functions* are ..."

Figure 2.2: A group of machines kinematically similar to excavators. Clockwise from top left: compact excavator, construction excavator, production excavator, tractor-loader backhoe, log feller buncher. Shown in approximately equal scale. *(Images courtesy Deere & Co.)*

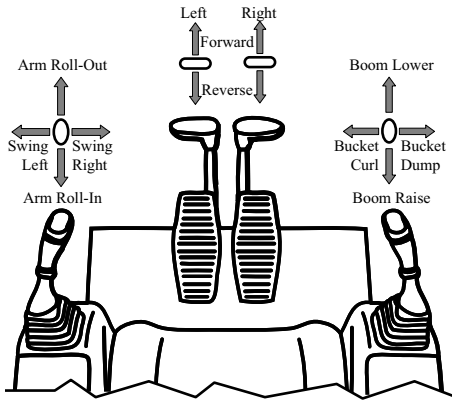


Figure 2.3: Operator controls in a typical excavator cockpit consist of levers that are displaced to command the rate of extension or retraction of the corresponding actuator

operations requiring simultaneous actuation of all functions. These common excavation tasks are generally repetitive; the cyclical nature will be exploited to aide in predicting the intended operator task. While the emphasis is on systems with hydraulic actuators designed for earthmoving, effort will be made to maintain relevancy of this thesis to industrial manipulators in general.

CHAPTER III

LITERATURE REVIEW RELATED TO CONTROL OF EXCAVATORS

This chapter presents a discussion on methods employed by industrial and academic researchers for increasing the performance of hydraulic multi-DOF manipulators. A primary focus will be on the applications to excavator control originating from literature within the domains of system dynamics and controls.

Researchers from a broad range of engineering disciplines have invested in projects to improve overall excavator performance. Their scope may focus on individual components, such as pumps [5, 6] and valves [7, 8], or may include entire systems or development of completely new hydraulic circuit topologies. For example, independent metering control promises improved efficiency by enabling a number of energy-efficient and high-speed operating modes [9–12]. Topologies including displacement controlled actuators reduce energy consumption by removing throttling valves altogether [13, 14]. Other hydraulic circuit layouts increase efficiency or actuator speed by *regeneration* of pressurized oil, such as redirecting high pressure hydraulic flow to other working functions or storing the flow in a hydraulic accumulator [15–19].

Real time optimal trajectory planning of hydraulic manipulators is difficult because of the high dimensionality of the machine configuration space, the nonlinearity of the rigid body dynamics, and the coupling between the degrees of freedom. The actuator system is typically a hybrid dynamic system having discrete operating modes[1] and coupled nonlinear dynamics. There is a tremendous void in works that contribute to trajectory planning or optimization, without abstracting away the most salient attributes of the response stemming from pressure, flow, and power limits.

[1]The flow path from the system pump to the actuator depends on the valve operating mode; transitions between modes may induce high-order effects including linkage oscillations which further complicate the analysis [12].

3.1 Developing Models of Excavators

Many computational tools exist for modeling multi-domain systems such as excavators [20–22]. The models are used for many tasks: to formulate and simulate actuator control laws, to model machine performance for design purposes, to determine trajectories that optimize a cost function, and to predict machine response during online operation. Basic models may only involve kinematics that relate joint angles (or cylinder extensions) to the position and orientation of the excavator bucket [23, 24]. Higher order models include the dynamics of the excavator system, and may include a combination of linkage dynamics, actuator dynamics, and dynamics of the bucket-soil interaction. For example, Koivo described an oft-cited and thorough excavator model [25], where models of both the linkage dynamics and digging reaction forces are developed. Zweire also discussed development and simulation of a similar dynamic model of a mobile front-end loader linkage and hydraulic control system [26, 27].

Owing to their complexity, simulation of the excavator dynamic models is computationally intensive and not practical for solving optimal control problems such as planning optimal bucket trajectories. A common simplification is to ignore the dynamics of the actuator system (engine, pumps, valves, conduits, etc.) by assuming the hydraulic cylinders to be simple force sources. This assumption allows classical manipulator control techniques to be employed. Unfortunately, the simplified models are somtimes applied to systems and scenarios for which they were not intended. For example, it is questionable to use these models to minimize the energy or time cost of a certain task, as the majority of large-scale excavators normally run in a power- and flow-limited operating condition not well described by these simplified models.

Krishna proposed the use of memory-based learning (MBL) to simplify the model of an excavator's hydraulic actuator system without completely eliminating it from the excavator model [28, 29]. MBL can efficiently determine the effects of the complex flow interactions which couple the functions of hydraulic actuator systems. The success of this technique depends on the accuracy of the model and the quality of the learned mapping (which may have no guarantee of convergence to the true mapping). The efficiency of MBL comes at the expense of abstracting any physical interpretation, and thus may not yield as much insight

into the problem as other methods do.

3.1.1 Operator Model

The human operator is present in all but fully autonomous systems. An ideal analysis of a human-in-the-loop control algorithm would include precise models of the operator's effects, so that the ultimate closed-loop dynamics of the human/machine/manipulator system could be simulated and predicted . Obviously, there does not exist human behavior and performance models of sufficiency fidelity to perform well in an arbitrary excavation scenario.[2] To model and understand excavator systems, a few authors include (to a greater or lesser extent) the operator, environment, and working task in addition to the machine itself. For instance, Filla developed a rule-based operator model for bucket filling of a wheel loader, where the task and the operator model are independent of the machine itself [30, 31]. Bernold proposed methods to quantify the skill of an operator which may be useful for comparing the effectiveness of competing control designs given variability in operator skill [32].

Real-time, human-in-the-loop simulations avoid the difficult requirement of accurately modeling human behavior, at the expense of less repeatability. More powerful computers have enabled much work in this area [33, 34], and are becoming very useful in the validation and justification of control designs. The experimental evaluations presented in Chapter 11 make use of a virtual reality human-in-the-loop excavator simulation testbed [35, 36].

3.2 Manual Control of Excavators

The vast majority of earthmoving machines are manually controlled by a human operator, with a direct one-to-one mapping between the user and the actuator commands. Manual control has been the norm since the earliest commercial excavator disclosed in 1881 [37]. Recently there are indications of increased commercial interest in severing the conventional human/excavator control link. However, the pace of adoption is considerably slower than

[2] If such high-fidelity models did exist, then many original problems in autonomous control of manipulators would be solved.

other industries, in contrast to the rapid adoption of modern fly-by-wire aircraft and drive-by-wire automobiles.

3.2.1 Efficient Manual Control

When limited to manual control, an excavator's efficiency and productivity are increased through operator training and hardware improvements. There are certain technologies being developed to improve the energy efficiency of the actuator systems of earthmoving machines. For example, independent metering control (IMC) promises improved efficiency by enabling more operating modes [9–12, 15, 38–43]. Other systems increase overall efficiency by re-generation of pressurized oil, or re-directing high pressure hydraulic flow to other working functions or storing the flow in an accumulator [15–19, 44–46]. Non-conventional hydraulic system topologies have been proposed, including the complete removal of energy-wasting control valves in favor of pump-displacement controlled actuators [13, 14].

3.2.2 Teleoperated Systems

Teleoperated systems have been heavily researched [47–52], even in the context of hydraulic excavation. Often the goal is to accurately track the operator command, so teleoperation includes manual control from a distance. Teleoperation may be augmented with extra layers of control, including coordinated control of the bucket [53–55]. Cooperative control schemes have been implemented in specialized cases, such as when a computer assists the operator in maintaining a specified grade when digging [56].

3.3 Automatic Control of Excavators

There are cases from as early as the 1960s where the direct human/excavator command link was broken. For example, Audemar suggested a cam-based system for controlling hydraulic machines in which the operator controlled the phasing of a series of cams, but it was the logic encoded in the cams which actually effected the final machine response [57]. Chytil gave an early mention of the possibility of autonomous computer-based excavation and recognizes that removing the operator from the control loop could have potential benefits including cost savings and more precise operation [58]. The following two sections discuss

two primary subsystems of an automatic earthmoving system: the motion planner, and the trajectory follower.

3.3.1 Motion Planning for Multi-DOF Manipulators

Motion planning implies there is some objective that the motion must fulfill, for example minimizing time or energy cost, avoiding obstacles, or maximizing the force exerted by the end effector. Common approaches include human-guided motion planning, automatic path planning, trajectory tracking, and fully autonomous excavators.

3.3.1.1 Human-Guided Motion Planning

In some applications, the manipulator motion is programmed directly by a human operator and replayed autonomously by the manipulator. In the case of excavation, researchers have studied systems that record the actuator motions during a portion of a dig cycle and then play them back during later cycles [59, 60]. These include systems capable of unloading a bucket of soil and/or returning to the trench automatically. Aside from the teaching phase, the human operator is removed from the control loop during certain phases of operation. Research into human psychology suggests that removing the operator from the control loop, even for small portions of a cycle, may be ill-advised because the operator'a overall attentiveness may drop as a result of the interruptions [61, 62].

3.3.1.2 Automatic Trajectory Generation

Trajectory generation and path-following are well-studied problems in robotics, but the control of hydraulic robots like excavators presents some interesting challenges. For one, the large, variable payload is usually a large fraction of the manipulator mass. Also, there are several simplifying assumptions often used for optimizing electrical industrial robotics, such as minimizing expelled energy by minimizing the time integral of net actuator torques, which is the basis behind most minimum energy trajectory planning techniques including methods utilizing dynamic programming [63, 64] and brute-force sequential optimization [65]. Unfortunately, the technique of minimization of actuator torques may not apply to hydraulic actuator systems, due to flow limiting, severe power limiting, multiple operating modes,

and having a substantial amount of energy consumed by the control system.[3]

Some have suggested that expert knowledge can be captured and used to plan excavation motions. Rowe introduced an approach that uses scripts, or templates, for planning and executing motions of an excavator in real time [67–70]. This method enables online local optimization of machine operation yet no discussion of this approach is given, though some vague experimental results are provided that demonstrate the script functionality on an full size excavator. The internal and external script parameters rely on a preprogrammed, expert system [68]. Shi and Lever also leveraged human knowledge to construct a finite state machine and define fuzzy logic rules for autonomous excavator control [71, 72]. The use of human expert knowledge in excavator control has interesting implications: if optimal solutions are to be discovered, then one must assert that the human behavior on which the method is based is itself close to optimal. The author of this thesis espouses that the definition of a trajectory must be based on sound scientific criteria, and should not solely rely on imitating what an operator does.

3.3.1.3 Optimizing the Digging Phase

Automation of digging involves more than just tracking a reference path, because of the strong bucket-soil interaction [73]. Researchers have presented techniques to decrease the energy consumed in excavating a certain volume of material, based on research into soil mechanics, excavator dynamics, soil/bucket interaction forces [74]. Hemami presented a systematic way of deriving, though not in real-time, the optimal cutting edge trajectory for a front-end loader [75]. Like most digging optimization techniques, a model of the digging forces must be developed first, and evaluation of these forces requires parameters that are soil dependent. In one technique for energy-optimal soil removal, the path and bucket rake angle follow the slip lines generated by the bucket during the previous shoveling stage [76]. While detecting and tracking these paths is often intuitive for very experienced operators,

[3]It is therefore surprising that so many researchers searching for energy-optimal trajectories still make this assumption and neglect the actuator system. For example, Ma [66] reported a real-time means of calculating the near energy-optimal trajectory of a robot by considering only the motion of the center of gravity. While computationally attractive, the method does not immediately ensure that the trajectory will even be reachable by the actuation system.

teaching novices or an electronic machine controller to perform the same is difficult because the stochastic nature of soil means the same sets of forces will generate different tool paths at different times. Bernold used robust force/torque sensors with impedance control to automate the digging process while optimizing the energy consumption per excavated volume, using a simple mass-spring model for soil-tool interaction modeling [77, 78]. Bernold introduced a pattern-matching technique for deducing soil properties from a resulting tool path under impedance control. Tan discussed an online numerical-based method for estimating soil parameters, with application to enabling accurate bucket path tracking under impedance control [79].

To effectively track a prescribed path while the bucket is engaged with the soil, the complexities arising during digging—such as the non-unique relationship between actuator forces and forces acting on the bucket—must be reconciled [80]. Vaha and Bodur used cognitive force control for following a preplanned trajectory, with the ability to change path online if the tool unexpectedly encounters too much resistance [81, 82].

3.3.2 Trajectory Following

Once the path is specified, a path tracking controller computes appropriate actuator commands to execute the motion. Generation of commands to move the bucket along the optimal path requires knowledge of the dynamic characteristics for each axis of movement, particularly the velocity and acceleration limits. Complications arising from the actuator system include high lag times and couplings associated with multiple actuators sharing a single pump. Some algorithms are able to track a reference path while simultaneously optimizing some cost, such as time or energy, while other methods are more concerned with very accurate tracking. Generally if the reference trajectory is not too aggressive, for example if it has been jerk-limited, then good tracking can be realized [83]. The path is often specified in the workspace and must be transformed to joint space or to cylinder space before tracking. Many practical methods suitable for practical implementation exist, with most efforts focusing on computationally efficient means of computing the inverse kinematics—for instance, Makkonen presented a computationally efficient way to compute

the joint angles of a six degree-of-freedom excavator using matrices of target points and simple terrain data [84].

Kahn reported an early solution to the time-optimal control problem for serial link manipulators [85]. Specifying the path essentially collapses the problem to a single-dimension optimization problem. This approach linearizes the dynamics about an optimal nominal trajectory, and uses a closed loop control about this nominal to achieve accurate tracking. Budny used non-linear programming to calculate the joint torques required to position an excavator bucket along a specified path in minimum time [86]. This method satisfies joint torque constraints but does not address the actuator couplings and flow-limited problem common to large hydraulic equipment: it is often not the torques that are constrained but the combined velocities of the actuators. Other researchers in fluid power [87–92] have considered various constraints on actuator torque, velocity, and acceleration, but often give no indication of applicability to hydro-mechanical systems with tightly coupled DOFs and multiple operating modes.

3.3.2.1 Sensing

Most algorithms assume feedback of the robot configuration to achieve very accurate tracking performance. However, few sensors are robust enough for typical excavation environments thereby limiting the wide-scale acceptance of controllers that rely on these measures. Some authors have presented decent path-tracking techniques based on accurate open-loop knowledge of the control valve system and do not rely on position feedback [93]. One common problem with estimating the manipulator state is that error may compound to an appreciable extent for long motions. The formulation developed for the research in this thesis uses relatively short-duration motion primitives to describe the motion and may prevent large errors in position estimates from compounding.

3.3.2.2 Transitions Between Rigid Contact with Soil

A challenging aspect of trajectory tracking is maintaining stability when the bucket transitions from free space to soil contact. Some schemes use hybrid position and force control, switching between modes when contact occurs. Richardson-Little used a position controller

to track in free space, with a compliance controller that modifies the reference trajectory—based on feedback from a model of soil-bucket interaction—to maintain an accurate path when resistance is encountered [94]. Others have used impedance control as a unified approach for both constrained and unconstrained motion [52, 55, 95]. Impedance control may address the issue of stable, accurate tracking yet it does not address any time- or energy-optimality issues. A seminal series of papers on impedance control is given by Hogan [96–98].

In large hydraulic excavators there is some latency between the instant a command is established and the instant that the actuator begins to move. This latency is due in part to the dynamics associated with opening a control valve, building pressure at a system pump, and accelerating a large inertia. Often this latency is on the order of 0.5 seconds and may account for an appreciable loss of productivity. To make up for this latency, at least one approach has been disclosed that anticipates future movement of some joints based on the actuation of another joint and access to a pre-programmed task time line [99].

3.4 Fully Autonomous Excavation Systems

Many researchers adopt an *all-or-nothing* approach to automation of excavation, as cataloged in several comprehensive surveys of autonomous excavation [100–103]. The previous sections have mostly presented piecemeal components required for automatically executing or optimizing some portion of the dig cycle. This section gives a brief discussion of fully integrated, autonomous excavation systems.

Likely the first autonomous excavator was built by the manufacturer Orenstein & Koppel and showcased at the 1983 BAUMA trade show in Germany. However, little data is available about this system.

The University of Wales developed an autonomous excavator called LUCIE [104, 105]. This system used a real-time artificial intelligence developed from extrapolations of observed behaviors of many expert operators. The system produced very high quality rectangular trenches, but the method is not easily adapted to perform general digging tasks, or for that matter, non-digging tasks such as craning or tamping.

Carnegie Mellon University developed an autonomous excavation system capable of

planning and executing digging tasks. The machine automated a complete truck loading task. The excavators control software decides where to dig in the soil, where to dump in the truck, and how to quickly move between these points while detecting and stopping for obstacles. Reportedly, multi-hour tests indicate the excavator loads trucks as fast as human operators [106, 106–111].

3.5 Summary

The excavation industry has a long, storied past. By necessity, the industry is generally risk-adverse and cost-conscious; innovations—especially to systems as fundamental as the controls and the human-machine interface—are adopted only after being thoroughly scrutinized. The cost and reliability of the components are major driving factors.

Most algorithms assume feedback of the manipulator configuration to achieve very accurate tracking performance, yet few sensors are robust enough for harsh excavation environments. One common problem with estimating the state is that error may compound to an appreciable extent for long motions. The formulation developed for this research uses relatively short-duration motion primitives to describe the motion and may prevent large errors in position estimates from compounding.

Autonomous machines require methods of localizing the digging face, the truck, and obstacles. As yet, fully autonomous systems are neither practical nor feasible for wide scale industry adoption. The Blended SC approach discussed in this thesis is well suited for uncertain, dynamic environments because the human operator is retained as the primary source of control authority.

This chapter discussed a broad overview of topics related to control of excavators. The next chapter takes a deeper look into using shared control to improve machine performance.

CHAPTER IV

SHARED CONTROL BACKGROUND

In this chapter, the general concept of shared control (SC) is discussed to motivate its benefits, problems, and typical contextual scenarios of use. Several varieties of SC are discussed and placed within a general framework of common SC architectures.

While SC may potentially be applied to any controlled system, the context here is the control of a multi degree-of-freedom manipulator. The manipulator is assumed to consist of two subsystems: the linkages and the actuators. In the SC paradigm, the actuator commands result from inputs derived by either a Human Agent (referred to as *human* or *HA*), an Electronic Agent (*EA*), or a combination of both. At the risk of violating standard terminology, the term *machine* will be synonymous with *Electronic Agent*. Hence, shared control will be principally characterized by a human and a machine sharing control of a manipulator.

4.1 General Motivation

Robotic manipulators are commonly used to perform tasks, particularly when the task is too dangerous, too mundane, or requires a scale of force otherwise unachievable by a human [112]. For manipulator control particularly, a human agent excels at reasoning, safety awareness, adaptability, and robustness while an electronic agent has superior numerical capacity, quicker reaction times, and greater precision. For any specific scenario, the goal of SC is to realize the unique benefits of both agents.

Shared control has been used in numerous applications. Generally, the impetus for adopting SC stems from at least one three factors [113].

- A physical or mental disability or cognitive limitation of the human operator make manual control unreliable. The operator may lack sufficient understanding of the controlled system, have finite processing capabilities, or be occupied with tasks of

higher importance

- The environment is unpredictable and rapidly changing.
- The task is too difficult, intricate, or occurs with a geometric or temporal scale unsuitable for manual control or direct teleoperation.

A key feature of SC, teleoperation, supervisory control, manual control, and human-machine interaction is that humans maintain some level of authority within the overall system control loop. Each topic has been well studied and many excellent books thoroughly address these topics, including Poulton [113] and Sheridan [112].

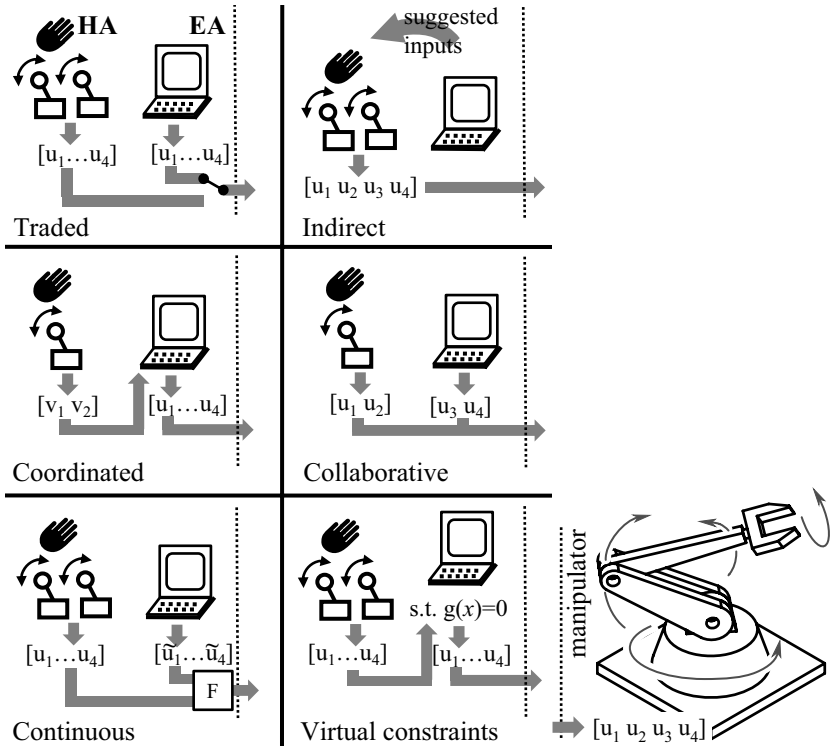
Generally, an autonomous agent is distinguished by its ability to make independent choices. A robot is considered to be autonomous when it can perform a given task in an unstructured environment without continual human guidance. Many industrial and household devices have some degree of autonomy but require, at least, human supervision to perform a task. Fully autonomous robots control the manipulator on their own, while SC approaches allow the human to make some decisions and effect change in the response.

4.1.1 Forms of Shared Control

There are many studies on the level of autonomy a robot assumes while interacting with a human. A primary question is to define the appropriate degree of autonomy. Within SC there exists a spectrum between manual control and autonomous control, which will be discussed next.

Consider the control of a generic 4-DOF manipulator, where the control vector $u = [u_1, u_2, u_3, u_4]^T$ represents the reference velocity tracked by each DOF. The numerous embodiments of SC found in literature can be categorized into the framework described in Figure 4.1 which consists of six distinct shared control forms: Traded, Indirect, Coordinated, Collaborative, Virtual constraints, and Continuous. Each will be discussed in turn, with citations to relevant examples from academic and industrial research.

Traded Control: The HA or EA may initiate, either on demand or automatically based on programed criteria or trigger event, a transfer of full authority to either agent. Control



(a) Mode of human-machine interaction

(b) The controlled machine

Figure 4.1: (a) Interactions between a Human Agent (HA) and Electronic Agent (EA) for sharing control of a general manipulator (b). Each of the manipulator's four degrees of freedom track a reference velocity input u.

is not actually shared by the human and machine at any particular time; rather, one or the other has control at some instant. The trigger event may be initiated by a button press or when the manipulator enters a critical region around an obstacle, for example. Applications include aircraft autopilot systems for which the operator cedes low-level control authority during cruising yet maintains full authority during takeoff and landing. In other approaches, the opeartor has full manual control of the robot while automatically granting the machine authority to prevent imminent collision [114]. Examples in the earthmoving industry include AutoDig [107], return to ready [60, 115, 116], and systems that allow recording a playing back of robot trajectories [59, 117].

In traded control, the human operator is removed from the control loop during certain phases of operation. Research into human psychology suggests that removing the

operator from the control loop, even for small portions of a cycle, may be ill-advised because the operator'a overall attentiveness may drop as a result of the interruptions [61, 62]. Among the consequences may be an increase in task time.

A class of automatic responses to the trigger event includes reactive algorithms, where simple robot behaviors are the result of coupling sensor readings and actions [118]. Kim *et al.* demonstrated another approach in a proof of concept paper [119]. Sensor readings drove manipulator reflexes modeled off of human behavior, such as a withdrawal reflex for obstacle avoidance and a grasp reflex for securely holding objects. The reflexes were triggered by sensor inputs and augment brain-controlled trajectories. While the command signal from the user retained continuous control of the end position, the robot reacted to local sensors detecting proximity, collisions, or other relevant information. Traded control approaches based on purely reactive behaviors tend to have several shortcomings, including abrupt jumps in response, falling into local "traps", and oscillating response.

Indirect Shared Control Through Cues: The EA may not directly influence the input. Rather, the EA derives sensory cues based on programmed criteria and displays these cues to the HA. These cues are designed to influence the input of the HA in a manner amenable to the applied stimulus. Examples include visual indications of suggested process inputs in the control of power plant systems [112] and haptic cues. Indirect SC has also been proposed as an effective way to teach users to perform certain complex tasks, such as minimally invasive telesurgery [120]. Results in vehicle lane tracking from Gillespie's lab suggested the benefits of Indirect Shared Control (including visual, auditory, and haptic cues) between a human and an autonomous system include improved performance on primary tasks, reduced perceptual demands, and freed attention available for secondary tasks [121]. Previous research by Kontz indicated haptic cues can improve digging performance in hydraulic excavation systems [122].

Typically, the purpose of Indirect SC is to increase the operator's perception of the

environment—e.g. revealing the presence and location of virtual or real obstacles—rather than the dynamics, behavior, or performance of the actual manipulator. While increased environmental perception may incidentally lead to decreases in the task cost, conventional approaches do not directly address the goal of decreasing task cost from the perspective of understanding the dynamics of the manipulator. Further, Indirect SC requires hardware capable of providing the sensory cues. Also, the cues require finite cognitive attention from the HA, who may already be overloaded with information. Benefits derived from the Indirect SC must outweigh the installation costs as well as the additional cognitive workload required to interpret and react to the stimulus.

Coordinated Control: The HA has full control of all actuators through a possibly lower dimensioned input v, with the EA transforming fewer high level commands to a greater number of lower level commands. For instance, the EA may enable simpler control of end effector motion by handling the calculations of inverse kinematics. This is often implemented by establishing a virtual or practical constraint such as a manifold of lower dimension than the total DOFs. The operator input then maps to a point on this lower-dimensioned manifold, which exists in the higher dimensioned space of the actuator inputs. The constraint may be a mathematical formulation or a specific mapping from the input space of the operator interface device to the output space of the manipulator. In the excavator domain, coordinated control of the bucket position and orientation has been explored by numerous researchers [50, 53, 56, 122, 123], who generally conclude that coordinated control decreases the amount of skill and time required to accurately execute a prescribed motion.

Collaborative Control: In this case, the HA issues a certain subset of inputs while the EA provides the remaining. Examples include automobile cruise control, where the HA controls steering while the EA modulates the throttle, and automatic parking [124], where the HA controls the throttle while the EA controls steering.

Virtual Constraint: The EA modifies or disallows a subspace of HA commands—as a

function of certain criteria x including speed, proximity to obstacles, or type of payload—to satisfy an arbitrary constraint on the state. For instance, an EA may prevent inputs that cause a wheelchair to collide with a wall while allowing all other inputs [125].

Hayati presented a controller that improves stability of teleoperated systems with transmission delay by imposing virtual constraints along with a coordinated shared control architecture [126]. When in free air, full teleoperation is allowed. When close to or contacting a surface, only teleoperator motions along the free direction (i.e., parallel to the surface) are allowed in order to prevent instability resulting from the time delay (rigid contact in time-delay systems is notoriously unstable). The EA handles motion in the direction normal to the surface to apply a controlled force to the surface [126].

In excavator control, virtual constraints have been tested as measns to allow for digging of level-bottomed trenches [23, 56], as well as introducing programmed regions where the bucket cannot enter [49].

Continuous Shared Control: The commands from the HA and the EA are combined through some functional relationship in a manner that changes in either the HA or EA input are immediately realized as input changes to the manipulator. This category holds the SC paradigm discussed in this thesis, and thus warrants a larger section for background.

Hybrid approach: In practice, a system may utilize multiple types of shared control. The different types may be used at different times, or on different subsets of degrees-of-freedom.

4.2 Continuous Shared Control Background

Continuous shared control is distinguished from the other modes of human-machine interaction because both the EA and HA directly effect, at every instant, the motion of all DOFs. The control is *continuous* because the interaction is immediate and does not have

the switching characteristics of a traded control strategy [119]. Further, unlike the categorization for coordinated or cooperative control, all active DOFs are simultaneously and directly affected. Though there are relatively few works that treat continuous shared control, there are two distinct architectures found in literature: methods based on potential fields, and methods based on command perturbations.

4.2.1 Potential Fields Approach to Continuous SC

Generating robot commands based on artificial potential fields (popularized by Khatib and commonly known as the Potential Fields Approach (PFA) [127]) is commonly used in robot/manipulator control. In this paradigm, the manipulator or robot moves along the least resisted path within a virtual field of forces. For continuous SC applications, the potential fields approach provides a natural and intuitive understanding to weighing and merging the commands of multiple agents. Both the human- and machine-derived commands produce attractive poles, i.e., local minima, in the potential field. Poncela *et al.* [128] propose a PFA method to merge commands of the machine and human at each instant for robot navigation. In essence, the operator's joystick direction is included as another vector in the potential field at each position. The weighting of human/machine inputs are calculated in terms of the efficiency of each agent, where this efficiency is evaluated in terms of local factors that depend on the desired behavior, such as distance to obstacle, path smoothness, and expected path length. Hence, if the robot perceives the operator command as more efficient, then greater authority is shifted to the human. The work of Poncela *et al.* does not guarantee that the resultant motion is of lower cost than the original human input. Furthermore, the study was restricted to planar motion with a two inputs, speed and direction.

4.2.2 Perturbed Input Approach to Continuous SC

Another paradigm for continuous shared control includes formulations utilizing command perturbations as the primary way of modifying the operator control signal. Either nominal manual (including teleoperated) trajectories are modified autonomously, or nominal autonomous trajectories are modified by the operator. For example, the operator may modify

pre-planned motion trajectories to track unmodeled target motions [129]. The Blended SC architecture described next is compatible with both approaches.

4.3 Blended Shared Control

This research introduces a new continuous SC architecture which is termed Blended Shared Control. The purpose of this section is to describe the general architecture; the details of each component of the architecture is discussed in subsequent chapters.

This architecture considered in Figure 4.2 consists of a Human Agent (HA) or operator, an Electronic Agent (EA), and a controlled system. The Blended Shared Control algorithms are implemented within the EA. The EA has three distinct functions: Task Identification, Task Optimization, and Blended Shared Control. The operator provides the input command $\bar{u}$ via a human interface device such as a joystick. The operator command is generally uncertain, but is a function of the intended task and the operator's perception of the machine response y through sensory feedback. A high-level EA modifies the original operator command through some general functional relationship to δu. Here the functional relationship is a simple perturbation by δu. The command perturbation is calculated by the *Blended Shared Control* module and may be a function of several terms including the optimal input u^* as calculated by the *optimization* module, the original input command $\bar{u}$, and machine response y. The optimized input u^* is determined by dynamic models of the system, and the estimated or measured machine state y, and a set of data consisting of constraints and objective functions which are specific to the particular task being completed by the operator. The constraints and objective function are determined by the *task identification* module of the robotic controller.

Several aspects of this architecture enable a positive synergy between the EA and HA: there are capabilities of a HA (e.g., reasoning, safety awareness, robustness, "ideal" cost function) that complement attributes of an EA (e.g., incorporation of complex system models, numerical capacity to solve those models, storage of much expert knowledge). These synergies of the Blended SC will nominally be leveraged to increase utility of the overall process. Realistically, there are several aspects in this process which may result in dis-utility

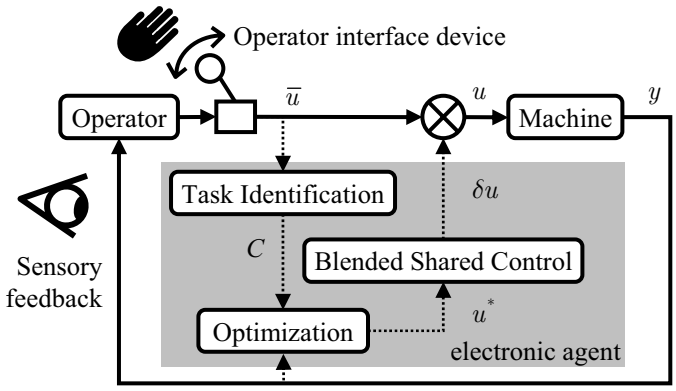


Figure 4.2: The architecture for *Blended Shared Control*. The Electronic Agent perturbs the operator command by an amount δu, based on three modules: Task Identification, Optimization, and Blended Shared Control.

and hence must be considered. Such unresolved issues include the effects of conflicting objectives between the EA and the HA (e.g., one agent values minimum time while the other wishes minimum energy), and under which conditions can the task cost be shown to decrease as a result of the SC perturbation.

4.3.1 Discussion

The purpose of this chapter was to introduce the general purpose Blended Shared Control paradigm, and to show its relation to a larger context of shared control and human/machine interaction. Several types of shared control have yielded benefits to a variety of traditionally manually controlled situations. The typical *modus operandi* is to relax the level of authority assumed by the human operator, while providing the machine with the ability to effect the response.

The nature of the application domain must be considered. The SC approach must not unduly inhibit the concentration of the operator, especially given the dynamic, uncertain, and high-risk environment of the sites at which excavators work. Human concentration is a limited resource that must be carefully managed.

The cost of new innovations to high-level controllers—measured in terms of dollars, risk, and complexity—must be offset by the advantages brought by the innovations. To its advantage, the Blended SC architecture may not require any major hardware changes or additional sensors. Further, the level of automatic assistance given to the operator can be

easily tuned and managed. Even substantial productivity and efficiency gains are rendered useless if the safety and effectiveness of the operator is unduly violated.

Looking forward, a formulation of a single-input example is given in the next chapter, Chapter 5, to illustrate the application of the Blended Shared Control architecture. Chapters 7 to 10 discuss the application of this architecture to the control of a four-DOF hydraulic manipulator.

CHAPTER V

BLENDED SHARED CONTROL OF ZERMELO'S NAVIGATION PROBLEM

Chapter 4 provided an overview of SC and introduced a concept for a perturbation-based Blended Shared Control (SC) architecture. The purpose of this chapter is to provide an extra application example illustrating the application of the Blended SC paradigm. The goal is to provide evidence that the approach has more broad applicability than the manipulator control which is the focus of the bulk of this thesis.

The example case considered here is Zermelo's navigation problem, which is far simpler than an excavator. The formulation and solution to the problem will be discussed, as will experimental results that illustrate the decrease in task completion time realized by the current approach. The architecture is experimentally shown to be superior to manual control, but does not perform as well as indirect shared control including haptic feedback and heads-up display.

5.1 Nomenclature

HA	The Human Agent in the shared control loop
EA	The Electronic Agent in the shared control loop
$\theta, \bar{\theta}, \theta^*$	Ship heading angle, heading commanded by the HA, and optimal heading commanded by the EA
θ_f	Final heading angle when ship approaches the origin
$\delta\theta$	Command perturbation introduced by EA
$\boldsymbol{x}$	$\boldsymbol{x} = [x, y]^T$ is the ship location
V	The speed relative to the water
$u(y)$	The linearly varying currents
d_0	Distance from the origin, beyond which the HA has full control

Δ	Error between HA command and EA command ($\Delta = \bar{\theta} - \theta^*$)
Δ_0	The maximum error in command, beyond which the HA has full control
$\bar{\phi}$	The joystick angle
$T(\boldsymbol{x})$	Minimum time to go; the time to reach origin along optimal path starting at $\boldsymbol{x}$
a	Blended shared control parameter; tunes the level of control authority between HA and EA
MC	Manual control
HUD	Heads up display (type of indirect shared control)
Haptic	Haptic feedback (type of indirect shared control)
SC2	Blended shared control, where control of θ is shared
SCJS	Blended shared control, where control of ϕ is shared

5.2 Blended Shared Control

The architecture for Blended SC of a single input is sketched in Figure 5.1. This architecture consists of a Human Agent (HA) or operator, an Electronic Agent (EA) with three distinct functions, and a controlled system. The operator issues input command $\bar{\theta}$ via a human interface device such as a joystick and perceives the boat response $\boldsymbol{x}$ through visual observation of the response. A high-level EA modifies the original operator command through some general functional relationship to $\delta\theta$. Here the functional relationship is a simple summation $\theta = \bar{\theta} + \delta\theta$. The command perturbation is calculated by the *Blended shared control* module and may be a function of several terms including the optimal input θ^* as calculated by the *optimization* module, the original input command $\bar{\theta}$, and machine response $\boldsymbol{x}$. The optimized command θ^* is determined by dynamic models of the system, the feedback $\boldsymbol{x}$, and a set of data C consisting of constraints and objective functions which are specific to the particular task being completed by the operator. The constraints and objective function are determined by the *task identification* module of the robotic controller.

There are several areas in this process that enable a positive synergy between the EA and HA. The HA has capabilities including reasoning, safety awareness, and robustness while

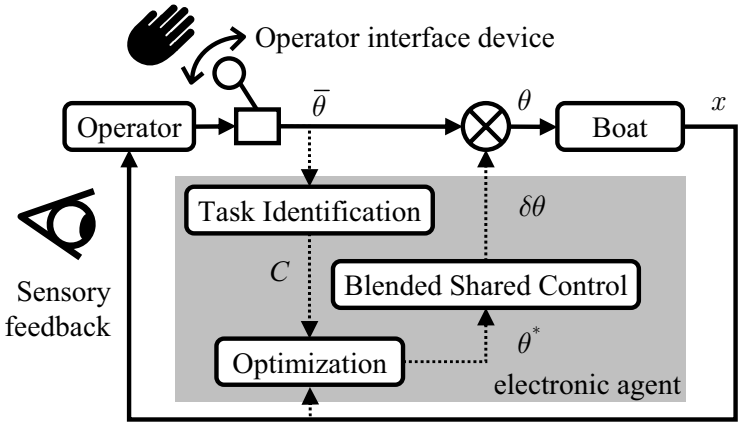


Figure 5.1: The architecture for perturbation-based *Blended Shared Control* of a single control input θ.

the EA has complimentary attributes such as incorporation of complex system models, high numerical capacity to solve those models, and storage of vast expert knowledge. These synergies of the Blended SC will nominally be leveraged to increase utility of the overall process. In reality, there are also several stages in this process which may result in dis-utility and hence must be considered. Such unresolved issues include the effects of conflicting objectives between the EA and the HA, e.g., one agent values minimum time while the other seeks minimum energy.

5.2.1 Shared Control Scheme

The Blended SC law for a system with a single control input θ is described next. The difference of the operator's command $\bar{\theta}$ and the optimal command θ^* calculated by the EA is

$$\Delta \equiv \bar{\theta} - \theta^* \tag{5.1}$$

The optimization as calculated by the SC module depends on the plant models and a cost function internal to the EA. A *command perturbation* $\delta\theta$ calculated by the SC module is added to the operator command giving

$$\theta = \bar{\theta} + \delta\theta$$

where θ is the control input to the plant. Designing the command perturbation is a major subject of the current investigation into Blended SC. In the case of a pursuit or interception problem, for example, the perturbation may be a function of any number of terms including an operator setpoint, distance to target, time on target, or Δ from (5.1). For example, choosing $\delta\theta = -a\Delta$ gives

$$\theta = \bar{\theta} - a\Delta \tag{5.2}$$

with the *Blended shared control parameter* $a \in [0, 1]$. Varying a on the interval $[0, 1]$ allows a continuum between full automation and full manual control, thus enabling a control engineer to scale the influence of the shared controller. When $a = 0$ the system is under manual control (i.e., $\theta = \bar{\theta}$) and when $a = 1$ the system is fully autonomous (i.e., $\theta = \theta^*$).

5.3 Zermelo's Problem: Time-Optimal Ship Navigation

A classic optimal control problem known as Zermelo's Problem is useful for studying the Blended SC law because of its known closed-form solution [130]. In addition, the task—minimize transit time to the origin—is easily defined and explained to a human operator. Hence, the *task* is known and does not need to be identified. The choice of Zermelo's problem as a prototype was arbitrary, aside from its convenience. The purpose is not to promote improved ship navigation, but to study the interaction between human and electronic agents during Blended SC.

In Zermelo's problem, a ship (modeled as a particle) travels with constant speed V relative to the water while navigating a region of strong currents. The captain modulates the ship's heading θ to minimize travel time T to the origin. The equations describing the optimal path for the case of linearly varying current velocity are [130]

$$\begin{aligned} \dot{x} &= V\cos\theta + u(y) \\ \dot{y} &= V\sin\theta \end{aligned} \tag{5.3}$$

and

$$\dot{\theta} = -\cos^2\theta \frac{du}{dy} \tag{5.4}$$

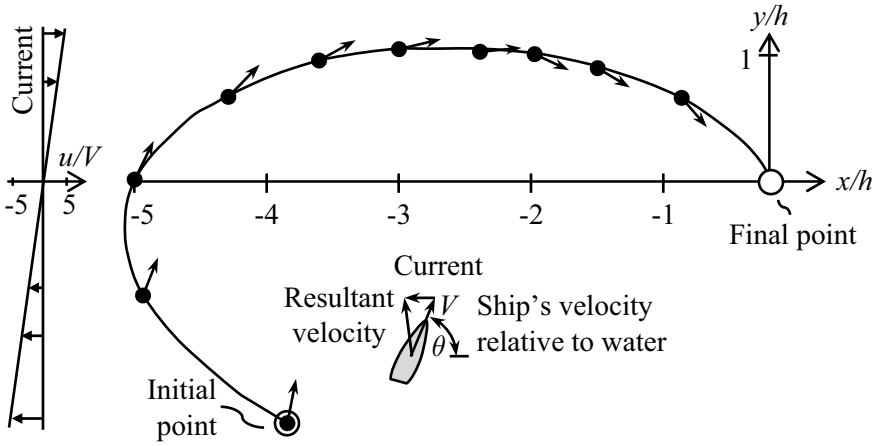


Figure 5.2: Minimum time path through a region of linearly increasing currents [130].

where θ is the ship's heading measured from the x-axis, (x, y) are its coordinates, and $u(y) = Vy/h$ is the velocity of the current, which varies linearly with the y displacement from the origin as in Figure 5.2 with h being a constant scaling factor. The initial value of θ is chosen so that the path passes through the origin. For the linearly varying current strength considered here, the optimal steering angle can be related to the ship position through a system of implicit feedback equations [130]

$$\frac{y}{h} = \sec\theta_f - \sec\theta \tag{5.5}$$

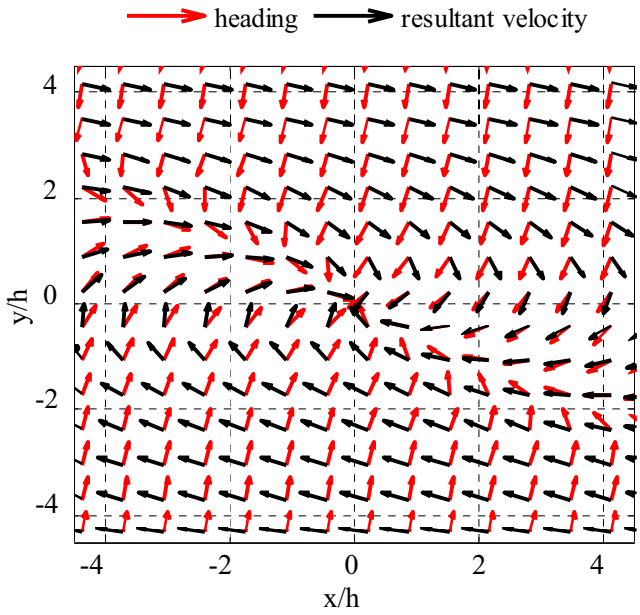
$$\frac{x}{h} = \frac{1}{2}\left[\sec\theta_f\left(\tan\theta_f - \tan\theta\right) - \tan\theta\left(\sec\theta_f - \sec\theta\right)\right] \tag{5.6}$$

$$+ \frac{1}{2}\log\frac{\tan\theta_f + \sec\theta_f}{\tan\theta + \sec\theta} \tag{5.7}$$

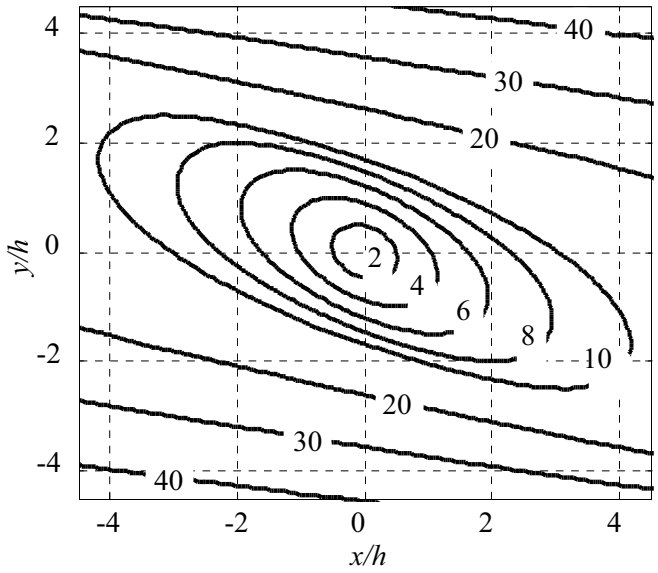
Figure 5.2 shows a particular optimal path along with the optimal heading angle at several points along the optimal path. A vector field of solutions to (5.7) are plotted in Figure 5.3.

The single-input perturbed Blended SC law (5.2) is used on this problem. In (5.2), the control designer has freedom to select the particular form of the SC parameter a. Suppose a is selected to be

$$a = \max\left(0, 1 - \frac{d}{d_0}\right) \cdot \max\left(0, 1 - \left(\frac{\Delta}{\Delta_0}\right)^2\right). \tag{5.8}$$



(a) Optimal heading (red arrows) and direction of resulting ship velocity (dark arrows)



(b) Optimal time to origin

Figure 5.3: Solutions to Zermelo's navigation problem. In (b), the optimal time to the origin at each location is expressed in units of h/V; plot shown for $h/V = 4$.

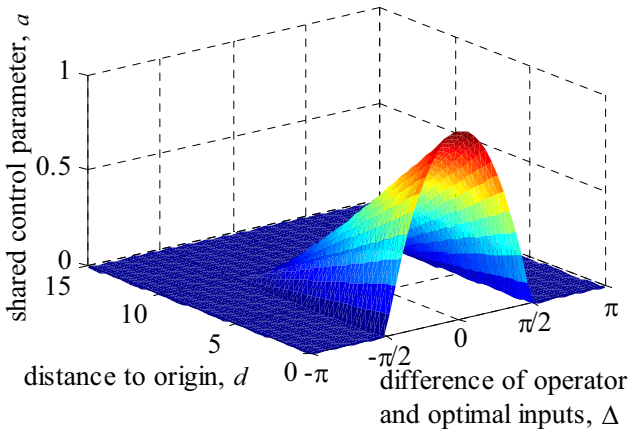


Figure 5.4: The SC parameter a in parabolic form (5.8) with $\Delta_0 = \frac{\pi}{2}$ and $d_0 = 10$.

Figure 5.4 shows plots of a for the parabolic form (5.8). This particular form in (5.8) allows manual operation ($a = 0$) if the ship is further than a critical distance d_0 from the origin, or if the input command deviates from the optimal by greater than Δ_0. Thus, the Blended SC relinquishes control authority to the operator in the presence of large error between the operator input and the optimal input calculated by the EA.

Choosing a as such is a first attempt at increasing overall system robustness by resolving the conflict that may arise between the independent agents; such conflict may stem from inaccurate plant or environment models, dissimilar cost functions used, or different goals altogether between the HA and EA. In the Blended SC paradigm the operator—rather than a complicated automatic controller requiring many feedback measurements—ultimately provides for the robustness and corrective action of the system.

5.4 Evaluation of Shared Control

This portion describes the experimental setup for evaluating the Blended SC. Each operator viewed a monitor (Figure 5.5 and Figure 5.6) depicting a ship moving in a simple virtual reality (VR) environment with dynamics governed by (5.3) (the VR environment was created using the VR Toolbox in MATLAB; the simulation runs at 1 KHz on an xPC Target computer). A green ring (see Figure 5.6) represented the origin to which the operator was instructed to navigate as quickly as possible. A green sphere drawn in front of the ship to represented the heading θ. Two static arrows illustrated the *direction* of the current's flow

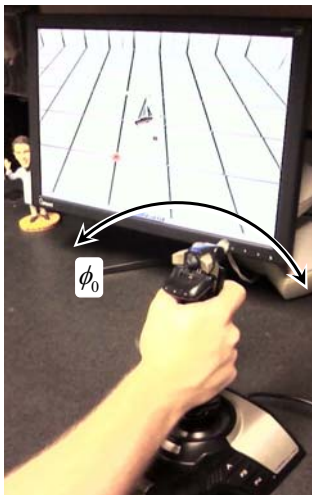


Figure 5.5: The operator video display and joystick

on either side of the origin. In the HUD mode (described in the next section), an additional red sphere was displayed to indicate the optimal heading angle.

The operator commanded the heading angle $\bar{\theta}$ by displacing the joystick an angle $\bar{\phi}$; the joystick angle controlled the rate of change of the heading through

$$\bar{\theta} = \alpha \int \bar{\phi} dt$$

where α is a constant for tuning the *snappiness* of the ship response to changes in joystick angle. A deadzone on the joystick input angle $\bar{\phi}$ was applied in software to prevent unintentional drift of the ship's heading.

5.4.1 Description of Control Types Evaluated

Five varieties of control were studied in this experiment and are summarized next.

Manual control (MC): Implemented by setting $a = 0$ in (5.2), thus the HA was in full control of the ship heading giving $\theta = \bar{\theta}$. No sensory cues were displayed to the operator, besides the standard VR interface. This control was used as a baseline for determining operator performance in absence of supplementary information or aiding controls.

Heads up display (HUD): The HA had manual control of the ship ($a = 0$ so $\theta = \bar{\theta}$).

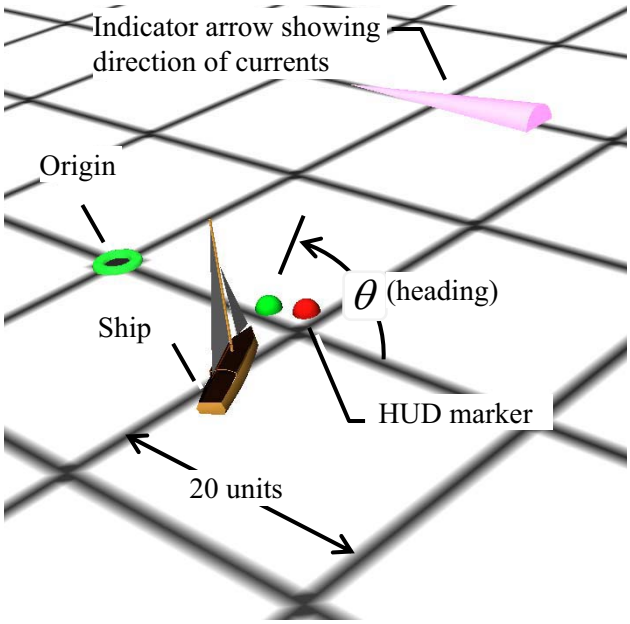


Figure 5.6: Close-up of operator display; labels are not shown during trials.

A red dot (as in Figure 5.6) was displayed to represent the optimal ship heading calculated by the EA. The operator was instructed before the experiment to align the green heading indicator dot with the red HUD marker. This control provided a baseline for determining the maximum operator capabilities, i.e., the capability the operator would have *if* the optimal solution was known to the operator. The HUD was a form of indirect SC.

Haptic feedback (Haptic): This was a second type of indirect SC. The HA had manual control of the ship. A Saitek Cyborg EVO Force joystick displayed a restoring force to the operator based on the input error Δ, with

$$F = -F_{\max} \cdot \min\left(\left|\frac{\Delta}{\Delta_{\max}}\right|, 1\right) \cdot \operatorname{sgn}(\Delta)$$

This resulting joystick force the operator's hand in a direction that caused θ to approach θ^*. For example, if $\Delta \leq 0$ then the joystick applied a force to the *right*, thus signaling the operator to decrease angle $\bar{\phi}$. The particular values used were $\Delta_{\max} = \pi/2$, $F_{\max} = 2.1$ N (measured at the joystick palm grip).

Shared control, heading (SC2): The HA and EA shared control of the ship heading through the relation

$$\theta = \bar{\theta} - a\Delta,$$

with

$$a = \max\left(0, 1 - \frac{d}{d_0}\right) \cdot \max\left(0, 1 - \left(\frac{\Delta}{\Delta_0}\right)^2\right)$$

as in Figure 5.4. No additional cues were displayed to the operator. The particular values during the experiment were $d_0 = 25$, $\Delta_0 = 3\pi/4$.

Shared control, rate (SCJS): Here, the original joystick input angle $\bar{\phi}$ was modified by the EA giving an effective joystick input angle of $\phi = \bar{\phi} + \delta_\phi \cdot u(\delta_\phi, \bar{\phi})$, where

$$\delta_\phi = \begin{cases} -\frac{\bar{\phi}}{2} & \text{for } |\Delta| \leq \theta_{th}\,, \\ -k \cdot \operatorname{sgn}(\Delta) & \text{otherwise}\,. \end{cases}$$

and the indicator function u is defined as

$$u(x, y) = \begin{cases} 1 & \text{if} \quad \operatorname{sgn}(x) = \operatorname{sgn}(y)\ , \\ 0 & \text{otherwise}\,. \end{cases}$$

Hence, the HA and EA shared control of the *rate*, ϕ, at which the ship's commanded heading changes. No additional cues were displayed to the operator. The specific values used were $\theta_{th} = \pi/12$, and $k = 0.5$.

The difference between SC2 and SCJS is subtle: in SC2 the operator's intended ship heading $\bar{\theta}$ is perturbed by the EA, whereas in SCJS the intended joystick *angle* $\bar{\phi}$ is perturbed.

5.4.2 Effect of Shared Control on Minimum Time-to-go

Let $T(\boldsymbol{x})$ be the *minimum time-to-go* at the location $\boldsymbol{x} = [x, y]^T$, that is, the time to reach the origin assuming the ship starts at $\boldsymbol{x}$ and follows the time-optimal path to the origin. It

can be shown that

$$T(\boldsymbol{x}) = \frac{h}{V}(\tan\theta(\boldsymbol{x}) - \tan\theta_f(\boldsymbol{x}))$$

where θ and θ_f are implicit functions of $\boldsymbol{x}$ from (5.7). Consider the manually controlled case $\theta = \bar{\theta} = \theta^* + \Delta$, with $\Delta = \bar{\theta} - \theta^*$ as defined in (5.1). The ship starts at location $\boldsymbol{x}$ and travels under manual control for a length of time dt. After time dt the minimum time-to-go will be

$$T_{MC} = T(\boldsymbol{x} + f(\boldsymbol{x}, \theta^* + \Delta)dt)$$

where $f(\boldsymbol{x}, \theta)$ is the vector form of equations of motion (5.3). If, on the other hand, the heading is under the SC law in (5.2) then $\theta = \theta^* + (1-a)\Delta$ and the minimum time-to-go after time dt can be written as

$$T_{SC} = T(\boldsymbol{x} + f(\boldsymbol{x}, \theta^* + (1-a)\,\Delta)dt)$$

The function $T(\boldsymbol{x} + f(\boldsymbol{x}, \theta)dt)$ is convex[1] in the variable θ (holding all other variables constant) for all $dt > 0$, with the minimum necessarily occurring at the optimal (i.e., where $\theta = \theta^*$). Thus, for $a \in [0, 1]$ and any $\boldsymbol{x}$,

$$T_{SC} \leq T_{MC}$$

This is illustrated in Figure 5.7, where $T(x + f(x, \theta)dt)$ is plotted for $\boldsymbol{x} = [10, 12]^T$ and $dt = 0.01$. Hence, the minimum time-to-go using SC never exceeds the time with manual control. For a *certain* operator input θ at $\boldsymbol{x}$, **the minimum time-to-go with Blended SC will never be strictly worse then that with manual control**. Obviously, for other systems where the cost function is *not* convex, the Blended SC (5.2) may result in greater cost than manual control alone, as the EA may *push* the HA commands to a higher cost. Also, it is assumed the operator input at (x, y) is independent of the type of control active, so the operator command is assumed to not depend on the control law. Finding ways to

[1]Because of the dependence of both θ and θ_f on changes in $\boldsymbol{x}$, showing convexity by proving $d^2/d\theta^2 T(\boldsymbol{x} + f(\boldsymbol{x}, \theta)dt) > 0, \forall\theta \in [-\pi, \pi]$ is straightforward yet tedious

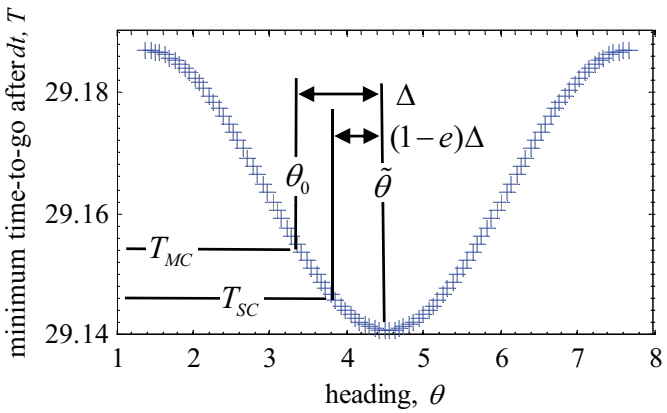


Figure 5.7: The minimum time-to-go, $T(x + f(x, \theta)dt)$.

Table 5.1: The optimal times to the origin and to the critical distance which terminated each trial.

Location	Optimal time to origin	Optimal time to $d = 1.5$
(12,12)	14.97	14.37
(0,17)	18.47	18.04
(12,-12)	7.43	6.63

settle these issues is the subject of the ongoing research described in this thesis.

5.4.3 Experimental Procedure

Before the experiment, the operator was allowed five practice runs starting from various locations in the field. During these runs the HUD control was active to provide instruction on navigating the currents. Each of the trials began with the ship at one of three locations: $(12, 12)$, $(12, -12)$, and $(0, 17)$. The constants were set to $h = 4$ and $V = 2$. The operator triggered a *start* button on the joystick and the simulation proceeded in real time with one of the five control laws active. A trial concluded when the ship was within distance $d = 1.5$ of the origin (to avoid severely penalizing a user that only slightly misses the origin, as it takes substantial time to loop back around). The optimal solutions from each location to the origin and to the $d = 1.5$ boundary are shown in Table 5.1.

At no time during the experiment was the operator explicitly informed which of the control laws was active. In cases when sensory cues were displayed, the operator was not told the intended *meaning* of the cue. The starting locations and controller orders

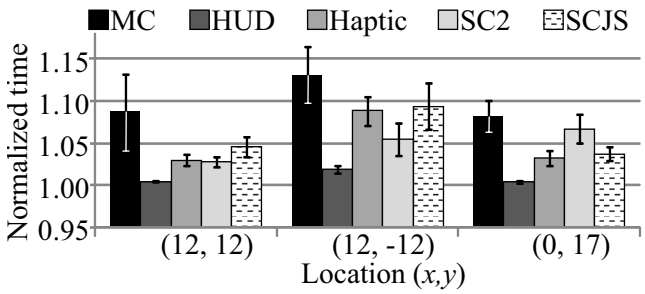


Figure 5.8: Summary of completion times

were randomized for each operator. To partially balance learning effects, each operator experienced each controller type once (but in a random order) before the controller types were repeated. Each operator visited each location exactly three times for each controller during the experiment, totaling 45 trials per participant.

5.5 Results

Eight computer literate participants volunteered for the experiment. Results summarizing the performance of all operators are summarized in Figure 5.8. The times are normalized with respect to the optimal time to origin from each location, then averaged among all operators for each controller. Error bars denote 95% confidence intervals. For each controller, at each location, $N = 24$. The optimal times to origin are 14.97 s, 7.43 s, and 18.47 s respectively for starting locations $(12, 12)$, $(12, -12)$, and $(0, 17)$. The mean HUD controller times were very consistent and only marginally exceeded the optimal time, as expected, presumably because the tracking skill of the operators was sufficient to track the displayed optimal command. While the HUD and Haptic control types produces superior results in this case, it may be impractical in reality considering the special hardware required and the cognitive attention/distraction that may be introduced to the HA. In contrast, the Blended SC methods require only a way of modifying the operator input based on the intended task and the machine state, both of which may be estimated in practice via the sensed inputs alone. The completion times of the other controllers show more variation than HUD; however, both of the Blended SC approaches generally surpassed the performance under manual control.

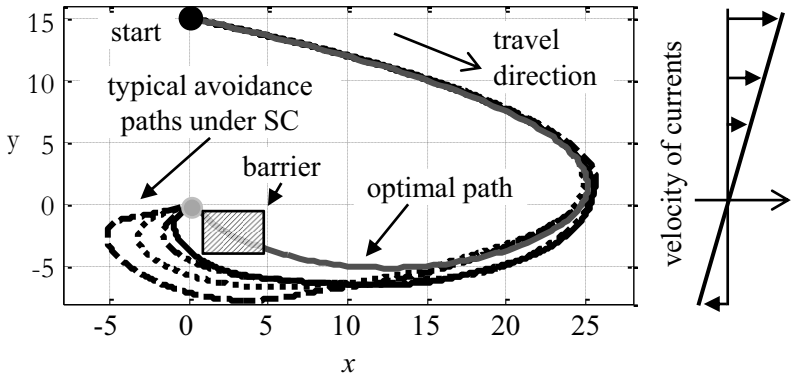


Figure 5.9: A *barrier* may obstruct the optimal path.

A fair criticism of Blended SC in general is that the HA (who, depending on experience and conditioning, may be very aware of a particular machine *feel*) cedes *too much* authority to the EA. This may at best lead to a benign sense that the machine is not responding in a manner consistent with operator expectations, and, at worst, lead to the machine failing to respond to an operator's safety-based evasive maneuvers.

To test for loss of control in this single-input example, four additional trials (two with MC, two with SC2) with each operator were performed starting from position $(15, 0)$. A barrier was intentionally placed to occlude the optimal path as in Figure 5.9; hence the EA essentially tried to cause the HA to collide with the barrier at the same time that the operator was working to avoid it. The operator performance with barriers present was 19.1 s and 19.8 s, respectively for MC and SC2; however, the data lacked sufficient statistical significant to clearly deem one approach superior to the other. Traversing the optimal path from position $(15, 0)$ to the origin in absence of the barrier takes 16.0 s; but the optimal path which avoids the barrier was not calculated. More significant was the fact that, under both manual and SC, only two trials among *all* experimental subjects resulted in collision with the barrier. No operator affirmed a feeling of *loss of control* when queried about navigating around the barrier.

5.6 Conclusions

An architecture for Blended Shared Control (SC) of a system with a single input was presented. This was an initial illustration of applying Blended SC to a class of problems,

including Zermelo's navigation problem, which have a well-defined task and a closed-form optimal solution which was globally convex in the input variable influenced by the SC.

Further, when the HA and EA had equivalent cost functions, the task time was lower with SC than with manual control. In the case of conflicting objectives, the results were not conclusive but did suggest that the HA was able to override the EA to prevent collision with an obstacle in the nominal path tracked by the EA.

For this class of problems, initial evidence indicated that the Blended SC approach is superior to purely manual control. Indirect SC included visual and haptic feedback and resulted in lower task completion times than Blended SC, but required both active attention from the operator and additional hardware to implement. Changes to the operator interface may be unacceptable in some applications where consistency between machines is tantamount.

The next Chapters discuss the application of SC to a four DOF manipulator, beginning with identification of the intended operator task in Chapter 7, continuing with dynamics and optimization of the excavator system in Chapter 8 and 9, and finishing with the Shared Control formulation in Chapter 10.

CHAPTER VI

BACKGROUND FOR TASK IDENTIFICATION AND ITS APPLICATION TO EXCAVATION

A key component of the Blended Shared Control architecture is estimation of the operator's intended task. This chapter discusses background into why it is necessary to know the operator's task; how the task estimation problem is solved in other domains related to robotics; and how this research introduced a simple, unique, and effective solution approach. The chapter concludes by presenting an example of task identification.

6.1 Background

Before modifying an operator command, it is important to first know not just the input at that time (which is measurable), but also the task that the operator intends to accomplish. Generally, this task is directly measurable. The goal is to derive the intended task from sensory information including commanded actuator velocities. Understanding the mapping from operator velocity commands to intended task has several applications, such as providing a compact description of fundamental tasks, enabling real-time task interpolation to predict future inputs, recognition of important or unexpected events, or automatically lowering the sensitivity of the input device when a high-precision task is expected.

6.2 Analysis of an Excavation Dig Cycle

This section introduces a typical dig cycle. Data from recorded field tests will be used to argue the merits of the task identification algorithm created through this research, and to highlight the shortcomings in the methods of conventional task identification methods.

Excavators are most often used for trenching, truck loading, and bulk digging. All of these are quasi-repetitive processes. Ancillary, non-cyclical uses of the machine, such as craning, hammering, or tamping, are not considered. The common trenching cycle may be delineated by four segments sketched in Figure 6.1: *dig* (fill the bucket with soil), *raise*

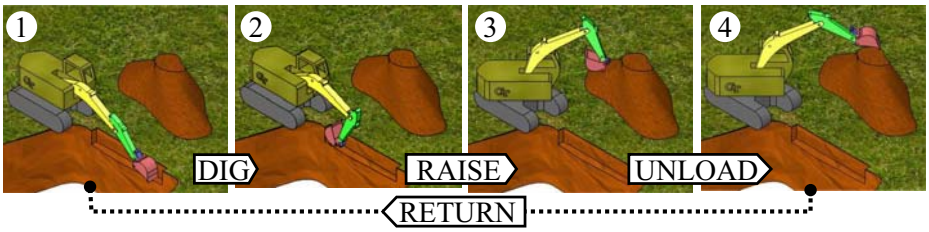


Figure 6.1: Illustration of common trenching dig cycle

(raise bucket out of trench and begin towards spoil pile), *unload* (empty bucket above spoil pile; often involves "flick" of the bucket), and *return* (return to trench). Each cycle is roughly the same as the preceding cycle, except for relatively minor differences necessary to account for process dynamics including varying trench depth, varying ground penetration locations, and changing dump locations.

Humans are imperfect operators. For example, in the case of manipulator control, it is difficult to repeatedly hit a small target with very great accuracy; there will be randomness in the final end effector position. To better understand the nature of these variations, an analysis was conducted of data from a fully instrumented 20-ton excavator operated by an expert with over 25 years of professional experience. The operator was instructed to dig a standard flat-bottomed trench to 2.5 m depth. The boom cylinder stroke and swing angle during the first 25 cycles is plotted in Figure 6.2. The extreme values of each function for each cycle are denoted with the gray dots. Both types of cycle variance are evident. The minimum boom cylinder stroke lengths generally trend downwards as more cycles are performed, due to the gradual increase in trench depth. Similarly, the maximum cylinder stroke increases as the spoil pile height slowly increases.

Observe that the variance in swing angle is larger during final positioning over the spoil pile. Over the spoil pile, the variance in swing angle equates to a 1.04 m standard deviation in final bucket position. Upon entering the trench, the final bucket position has a standard deviation of 0.16 m. There is a higher penalty associated with missing the trench than with missing the center of the spoil pile. Fitt's law predicts that the manual positioning of an end effector over a larger target will be subject to a greater amount of error than an equivalent positioning task but with a smaller target [113]. This phenomenon is evident in

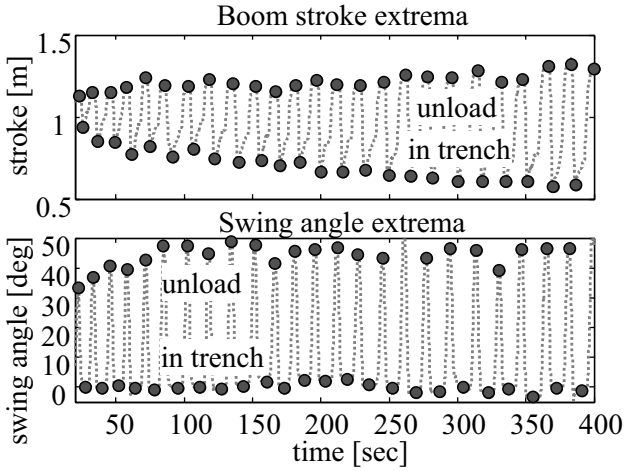


Figure 6.2: The extreme values in boom cylinder stroke and swing angle for each cycle during the first 25 cycles of trench digging.

the observed response.

6.3 Task Identification in the Mobile Hydraulics Application Domain

In the mobile fluid power domain, there exist techniques are technically considered online task identification and hence deserve mention here for completeness. For example, certain logic conditions acting on the operator input may prompt a system to automatically divert flow from a function to cause another function to speed up. In other cases, the task is directly communicated by the operator. The operator may manually select from a predefined set of operating modes, for example [131]. None of the existing methods are suitable for predicting the gross motion of the machine, however.

6.4 Task ID in the Robotic Manipulator Domain

The problem of deducing intended task based on sensed operator inputs may be considered a sub-problem of general pattern recognition. Classical approaches to solve these problems include neural networks, hidden Markov models, and linear and nonlinear variations within the vein of principal component analysis. These are briefly discussed next.

This section gives a brief overview of three popular techniques (Dimensionality reduction, Hidden Markov models, and Neural networks) for identifying the task of an human-controlled, multi-DOF robotic manipulator.

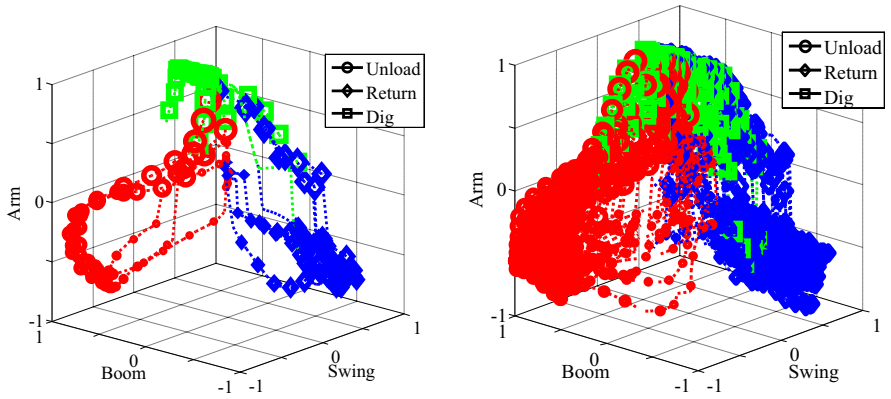
6.4.1 Dimensionality Reduction

Principal component analysis (PCA) is a classical technique for dimensionality reduction and is simple to implement and guaranteed to represent the true structure of data near a linear subspace of the high-dimensional input space [132]. PCA is used in many domains to transform a set of correlated variables from some large-dimension space into a smaller set of uncorrelated principal components. A well-known example is the use of PCA in facial recognition [133]. The most relevant application domain to the excavator task identification problem is movement classification, where a set of measured (human) joint angles are used to classify observed movements [134, 135].

In Jenkins' work [134, 135], a measured time series of joint angles of a human's four DOF arm are segmented and the principal components calculated using a spatial-temporal extension of Tenenbaum's innovative Isomap algorithm [136]. The measured projections are classified by comparing the measured principal components with a set of pre-programmed primitives (e.g., one primitive may be *reach out* while another may be *reach up*). The classical Isomap algorithm was applied to digging data for this research, but the results were very sensitive to the topology of the data—as warned by Jenkins—and thus did not yield acceptable results to be presented here. The spatial-temporal algorithm was not applied to the excavator task identification problem.

6.4.2 Pattern Recognition Using a Neural Network

A four-dimensional plot of operator input commands during trenching is shown in Figure 6.3 (note that arm, boom, and swing commands are normalized to $[-1, 1]$ and graphed; and that markers of varying size denote the bucket velocity command). The line color and marker shape indicate the manually-categorized task (here the tasks correspond to a dig cycle segment) at each instant. The tight clustering of trajectories in Figure 6.3a hints at the feasibility of task identification based on operator input. On the other hand, Figure 6.3b illustrates the complexity induced by variations encountered during an extended digging duration. Observe that the same point in input space may map to multiple subtask outputs.



(a) N=3 cycles (b) N=24 cycles

Figure 6.3: Operator input commands during consecutive trenching cycles. Marker symbols (and line color) denote segment of dig cycle. Size of marker denotes Bucket command (smallest symbol corresponds to bucket command of -1, largest symbol to command of $+1$. Equivalent points in the input space map to different subtasks.

Neural networks (NNs) have been used for many applications including pattern recognition and next-in-sequence prediction [137]. Hertz provides an excellent resource for NNs [138]. For this example, a 2-layer NN having 4 inputs, 1 output, and 20 neurons in the hidden layer is used in an initial attempt to classify the input commands according to subtask. Training data inputs consists of the measured and filtered velocity commands from the first 183 s (13 dig cycles) of a trenching task, and the network training targets were the actual subtask performed (either *dig, unload*, or *return*), as classified from manual inspection of the data. The data were resampled to 10 Hz. A standard back propagation algorithm trains the NN. To check the performance and robustness the (trained) NN is applied to the original data as well as the second half (i.e., *unseen portion*) of the trenching data. The simulation results are shown in Figure 6.4. It should be noted that while the task and subtasks remain the same, the nature of the task changes as the cycle progresses (e.g., as the trench gradually deepens the *boom up* command duration increases as well). Comparing each time step, the network had 97.3% accuracy for the trained set, and 94.2% accuracy for the unseen set. For this operator, the identification performance seems acceptable, considering that the input varies appreciably from one cycle to the next.

Input trajectories for completing a task will vary among operators. Data from a fully

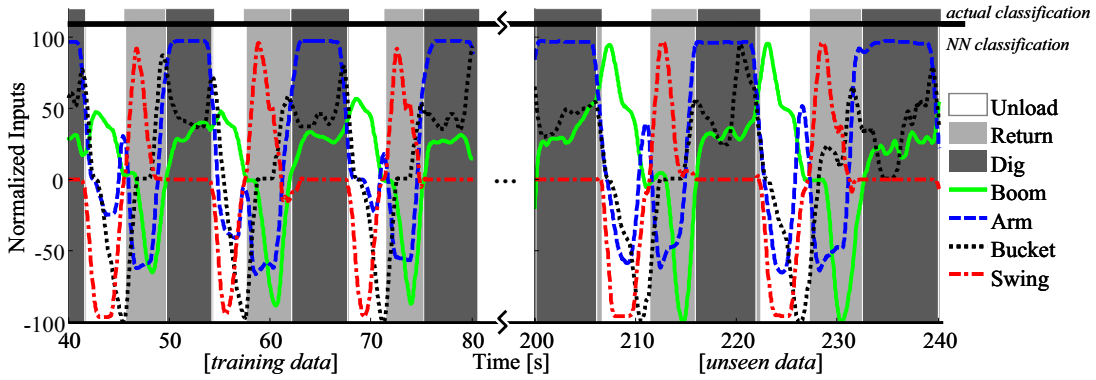


Figure 6.4: Input signals and subtask classifications for a long trenching cycle. Above the heavy line is the actual manually assigned subtask classification; below is the classification from the neural network after training. The NN was trained with training data of the initial 183s (13 dig cycles). The final 183s is data not seen during the training process.

instrumented excavator was obtained during an experiment performed by two expert excavator operators. Both operators were instructed to dig his own level-bottom trench to a depth of 3.5 m. The same fully instrumented excavator was used by both operators. One of the study's objectives was to determine the extent to which the operator *style* differed for completing identical tasks. Though anecdotally each performed equally well, two very different command sequences resulted. The operators' differing technique is markedly apparent in histograms of the fraction of time for which certain control patterns are input. Figure 6.5 shows that both operators use the *AR, BM, BK* (arm, boom, bucket) pattern as the major component of the dig cycle (this pattern generally, but not exclusively, maps to the *digging* subtask). Besides this commonality, the rank-ordering of common input patterns between these operators is dissimilar. The NN discussed above is not robust to this inter-operator difference in digging *style*. Thus, a method of task identification that is robust to the style of many operators is needed.

6.4.3 Summary of Task Identification Background

This chapter summarized conventional approaches of associating manipulator commands with high-level tasks. These approaches, including dimensionality reduction and neural networks, have thus far proven insufficient for identifying the task of human-controlled, four DOF manipulators. Many methods rely on matching basic templates of the observed

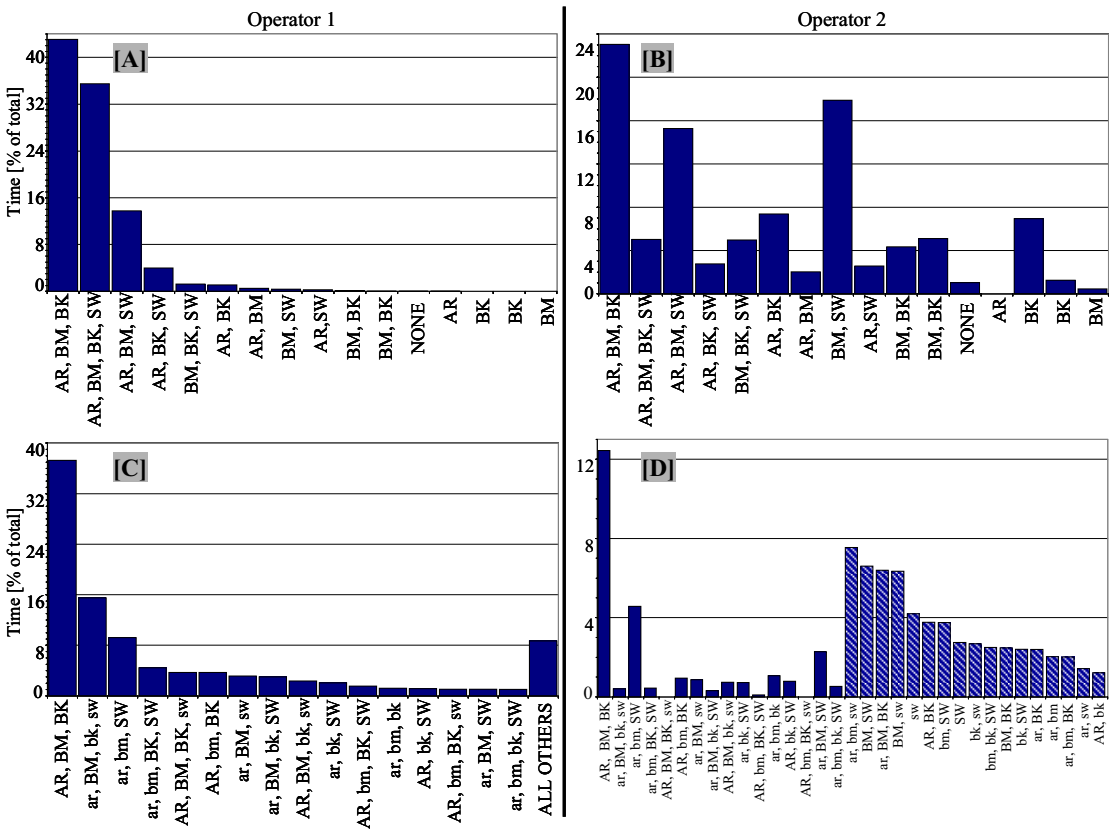


Figure 6.5: Relative lengths of actuation time for various combinations of functions used during a trenching cycle performed by two expert operators. [A], [B]: in any direction. [C], [D]: the symbol case denotes direction (e.g., AR = *extend arm cylinder* and ar = *retract arm cylinder*). The 16 most common Operator 1 patterns are shown in [C], while [D] shows the same 16 patterns for Operator 2. The hashed boxes in [D] are the next 16 most common patterns for Operator 2; note that *neither* of these appear in the top 16 for Operator 1.

motion to previously seen motions. However, this approach is not compatible with tasks that can be completed in many different ways. In particular, Figure 6.5 showed that a simple trenching task can be completed with vastly different operator input styles. In the next chapter, a simple and effective technique to map operator inputs to manipulator tasks is introduced. This method will require no prior knowledge of the task or the operator style, and can adapt to variations in the task parameters over time.

CHAPTER VII

IDENTIFICATION OF INTENDED OPERATOR TASK: ESTIMATING THE DISPLACEMENT OF MOTION PRIMITIVES

This chapter discusses the identification of the intended operator task. The first section presents a low-overhead method to describe point-to-point motions of general manipulators. The motion of the actuators from one point to another will be described by *motion primitives*. Any general manipulator trajectory will have a unique representation as a sequence of motion primitives. The motion primitives will have a *direction*, Ω, and a *length*, x. Ω indicates the direction each actuator moves and x indicates the relative actuator displacement in that direction.

The remaining sections describe a means of estimating the length of a motion primitive—and, hence, the future motion of the manipulator—given the primitive category and the manipulator state. In this way, the intended operator motion at any moment is estimated.

7.1 Nomenclature

n	Number of actuators
q	Vector in R^n of generalized coordinates for actuator position
x	Vector R^n of normalized position coordinates. x measures the remaining displacement until the actuators reach the origin.
$\hat{x}$	Estimated duration of a motion primitive
Ω	The "direction" of a motion primitive
P_x^{Ω}	A motion primitive having displacement x and direction Ω
$(\cdot)_k$	Indicates that variable $(\cdot)$ has been encountered k previous times since the manipulator began its motions. $(\cdot)_k$ *does not* indicate the kth component of $(\cdot)$.
y_k	Measurement or observation vector at start of primitive, $y_k = [q_k, \xi_k]$
q_k	Value of the coordinate q at the *start* of occurrence number k

ξ_k	An additional set of observations made at the beginning of a primitive
x_k	The expected displacement of the kth occurrence of the primitive
$h_j(y_k)$	Regressor function for estimating component j of x
$H(y_k)$	Block diagonal matrix of regressors
Θ_k	Vector of unknown modeling coefficients
R	Noise covariance matrix
JS	Joystick; the operator interface device
RLS	Recursive least squares
SVD	Singular value decomposition
rampe	Recursive Algorithm for Motion Primitive Estimation

7.2 Representing Point-to-point Motions with Motion Primitives

This section presents a method to represent actuator motion in a compact way. This research considers the class of quasi-repetitive, point-to-point type motions in an obstacle free environment; estimating motions of an arbitrary trajectory is outside the scope.

Consider the point-to-point motion illustrated in Figure 7.1. This motion is similar to the motion of the boom and swing actuators during a common trenching cycle. The manipulator coordinates $q_1(t)$ and $q_2(t)$ are shown in Figure 7.2a. Observe that the motion is essentially repetitive, although there are minor variations from one cycle to another.

The formulation considered here divides the motion of the actuators into a sequence of segments called *motion primitives*. A motion primitive denotes the direction and the relative displacement of an actuator coordinate through the motion. The coordinate may be the actuator length, the joint angle, or any similar generalized position variable. Since only the relative displacement of a trajectory is saved, each motion primitive will be piecewise monotonic in the actuator coordinate. The transition between segments is defined by some trajectory-dependent criteria, for example changes in actuator velocity, changes in actuator position, or changes in valve operating mode.

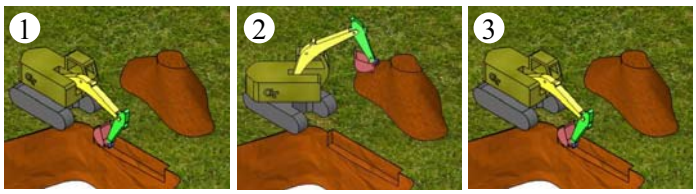


Figure 7.1: Illustration of a two-point motion sequence involving boom and swing

For example, alternating regions in Figure 7.2 indicate a segmentation based on changes in actuator velocity. The velocity is plotted in Figure 7.2b. A transition occurs at the time either event below happens:

1. Transition to Primitive Category $\Omega = 1$: Occurs when Swing velocity u_1 changes sign *from* positive *to* negative.

2. Transition to Primitive Category $\Omega = 2$: Occurs when Boom velocity u_2 changes sign *from* negative *to* positive.

The unshaded regions in Figure 7.2b denote $\Omega = 1$, while the shaded regions denote $\Omega = 2$. The category of motion indicates the direction each actuator moves, in particular:

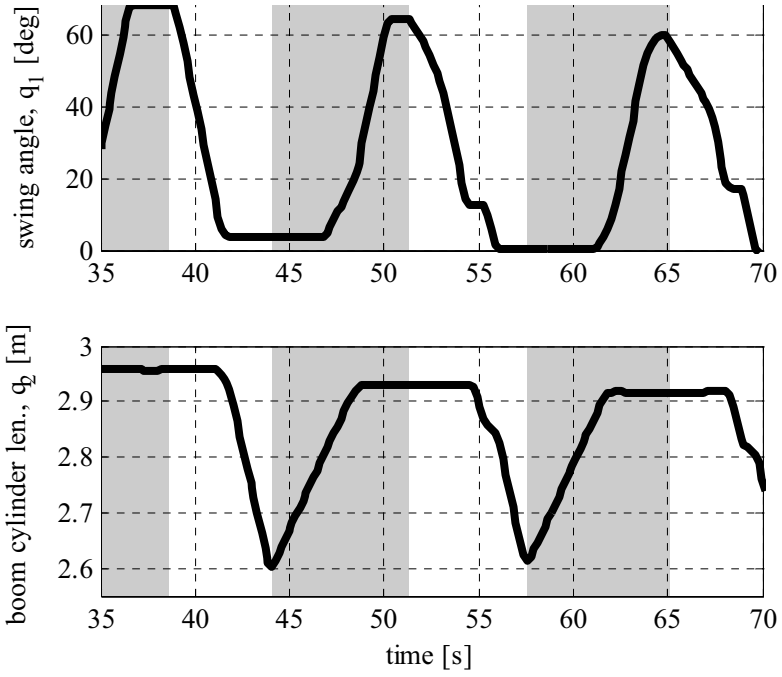
$$\Omega = 1 \longmapsto \dot{q}_1, \dot{q}_2 < 0 \quad \text{(decrease swing angle and decrease boom cylinder length)} \tag{7.1}$$

$$\Omega = 2 \longmapsto \dot{q}_1, \dot{q}_2 > 0 \quad \text{(increase swing angle and increase boom cylinder length)} \tag{7.2}$$

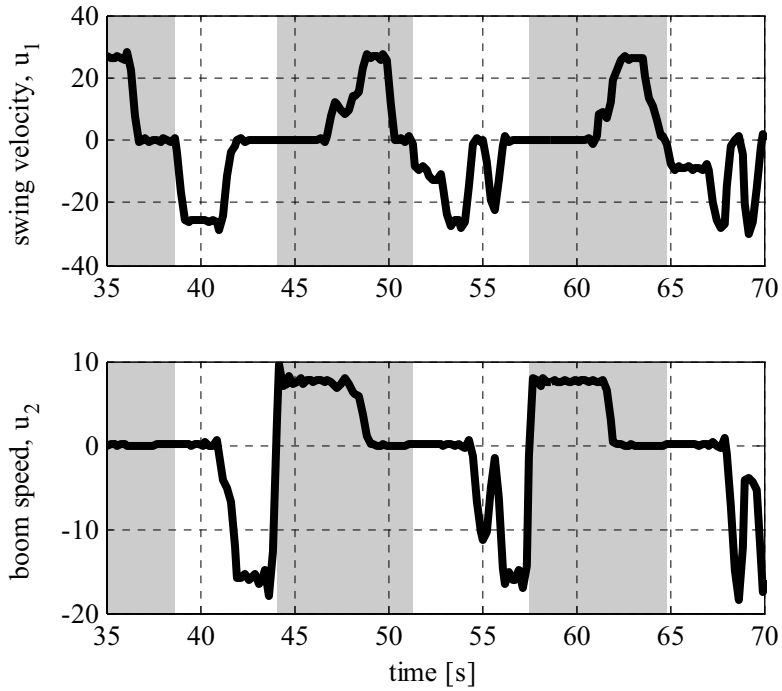
The motion primitive has a category which denotes the direction of displacement, and a magnitude which denotes the displacement of each actuator. The category is denoted Ω and the absolute displacement is denoted x. The value of x at any time t represents the remaining actuator displacement, measured in absolute value, until the end of the present motion primitive. For the example motion considered here, the relative displacement $x(t)$ is given in Figure 7.3.

P_x^Ω denotes a motion primitive; the superscript Ω indicates the category of motion, or the direction the actuators are moving. The subscripted value x indicates the relative actuator displacement.

For example, the motion considered in Figure 7.2 is decomposed into the approximate



(a) position



(b) velocity

Figure 7.2: Boom and swing coordinates for two-point motion sequence. The unshaded regions denote Primitive Category 1; the shaded regions denote Primitive Category 2.

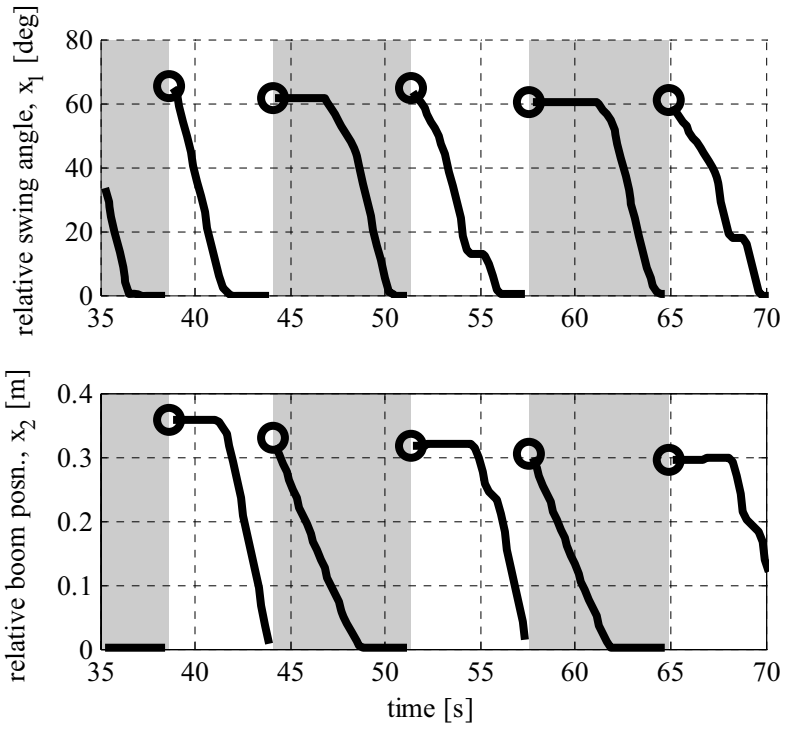


Figure 7.3: Relative actuator displacement x for the motion in Fig. 7.2. The unshaded regions denote Primitive Category 1; the shaded regions denote Primitive Category 2.

repeating sequence of motion primitives

$$\left\{P^1_{[60,0.32]}, P^2_{[60,0.32]}\right\}$$

indicating that first, the swing angle should decrease by 60 degrees and the boom cylinder length should decrease by 0.32 m; and second, the swing angle should increase by 60 degrees and the boom cylinder length should increase by 0.32 m.

Each generalized coordinate may be increasing, stationary, or decreasing so for an n-DOF manipulator there will be 3^n categories. The n-dimensional vector x represents the relative displacement of each coordinate. Each component of x is shifted such that the displacement x will always be positive and so end of the motion corresponds to $x(T) = 0$.

The primitive specifies the direction and absolute value of the change in each coordinate throughout the motion. Unfortunately, the mapping from trajectory to motion primitives is non-injective, as every manipulator trajectory defines a unique sequence of motion primitives

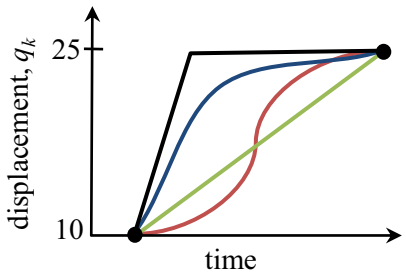


Figure 7.4: Four trajectories with the same motion primitive decomposition

but a particular sequence of motion primitives does not describe a unique trajectory, since there exist an infinite number of monotonic functions between any two endpoints. For example, eigure 7.4 shows four trajectories that have the same motion primitive. There, the motion primitive category is $\Omega = \textit{increase component } k$ and the displacement is $x = 15$ units.

The motion primitive formulation naturally has a two-part identification process: first is determining the motion primitive category, Ω; second is estimation of the expected displacement. The remaining sections of this chapter proceed in that order.

Note that the particular segmentation rules applied will depend on the specific application.

7.3 Identifying the Motion Primitive Category

The motion primitive category Ω is determined by the sensed operator velocity commands, $\bar{u}$. The sensed commands are subject to an appreciable amount of noise, due to a combination of measurement noise incident present on the input signal and input noise from the operator due to "glitches" or other unintentional joystick (JS) inputs. A "glitch" occurs as the result of an unintentional operator input. For example, an operator may inadvertently retract the arm cylinder; when the operator observes the function moving in the direction opposite of the intended motion, the operator quickly corrects the input. This glitch phenomenon is much more common in novice operators, and is a prime motivation cited by proponents of coordinated control [113]. Glitches do not represent the true intended operator task and are detrimental to the task identification process.

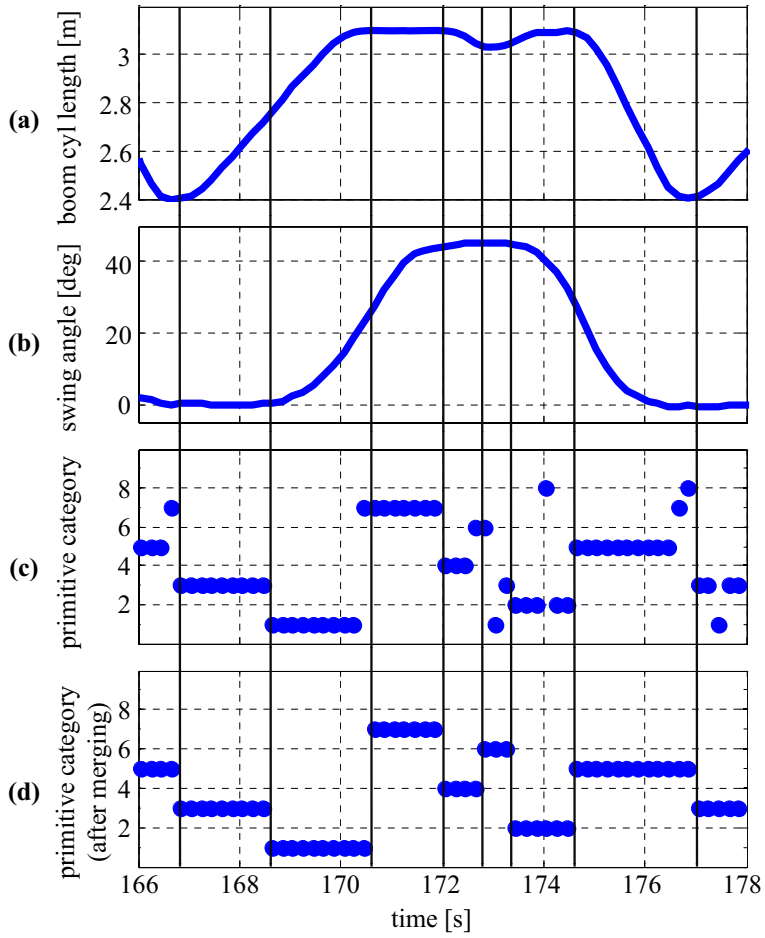


Figure 7.5: Illustration of Backward Merge Algorithm applied to actual trenching data. (a): Boom cylinder length. (b): Swing angle. (c): Initial classification of motion primitives. (d): Classifications after Backward Merge operation. The two-point, two-function cycle was performed by an expert operator on a 20-ton excavator. Note that the primitive category classification samples the operator input once per 100 ms. The solid vertical lines represent the boundaries of each motion primitive.

7.3.1 Backward Merge Algorithm

Figure 7.5 shows the boom and swing trajectory and the classification of the primitives for the two-function task performed by an expert operator. An initial classification round (Figure 7.5c) based solely on instantaneous velocity inputs exhibits considerable noise. Noise may be introduced if the operator inadvertently overshoots and compensates by reversing course, or by noise present in the measurement. One successful approach to filter unintended motions is the Backward Merge algorithm.

The Backward Merge algorithm combines two primitives from different categories when

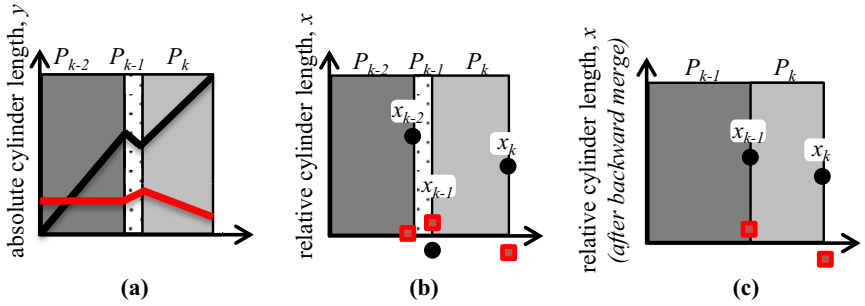


Figure 7.6: Illustration of the Backward Merge process to prevent unintentional motion primitive classifications. (a): The current motion is a P_j primitive. (b): The previous primitive P_{j-1} produced small changes x_{j-1} relative to its prior primitive. (c): The category of P_{j-1} is changed to P_{j-2}, and the origin of this primitive is shifted by x_{j-1}.

only small (and, presumably, unintentional) relative displacements in cylinder position occur between them.

(0) Suppose the primitive category has just changed from Ω_{j-1} to Ω_j.

(1) Determine the size of the displacement of the previous primitive, if it is "small" according an appropriate metric, e.g., $\|x_{j-1}\|_2 \leq k$ (where k is a tuned constant), then the previous primitive was an unintentional "glitch". To correct for this,

(2) Shift the origin of the previous primitive: $x_{j-2} = x_{j-2} + x_{j-1}$ and

(3) Re-code the category of the previous motion primitive: $\Omega_{j-1} = \Omega_{j-2}$.

Figure 7.6 illustrates this Backward Merge algorithm, showing the merging of two prior primitives if the norm of the displacement is small. Figure 7.5d is the classification resulting from application of the Backward Merge operation. Observe that the short-duration classifications around 172 s to 174 s are effectively filtered by the algorithm.

The Backward Merge algorithm can be described in the notation of primitive sequences by the operation

$$\text{BackMerge}(j) : \left\{ P^{\Omega_{j-2}}_{x_{j-2}}, P^{\Omega_{j-1}}_{x_{j-1}}, P^{\Omega_j}_{x_j} \right\} \mapsto \left\{ P^{\Omega_{j-2}}_{x_{j-2}+x_{j-1}}, P^{\Omega_j}_{x_j} \right\} \text{ if } \|x_{j-1}\|_2 < k$$

7.3.2 Boundary Entry Algorithm

Due to the motion primitive formulation, the actuator motion is piecewise monotonic. There is a bit of arbitrariness associated with how a null input is handled. The following convention

is used to specify when two primitives are merged: if a function goes to a null velocity (with the exception of a null input of all DOFs), then the category of motion will be the same as the previous category, *unless* a change in *direction* of some function has occurred. To illustrate, consider the two primitives $P_1 = P_{x_1}^{(-,BM)}$ and $P_2 = P_{x_2}^{(SW,BM)}$ in the sequence $\{P_1, P_2\}$. Suppose a single joystick actuates both the swing and the boom functions. The motion category $(-, BM)$ is achieved by moving the joystick forward, and the category (SW, BM) is achieved by moving the joystick forward and right. See Figure 7.7a. By moving the joystick from position 0 to 1 to 2 and back to 0, the manipulator follows a path outlined in Figure 7.7b. Since the transition from motion category $(--, BM)$ *into* category (SW, BM) resulted in the activation of a function (swing), hence the *Boundary Entry* rule does not apply in this case.

In contrast, consider the sequence $\{P_2, P_1\}$ with $P_1 = P_{x_1}^{(-,bm)}$ and $P_2 = P_{x_2}^{(sw,bm)}$. This sequence is realized by displacing the joystick back and left, as in Figure 7.7c. By moving the joystick from position 0 to 1 to 2 and back to 0, the manipulator follows a path outlined in Figure 7.7d. Since the transition from motion category (sw, bm) *into* category $(-, bm)$ resulted in the deactivation of a function (swing), thus in this case, the *Boundary Entry* rule applies, and the motion primitives are merged accordingly:

$$\left\{ P_{x_2}^{sw,bm}, P_{x_1}^{(-,bm)} \right\} = \left\{ P_{x_2+x_1}^{(sw,bm)} \right\}$$

The boundary entry filter has a subtle but important consequence. Namely, it allows description of actual endpoints of a point-to-point motion, even though the operator may have completed the motion imperfectly. For example, consider the displacement q_1 and q_2 of two actuators as shown in Figure 7.8. The goal is to arrive at point A in minimum time. For whatever reason, perhaps an unfamiliarity with manipulator control, the operator may command the trajectory sketched in the solid line, where actuator 2 reaches its goal state before actuator 1. Thus, there is a "dogleg" in the resulting trajectory. In absence of the boundary entry filter (i.e., defining the motion primitives strictly based on the region of the u-plane), the path consists of two motion primitives. In essence, the motion is incorrectly

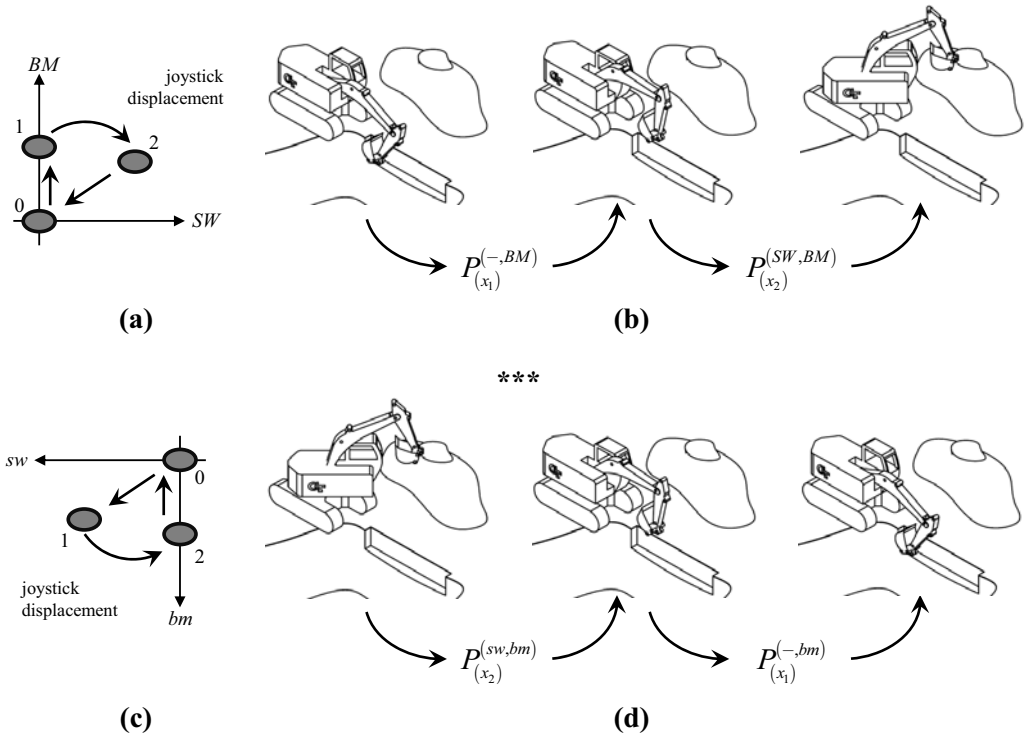


Figure 7.7: Two similar manipulator motions that illustrate the different cases of the Boundary Entry algorithm. (a),(c): Sketch of the sequence of joystick displacements for each motion. (b),(d): The manipulator positions at intermediate points along motion

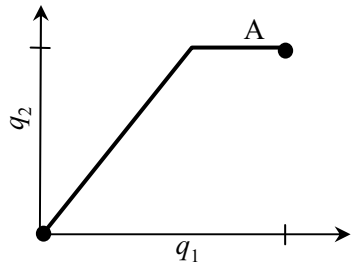


Figure 7.8: A manipulator path plotted in the q_1, q_2-plane

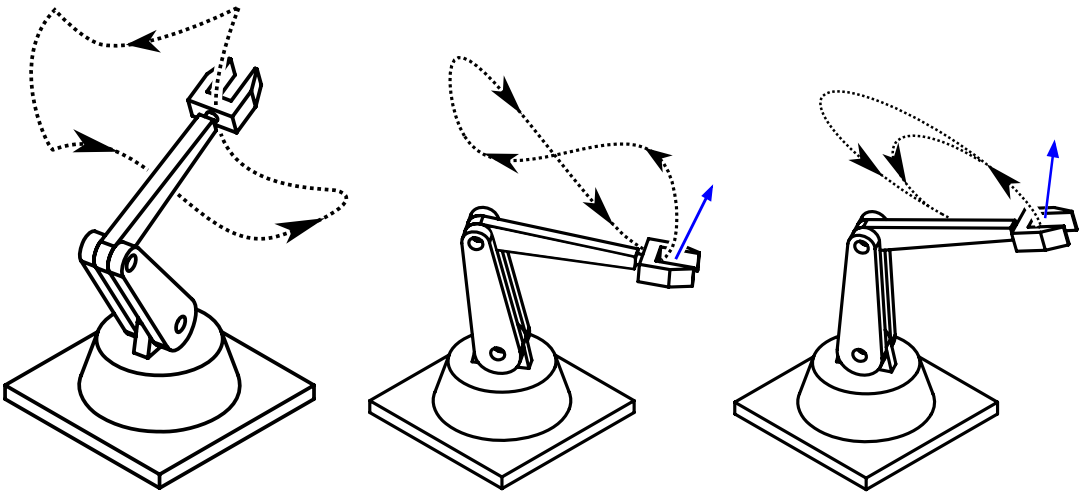
identified as two distinct paths. However, application of the boundary entry filter correctly identifies the parameters of the fundamental task. Of course, if the operator *did* intend the original motion—for example, perhaps the dogleg was to avoid a workspace obstacle—then the task would be misidentified.

7.4 Limitations on Trajectory Types

The motion primitive formulation implicitly assumes that the manipulator path is uniquely parameterized by the generalized position q and the motion category Ω. The coordinate q may be expressed in any domain, e.g., actuator lengths or joint angles, as long as there is a one-to-one correspondence between the chosen coordinate and the degrees of freedom. Paths which are not parametrized by q and Ω are not suitable for the motion primitive formulation. Any non-intersecting path, such as Figure 7.9a, is suitable for decomposition into a sequence of motion primitives. Also, any path which intersects at an oblique angle, such as Figure 7.9b, is suitable because knowledge of the motion direction at the intersection allows unique specification of the future path manipulator position. However, paths which intersect and are tangent (i.e., the velocity of the path at the intersection is point is degenerate) are *not* parametrized by q and Ω alone, such as in Figure 7.9c where it is impossible to tell which "loop" the manipulator will trace given only the position and velocity at the intersection point.

7.5 Algorithm for Motion Primitive Estimation

While the motion primitive category can be directly determined from measured and filtered operator inputs, the length or magnitude of a given primitive is unknown. The RAMPE



(a) Non-intersecting path (b) Oblique intersection (c) Tangent intersection

Figure 7.9: Example manipulator paths suitable for the Motion Primitive formulation. (a) and (b) are uniquely parametrized by generalized position q and motion category Ω, whereas (c) is not.

algorithm described next estimates the length using the machine state at the start of the primitive and the previous durations of that primitive.

This section describes the Recursive Algorithm for Motion Primitive Estimation—named RAMPE—for estimating the displacement of a motion primitive. The problem can be summarized by

> ***Problem Statement:*** At the beginning of the kth occurrence of a primitive within category Ω, e.g., $P_{x_k}^{\Omega_k}$, find the expected duration, $x_k \in R^n$, given the observation $y_k, y_{k-1}, \cdots, y_1$ with $y_k \in R^{n+m}$.

For convenience, the symbols (x, Θ, P, etc.) are used for all categories of motion, and it is understood that for *each* category of motion there will be stored in memory the full set of variables necessary for the RAMPE algorithm.

The displacement x_k depends on the manipulator path. Subscript k denotes that the motion primitive category has been encountered k previous times during the dig cycle. Note the subscript k denotes the number of times the particular primitive was previously encountered during the motion; in this context, k does not refer to the kth component of the vector. The manipulator has n degrees-of-freedom, so x_k is a real vector with n components.

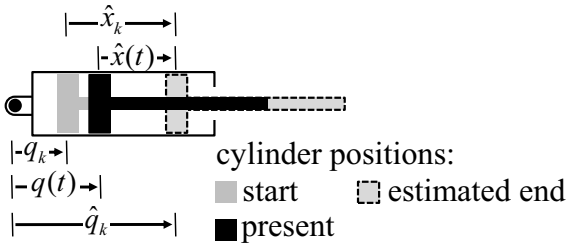


Figure 7.10: Definition of actuator variables for describing the absolute and relative actuator position. Subscript k denotes a measurement at the start of the kth occurrence of the motion primitive; a *hat* denotes estimated values.

In reality, the displacement is some unknown parametric function of the intended operator path. For estimation purposes, the displacement x_k is assumed to be well approximated by the unknown function

$$x_k = f(\Omega, y_k) \tag{7.3}$$

where Ω is the primitive category and $y_k = [q_k, \xi_k] \in R^{m+n}$ is an observation vector taken at the start of the motion primitive. The observation always consists of the generalized coordinates $q_k \in R^n$ which describe the initial manipulator configuration, and the optional additional measurements $\xi_k \in R^m$ which may include additional measures such as the total commanded flow rate. Figure 7.10 gives the convention for defining the cylinder stroke. The model f ignores any direct time dependence, although the additional measure ξ can include such time-dependent terms including k or t.

The unknown model f in (7.3) is expanded as a linear function of the observation y_k and unknown parameters θ_{ij}

$$x_k = H(y_k)\Theta \tag{7.4}$$

where H is a block diagonal regressor matrix defined as

$$H(y_k) = \begin{bmatrix} h_1(y_k) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & h_n(y_k) \end{bmatrix} \tag{7.5}$$

The n elements $h_j(y_k)$ are row vectors used as regressors to estimate the jth component of

x_k. In general, each regressor h_j can have a different structure and order for each component of x_k. For example, the first component of x may be found to have no dependence on q_3, so that component may be left out of the regressors h to save computation and storage cost.

The vector Θ is a vector of unknown parameters θ_j:

$$\Theta = \begin{bmatrix} \theta_1 \\ \vdots \\ \theta_n \end{bmatrix}$$

where each θ_j is itself a vector of unknown parameters θ_{ij} for estimating the jth component of x_k. The number of elements in θ_j is equal to the dimension of h_j.

An estimate for the length of occurrence k of the motion primitive is formed by computing the model (7.4):

$$\hat{x}_k = H(y_k)\Theta_{k-1} \tag{7.6}$$

where $\hat{x}_k$ is the estimated duration. The model parameters in Θ_{k-1} are chosen so that the model (7.4) fits the *observed* duration of the previous $k-1$ occurrences, in the least squares sense. The model parameters Θ_{k-1} are updated recursively before a new estimate is computed, using the update law [139]

$$\Theta_{k-1} = \Theta_{k-2} + K_{k-1}\left(x_{k-1} - H(y_{k-1})\Theta_{k-2}\right) \tag{7.7}$$

with

$$K_{k-1} = P_{k-1}^{-1} H(y_{k-1})^T G \tag{7.8}$$

The matrix P_{k-1} is updated recursively as

$$P_{k-1} = P_{k-2} + H(y_{k-1})^T G H(y_{k-1}) \tag{7.9}$$

Matrix G is the inverse of the noise covariance matrix R

$$G^{-1} = R = \begin{bmatrix} \sigma_1 & \cdots & \sigma_{1n} \\ \vdots & \ddots & \vdots \\ \sigma_{n1} & \cdots & \sigma_n \end{bmatrix} \tag{7.10}$$

where R is assumed constant, but is in reality a function of the state q. For instance, Figure 6.2 indicates the final bucket position over the trench ($q_1 \approx 0$) has lower variance than the final positioning over the spoil pile. One key role of R is to scale the elements of the vector x, as each element may have very units of different magnitudes (i.e., *degrees* versus *meter*)

The matrix P_{k-1} may be singular or close to singular due to the regressors in $H(y_{k-1})$ being nearly linearly dependent. In this case, the inverse $(P_{k-1})^{-1}$ required in computing (7.8) may be numerically ill-conditioned (if P_{k-1} close to singular), or not unique all-together if H is rank-deficient.

As a remedy, $(P_{k-1})^{-1}$ is computed using the singular value decomposition (SVD) of P_{k-1}. First, P_{k-1} is expanded as

$$P_{k-1} = U\Sigma V^T$$

where U and V are sets of orthonormal basis vectors and Σ is the diagonal matrix of singular values. Next, the diagonal pseudoinverse Σ^{-1} of the matrix of singular values Σ is calculated by transposing the matrix obtained after inverting each element along the diagonal of Σ. When the matrix P_{k-1} is nearly singular, some of the elements in Σ will be close to zero. Only those diagonal elements larger than a given tolerance *tol* will be inverted; the other elements are set to zero. Finally, the pseudoinverse of P_{k-1} is computed as

$$P_{k-1}^{-1} = V\Sigma^{-1}U^T \tag{7.11}$$

7.5.1 Initialization

The parameters in Θ must be initialized before computing an estimate. This is done by waiting until the primitive Ω is encountered N_{init} times before making the first prediction.

For $k = 1, \cdots, N_{\text{init}}$, no predictions for x_k are made. Rather, the data is stored to form an estimate x_k beginning with $k = N_{\text{init}} + 1$.

At the initialization step, a total of N_{init} measurements of x, each with a corresponding observation y, are accumulated. A large regressor matrix H_0 is formed by placing the submatrices $h_{j,\text{init}}$ along its main diagonal, where

$$h_{j,\text{init}} = \left[h_j(y_1)^T, h_j(y_2)^T, \cdots, h_j(y_{N_i nit}))^T\right]^T$$

Similarly, the vector x_{init} is formed by "unwrapping" the previous measurements to produce

$$x_{\text{init}} = [x_{1,1}, \cdots, x_{1,N_{\text{init}}}, x_{2,1}, \cdots, x_{2,N_{\text{init}}}, \cdots, x_{n,1}, x_{n,N_{\text{init}}}]^T$$

where $x_{j,k}$ is component j of the kth sample of x_k. The initial estimate of the parameter $\Theta_{N_{\text{init}}-1}$ is obtained by solving the system of equations in (7.6).

SVD is used to solve the system (7.4) by direct computation using

$$\Theta_{N_{\text{init}}-1} = Vr \tag{7.12}$$

where r is the solution to the diagonal system

$$\Sigma r = U^T G_0 H_0 x_{\text{init}}$$

and U, Σ, V are from the singular value decomposition of matrix $[G_0 H_0]$. As before, inversion of r is carried out by setting to zero the components of r corresponding to the diagonal elements of Σ which are below a specified threshold. G_0 is the inverse of the initial covariance matrix R_0, where R_0 is a padded version of (7.10) and H_0 is the initial set of regressor matrices, defined earlier. The initial matrix $P_{N_{\text{init}}-1}$ for use in (7.9) is

$$P_{k-2} = H_0^T G_0 H_0$$

To compute the initial estimate, first compute $\Theta_{N_{\text{init}}-1}$ using (7.12), then compute $\hat{x}_k$

using (7.6).

A flow chart for calculating the estimate $\hat{x}_k$ is shown in Figure 7.11. A description of each of the modules is provided in Table 7.1.

7.5.2 Engineering Tests with the rampe Algorithm

A simple experiment with a 2 DOF kinematic excavator was performed to benchmark the accuracy at which the RLS algorithm estimates the primitive displacement x_k. The regression model used assumes all components of x are linearly dependent on q, as in

$$h_j(q_k) = \left[q_k^T, 1\right] \tag{7.13}$$

Each vector of unknown parameters, θ_j, has 3 components in this case.

The operator moved the bucket through a cycle defined by the sequence of targets shown in Figure 7.12. The path follows the sequence

$$A \to B \to C' \to C \to C' \to B \to A \to \cdots$$

and then repeats. The coordinates, q, which define the locations of each target in actuator space, and the minimum displacement, x, between targets is summarized by Table 7.2.

The generalized coordinates q for the swing and boom during a portion of the cycles are shown in Figure 7.13. A total of 23 cycles was completed and used for studying the identification problem, the results of which are described next.

Only the swing and boom functions are active. As discussed earlier, the motion can be described by a displacement x in the direction of primitive category Ω, which, for $n = 2$, can assume nine possible values as defined in Figure 7.14.

The operator inputs are processed with the Boundary Entry and Backward Merge algorithms as discussed earlier. The trajectory in Figure 7.15 plotted in the q-plane has been coded at each time instance to reflect the instantaneous motion primitive category Ω. The purpose of Figure 7.15 is to show: the human operator is not precise in commanding the motion, as evidenced by the general "width" of the plot; some motion primitives only occur

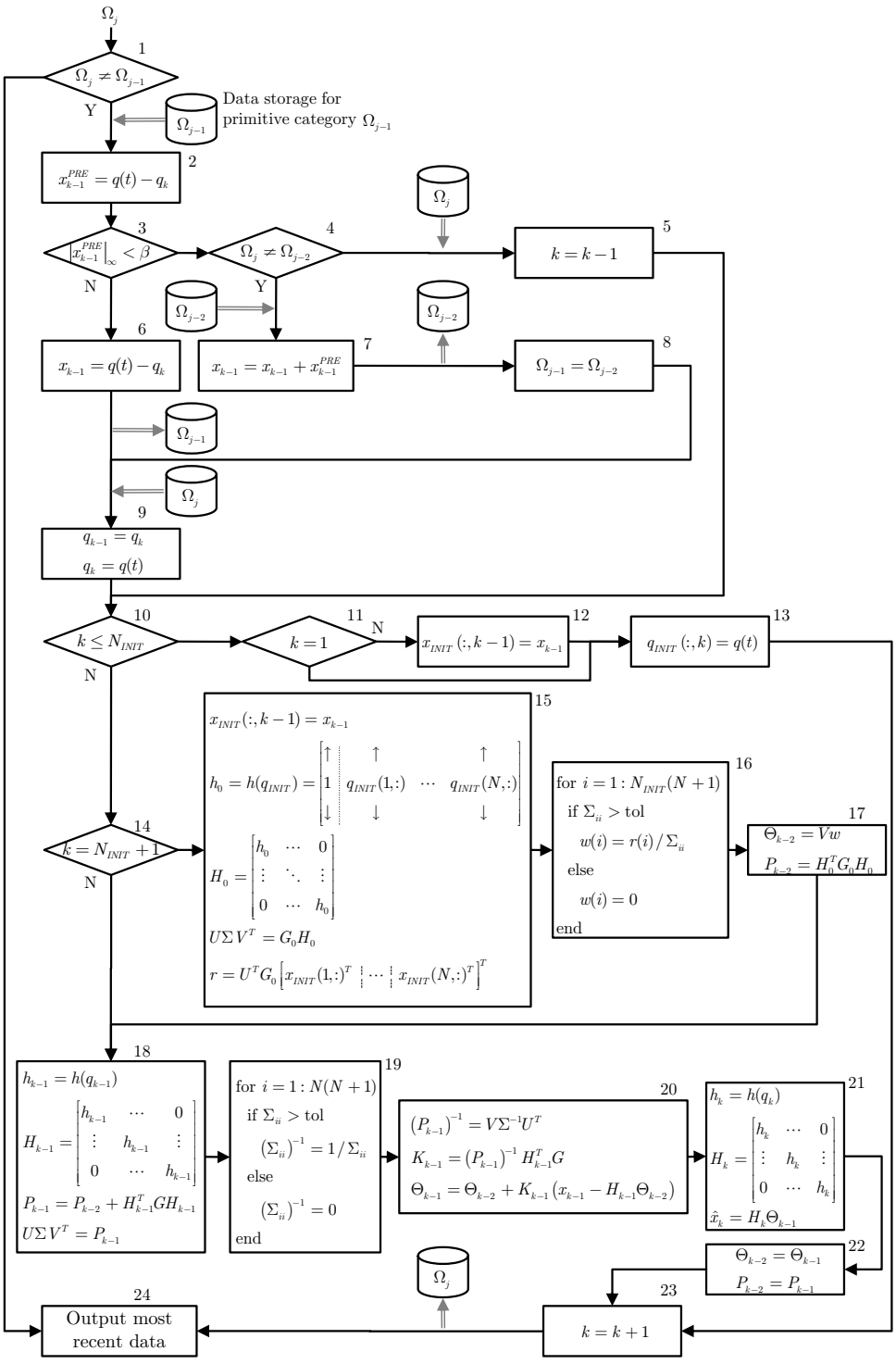


Figure 7.11: Flowchart for the RAMPE algorithm used in estimating the expected duration $\hat{x_k}$ of the present primitive Ω_j, using recursion. A description of each block is provided in Table 7.1. The double lines denote the transfers into or out of memory for all data (x_k, q_k, Θ, etc.) of a particular Ω.

Table 7.1: Description of the flow chart blocks in Figure 7.11

Description of flow chart elements in Figure 7.11

1. Check if the primitive category has changed.

2. Save the length of the primitive that just ended as x_{k-1}^{PRE}.

3. Test if the previous primitive needs to be merged with its prior.

4. If the current primitive Ω_j is of the same category as the one twice before (Ω_{j-2}), then the last primitive (Ω_{j-1}) was merely a "glitch".

5. Since the primitive that occurred prior to the "glitch" was only interrupted by the glitch, the occurrence count for that category was wrongly incremented. Thus, the count is decremented.

6. The length of primitive Ω_{j-1} is calculated as $x_{k-1} = q(t) - q_k$, where $q(t)$ is the value of the current generalized coordinate, and q_k is the (recorded) starting value of primitive Ω_{j-1}. This updated information for Ω_{j-1} is saved.

7. The data for primitive Ω_{j-2} is loaded. The end point is adjusted by the length of the (short) duration of Ω_{j-1}. The data for primitive Ω_{j-2} is resaved.

8. Since the previous primitive was combined with its prior through the *Backward Merged* process, the category of the previous primitive must be updated.

9. Load data for primitive Ω_j and update the starting point of the previous iteration ($q_{k-1} = q_k$) and the current iteration ($q_k = q(t)$).

10. Check if this primitive category is still in the learning phase.

11. Check if this is the first time primitive Ω_j is encountered.

12. Append the initialization data for the primitive length x.

13. Append the starting point q with the previous cycles.

14. Check if this will be the first estimate of $\hat{x}$

15. Prepare variables to use Least Squares via SVD to compute the initial estimate.

16. Compute the projection vector w for the Least Squares estimate by computing the (reduced) pseudo-inverse the singular values matrix.

17. Compute the least squares estimate of the parameter vector Θ and the matrix P to be used in computing the initial estimate of $\hat{x}_k$.

18. Compute the necessary terms to update the recursion relationship for RLS estimation. This includes calculating the SVD of P_{k-1}.

19. To invert P_{k-1}, the reduced pseudo-inverse of the matrix of singular values Σ is used.

20. Update the variables for the RLS estimation of x_k.

21. Compute the estimate of the primitive length, $\hat{x}_k$ using the present generalized coordinates q_k as a basis, and the most up-to-date parameters Θ_{k-1}.

22. Store the variables for use during the next iteration.

23. Update the number of times primitive category Ω_j has been encountered. Push all data for Ω_j back to memory.

24. Output most recent estimates, including $\hat{x}_k$.

Table 7.2: Description of 2 DOF Cycle

	Generalized coordinate, q		Displacement x to next target	
Target	SW [deg]	BM [m]	SW [deg]	BM [m]
A	64.4	2.79	-41.7	0.16
B	22.7	2.95	-22.7	-0.16
C'	0	2.79	0	-0.19
C	0	2.60	0	0.19

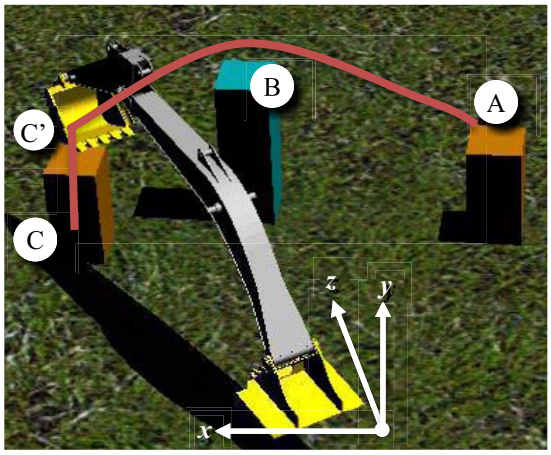


Figure 7.12: A trajectory that mimics a two-DOF dig cycle is used to demonstrate the RAMPE formulation of task identification.

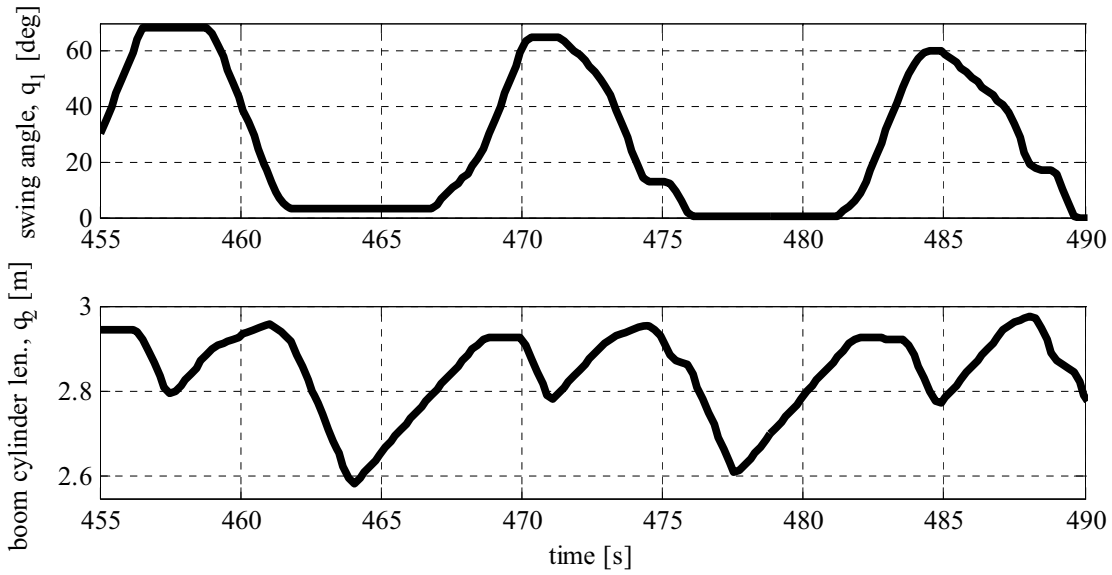


Figure 7.13: Swing and Boom coordinates, q, during a portion of the 2-DOF test cycle

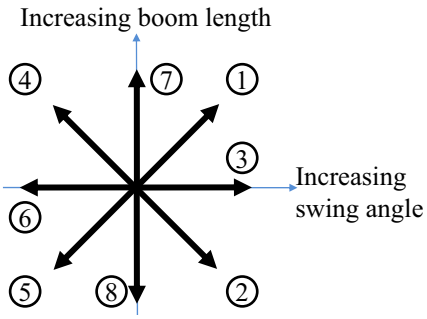


Figure 7.14: The nine categories of motion primitives for two degrees of freedom; circled numbers indicate the category of motion, Ω.

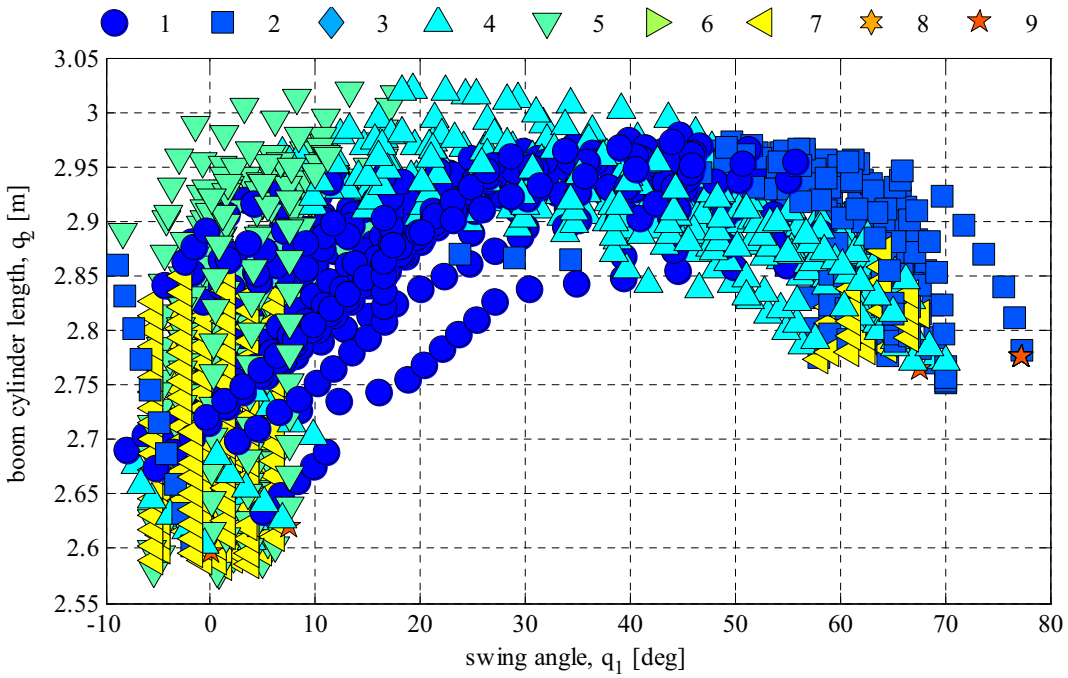


Figure 7.15: Trajectory of an example motion, plotted in the q-plane. The markers denoting the instantaneous motion primitive category, Ω. Each symbol denotes a particular primitive category, as determined by the legend at the top of the frame.

in one region of the q-plane, while others occur at multiple regions.

The RLS algorithms described above were used to estimate the duration $\hat{x}_k$ of a motion primitive when a new category of motion is encountered. Figure 7.16 illustrates the performance of these methods. Only results for the five most encountered primitives are shown. Green circles represent the state q_k at the beginning of the motion primitive; cyan triangles mark the estimated end point, $\hat{x}_k + q_k$; red squares mark the actual end point of the motion primitive. Three iterations were completed during the initialization phase (since $N_{\text{init}} = 3$); the estimates during the learning phase were simply set to $\hat{x}_k = 0$ for $k = 1, \cdots, N_{\text{init}}$. Notice that for $\Omega = 1$, $\Omega = 4$, and $\Omega = 7$ there are two distinct clusters from which the motion primitive begins. The displacement x for each of these starting points is different. By defining a structure such that regressors depend on the initial state q as in (7.13), the functional dependence of x_k on q_k can be captured.

The estimation error $E = x_k - \hat{x}_k$ is large during the initialization phase. The error generally improves by increasing the number occurrences a particular primitive is encountered,

as shown in Figure 7.17. Since the process is controlled manually by a human operator, it is reasonable to expect the error to never settle to zero (as a perfect estimate would require exact knowledge of the future operator input). Generally, the error in estimating the boom cylinder displacement is within 15 percent of the full stroke of the cylinder (E_1 is typically less than 0.2 m, and the maximum boom cylinder displacement is 1.4 m).

7.6 Summary

The motion primitive formulation can describe large-scale manipulator motions in the workspace as a sequence of shorter actuator displacements. The motion primitives are delineated based on a piecewise monotonic decomposition of the original actuator trajectory. Only the displacement of the actuator across each primitive is saved, thus simplifying the description of larger actuator trajectory. Further, the motion is described by relatively short-duration motion primitives. This may prevent large errors in position estimates from compounding, as the error in relative actuator displacement is reset whenever a new motion primitive is encountered.

Any particular motion primitive can describe an infinite number of possible trajectories. Thus, this formulation is not appropriate for describing the temporal dependence of the actuator motion or for imposing constraints on the actuator motion, e.g., to describe particular regions in the actuator space to avoid in case of the presence of obstacles.

The motion primitive formulation is well suited for describing motions within an obstacle free environment. Further, the motion is conveniently decoupled into two elements: the motion primitive category, Ω, and the relative actuator displacement, x. The category Ω is determined by sensing and filtering the operator input commands, while the expected displacement, $\hat{x}$, is estimated using previous observations in a recursive algorithm.

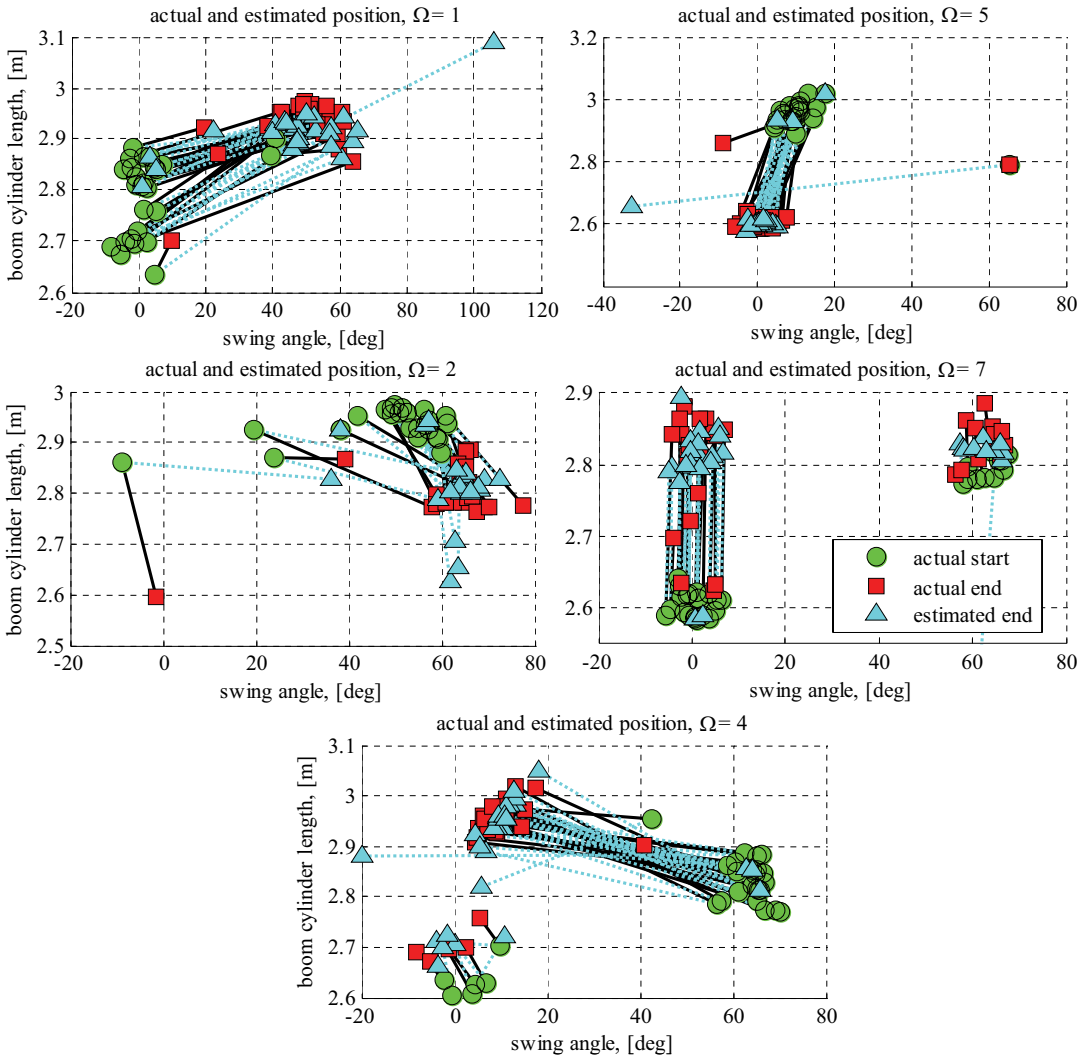


Figure 7.16: Estimating the end state at the onset of a new motion primitive. Green circles represent the state q_k at the beginning of the motion primitive; cyan triangles mark the estimated end point, $\hat{x} + q_k$; red squares mark the actual end point of the motion primitive. The solid black and dashed cyan lines link the start and end points for a particular cycle iteration k.

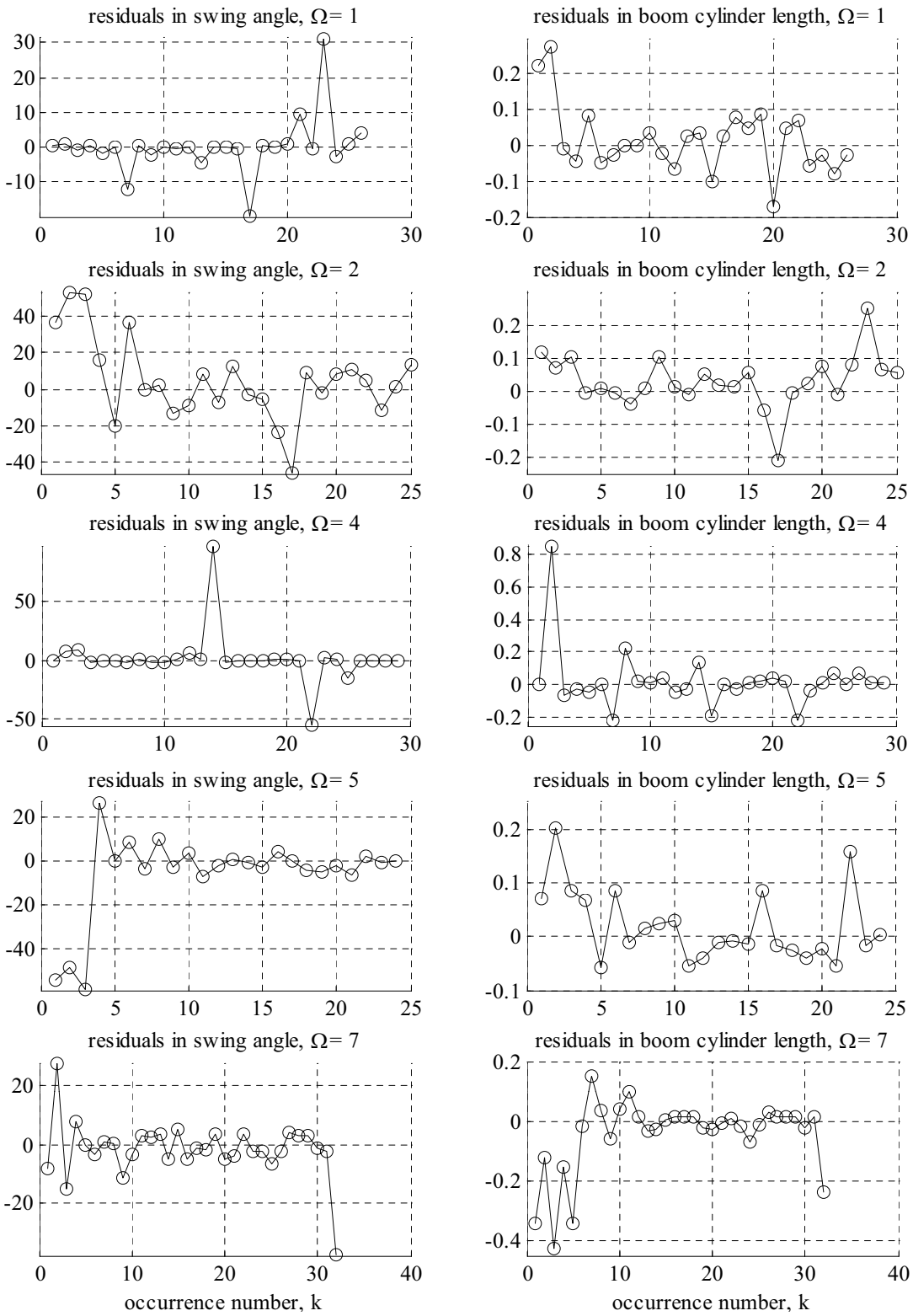


Figure 7.17: Residual error $E = \hat{x}_k - x_k$ for the example trajectory. The abscissa marks the number of prior occurrences of the primitive category Ω. Units for swing angle and boom cylinder length are degrees and meters.

CHAPTER VIII

DYNAMIC MODELS OF THE HYDRAULIC MANIPULATOR SYSTEM

In this chapter, two machine models are discussed. A dynamic model of the excavator system was used to simulate the response of the machine during human factors testing. The model used for simulation is relatively high-fidelity but remains capable of running in real time. However, this model is unsuitable for use in the task optimization problem. For this, a quasi-dynamic model was developed to approximate the dynamics and salient motion constraints over moderate time periods of individual motion primitives.

The testbed was a Bobcat 435 compact excavator, pictured in Figure 8.1. The excavator model described in this chapter was based on the stock machine characteristics, with the exception of the hydraulic system which was been modeled after a hypothetical independent metering control system.

8.1 Nomenclature

n	Number of actuators
$q, \dot{q}$	Generalized position and velocity of actuators in cylinder space
$\theta, \dot{\theta}$	Joint space angle and velocity
$y, \dot{y}$	Task space coordinates of end effector position and velocity
F	External force acting on actuator
L_p	Differential pressure load applied by actuator ($F = A_b L_p$)
P_s, P_r	Supply and return pressures, respectively
P_a, P_b	Pressure on head and rod side of actuator, respectively
A_a, A_b	Effective cylinder areas of the actuator
Q	Flow rate of hydraulic fluid
$K, K_{\mathbf{cmd}}$	Actual and commanded valve flow conductance

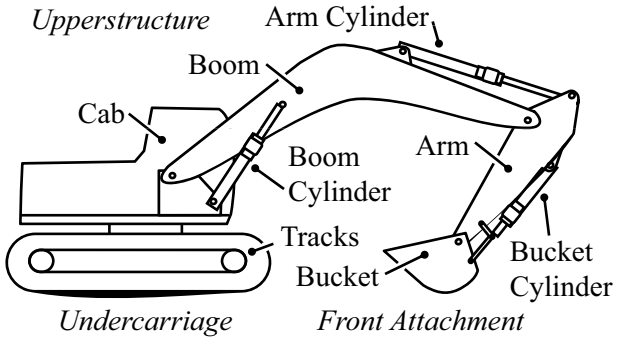
Figure 8.1: The Bobcat 435 compact excavator research testbed *(photo courtesy of Mark Elton)*

K_{eq}, P_{eq}	Equivalent flow and pressure for the particular valve operating mode
D_k	Maximum speed of actuators, for the particular operating mode
ψ	Flow-velocity ratio the actuator velocity to the flow into the valve
u	Nominal actuator velocity
$\bar{u}$	Commanded actuator velocity
r_k	Normalized joystick displacement, in $[-1, 1]$, for function k
$\hat{R}_k(r_k)$	Function mapping joystick command to fraction of maximum actuator speed
U	Domain of achievable actuator velocities

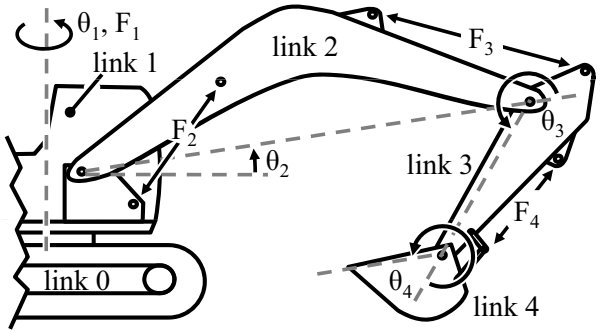
8.2 Overview of Excavator Dynamic Model

The hydraulic excavator consists of two primary subsystems: the *linkages* and the *actuators*. The swing, boom, arm, and bucket functions comprise the linkage system in a typical excavator, as in Figure 8.2b. Dynamic models of fluid power systems are common in literature, but vary in degrees of complexity and fidelity. The model used for this research is described in this chapter.

A block diagram in Figure 8.3 shows the interactions of the major components in the excavator dynamic model. Each block is briefly described in Table 8.1. The next section gives further details about the core elements of the model.



(a) Major components comprising the excavator linkage system



(b) Definition of forces and joint angles

Figure 8.2: Components and notation for excavator linkage modeling

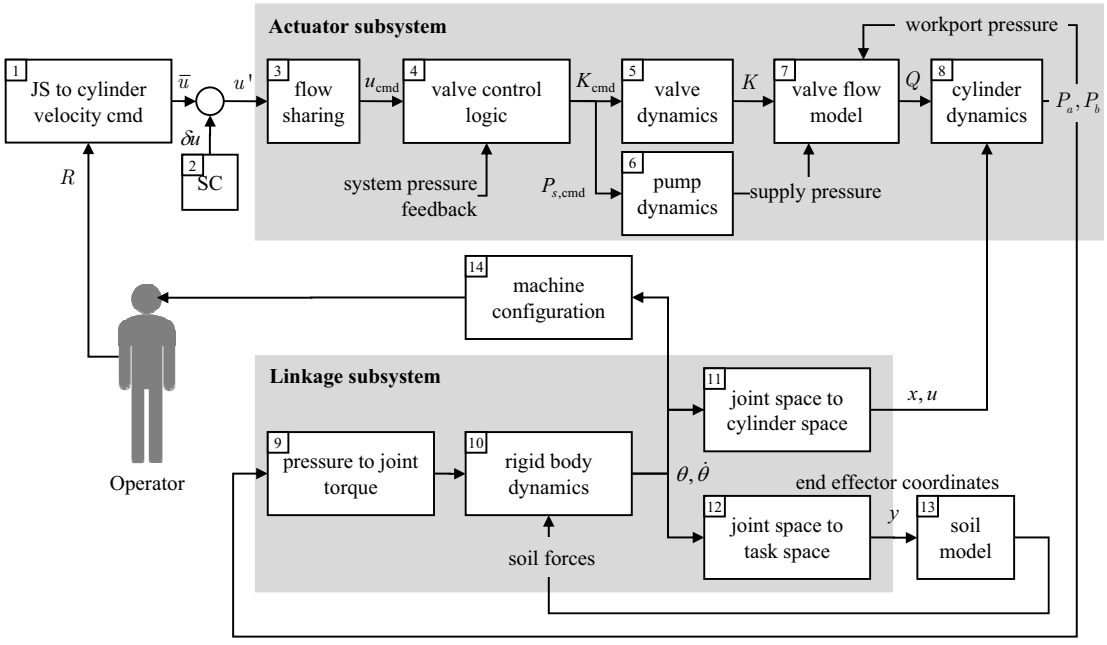


Figure 8.3: Major components of the excavator dynamic model

Table 8.1: Description of the system flow chart blocks in Figure 8.3

Description of flow chart elements in Figure 8.3
1. The angular displacement R of the four joysticks is converted into a nominal velocity command $\bar{u}$
2. A shared control algorithm (discussed in Chapter 10) perturbs the operator input by an amount δu
3. The velocity command may exceed the machine capabilities, due to pump and valve flow limits. Infeasible velocity commands must be projected into the feasible region.
4. A control system calculates the required system pump pressure $P_{s,\text{cmd}}$ and the orifice flow conductance K_{cmd} for each orifice in each valve for all functions. The valve openings are a function of velocity command u_{cmd} and the sensed system pressures. This logic may, for example, be implemented by hydro-mechanical logic (e.g., in pressure-compensated load-sense architectures), or, in the case of this system, a proprietary control algorithm.
5. The electro-hydraulic poppet valves have a finite response time that must be accounted for.
6. The pump dynamics include the pump itself (consisting of leakage and first order swash plate dynamics), as well as the electronic pump pressure controller.
7. Given the incident system pressures and the valve openings, the flow through the valve is computed using standard orifice models.
8. The pressure within the cylinders (and swing motor) is computed by integrating the pressure dynamic equations. The model includes lumped parameters for conduit volume, as well as seal friction.
9. Computing joint torque from cylinder pressure (and motor torque) requires knowledge of the inverse kinematic mapping between *cylinder space* and *joint space*.
10. The linkage response is a function of the torques acting on the links. Lagrangian methods are used to compute the angular acceleration, which is integrated to obtain the position and velocity of the joints. If any cylinder is near the end of stroke, the model sets the corresponding velocity and acceleration to zero.
11. The response θ and $\dot{\theta}$ is converted from *joint space* to *cylinder space* (x and u) using a forward kinematic mapping.
12. The bucket position relative to the ground is computed using a forward kinematic model.
13. Soil reaction forces are computed based on the current soil state, and the speed and velocity of the bucket.
14. The operator observes the machine configuration and adjusts the velocity commands to achieve the desired response.

8.3 Dynamics of the Linkage System

Many resources are available for modeling multi-domain systems such as excavators [21, 22]. In the Intelligent Machine Dynamics Laboratory at Georgia Tech, several researchers have contributed to a C++ based package containing general-purpose models of common linkage systems actuated by hydraulic cylinders [7, 35, 49]. The models are usable for a broad class of kinematically similar machines including hydraulic manipulators, backhoes, and excavators. The package includes models for the equations of motion of the linkages, as well as the forward and inverse kinematics relating variables within three frames: the *cylinder space* (lengths of the hydraulic actuators, given symbol q), the *joint space* (generalized angle of each rigid body link, with symbol θ), and the *task space* (the end effector position and orientation, given symbol y). To use this software library, the model parameters need only to be set up to match the manufacturer's CAD specifications. Since the models have been presented elsewhere, and because the proprietary nature of the manufacturer data, only a brief overview of the linkage dynamics is given in this chapter.

The linkage response to an applied torque is expressed in closed form as

$$\ddot{\theta} = [M(\theta)]^{-1}\left(T - C\left(\theta, \dot{\theta}\right)\right) \tag{8.1}$$

where $\theta = [\theta_1, \theta_2, \theta_3, \theta_4]^T$ are the linkage joint angles as in Figure 8.2 $[M(\theta)]$ is the inertia matrix (including gravitational terms); T is the vector of applied torque in joint space; and C is the vector including torques due to friction, applied external loads such as ground forces, and Coriolis effects. Figure 8.2b defines the convention for describing the joint angles θ and the actuator forces F. Equation (8.1) has an identical form when applied to a wide class of machines ranging from 10-ton compact excavators to 40-ton full-size excavators. The joint torque T is a function of the generalized actuator forces $F = [F_1, F_2, F_3, F_4]$, the θ-dependent moment arm of the linkage (or, for the swing function, the effective gear ratio between motor and joint), and the joint friction.

When the joint angles are within allowed limits (i.e., neither of the cylinders is at its stops), the equations of motion (8.1) are integrated as usual. If either of the actuators are

at its end of stroke, then an unknown reaction force acts on the system and the equations of motion (8.1) are invalid. In this case, the angular velocity $\dot{\theta}$ and acceleration $\ddot{\theta}$ are set to zero to represent the effects of the hard stops while avoiding the stiffening effects often associated with integrating events having short timescales.

8.4 Dynamics of the Actuator System

An engine powers a drive train and one or more pumps. The actuator system uses pressurized oil to transfer power to the linkages using an arrangement of pumps, valves, conduits, and actuators such as hydraulic cylinders and motors. While the linkage systems of many hydraulic manipulators are kinematically and dynamically similar, the actuator systems are markedly dissimilar. Consider the hydraulic architecture described in Figure 8.4 having a single system pump and a simple parallel arrangement of valves. Actual hydraulic circuits are more complex than the one illustrated, and typically may include additional valves, energy storage devices, additional interconnections between functions, and circuitry for pilot and charge flow. Many larger excavators also have more than one system pump.

A stock Bobcat 435 excavator uses conventional spool valves to distribute the flow. In contrast, the excavator studied here utilizes a hypothetical independent metering control (IMC) architecture. Both the valve arrangement and high-level valve control logic are based on a technology called INCOVA which was developed by HUSCO Int'l. The specific architecture of the hydraulics is mostly inconsequential and the analysis and results presented elsewhere in this thesis can be applied to other hydraulic architectures as well.

In short, the INCOVA Valve Controller reads a velocity reference u and computes the necessary valve control signals to cause the hydraulic motor and cylinders to exert forces on the linkage system in order for the functions to respond as dictated by the reference u. The following subsections will describe models for the valve controller, the valves, the pump, and the cylinder.

8.4.1 The Hydraulic Cylinder and Motor

Hydraulic actuators include motors (*rotational actuators*) and cylinders (*linear actuators*). The static and dynamic behavior of these elements are similar (both convert a pressure

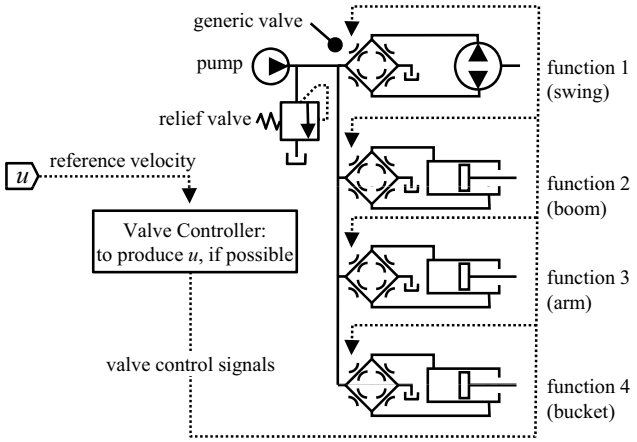


Figure 8.4: The hydraulic system architecture considered in this research

within a given fluid volume into a mechanical force or torque).

The actuators in Figure 8.5 produce the forces F in (8.1). The force balance on the piston is

$$P_a A_a - P_b A_b - F - f = m\ddot{x}$$

where P_a and A_a are the pressure and area of the "head" side, while P_b and A_b are the pressure and area of the "rod" side of the cylinder and f is the lumped parameter friction force. Typically the acceleration and piston mass, m, are negligible allowing the cylinder force to be approximated with

$$F = P_a A_a - P_b A_b - f \tag{8.2}$$

8.4.1.1 Cylinder behavior at limits of stroke

Any real cylinder has finite range. When the cylinder piston reaches the end of stroke, a hydro-mechanical cushion mechanism activates to quickly arrest the motion. Accurately modeling these effects requires complicated equations involving fluid dynamics and solid mechanics. The resulting dynamics are often numerically stiff and difficult to solve in realtime. A good approximation is to assume that there are no dynamics associated with the stops, and that the piston immediately stops at the limits of the cylinder stroke. During simulation, this constraint is realized by checking if the stroke is within limits. If, for

example, the cylinder is at maximum stroke, then the velocities and accelerations that cause the cylinder to extend further are set to zero.

8.4.1.2 Pressure dynamics

The pressure change within a chamber of fluid having volume, V, is

$$\dot{P} = \frac{\beta}{V} \sum Q$$

where β is the effective bulk modulus of the fluid and $\sum Q$ is the net flow entering this volume [21]. For a cylinder extending at rate $\dot{q}$, with a flow Q_a entering the head side chamber through workport A, the pressure P_a will vary as

$$\dot{P}_a = \frac{\beta}{V_a(q)} \left(Q_a - A_a \dot{q} - Q_{\text{leak}} \right) \tag{8.3}$$

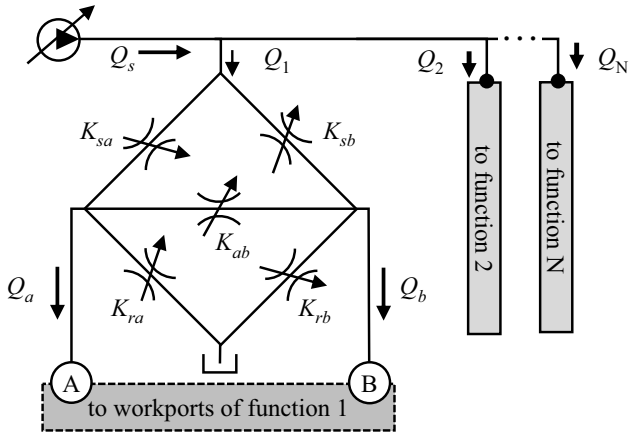
where Q_{leak} is the (typically very small) leakage between chambers, and $V_a(q)$ is the net volume of fluid in the chamber, which depends on the cylinder position and the constant dead volume of the conduits. Pressure dynamics for P_b have a form equivalent to (8.3).

8.4.1.3 Pressure dynamics at maximum/minimum pressure

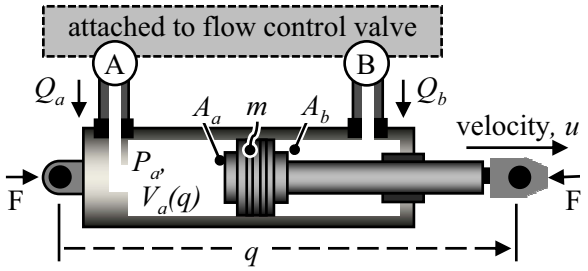
Pressure relief valves are installed in hydraulic circuits to prevent the working pressure from exceeding the safety limits of the hardware. In the Bobcat 435 excavator, there are relief valves installed between each workport and the tank.

Check valves are installed between some workports and the tank (or charge pressure circuit). A check valve allows unidirectional flow of hydraulic oil from the tank in case the line pressure drops below the tank pressure.

The dynamics of relief and check valves are nonlinear, but usually occur across short time intervals. Both types of valves act as pressure limiters: the maximum workport pressure is governed by the relief valve setting, while the minimum pressure is determined by the tank (or charge circuit) pressure. Since the valves are assumed to respond infinitely fast, the limits on allowed pressure are accounted for by integrating (8.3) with a saturating integral.



(a) A generic flow control valve consisting of multiple controlled orifices.



(b) A typical hydraulic linear actuator. Rotary actuators are nearly identical in concept.

Figure 8.5: A parallel arrangement of hydraulic flow control valves

8.4.1.4 Fluid conduits

The fluid conduits between the system pump and the actuators generally are of moderate length and diameter. The small fluid volume means the fluid mass, and hence also the inductance, is negligible. The conduits also have a moderate diameter, so only a negligible pressure drop across the lines occurs due to flow resistance. Both assumptions are valid given the characteristics of the actual system. The conduit does, however, have an appreciable effect on the capacitance of the system because of the fluid volume and line compliance. To account for this, the volume of the lines between valve and actuator is included in the volume of the cylinder chamber. The volume of the main system supply line is included in the pump model.

8.4.2 The Hydraulic Control Valves

The flow into (and out of) the hydraulic actuators is often controlled by a valve having controllable orifices, such as the one shown schematically in Figure 8.5a. Nodes A and B of the valve attach to the respective workports of the actuator. The (variable) orifices K_{ij} throttle or "meter" the flow between nodes i and j according to the equation for turbulent flow through a sharp-edged orifice [21],

$$Q_{ij} = K_{ij}\sqrt{|P_i - P_j|} \cdot \text{sign}\,(P_i - P_j) \tag{8.4}$$

where Q_{ij} is the flow from node i to node j and $P_i - P_j$ is the pressure difference between nodes i and j.

Equation (8.4) is valid for a general valve configuration (as long as the basic assumption of turbulent flow across a sharp edge is met). In some valve designs, the flow conductances K of various orifices are coupled, e.g., by the physical geometry of the lands cut into the spool, and thus controlled by a single control input. In other valve designs the conductance of each orifice is independently controlled, such as IMC discussed in literature [10, 12, 41].

In certain applications of IMC, the flow conductance coefficients of the valve, K_{ij}, are independently controlled by electrohydraulic poppet valves (EHPVs) [8]. These EHPVs have several benefits over conventional spool valves, including more accurate control, potentially lower manufacturing costs, and enabling of more energy efficient operating modes. However, one downside is the increased complexity required to control the valves. Research at the industrial and academic levels has been done to characterize these valves in order to enable accurate control of the flow conductance [140]. A cross-sectional view of one such valve manufactured by HUSCO Int'l is shown in Figure 8.6.

The dynamic relationship between the actual EHPV flow conductance, K, and a commanded flow conductance, K_{cmd}, has been determined by experimental data to be well approximated by a second-order response under standard operating conditions [7]

$$\ddot{K} + 2\zeta\omega_n\dot{K} + \omega_n^2 K = \omega_n^2 K_{\text{cmd}} \tag{8.5}$$

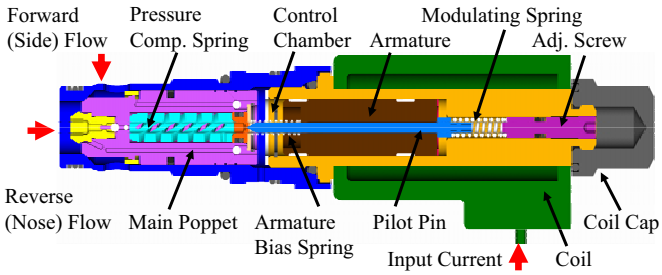


Figure 8.6: Cross-sectional diagram of electrohydraulic poppet valve (EHPV). *Figure courtesy of Patrick Opdenbosch and reproduced from [7]; originally based on [8, 141].*

where typical values are $\omega_n \approx 70$ rad/s and $\zeta \approx 1.25$. The control input to the EHPV is a PWM current signal with a duty cycle determined by the Valve Controller, as discussed in the next subsection. The valve has a finite flow conductance, so (8.5) is valid only when $K_{\text{cmd}} < K_{\max}$, where $K_{\max}$ is the flow conductance at the maximum valve opening.

8.4.3 The Valve Controller

IMC is a hydraulic control architecture that offers increased flexibility, controllability, and efficiency and has attracted the attention of most manufacturers of hydraulic earthmoving machines including Caterpillar [9], Deere [40], and HUSCO [41].

The particular implementation of IMC used in this research is the INCOVA control logic developed by HUSCO, Int'l [41–43, 142, 143]. Contributions to the control of EHPVs has been the subject of two related theses by Opdenbosch [7] and Shenouda [12] at Georgia Tech.

The INCOVA master controller receives velocity commands from the operator, typically communicated via two sets of electric joysticks in the cab, and converts these to nominal velocity commands for each of the four functions. The velocity command may be unattainable due to flow limits, power limits, or other factors, so the master controller may change the velocity command based on a flow sharing algorithm or function priority. The master controller also computes the pressure setpoint for the primary system pump. Locally distributed function controllers then calculate the commanded orifice flow conductances K_{ij} based on the numerical mappings relating control current to flow conductance command, sensed workport pressure of the local function, and the velocity command received from the

master. The specific computation of the valve flow conductance, given velocity command and sensed pressures, was implemented in Simulink; however, the proprietary nature precludes presentation of this work here. The algorithm is discussed at a high level in patents issued by HUSCO [41, 42].

8.4.4 Joystick Input

An operator seated in the cockpit of the excavator provides nominal actuator velocity commands by displacing a pair of joysticks (JS). The JS displacement maps to the nominal velocity command of the corresponding actuator, according to

$$\bar{u} = R(r)$$

where r is the normalized velocity command vector with components $r_k \in [-1, 1]$. The mapping is viewed by industry as an important aspect of excavator design and "feel". Typically R is a constant, monotonic function. Further, each component $\bar{u}_k$ only depends on the corresponding component of r_k, so the commanded velocities are independent. Note, however, that the actual cylinder velocities are not independent. Excavator manufacturers often consider the mapping between JS throw and nominal velocity command to be a proprietary competitive advantage. It is known that usually there is a deadband about the null position, then a linear region with high slope (often termed *hop-on* in industry circles), followed by a nonlinear portion. The function R is continuous and smooth (continuous first derivatives) except for the transition out of the deadband. The deadband and hop-on effects are the same for all functions, so the mapping $R = [R_1(r_1), \cdots, R_N(r_N)]^T$ has the form

$$\bar{u}_k = R_k(r_k) = \begin{cases} D_k^+ \hat{R}(r_k) & \text{if } r_k > 0, \\ -D_k^- \hat{R}(r_k) & \text{if } r_k \leq 0 \end{cases}$$

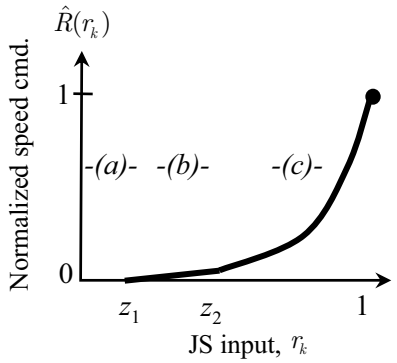


Figure 8.7: The normalized joystick velocity map, with three regions. Region (a): deadzone; region (b): linear *hop-on*; region (c): quadratic rise.

where D_k^+ and D_k^- are the maximum speeds of function k for extending and retracting, respectively, and $\hat{R}$ is the normalized JS mapping function defined as

$$\hat{R}(r_k) = \begin{cases} 0 & \text{if } |r_k| < z_1, \quad \text{(Deadzone)} \\ a(r_k - z_1) & \text{if } z_1 < |r_k| < z_2, \quad \text{(Hop-on)} \\ b(r-c)^2 + d & \text{otherwise} \quad \text{(Quadratic rise)} \end{cases}$$

where z_1, z_2, and a are parameters defined by the particular JS mapping and the constants b, c, and d are readily solved for by imposing continuity of the function and its first derivate at z_2 to give

$$b = \frac{a(z_1 - 1) + 1}{(z_2 - 1)^2} \tag{8.6}$$

$$c = \frac{a\left(z_2^2 - 2z_1 z_2 + 1\right) - 2z_2}{2a\left(z_1 - 1\right) + 2} \tag{8.7}$$

$$d = -\frac{a\left(4z_1\left(z_1 - z_2 - 1\right) + \left(z_2 + 1\right)^2\right) + 4\left(z_1 - z_2\right)}{4\left(a\left(z_1 - 1\right) + 1\right)}a \tag{8.8}$$

A sketch of the function $\hat{R}(r_k)$ is shown in Figure 8.7.

8.4.5 Pump Dynamics

In the stock Bobcat 435 excavator, there are four pumps— one for each of the tracks, one charge pump, and a primary pump. The primary pump is a single variable displacement

axial piston pump that provides this pressurized hydraulic oil to actuate the the swing, boom, arm, and bucket functions. A prime mover, such as a diesel engine, turns the shaft of the pump.

A schematic of the system pump model is shown in Figure 8.8 and a block diagram of the pump dynamics and pressure controller is given in Figure 8.9. The goal is to control the pump supply pressure, P_s. The pump is given a displacement command d, in units of volumetric flow per revolution. The actual pump displacement is finite and limited by the pump geometry. Furthermore, the friction and inertia of the pump's swash plate is modeled as a first-order lag [5]. The prime mover which drives the pump turns at a fixed rotation rate, N, and the pump has a net volumetric efficiency η_p, so the pump flow is

$$Q_p = N\eta_p d$$

For lubrication and heat transfer, a finite amount of leakage exists between the pump workports and the pump case; this leakage is non turbulent and depends linearly on the pump pressure differential as

$$Q_{\text{leak}} = G_{\text{leak}}(P_s - P_r)$$

The downstream system components require a total flow Q_s. Thus the net flow *into* the supply conduit is

$$\sum Q = Q_p - Q_s - Q_{\text{leak}}$$

This net flow into the conduit of volume V_0 causes the supply pressure to change according to the standard dynamics that determine the pressure rise within a closed volume of fluid

$$\dot{p} = \frac{\beta}{V_0} \sum Q$$

A standard proportional-integral feedback controller modulates the supply pressure, P_s, in the presence of the disturbance flow Q_s, as illustrated in Figure 8.9.

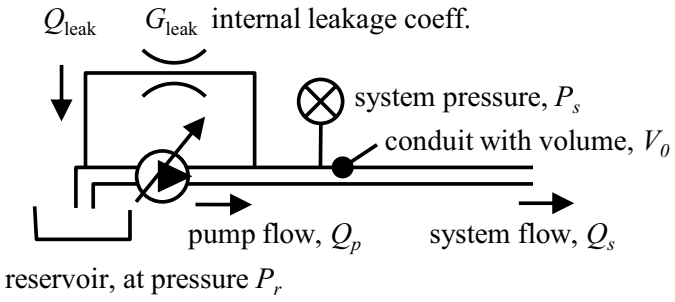


Figure 8.8: Schematic representation of system pump model

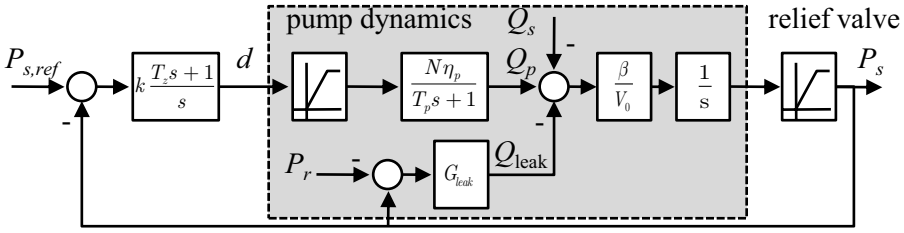


Figure 8.9: Block diagram of pump and pressure controller dynamics

8.5 Simplified Excavator Model

The dynamic model presented in the preceding sections represents a relatively high-fidelity model that captures effects due to inertia, friction, acceleration, and finite response times. This model serves as a surrogate to an actual excavator during the human-in-the-loop evaluations. However, the complexity of this model is unsuitable for optimizing the gross manipulator motion. For this purpose, a low-order steady state model which ignores the complex transients in the manipulator response is used. There are several dynamic effects that can be neglected because they occur over a short time span: the pump response time, the acceleration time, the valve response time, and the dynamics associated with pressure rise with a closed fluid volume.

The critical effects that must be retained by the steady state model include the pressure, flow, and power limits of the pump, as well as the specific fluid path between the pump and the actuator.

Other researchers have characterized the steady-state response of hydraulic excavators. Krishna used a memory-based learning technique to construct a mapping between

inputs of actuator loads and valve commands and the resulting steady state machine response [29, 144]. This model described the interaction between functions well. For example, the tendency for the function having the lower load to scavenge all the flow from a higher-loaded function when in a flow-limited situation. The approach required accurate training data and relied on convergence of the memory map, which cannot be guaranteed. Since the physical meaning of the model is not readily apparent for this black box model, little insight into the underlying phenomena that drive the response. The model presented in the next section does not depend on pre-initialized models. Further, the model is well suited for application of the optimization theorems in Chapter 9.

8.5.1 Valve Operating Mode

There are two constraints which limit the achievable velocity of all the actuators. The *multi-function constraint* limits the combined flow consumed by all functions, and the *single-function constraint* limits the velocity that any particular function can attain. Both of these constraints are a function of the valve operating mode, which will be discussed in this section.

A hydraulic control valve directs flow to an actuator. The valve consists of controlled orifices which, depending on the type of valve, may be independently controlled or coupled. The following analysis assumes the valve consists of four controlled orifices in a Wheatstone bridge arrangement; many types of valves can be modeled with this structure [21]. This arrangement enables five basic operating modes: Standard Extend (SE), High-side Regeneration Extend (HSRE), Low-side Regeneration Extend (LSRE), Standard Retract (SR), and Low-side Regeneration Retract (LSRR). Operating modes using different combinations of valves are also possible, but not discussed here. The operating modes are distinguished by the flow path of the fluid, as sketched in the circuit diagrams within Table 8.2 and Table 8.3 on pages 91 and 92. Each mode has its own advantages, disadvantages, and performance capabilities such as higher operating force, greater energy efficiency, or higher achievable speed. A high-level system controller selects the particular operating mode based on criteria including velocity command, load, and available flow [42].

Let the flow Q_b be the flow out of the rod side of the cylinder. Q_b is a function of the valve operating mode, the source and return pressures (P_r and P_s), and the external cylinder load, F. At steady state, the external load F balances the hydraulic force, so that

$$F = P_a A_a - P_b A_b \tag{8.9}$$

$$= A_a \left(P_a - \frac{A_b P_b}{A_a} \right) \tag{8.10}$$

$$= A_a L_p \tag{8.11}$$

where the symbol L_p denotes the *pressure load*. If $L_p > 0$, then there is an external force pushing the cylinder *in*, and if $L_p < 0$, this force is pulling the cylinder *out*. If L_p is opposing the cylinder velocity, then the load is said to be *resistive*; if L_p is aligned with the velocity, the load is *overrunning*. Certain operating modes are only possible for overrunning loads, for example. Q_b can be expressed as the flow through a single orifice, as in Figure 8.10 with the standard relation

$$|Q_b| = K_{eq} \sqrt{P_{eq}}$$

where the equivalent pressure, P_{eq}, and equivalent flow conductance, K_{eq}, are determined from the valve operating mode [12]. In general, the equivalent pressure for an operating mode is

$$P_{eq} = \text{sign}\,(u) \cdot (R\,(\textbf{Head-side source pressure}) - (\textbf{Rod-side source pressure}) - RL_p)$$

where $R \equiv A_a/A_b$. **Head-side source pressure** refers to the pressure of the source of the fluid that is ultimately entering the head of the cylinder. Likewise, **Rod-side source pressure** refers to the pressure of the fluid source that enters the rod side. For example, in HSRE (see Table 8.2), fluid enters the head side through valve K_{sa} which connects to source P_s, while fluid leaves the rod side through valve K_{sb} which connects to P_s. Thus for HSRE, $P_{eq} = RP_s - P_s - RL_p = (R-1)P_s - RL_p$.

The equivalent valve flow conductance is a function of the valve openings and is given

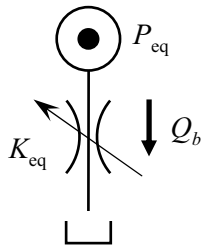


Figure 8.10: The flow Q_b due to a valve operating mode is mathematically equivalent to the flow through a single orifice having flow conductance K_{eq} and pressure differential P_{eq}.

by

$$K_{eq} = \frac{K_a K_b}{K_a^2 + R^3 K_b^2}$$

where K_a is the flow conductance of the valve connected to the head-side, and K_b is the conductance of the valve connected to the rod-side. Table 8.2 (for extending modes) and Table 8.3 (for retracting modes) list P_{eq} and K_{eq} for each of the common operating modes. The dark lines in each figure represent the active flow path and the direction of flow is inferred from the motion direction.

The actuator speed is computed using the equivalent pressure and conductance using

$$|u| = \frac{|Q_b|}{A_b} = \frac{1}{A_b} K_{eq} \sqrt{P_{eq}} \tag{8.12}$$

The sign of u is inferred from the operating mode (i.e., $u > 0$ if the valve is in an *extend* mode).

8.5.2 Limits on Cylinder Velocity

Equation (8.12) gives the cylinder speed as a function of load, valve conductances, and supply and return pressures. The maximum velocity of the kth function is achieved when the system pressure and valve flow conductances are maximal ($P_s = P_s^{\max}$ and $K_k = K_k^{\max}$) for each valve. The maximum actuator speed is a positive number denoted D_k, and is given by evaluating (8.12) at $P_s^{\max}$ and $K_k^{\max}$, thus constraining the maximum speed as

$$|u_k| < D_k \tag{8.13}$$

Table 8.2: Equivalent pressure and equivalent flow conductance for common extend operating modes. Dark lines in hydraulic circuit drawing represent fluid path.

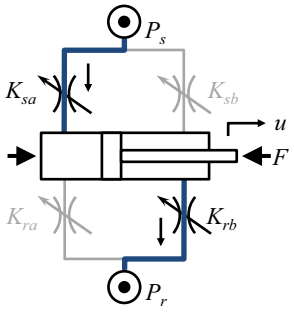
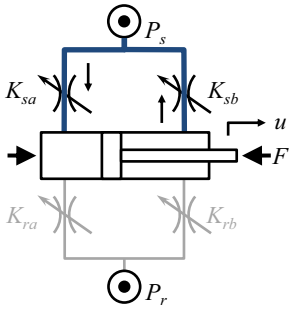
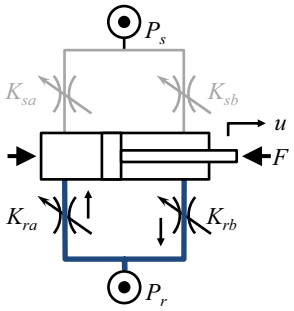
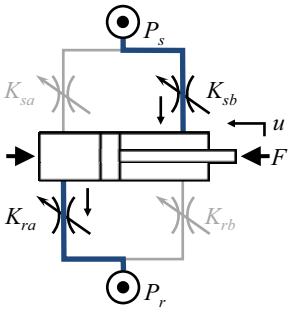
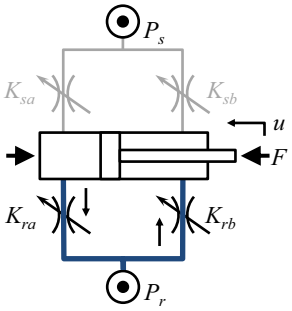
Operating modes for $u > 0$ (Extend)	
Powered Extend (PE)	 $P_{eq} = RP_s - P_r - RL_p$ $K_{eq} = \frac{K_{sa}K_{rb}}{\sqrt{K_{sa}^2 + R^3K_{rb}^2}}$ $\psi = A_a$
High-side Regeneration Extend (HSRE)	 $P_{eq} = (R-1)P_s - RL_p$ $K_{eq} = \frac{K_{sa}K_{sb}}{\sqrt{K_{sa}^2 + R^3K_{sb}^2}}$ $\psi = A_a - A_b$
Low-side Regeneration Extend (LSRE)	 $P_{eq} = (R-1)P_r - RL_p$ $K_{eq} = \frac{K_{ra}K_{rb}}{\sqrt{K_{ra}^2 + R^3K_{rb}^2}}$ $\psi = 0$

Table 8.3: Equivalent pressure and equivalent flow conductance for common retract operating modes. Dark lines in hydraulic circuit drawing represent fluid path.

Operating modes for $u < 0$ (Retract)	
Powered Retract (PR)	
	$P_{eq} = P_s - RP_r + RL_p$ $K_{eq} = \frac{K_{ra}K_{sb}}{\sqrt{K_{ra}^2 + R^3K_{sb}^2}}$ $\psi = A_b$
Low-side Regeneration Retract (LSRR)	
	$P_{eq} = (1-R)P_s + RL_p$ $K_{eq} = \frac{K_{ar}K_{br}}{\sqrt{K_{ar}^2 + R^3K_{br}^2}}$ $\psi = 0$

with

$$D_k = (K_{eq,k})_{\max} \sqrt{(P_{eq,k})_{\max}}$$

8.5.3 Multi-function Flow Constraint

The portion of system flow consumed by a single actuator will be related to the function's speed with the definition

$$\psi_i \equiv \frac{Q_i}{|u|} \tag{8.14}$$

The variable ψ will have special significance later in this thesis, so it is given the special name *Flow-Speed Ratio* (FSR). For the modes discussed here, ψ is constant throughout the motion. The FSR for each of the metering modes is given in Table 8.2 and Table 8.3. Notice that for LSRE and LSRR, $\psi = 0$, implying from (8.14) that no flow from the pump is used to move the actuator in these modes.

The net flow from each function is limited by the maximum flow available from the system pump. If the net flow delivered by the pump is Q_s, and the portion of system flow consumed by each function is Q_i, then

$$Q_s = \sum Q_i = \psi_1 u_1 + \psi_2 u_2 + \cdots = \psi^T u < C_0$$

where $u = [u_1, \cdots, u_n]^T$, $\psi = [\psi_1, \cdots, \psi_n]^T$, and C_0 is the physical flow limit of the pump, determined by the pump shaft speed, the volumetric displacement, and the pump volumetric efficiency at the operating pressure.

8.5.4 Power Limits

The power delivered by the system pump is limited, thus

$$\text{Power} = P_s Q_s = P_s \sum Q_i = P_s \psi^T u < W$$

where W is the maximum power capabilities of the prime mover. If the demanded power exceeds to available power, the prime mover will decrease speed until the power limit is satisfied.

8.5.5 Summary of Constraints on Actuator Velocity

The actuator velocity must satisfy the constraints

$$\psi^T u < C \tag{8.15}$$

$$|u_k| < D_k \tag{8.16}$$

where

$$C = \min\left(C_0, \frac{W}{P_s}\right)$$

is the maximum pump flow the pump can provide, adjusting for power constraints, if necessary. The velocity constraints (8.16) define a feasible region, U, for which the simplified model is valid.

This kinematic model implicitly or explicitly assumed satisfaction of several assumptions. Of note were assumptions that the actuators can exactly track the commanded velocity. This assumes the valve dynamics are negligible, the mapping between commanded and actual valve flow conductance is exact, the fluid is incompressible, and the actuator paramters (i.e., A_a, A_b, eletric current to flow conductance mappings, etc.) are known exactly. Further, the system pump is treated as a flow-limited pressure source with no dynamics which neglects dynamics of the pump swash plate angle, the pump controller, and effects related to the internal geometry including flow ripple. The inertial effects of the linkages are also neglected.

In the following chapter, this simplified model is used for optimizing the motion primitives.

CHAPTER IX

OPTIMIZATION OF VELOCITY-CONSTRAINED ACTUATORS

Most hydraulically actuated manipulators, including excavators common in the earthmoving industry, have constraints on the net sum of the actuator speeds. Thus, the combined velocity achieved by all functions is limited, in addition to the conventional rate limits of individual actuators. A velocity-constrained kinematic model, $\dot{x} = -u$, sufficiently describes the manipulator motion over short time intervals. This chapter gives a necessary condition for the velocity control input $u(t)$ to be time-optimal for point-to-point motion in an obstacle free environment. The optimal solution is almost never unique.

A special point, u^*, is introduced. In the case where the input $u(t) = u^*$ for all time, the motion is necessarily optimal. In the case where $u(t) \neq u^*$ for some time, the input may or may not be optimal; the location of u^* in the input plane provides a convenient test for optimality, and can discriminate an input $u(t)$ as sub-optimal even before the input violates a necessary condition for optimality.

9.1 Nomenclature

n	Number of actuators moved during the particular motion
C	Maximum pump flow rate, *scalar*
D	Vector of single-function flow constraints
ψ	Vector of the volumetric efficiency of each function
u	Velocity of actuators
$\bar{u}$	Operator velocity command
u^*	The particular optimal input that is also stationary
x	Actuator position coordinate, normalized so $x(T) = 0$
q	Generalized position coordinate
T	Final task time

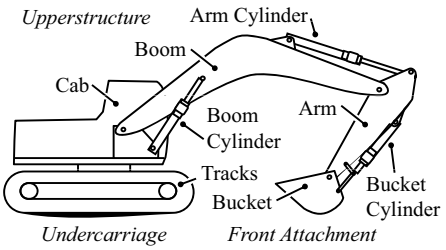


Figure 9.1: Major components of an excavator linkage system typically include four actuated *functions* including *swing, boom, arm, bucket.*

$(\cdot)^T$ Transpose of $(\cdot)$

$(\cdot)_k$ The kth element of vector $(\cdot)$

$[\,(\cdot)\,]$ A diagonal matrix with $(\cdot)$ along the diagonal

All vector quantities exist in R^n

9.2 Introduction and Background

Hydraulic actuators are often used in applications requiring high power density at low to moderate speeds. Many large-scale industrial manipulators are controlled with fluid power, including manipulators for factory automation and for earthmoving. The manipulator considered here is the common earthmoving excavator (Figure 9.1). Excavators generally have at least four degrees-of-freedom (DOFs) arranged in an open kinematic chain and are typically manually controlled via joysticks by a human operator seated in the cab.

A human operator interacts with the control system—possibly consisting of electronic- or hydromechanical-based logic—that drives the actuator system. The operator sits within the cab and clenches two joysticks as in Figure 9.2, each able to pivot in two directions; the angular displacement (e.g., fore/aft or left/right) maps to the (nominal) commanded velocity $\bar{u}$ of the corresponding hydraulic actuator.

The linkages are accelerated by a hydraulic actuator system comprised of cylinders, conduits, controlled orifices, pumps, accumulators, and a prime mover. Figure 9.3 shows a simplified schematic of the typical connection of actuators for the type of systems considered here. The valves represent a generic arrangement of electronically controlled orifices. The

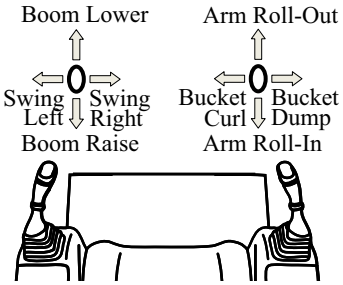


Figure 9.2: The operator interface of a typical excavator

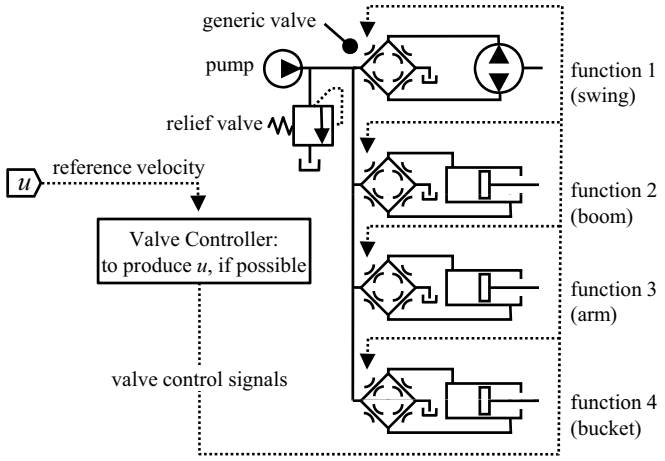


Figure 9.3: Simplified hydraulic circuit for a multi-DOF hydraulic manipulator

orifices comprising each valve may be independently controlled (e.g., the valve may consist of four or more electro-hydraulic poppet valves) or they may be coupled (e.g., the valve may consist of a spool valve with a single degree of freedom). An open-loop electronic *Valve Controller* produces control signals to cause the actuators to track the reference velocity input $\bar{u}$.

All robotic manipulators are subject to motion constraints, including power limits and joint torque limits. In addition, multi-DOF hydraulic manipulators driven by a single source of pressurized oil are often subject to limits on the combined velocity achievable by all functions. This constraint does not generally affect electrically actuated robots, and consequently the literature is deficient in methods of control to deal with this problem.

As discussed in Chapters 5 and 10, Blended Shared Control (SC) is a way to include artificial machine intelligence in conventionally manually controlled systems. This approach may be a low overhead way to decrease cycle times of repetitive tasks. Critical to blended

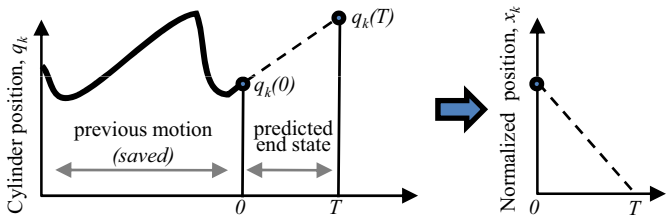


Figure 9.4: Piecewise monotonic motion segments are normalized so the value at the final time is zero.

SC is a method of calculating the time-optimal control to move the manipulator from one configuration to another. The optimization must be completed in real-time and for a variety of configurations. Methods presented in literature are too specialized [145] or appear to be unsuitable for real-time implementation [78, 93, 146].

The purpose of this chapter is to derive the optimal control input for a sub-class of velocity constrained manipulators performing basic point to point motions.

9.3 The Manipulator Task

Let $q(t) = [q_1(t), \cdots, q_n(t)]^T$ be the generalized position of the actuators. The cab rotation (or *swing*) angle is q_1, and the length of the boom, arm, and bucket cylinders are q_2, q_3, q_4, respectively. In absence of kinematic singularities, any end effector path through the workspace is equivalently described by the displacement q of the actuators. Further, the actuator trajectory may be uniquely defined by a sequence of piecewise monotonic (in the actuator position coordinate q) segments termed *motion primitives*. A sequence of motion primitives can be described equivalently by a sequence of endpoints $q(T)$. The goal is to find the velocity command $u(t)$ that minimizes the time to reach a given final configuration $q(T)$ from an initial configuration $q(0)$. The final configuration $q(T)$ is assumed to be precisely known or well approximated, for example by means of the RAMPE process in Chapter 7.

A change of variables helps to simplify notation. Defining

$$x(t) = (q(T) - q(t)) \operatorname{sign}(q(T) - q(0))$$

as illustrated in Figure 9.4 means the motion from $q(t)$ to $q(T)$ is equivalent to the motion

from $x(t)$ to $x(T) = 0$. The sign $(\cdot)$ term guarantees that $x(t) \geq 0$, since $q(t)$ is monotonic over the period. Given the change of variables, the problem is to drive the $x(t)$ system from an initial position $x(0)$ to the origin, $x(T) = 0$, with minimum final time T. Note that $x(t)$ denotes the expected remaining actuator displacement before reaching the origin.

9.3.1 System Model

Excavators have dynamics that occur over very different time scales, ranging from the very fast pressure rise within a closed volume of fluid to the slower pump and rigid-body linkage dynamics [21].

The quasi-static dynamics derived in Chapter 8 are used:

$$\dot{x} = -u \tag{9.1}$$

with the requirement that

$$u \in U$$

where the velocity control input to the cylinders is u, and U is the set of allowable inputs. From Chapter 8,

$$U = \left\{ u : \psi^T u \leq C, \quad 0 \leq u_k \leq D_k \right\}$$

where C is the max flow rate of the pump, ψ is the vector of flow-velocity ratios for the valve operating mode, and D_k is the maximum velocity of the actuator as given in Tables 8.3 and 8.2 on pages 92 and 91. The set U is simply connected and always nonempty.

9.4 Projection of Non-allowable Inputs to Allowable Region

From the equations of motion (9.1), the actuators track the input u perfectly for $u \in U$. Since the velocity $\bar{u}$ commanded by the operator might be outside the allowable region U, a suitable projection must be defined to make an infeasible $\bar{u}$ feasible.[1] A common technique

[1] In industry, this problem is often termed *flow sharing*.

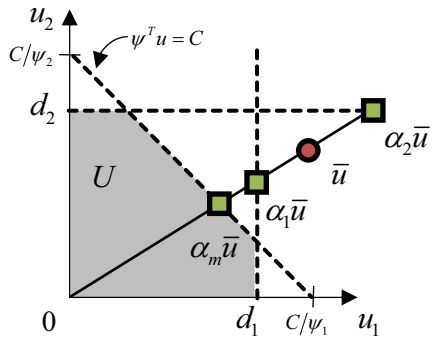


Figure 9.5: An infeasible input $\bar{u}$ is projected into the feasible region U by proportionally scaling each component of $\bar{u}$.

in industry [41] is to scale each component of $\bar{u}$ by a common factor α so that

$$\bar{u}_p = \text{proj}_{\partial U}\bar{u} \tag{9.2}$$

$$= \alpha\bar{u} \tag{9.3}$$

where $\bar{u}_p$ is the projection of $\bar{u}$ onto the boundary ∂U of U. With reference to Figure 9.5,[2] ∂U may consist of points along the multi-function constraint line $\psi^T u = C$, or the single-function constraint lines $u_k = D_k$, or both. The scalar α is

$$\alpha = \min_j (1, \alpha_m, \alpha_k) \tag{9.4}$$

where $\min_j(\cdot)$ indicates the minimum of the arguments considering all values $j = 1, 2, \cdots, n$ and

$$\alpha_m = \frac{C}{\psi^T\bar{u}} \tag{9.5}$$

$$\alpha_j = \frac{D_j}{\bar{u}_j} \tag{9.6}$$

are the scaling factors necessary to intersect the multi- and single-function constraint lines. Note that in all cases $\alpha \leq 1$; this prevents the input from being scaled if $\bar{u} \in \text{int}(U)$.

[2]Note: all figures will be drawn in the u_1-u_2 plane for clarity; however, in general, $u \in R^n$

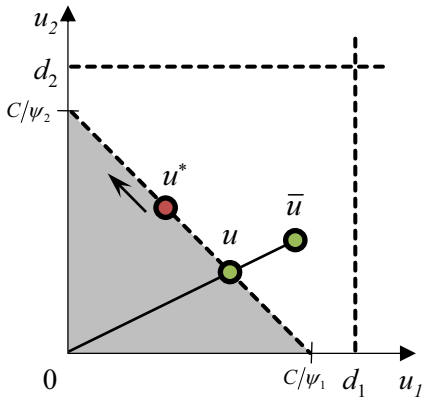


Figure 9.6: Allowable region U for Undersized Pump ($D_k \geq C \quad \forall k$). An infeasible $\bar{u}$ is projected to the boundary of U by scaling each component equally.

9.5 Optimal Input - Special Case of Undersized Pump

Before introducing a theorem for the time-optimal input of the general case, a special case will be considered. The useful point u^* is introduced.

If $D_k \geq C \quad \forall k$, then the pump is *undersized*, and the feasible region in the u-plane is triangular, as sketched in Figure 9.6. The pump is called undersized because any particular valve is able to consume the entirety of the pump flow. In this case the optimal solution $u^*(t)$ must satisfy

$$\psi^T u^*(t) = C \tag{9.7}$$

for all time. The optimal control for the case of an undersized pump can be arbitrarily chosen such as long as (9.7) is satisfied.

There is a particular u^* such that all functions reach the origin simultaneously. This input is constant for all time throughout the trajectory from $x(0)$ to the origin.

Theorem 1. *The input*

$$u^* = \frac{C}{\psi^T x} x \tag{9.8}$$

will drive the system (9.1) from $x = [x_1, \cdots, x_n]^T$ *to the origin in minimum time*

$$T^* = \frac{\psi^T x}{C}$$

and will be constant for all x along the trajectory.

Proof. The total fluid volume required from the system pump is related to the flow-velocity ratio ψ of each function and the remaining distance x each function must travel as

$$V = \psi_1 x_1 + \psi_2 x_2 + \cdots + \psi_n x_n = \psi^T x$$

With input u^* the pump always delivers maximal flow rate C because $\psi^T u^* = C\frac{\psi^T x}{\psi^T x} = C$. Hence, the minimum time to deliver the net volume V is

$$T^* = \frac{\psi^T x}{C}$$

Given a constant input u_k, the time required for the kth function to go from x_k to the origin is

$$T_k = \frac{x_k}{u_k}$$

Equating $T_k = T^* = \frac{\psi^T x}{C}$ and solving and solving for u_k, gives

$$u_k^* = \frac{C}{\psi^T x} x_k$$

for all $k = 1, \cdots, n$, which becomes Equation (9.8) when expressed in vector notation . □

9.5.1 Motion of u^* for Suboptimal $\bar{u}$

If the operator input $\bar{u}$ differs from u^*, then u^* will in general not be stationary. Understanding the dynamics of u^* is helpful for analysis and for designing a shared controller. The motion of u^* will depend on the state x and the input u, as in Theorem 2.

Theorem 2. *The point u^* defined by (9.8) is a dynamic function of the state x and input u, having velocity $du^*/dt = \dot{u}^*$, where*

$$\dot{u}^* = \frac{C}{\left(\psi^T x\right)^2} \left(\left(\psi^T u\right) x - \left(\psi^T x\right) u \right) \tag{9.9}$$

Proof. This result follows directly from

$$\frac{du^*}{dt} = \nabla u^* \frac{dx}{dt}$$

by using the dynamics in (9.1) and

$$\nabla u* = \frac{C}{\psi^T x}[I] - \frac{C}{(\psi^T x)^2} x\psi^T$$

where $[I]$ is the identity matrix. The velocity is always directed parallel to the manifold $\psi^T u = C$ because

$$\psi^T \dot{u}^* = \frac{C}{(\psi^T x)^2}\left(\psi^T\left(\psi^T u\right)x - \psi^T\left(\psi^T x\right)u\right) \equiv 0$$

□

Remark 1. *The point u^* is stationary for any u along the line from the origin to u^*. This is true because $\dot{u}^* = 0$ whenever $u_k/x_k = \left(\psi^T u\right)/\left(\psi^T x\right)$.*

Remark 2. *The velocity $\dot{u}^*$ is always on the constraint manifold $\psi^T u = C$ and points in a direction "away" from u. This result is seen by writing (9.9) as*

$$\begin{aligned} u^* &= \frac{\psi^T u}{\psi^T x}\frac{C}{\psi^T x}x - \frac{C}{\psi^T x}u \\ &= \frac{\psi^T u}{\psi^T x}u^* - \frac{C}{\psi^T x}u \\ &= \frac{\psi^T u}{\psi^T x}(u^* - u) - \frac{C - \psi^T u}{\psi^T x}u \end{aligned}$$

Thus, there is always a component of velocity in the direction $(u^* - u)$. The motion of u^* for the case of suboptimal operator input ($u = \bar{u}$) is illustrated in Figure 9.7; for the planar case, u^* moves along the line $\psi^T u = C$.

The usefulness of point u^* will be apparent when analyzing the general case.

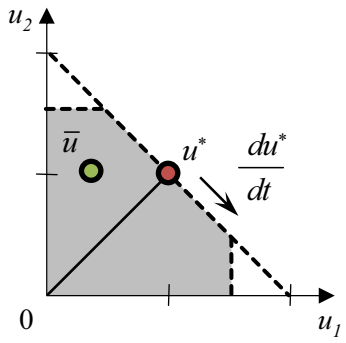


Figure 9.7: The point u^* is not stationary when the input $u = \bar{u}$ differs from u^*

9.6 Optimal Input for General Case

Consider the problem where $\psi, D \in R^n$, and $C \in R$ are given constants. Further, the input $u = \bar{u} \in U$ and the value $x(t)$ are given, and the input is assumed to follow the switched input control. The task time T is the time required for all components x_k to reach the origin, so

$$T^* = \max_i \frac{x_i}{u_i}$$

Choosing $u = u_p^*$, where u_p^* is the the projection of $u^* = \frac{C}{\psi^T x} x$ onto the feasible region using Equation (9.3), gives

$$u_p^* = \min_j \left(1, \frac{C}{\psi^T u^*}, \frac{D_j}{u_j^*}\right) u^*$$

so that

$$T^* = \max_i \left(\frac{\frac{x_i}{u_i^*}}{\min_j \left(1, \frac{C}{\psi^T u^*}, \frac{D_j}{u_j^*}\right)} \right)$$

where

$$\begin{aligned}
\frac{C}{\psi^T u^*} &= \frac{C}{\psi^T x} \frac{\psi^T x}{C} = 1 \\
\frac{D_j}{u_j^*} &= \frac{D_j}{\frac{C}{\psi^T x} x_j} = \frac{\psi^T x}{C} \frac{D_j}{x_j} \\
\frac{x_i}{u_i^*} &= \frac{x_i}{\frac{C}{\psi^T x} x_i} = \frac{\psi^T x}{C}
\end{aligned}$$

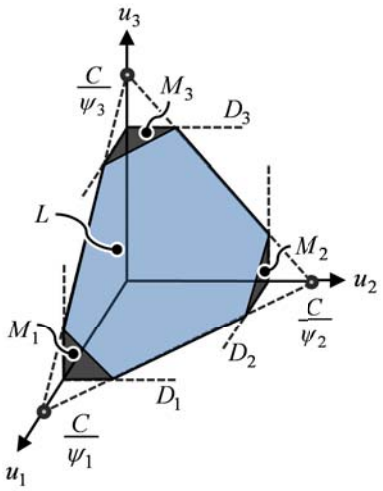


Figure 9.8: The regions L and M_k in the domain U for $n = 3$

Thus the completion time can be written as

$$T^* = \max_i \left(\frac{\psi^T x / C}{\min_j \left(1, \frac{\psi^T x}{C} \frac{D_j}{x_j}\right)} \right) \tag{9.10}$$

$$= \left(\frac{\psi^T x / C}{\min_j \left(1, \frac{\psi^T x}{C} \frac{D_j}{x_j}\right)} \right) \tag{9.11}$$

$$= \max_j \left(\frac{\psi^T x}{C}, \frac{x_j}{D_j} \right) \tag{9.12}$$

9.7 Optimality Conditions

There are two subsets of ∂U that are of interest. Let L and M_k be the regions defined as

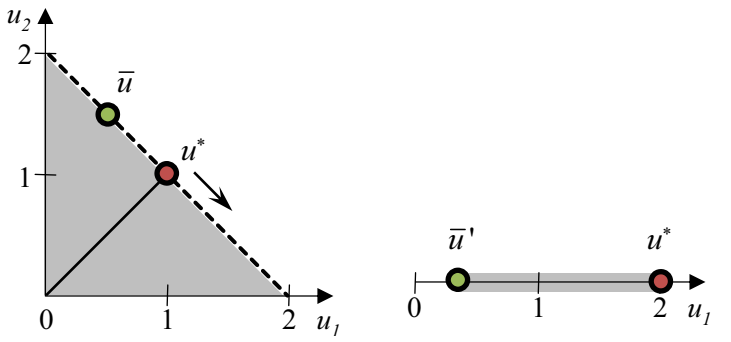
$$L = \left\{ u : \psi^T u = C, \quad u_k \leq D_k \right\} \tag{9.13}$$

$$M_k = \{ u : u_k = D_k, \quad u \in U \} \tag{9.14}$$

$u \in L$ requires the maximum pump flow C, while $u \in M_k$ implies that u_k is at maximum value, D_k, for actuator k. Figure 9.8 shows a sketch of regions L and M_k for the case of three actuators ($n = 3$). Note, it is not necessary for $M_i \cup M_j = \emptyset$.

Claim

If $u^* \in L$, then the minimum task time is $T^* = \frac{\psi^T x}{C}$. The optimality conditions are



(a) 2DOF motion $(x_1, x_2 > 0)$ (b) 1DOF motion $(x_1 > 0, x_2 = 0)$

Figure 9.9: Example of dynamics within u-plane for 2 degree-of-freedom manipulation

1. $u(t) \in L \quad \forall t \in [t_0, t_0 + T]$

2. $u^*(t) \in L \quad \forall t \in [t_0, t_0 + T]$

Note that (1) $\Rightarrow$ (2).

Claim

If the projected point $u_p^* \in M_k$, then a necessary condition for optimality is that $u_p^*(t) \in M_k \quad \forall t \in [t_0, t_0 + T]$. This will be illustrated by three example cases.

9.7.1 Examples

Consider manipulation of two degrees-of-freedom, with $\psi_1 = \psi_2 = 1$, $C = 2$, $D > 2$. Suppose the actuators are just beginning a motion with $x(0) = x = [1, 1]^T$ Using (9.8), the stationary optimal input is $u^* = [1, 1]^T$. If the operator input is $\bar{u} = [\frac{1}{4}, 1\frac{3}{4}]^T$, then the point u^* moves *away* from $\bar{u}$ in the direction shown in Figure 9.9a.

The operator input $\bar{u}$ satisfies the necessary condition for optimality ($\psi^T \bar{u} = C$) up to the moment that u^* enters the $u_2 = 0$ plane. At this point, the dimensionality of the problem is reduced to a line as in Figure 9.9b. The original input $\bar{u}$ is no longer optimal since $\psi_2 \bar{u}_2 < C$. To remain optimal, the operator input must always lie within the locus of point $\psi^T u = C$. If the operator does not immediately change inputs when the dimensionality is reduced, then there will be a time for which the input is not optimal.

The methods of analysis are illustrated by three examples. Consider three hypothetical trajectories for a 2DOF manipulator moving from the initial state $x = [1, 1]$ to the origin.

The illustrations will assume for $C = 1$, $\psi = [1, 1]^T$, and $D = \left[\frac{3C}{4}, \frac{3C}{4}\right]^T$.

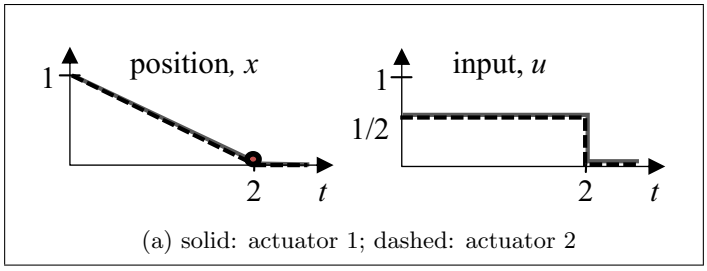
Case 1, in Figure 9.10 The input is chosen to be $\bar{u} = u^*$, so the necessary condition for optimality ($\psi^T u = C$) is satisfied everywhere. In the u-plane, Figure 9.10b, the point u^* is stationary.

Case 2, in Figure 9.11 Here, $\bar{u} \neq u^*$; however, the necessary condition for optimality is satisfied ($\psi^T \bar{u} = C$ always). Consider the behavior of u^* in the u-plane. Initially, $\bar{u} = [\frac{1}{4}, \frac{3}{4}]^T$ as shown in Figure 9.11b. The point u^* has a velocity away from $\bar{u}$, so it slides *down* the line $\psi^T u = C$. Just before $t = 1$, the point u^* is on the verge of leaving the feasible region as shown in Figure 9.11c. At that instant, $\bar{u}$ is suddenly changed to $\bar{u} = [\frac{3}{4}, \frac{1}{4}]^T$, which *pushes* u^* back into the feasible region. u^* remains feasible for the duration of the motion.

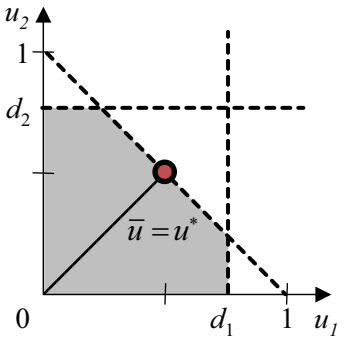
Case 3, in Figure 9.12 This is a suboptimal trajectory. The input is initially at $\bar{u} = [\frac{1}{4}, \frac{3}{4}]^T$, which again causes u^* to move down the curve. Since $\bar{u} \neq u^*$, u^* moves *away* from $\bar{u}$ (see Figure 9.12b) according to Equation (9.9). The necessary condition of optimality is satisfied through $t = 1$, after which u^* leaves the feasible region (Figure 9.12c). The trajectory is confirmed to be suboptimal immediately after u^* leaves U—the suboptimality is proven even before $\bar{u}$ violates the necessary condition. Once the point u^* leaves U, then *eventually* the input $\bar{u}$ must become sub-optimal. At $t = \frac{4}{3}$, actuator 2 reaches the origin and the dimension is reduced by one, as in Figure 9.12d. Due to the constraint $u_1 < d_1$, it is impossible for $\bar{u}$ to continue satisfying the optimality condition. Indeed, x reaches the origin with $T = 2\frac{2}{9}$, which is 11 percent longer than the optimal cases. This illustrates the potentially counter-intuitive result that an operator commanding inputs that at all times yield the maximum speed of a function will not necessarily yield a time-optimal trajectory.

9.8 Summary

This chapter gave the optimality conditions on the input $u(t)$ for completing a motion primitive with a displacement $x(t)$ with single-function constraints on the allowable input.

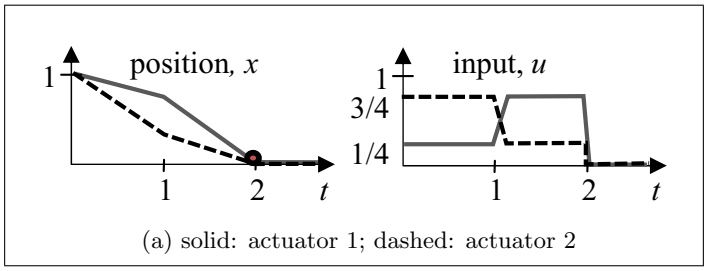


(a) solid: actuator 1; dashed: actuator 2

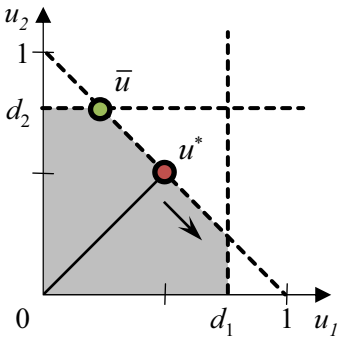


(b) $\bar{u} = u^* \quad \forall t \in [0, 2]$

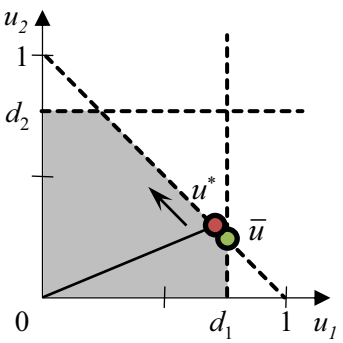
Figure 9.10: Case 1: An optimal trajectory with $\bar{u} = u^*$



(a) solid: actuator 1; dashed: actuator 2

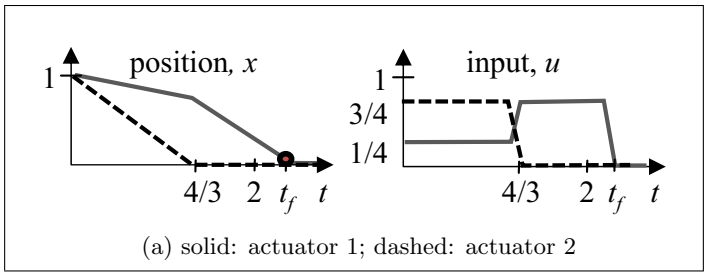


(b) $\bar{u} = [\frac{1}{4}, \frac{3}{4}]^T \quad \forall t \in [0, 1)$

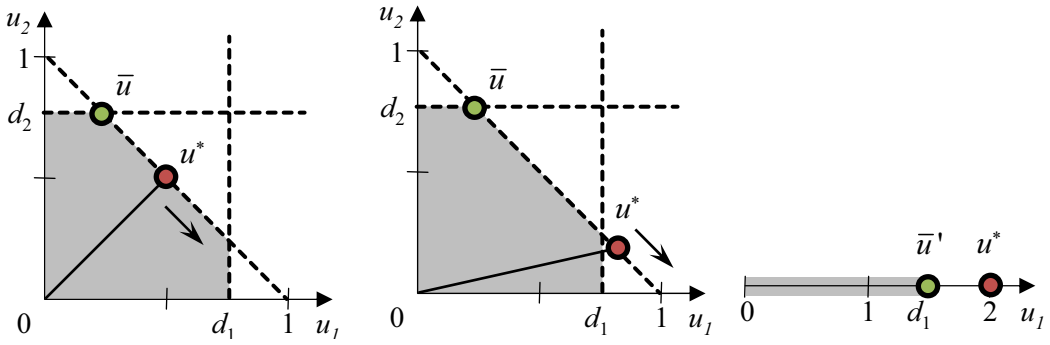


(c) $\bar{u} = [\frac{3}{4}, \frac{1}{4}]^T \quad \forall t \in [1, 2]$

Figure 9.11: Case 2: An optimal trajectory with $\bar{u} \neq u^*$



(a) solid: actuator 1; dashed: actuator 2



(b) $\bar{u} = [\frac{1}{4}, \frac{3}{4}]^T \quad \forall t \in [0, \frac{4}{3})$ (c) at $t = 1$, $u^* \notin U$ (d) $\bar{u} = [\frac{3}{4}, 0]^T \quad \forall t \in [\frac{4}{3}, T]$

Figure 9.12: Case 3: A suboptimal trajectory

The optimal solution is almost never unique, but optimality can be tested by considering a special point, u^*. The location of u^* in the input plane provides a convenient test for optimality, and can discriminate an input $u(t)$ as sub-optimal even before the input violates a necessary condition for optimality, as demonstrated in the three examples presented.

In the next chapter, the operator input $\bar{u}$ will be subject to perturbation based on the input u^*.

There are some clear caveats to this optimization approach. First, the variable $x(t)$ was assumed to be precisely known. In reality, this $x(t)$ must be estimated online, for example with the RAMPE methods presented in Chapter 7, and is subject to error. This error manifests as error in u^*. The effects of this error on task completion time should be studied. Second, the optimization occurs relative to each motion primitive; whether sequential optimization of a trajectory's constituent primitives leads to a lower overall task cost remains to be shown. Third, the controls engineer must weigh whether minimizing task time is appropriate for a given application, especially since the energetic expense of a manipulator trajectory tends to increase with the speed.

CHAPTER X

BLENDED SHARED CONTROL TO DECREASE TASK TIME

This chapter discusses a method to incorporate computer assistance into a traditionally human controlled system. The aim is to enhance the human input, without overwhelming the operator or revoking all of the operator's control authority. Specifically, assistance is provided by modifying the original operator input commands in such a way that the anticipated task is accomplished with lower cost, where task cost will be measured in terms of the completion time of a motion primitive, as introduced in Chapter 7.

10.1 Nomenclature

n	Number of actuators
x	Actuator position, normalized so at the final time $x(T) = 0$
T	Final task time
U	Domain of allowable control inputs
u	Actuator velocity
$\bar{u}$	Operator commanded velocity
δu	Control perturbation
u^*	An optimal input for completing a specified task
$(u^*)'$	Projection of u^* onto a constant-flow manifold
ψ	Flow-velocity ratio relating flow into valve to actuator velocity
C	Maximum pump flow rate
D	Vector of single-function flow constraints
a	Scalar shared control parameter
R	Function mapping joystick input coordinates to velocity command
r	Normalized joystick displacement

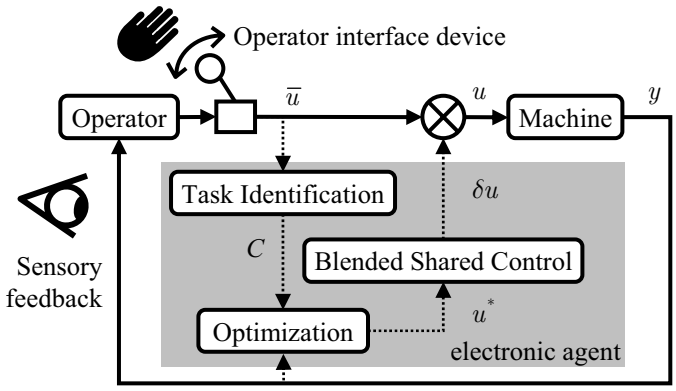


Figure 10.1: The *Blended Shared Control* architecture. The Electronic Agent perturbs the operator command by an amount δu, based on three modules: Task Identification, Optimization, and Blended Shared Control.

F, F_0	Applied joystick force, and maximum joystick force
$\|(\cdot)\|_p$	The p-norm of $(\cdot)$
$(\cdot)_k$	Component k of vector $(\cdot)$

10.2 Introduction and Background

This chapter uses the perturbation-based Blended Shared Control (SC) architecture, shown in Figure 10.1, that was introduced in Chapter 4. For a general overview of SC, see Chapter 4. The methods in Chapter 7 are used to provide an estimate of the actuator displacement x for the task. Given x, the methods in Chapter 9 are used to compute u^*, which satisfies the optimality conditions for minimum time. This chapter computes a command perturbation, δu, based on u^* and the operator command $\bar{u}$.

The input $\bar{u}$ from the Human Agent (HA), or operator, may be suboptimal.[1] The Blended SC approach discussed in this chapter continually modifies this input to bring the response closer to optimal. This can either be a direct modification or an indirect modification. *Direct shared control* will refer to cases where the HA input $\bar{u}$ is modified by the Electronic Agent (EA) before the velocity command is given to the manipulator. *Indirect shared control* refers to cases in which the EA provides sensory cues that represent

[1]Without knowing exactly the circumstances of any given input, one cannot claim with certainty that the input is *not* optimal, so the statement is technically an assertion.

Table 10.1: Control perturbations for Blended Shared Control

Type	$\delta u =$
Proportional (PSC)	$a\,(u^* - \bar{u})$
Constant Flow (CFSC)	$a\left(\frac{\psi^T \bar{u}}{\psi^T x} - \bar{u}\right)$
Haptic Cues (HC)	$F = aF_0 \frac{R^{-1}(u^*) - \bar{r}}{\|R^{-1}(u^*) - \bar{r}\|}$

suggested input changes to the HA. The purpose of the cues is to indicate to the operator *suggested* input changes. These cues may include haptic force cues, auditory tones, or visual signals. A benefit is that there is no direct conflict between the HA and EA, i.e., the HA can disregard those inputs which are deemed unacceptable. However, the HA must allocate resources to interpret these cues, and there is no guarantee that the HA interpretation of the cues will be as intended by the EA.

When using direct SC, the EA perturbs the HA input by δu so that the reference input to the manipulator is $u = \bar{u} + \delta u$. Two types of direct control perturbations, δu, will be discussed in this chapter, and are previewed in Table 10.1. These are the *Proportional* and the *Constant Flow* Shared Control perturbations. The Blended SC parameter a appears in both formulations. The parameter determines the amount of aid given by the Electronic Agent, and thus is another "knob" for the controls engineer to tune. Several formulations for limiting the value of a based on specific criteria are discussed later in this chapter.

Recognizing that any kind of analytical control analysis when a human is present in the control loop is difficult, certain assumptions will be proposed based on operator control behavior. With these assumptions, the Blended SC control laws will be shown to decrease the task time. While the assumptions on the operator inputs will allow for convenient analysis, it should be noted that strict performance optimality can not be guaranteed, as there is no way to guarantee one human response versus another. Thus, the reader should also refer to the experiments presented in Chapter 11 to understand the true effect of Blended SC on the task cost.

10.3 Direct Shared Control

Consider a single point-to-point motion task. As done previously, the displacements are scaled such that the variable $x(t)$ measures the remaining displacement until the actuators reach the origin $x = 0$. The operator input $\bar{u}$ controls the actuator speed as $\dot{x} = -u$, where $u = \bar{u} + \delta u$. The commanded velocity of the actuators is $\bar{u} = [\bar{u}_1, \cdots, \bar{u_n}]^T$ where $\bar{u}_k$ is the commanded velocity of actuator k.

The operator input $\bar{u}$ is assumed constant throughout the motion primitive. This assumption implies that the input has reached steady state, that the operator goal does not change during the course of the motion, and that the operator input does not change in response to the modifications of SC. These are each strong assumptions, but are helpful in understanding the effect of Blended SC on the task time.

The Electronic Agent will perturb the operator input by δu, so the commanded velocity ultimately realized by the manipulator is $u = \bar{u} + \delta u$, assuming $u \in U$.

10.3.1 Operator Input Model

The operator command $\bar{u}$ is assumed to be quasi-stationary, such that the input will go to zero when the corresponding function reaches the origin. Define a diagonal switching matrix to be

$$H(x) = \text{diag}(h(x_1), \cdots, h(x_n))$$

where $h(x_i)$ is an indicator function that assumes the value of 1 if the actuator position has not reached the goal, and 0 if it has reached the goal:

$$h(x_i) = \begin{cases} 0 & \text{for } x \leq 0\,, \\ 1 & \text{for } x > 0\,. \end{cases}$$

Then, with the switching taking place, the operator input $\bar{u}$ will transition according to

$$\bar{u}(t + \Delta t) = H(x(t))\bar{u}(t) \tag{10.1}$$

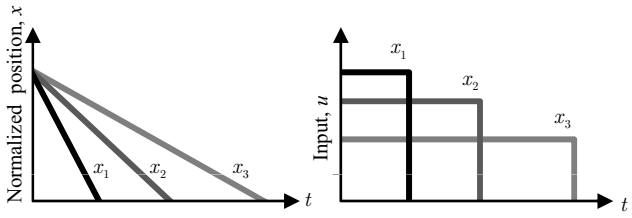


Figure 10.2: Trajectory and inputs for the switched input

Figure 10.2 shows an example of this case where an input component is deactivated when the corresponding component of the trajectory x equals zero. This assumption claims that the operator will not overshoot the goal.

This model ignores several psycho-motor phenomena associated with manual control of manipulators, including the tendency to overshoot the target and the hunting effects, during which the operator will slow down as the end effector nears the goal. This model is visualized by considering an operator holding the joysticks at a constant displacement, and, when a function gets to the desired position, the operator returns the corresponding joystick axis to the null position.

Basic Problem Assumptions:

The HA gives the velocity command $\bar{u}$ to drive the actuators such that the normalized displacement x goes from the current position $x(t)$ to the origin $x(T) = 0$ with the goal of minimizing the final time T. The operator input $\bar{u}$ is assumed to follow the operator input transition law given by (10.1). Given the normalized displacement x with dynamics

$$\dot{x} = -u$$

with the requirement that $u \in U$, where U is the region of allowable control inputs derived in Chapter 9 to be

$$U = \left\{ u : 0 \leq u_k \leq D_k, \quad \psi^T u \leq C \right\}$$

and ψ, D, C are positive and constant over the motion. The operator input $\bar{u}$ is assumed to be feasible $\bar{u} \in U$. An infeasible $\bar{u}$ is projected to U using Equation (9.3).

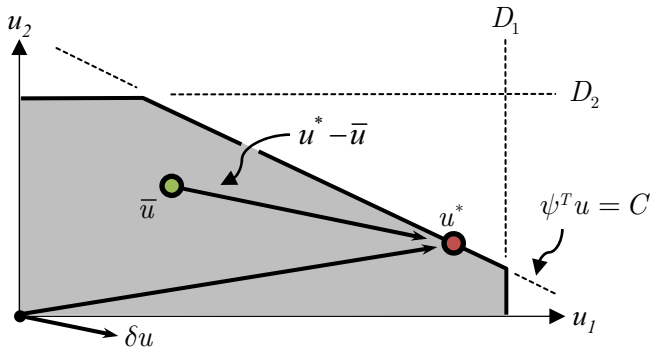


Figure 10.3: Proportional command perturbation

10.3.2 The Proportional Blended Shared Control Law

This section gives a form of δu that is proportional to the error vector between the operator input and the optimal command, so

$$\delta u = a(u^* - \bar{u})$$

This will be known as the *Proportional Shared Control Perturbation* (PSC). By Theorem 3, the PSC law is shown to reduce the task completion time compared to the unperturbed command. A geometric interpretation of this control law is shown in Figure 10.3 for the planar case.

Theorem 3. *Let $a \in (0, 1)$ be given. For the system described in the Basic Problem Assumptions, a control perturbation of the form*

$$u = \bar{u} + \delta u$$

with

$$\delta u = a(u^* - \bar{u})$$

(where $u^* = \frac{C}{\psi^T x} x$), will be an allowable control perturbation (i.e, $u \in U$) and will have a task completion time that never exceeds the unperturbed task completion time:

$$T(\bar{u} + \delta u, x) \leq T(\bar{u}, x)$$

Proof. Outline of proof: it will be shown that for some components, the component task completion time will decrease for all $a \in (0,1)$, but will be bounded from below by the minimum task time T^*. For the remaining components, the control perturbation will cause the task completion time to increase for all $a \in (0,1)$; however, this increase will be bounded above by the minimum task time T^*. Since the overall task completion time is the time it takes for the *last* function to reach the origin, that is

$$T(u,x) = \max_i (T_i(u_i, x_i))$$

an overall reduction in task time can be shown.

The minimum task completion time for any x is $T^* = \psi^T x/C$ from Theorem 1 on page 102 in Chapter 9. Denote the single-function task completion time of function k as $T_k(u,x)$, where x is the initial state and u is the input. Without the perturbation δu, and assuming the operator input follows (10.1), then the task time for the kth component is

$$T_k(\bar{u}, x) = \frac{x_k}{\bar{u}_k} \tag{10.2}$$

With an applied command perturbation $\delta u = a(u^* - \bar{u})$ the task time is

$$T_k(\bar{u} + \delta u, x) = \frac{x_k}{\bar{u}_k + a\left(\frac{C}{\psi^T x} x_k - \bar{u}_k\right)}$$

Using (10.2) this task time is written

$$T_k(\bar{u} + \delta u, x) = \frac{T_k(\bar{u}, x)}{1 + a\left(\frac{u_k^*}{\bar{u}_k} - 1\right)}$$

which, after rearranging the fractions, becomes

$$\frac{T_k(\bar{u}, x)}{T_k(\bar{u} + \delta u, x)} = 1 + a\left(\frac{u_k^*}{\bar{u}_k} - 1\right)$$

If $\bar{u}_k \leq u_k^*$ then the term in parentheses is not negative, so for any choice of $a > 0$

$$\frac{T_k(\bar{u}, x)}{T_k(\bar{u} + \delta u, x)} \geq 1$$

which implies

Claim 1:

$$T_k(\bar{u} + \delta u, x) \leq T_k(\bar{u}, x) \qquad (\forall k \text{ s.t. } \bar{u}_k \leq u_k^*)$$

That is, the task completion time is reduced as a result of the SC perturbation for each function having an operator input less than optimal.

Further, the function $T_k(\bar{u} + \delta u, x)$ is a monotonic function in the parameter a for $a \in (0, 1)$. The function is either increasing or decreasing, depending on the relative magnitude of $\bar{u}_k$ and u_k^*. At the limit $a = 1$, the task time is $T_k(\bar{u} + \delta u) = T^*(x)$ for each component k. This implies

Claim 2:

$$T_k(\bar{u} + \delta u, x) < T^* \qquad (\forall k \text{ s.t. } \bar{u}_k > u_k^*)$$

That is, the task completion time is bounded from above by T^* for those components having operator input greater than optimal.

Similarly for the components having operator input less than optimal:

$$T_k(\bar{u} + \delta u, x) > T^* \qquad (\forall k \text{ s.t. } \bar{u}_k < u_k^*)$$

which implies

Claim 3:

$$T^* \leq \max_i \left(T_i(\bar{u}, x)\right) = T(\bar{u}, x)$$

That is, the actual time required for *all* functions to reach the origin is bounded from below by the optimal time.

Thus, even though the control perturbation δu will increase the completion time for those components having $\bar{u}_k > u_k^*$, Claim 3 and Claim 2 guarantee that this perturbed time

will be less than the original task time $T(u, x)$. All the remaining components will have $\bar{u}_k \leq u_k^*$, for which Claim 1 guarantees the task time is reduced.

Therefore

$$T_k(\bar{u} + \delta u, x) \leq \max_i(T_i(\bar{u}, x)) \quad \forall k$$

□

10.4 Constant Flow Rate Shared Control Law

With input $\bar{u}$, the operator commands a total pump flow $\bar{Q} = \psi^T \bar{u}$. There are circumstances where it is desired to not change the net flow delivered by the pump. For example, in a system with multiple actuators there may be a fixed allocation of flow for a particular subgroup of functions. The flow due to the perturbed command equals $\bar{Q}$, so, with this requirement that the flow is unchanged, the induced constraint equation $\psi^T(\bar{u}+\delta u) = \bar{Q}$ must always be satisfied. If the proportional Blended SC law were used, that is, $\delta u = a(u^* - \bar{u})$, then only the trivial solution $\delta u = 0$ is an admissible constant-flow command perturbation when $\psi^T \bar{u} \neq C$. This section will introduce another class of command perturbations which satisfy the constant flow constraint but still reduce the task completion time.

Any control perturbation that lies on the orthogonal projection of ψ will not change the flow rate, since by definition $\psi^T \delta u = 0$ for all δu in the orthogonal projection of ψ.

A special case of the constant-flow projection is considered here. Let $(u^*)'$ be the intersection between the hyperplane $\psi^T \bar{u}$ (also an orthogonal projection of ψ) and the line from the origin to u^*, as in Figure 10.4, so that

$$(u^*)' = \frac{\psi^T \bar{u}}{\psi^T x} x$$

where, as always, x is the coordinate of the actuators. The point $(u^*)'$ is admissible for all x and $\bar{u}$, and, since $(u^*)'$ is on the line from the origin to u^*, it is stationary for $u = (u^*)'$ (see Remark 1 on page 103 of Chapter 9). The command perturbation δu is chosen to be

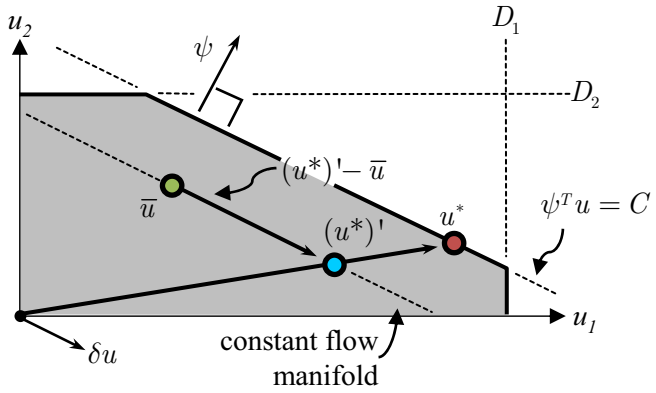


Figure 10.4: Constant-flow command perturbation

proportional to the vector $(u^*)' - \bar{u}$ as in

$$\delta u = a\left((u*)' - \bar{u}\right) = a\left(\frac{\psi^T \bar{u}}{\psi^T x}x - \bar{u}\right)$$

Theorem 4. *Let $a \in (0,1)$ be given. For the system described in the Basic Problem Assumptions, a control perturbation of the form*

$$u = \bar{u} + \delta u$$

with

$$\delta u = a\left(\frac{\psi^T \bar{u}}{\psi^T x}x - \bar{u}\right) \tag{10.3}$$

will be an allowable control perturbation (i.e., $\bar{u}+\delta u \in U$) and will result in a task completion time that never exceeds the unperturbed task completion time:

$$T(\bar{u} + \delta u, x) \leq T(\bar{u}, x)$$

Further, the weighted sum of velocities $\psi^T \bar{u}$ (i.e., the system pump flow) will be unchanged by this perturbation.

Proof. Before the control perturbation, the weighted sum of velocities is

$$\bar{Q} = \psi^T \bar{u}$$

With the control perturbation, the weighted sum of velocities is

$$\bar{Q}' = \psi^T (\bar{u} + \delta u) = \bar{Q} + \psi^T \delta u$$

but the term $\psi^T \delta u$ becomes

$$\psi^T \delta u = a\psi^T \left(\frac{\psi^T \bar{u}}{\psi^T x} x - \bar{u} \right) \equiv 0$$

and hence

Claim 1: The weighted sum of velocities is unchanged by the control perturbation

Using a result from Theorem 1 on page 102 of Chapter 9, the minimum task completion time for any x and a constant flow rate $\bar{Q}$ is

$$T^*(\bar{Q}) = \frac{\psi^T x}{\bar{Q}}$$

Define the point u' to be the intersection between the constant-flow constraint manifold $\psi^T u = \bar{Q}$ and the unconstrained optimal input given as

$$u^* = \frac{C}{\psi^T x} x$$

from Theorem 1. Since u' is on the line between the origin and u^*, one can write $u' = \lambda u^*$. Imposing the constraint

$$\psi^T (u') = \psi^T \bar{u} = \bar{Q}$$

gives $\lambda = \psi^T \bar{u}/C$, so that

$$u' = \frac{\psi^T \bar{u}}{\psi^T x} x$$

Denote the single-function task completion time of function k as $T_k(u, x)$, where x is

the initial state and u is the switched input. Without the perturbation δu, the task time is

$$T_k(\bar{u}, x) = \frac{x_k}{\bar{u}_k}$$

and with the perturbation δu the task time is

$$T_k(\bar{u} + \delta u, x) = \frac{x_k}{\bar{u}_k + a\left(\frac{\psi^T \bar{u}}{\psi^T x} x_k - \bar{u}_k\right)}$$

After some manipulation, consider

$$\begin{aligned} \frac{T_k(\bar{u}, x)}{T_k(\bar{u} + \delta u, x)} &= 1 + a\left(\frac{\psi^T \bar{u}}{\psi^T x}\frac{x_k}{\bar{u}_k} - 1\right) \\ &= 1 + a\left(\frac{u'_k}{\bar{u}_k} - 1\right) \end{aligned}$$

If $\bar{u}_k < u'_k$, then the RHS of the equation above is greater than 1, which implies

Claim 2:

$$T_j(\bar{u} + \delta u, x) \leq T_j(\bar{u}, x), \quad (\forall j \text{ s.t. } \bar{u_j} \leq u'_j)$$

Further, the function $T_k(\bar{u} + \delta u, x)$ is a monotonic function in the parameter a for $a \in (0, 1)$. The function is either increasing or decreasing, depending on the relative magnitude of $\bar{u}_k$ and u^*_k. At the limit $a = 1$, the task time is $T_k(\bar{u} + \delta u) = T^*(x)$ for each component k. This implies

Claim 3:

$$T_k(\bar{u} + \delta u, x) < T^* \qquad (\forall k \text{ s.t. } \bar{u}_k > u^*_k)$$

That is, the task completion time is bounded from above by T^* for those components having operator input greater than optimal.

Similarly for the components having operator input less than optimal:

$$T_k(\bar{u} + \delta u, x) > T^* \qquad (\forall k \text{ s.t. } \bar{u}_k < u^*_k)$$

which implies

Claim 4:

$$T^* \leq \max_i(T_i(\bar{u}, x) = T(\bar{u}, x)$$

That is, the actual time required for *all* functions to reach the origin is bounded from below by the optimal time.

Thus, even though the control perturbation δu will increase the completion time for those components having $\bar{u}_k < u_k^*$, Claim 4 and Claim 3 guarantee that this perturbed time will be less than the original task time $T(\bar{u}, x)$. For all the remaining components ($\bar{u}_k \geq u_k^*$), Claim 2 guarantees the task time to be reduced. Claim 1 ensures that the weighted sum of velocities is unchanged.

Therefore

$$T_k(\bar{u} + \delta u, x) \leq \max_i(T_i(\bar{u}, x)) \quad \forall k$$

and

$$\psi^T \delta u = 0$$

□

10.5 Discussion About the Direct Shared Control Perturbations

By varying the SC parameter a, the degree of control authority attributed to the HA and EA is scaled. For example, when high resolution of motion is necessary, one may decrease the influence of the EA by choosing a smaller value for a. Also, when unrestricted motions in a large obstacle-free workspaces are involved, the EA may be given greater authority. Generally, a higher degree of autonomy is warranted if the task is well defined, performed in a known environment, is described by a small number of motion primitives, and/or has long time delays [112]. With excavation in particular, the environment and task requirements are unpredictable, so a lesser degree of autonomy may be warranted for some portions of the dig cycle.

The perturbation-based shared control approaches introduced in this chapter are guaranteed to reduce the task completion time for all x and $\bar{u}$ that satisfy the basic assumptions.

While the Proportional and Constant Flow perturbations have a convenient geometric interpretation in the u-plane, they certainly are not unique. There likely exist many other formulations for δu which also reduce the task completion time. More complicated forumlations of δu were evaluated in the course of this research. These included a persuit-evasion formulation, whereby the operator input $\bar{u}$ was perturbed by δu in such a way to produce a minimum-time interception of the optimal point u^*. Proving the cost benefit—and even convergence—of these other approaches was much more involved than the Proporitonal and Constant-flow perturbations presented earlier.

While the assumptions on the operator inputs will allow for convenient analysis, it should be noted that strict performance optimality can not be guaranteed, since there is no way of guaranteeing one human response versus another. Thus, one must also refer to the experiments presented in Chapter 11 to understand the true effect of Blended SC on the task cost.

10.6 Choosing the SC Parameter a

As yet, the only restriction placed on the SC parameter a is that $a \in (0, 1)$. Theorems 3 and 4 guarantee that any a within the allowable domain will result in a lower task. However, the control perturbations must always be "acceptable" to the operator. In some cases, it may be necessary to maintain the control perturbations below some critical level. In another case, it may be prudent to limit the resulting change in perceived end effector velocity so as to remain below a tolerable, noticeable level or even to keep the velocity change below what is even noticeable to the operator. This section discusses some alternatives for choosing a in order to satisfy certain constraints.

Here, several alternatives for restricting a are given. Rather than constraining the vector δu directly, it may be prudent to first transform it with either a linear mapping A. Such a transformation may be useful, for example, if the units of measure of certain components differ or if deviations of a particular actuator are of greater concern than others.

All of the SC formulations require that $a \in (0, 1)$. The additional constraints presented below are not to supersede this constraint. These will provide an additional upper-bound

on a.

10.6.1 Constraint class 1: The velocity difference of the cylinders is constrained

The norm provides a convenient distance measure for a vector space, and there are several types of norms that can be used. Here, the p-norm of δu is constrained such that

$$\|A\delta u\|_p < b$$

where A is a constant square matrix used to weight the elements of δu. When δu is defined as

$$\delta u = a\,(u^* - \bar{u})$$

then the upper bound of a is found

$$a < \frac{b}{\|A\,(u^* - \bar{u})\|_p}$$

In other cases, constraints to changes in total pump flow are necessary. For a given system pressure, this is equivalent to constraining the change in mechanical power delivered by the prime mover. If, for example, the increase in system flow is limited by ΔQ, then the allowed change in actuator velocity is limited by

$$\psi^T \delta u < \Delta Q$$

which gives

$$a < \frac{\Delta Q}{\psi^T\,(u^* - \bar{u})}$$

10.6.2 Constraint class 2: The task-space velocity is constrained

Instead of constraints on the changes in actuator velocities δu, constraints can apply to the change in workspace velocity $\dot{X}$ of some point on the linkages themselves, such as the end effector. This makes sense since the operator is typically concerned not with the motion of the actuators but with the motion of the linkages. Further, the operator may be more

sensitive to deviations in one direction more than another. For example, deviations in motions in the fore/aft direction may be less noticeable than equivalent motions in the left/right direction (where the operator not only "sees" the change in velocity, but may also feel it, since often the cab rotates with the end effector). This effect is illustrated in Figure 10.5 by the elliptical constraint for deviations in end effector velocity.

The task-space end-effector coordinate velocities $\dot{X}$ are related to the actuator velocities $\dot{q}$ by

$$\dot{X} = J\Big|_q \frac{\partial \Gamma}{\partial q}\Big|_q \dot{q}$$

where J is the (state-dependent) Jacobian relating the end-effector linear velocity $\dot{X}$ to the joint velocities $\dot{\theta}$ and Γ is the function to resolve the actuator-space variables to the joint-space variables. The notation $(\cdot)|_q$ denotes evaluation of the enclosed term at the point q. Both J and Γ are evaluated at the current actuator configuration q. Thus, some measure of the change in end-effector velocity is limited, as in

$$\left\| A\delta\dot{X} \right\|_p < V \tag{10.4}$$

$$\left\| AJ|_q \frac{\partial \Gamma}{\partial q}\Big|_q \delta u \right\|_p < V \tag{10.5}$$

In this case a is readily obtained

$$a < \frac{V}{\left\| AJ|_q \frac{\partial \Gamma}{\partial q}\Big|_q (u^* - \bar{u}) \right\|_p}$$

Recall that $\|(\cdot)\|_p$ denotes the p-norm of $(\cdot)$. As before, a weighting matrix may be added to the above constraint to account for the fact that an operator's just noticeable difference (JND) may be a function of the configuration of the system, as well as the direction of motion. For example, the *in-out* direction may be less sensitive to the *left-right* direction.

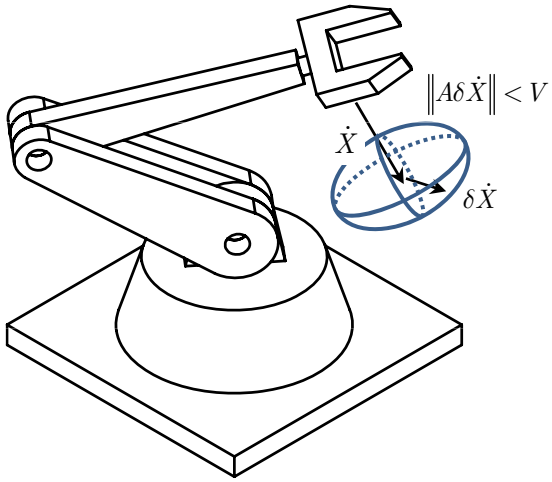


Figure 10.5: Constraints on the allowed change in end-effector velocity can be visualized as an ellipse in the task space.

10.6.3 Constraint class 3: The effective joystick angle deviation is constrained

For a given perturbation δu, there is an effective joystick angle deviation. Recall from Chapter 8 the actuator reference velocity $\bar{u}$ is related to the operator's joystick displacement r by

$$\bar{u} = R(r)$$

where $r \in [-1, 1]$ is the vector of normalized joystick displacements. Typically the components of $\bar{u}$ are independent so that each component $\bar{u}_k$ only depends on r_k. The function R maps the percentage of joystick throw along a particular axis to some fraction of the maximum (or minimum) speed for that function. Usually R is a constant monotonic function calibrated such that the maximum and minimum value of its range correspond to the maximum and minimum achievable velocity of the corresponding function.

A change in actuator velocity δu can be related to an effective change in joystick input r. The effective change in joystick input can be constrained. For instance, there is a finite precision with which a human hand can position a joystick [113]; this produces a region within which it may be permissible to deviate, as illustrated in Figure 10.6. Presumably, the operator is accustomed to the deviation stemming from the operator's own limitations. Knowledge of this tolerance in the r-space can be mapped to a corresponding range for δu.

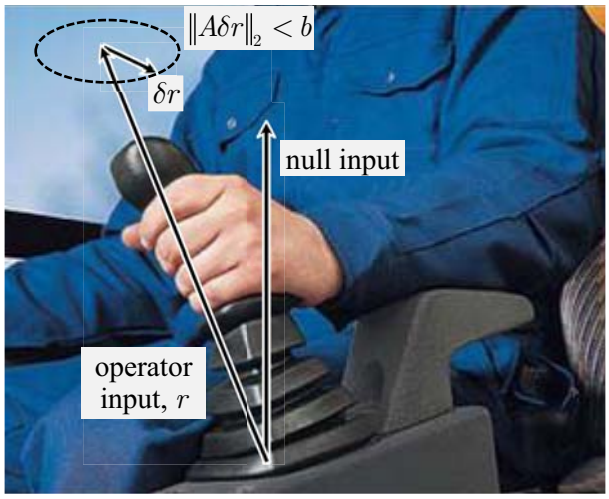


Figure 10.6: The constraint on joystick deviation mapped to the joystick input planes

Whether this is an appreciable effect needs to be determined.

For a given perturbation in the joystick input there will be a corresponding change in the commanded velocity

$$\bar{u} + \delta u = R(r + \delta r)$$

By assuming the change in joystick input to be small, R is expanded by retaining the linear terms of its Taylor series to give

$$\bar{u} + \delta u = R(r) + \left.\frac{\partial R}{\partial r}\right|_r \delta r$$

Recognizing that $\bar{u} = R(r)$, the joystick perturbation can be solved for in terms of the equivalent speed perturbation δu to give

$$\delta r = \left(\left.\frac{\partial R}{\partial r}\right|_r\right)^{-1} \delta u$$

The matrix $\partial R/\partial r$ is invertible everywhere outside the deadband.

Assuming the effective joystick input deviation δr is constrained according to

$$\|A\delta r\|_p < V$$

which equals

$$\left\| A \left(\frac{\partial R}{\partial r}\Big|_r \right)^{-1} \delta u \right\|_p < V$$

Using the assumption that the perturbation to the function velocity is $\delta u = a(u^* - u)$, the parameter a is found to be

$$a < \frac{V}{\left\| A \left(\frac{\partial R}{\partial r}\big|_r \right)^{-1} \delta u \right\|_p}$$

10.6.4 Discussion

None of the restrictions supersede the requirement that the SC parameter remain within the interval $(0, 1)$. Some directions, whether interpreted in joystick r-space, cylinder q-space, or manipulator task space X may have a greater tolerance for velocity deviations than other directions. For example, an operator may be more perceptive to deviations in the slew rate since the operator rotates with in the cab; in this case, the corresponding diagonal elements of A could be chosen to weight the swing components by a greater amount than the other functions.

Choosing the nominal value for a: The restrictions above provide an upper bound for a. The control engineer is still faced with choosing a nominal value for a.

a as a constant: Perhaps chosen based on the expected task type. All else equal, repetitive tasks for which the task can be easily identified would have a higher a than unpredictable tasks.

a as a function of task certainty An extension not discussed here is varying a based on the estimation accuracy. If there is high confidence in the prediction of the operator's task, then this can be a cue to increase the influence of the EA. Similarly, more authority is automatically provided to the operator if the cycle is unpredictable. Estimation accuracy may be measured in terms of the covariance matrices in the RAMPE algorithm, or by simply comparing the mean square error of the past few cycles. Another way of measuring task certainty is to consider the difference between $\bar{u}$ and u^*, as was done in the SC of Zermelo's problem (Chapter 5). A large difference could be

attributed either to operator ignorance (i.e., the EA *knows* the operator's task, while the operator does not know how to implement it) or the EA mis-identifying the task.

a **manually set by the operator** The operator chooses a setpoint to specify the level of autonomy to grant the EA.

10.7 Indirect Shared Control

With *direct* SC, the HA command $\bar{u}$ is modified through software by the EA. In contrast, *indirect* SC relies on the HA to actually change the input. This change in input is done according to feedback displayed to the HA. The EA calculates the specific nature of the stimulus to suggest the best manner by which the HA should respond.

10.7.1 Haptic Cues

A haptic cue is displayed to suggest to the operator the direction in which the inputs should be changed. Similar to the approach used for the proportional SC perturbations, the displayed force is in the direction of u^*. Since the force acts in the joystick r-space, the optimal command u^* must be transformed into this space. Let the optimal joystick displacement be

$$r^* = R^{-1}(u^*)$$

where R is the function mapping the joystick displacements to the nominal velocity commands, as described in Chapter 8.

The selection of the particular *direction* that the operator's hand is pushed in has several options. For example, one may be able to model the dynamics of the operator input in response to force provided by the haptic cue. If an adequate model is known, then the force can be directed in such a way that intercepts the *moving* point u^* in minimum time. The general problem is one of target pursuit/interception, the simplest strategy of which is a line-of-sight strategy, as discussed next.

Choose the force so that the operator's hands are pushed in the direction of the optimal

input. In the joystick r-space, this force will then be

$$F = aF_0 \frac{r^* - \bar{r}}{\|r^* - \bar{r}\|} \tag{10.6}$$

where $a \in (0, 1)$ is the SC parameter that scales the nominal (and constant) force magnitude F_0. (Note that conventional excavators are controlled by two independent joysticks; thus the joystick F must be projected into the separate joystick input planes.)

10.7.2 Choosing the SC Parameter a

The force magnitude must be carefully chosen to avoid destabilizing the system or confusing the operator. Transitions between motion primitives are not guaranteed to result in continuous u^*. Consequently, $F(t)$ may be discontinuous at the transition time. Here, the force magnitude is ramped down as while approaching the end of the motion primitive. The SC parameter a in (10.6) scales the force magnitude; its nominal value is

$$a_0 = \begin{cases} 0 & \text{for} \quad \rho < \rho_{\min} \\ f(\rho) & \text{for} \quad \rho_{\min} \leq \rho \leq \rho_{\max} \\ 1 & \text{for} \quad \rho > \rho_{\max}. \end{cases}$$

where $\rho_{\min}$ and $\rho_{\max}$ are thresholds for activating and deactivating the force, and the function $f(\rho)$ is a smooth, monotonic function such that $f(\rho_{\min}) = 0$ and $f(\rho_{\max}) = 1$. The distance ρ is computed based on the expected remaining actuator displacement

$$\rho = \sqrt{x^T \left(A^T A\right) x}$$

where, as before, the matrix A weights the individual components of x.

At the start of a new motion primitive, x changes in magnitude discontinuously. To prevent a sudden increase in force, changes to a are *rate limited* so that $|\dot{a}| < \gamma$ where $\gamma > 0$

is the maximum slew rate. At runtime, the SC parameter is computed as

$$a(t) = \begin{cases} a_0(t) & \text{for} \quad |\dot{a}| \leq \gamma, \\ a(t - \Delta t) - \gamma \Delta t & \text{for} \quad \dot{a} < -\gamma, \\ a(t - \Delta t) + \gamma \Delta t & \text{for} \quad \dot{a} > \gamma. \end{cases}$$

where Δt is the sampling interval.

10.8 Discussion and Chapter Summary

This chapter presented three perturbation-based Blended SC formulations, summarized in Tables 10.2. Modest assumptions on the dynamics of the operator command $\bar{u}$ allowed for an exploration of the effect that these perturbations has upon the task completion time. The perturbations follow simple, consistent laws as summarized in Table 10.2a.

In each formulation, the shared control parameter a must be chosen. The optimality conditions required that $a \in (0, 1)$. When $a = 0$, the Human Agent has full control of the manipulator velocity and when $a = 1$, the Electronic Agent has full control. Table 10.2b summarizes several formulations of constraints on a; these constraints on a are induced from constraints defined within different domains, including the actuator-space, the task-space, and the joystick-space.

Regardless of the value of a, the architecture only acts in response to an operator input. Thus, even when $a = 1$ and the EA has full control of the manipulator, the commands which the EA issues are based upon the commands of the HA. Thus, if the HA is commanding a null motion (by not moving the joysticks), the EA will issue a null command. Similarly, if the HA commands the boom and arm cylinders to retract, then the EA must also only give commands that cause the boom and arm cylinders to retract. These convenient constraints are a result of defining the manipulator task using the motion primitive formulation discussed in Chapter 7. Therefore, when the EA is given full authority, that authority is inherently restricted to a u-plane which is defined by the operator's present inputs. This has interesting implications, in that now the HA can use the displacements of the joysticks to "conduct" a large variety of autonomous manipulator motions, in a manner reminiscent

of the way a maestro conducts the flow of a melody.

The Blended SC formulation discussed here assumes that nominal operator commands are perturbed by an autonomous Electronic Agent. The contrasting case is when nominal autonomous trajectories are perturbed by an operator. The architecture developed in this chapter is compatible with both approaches, and an interesting extension is to consider the alternate case.

The design space for the Blended Shared Control is large, and rich with interesting possibilities.

Table 10.2: Summary of parameters used for Blended Shared Control

(a) Control perturbations for Blended Shared Control.

Type	δu	Notes
Proportional (PSC)	$a\,(u^* - \bar{u})$	In the u-plane, the perturbation is in the direction of the optimal input u^*
Constant Flow (CFSC)	$a\left(\frac{\psi^T \bar{u}}{\psi^T x} - \bar{u}\right)$	The perturbation lies in the orthogonal complement of ψ. The net pump flow is unchanged by the control perturbation
Haptic Cue (HC)	$F = aF_0 \frac{R^{-1}(u^*) - \bar{r}}{\|R^{-1}(u^*) - \bar{r}\|}$	A force-reflecting joystick applies force F in the direction implies a change in $\bar{u}$ towards the direction of u^*

(b) Summary of limits on shared control parameter a given induced constraint on other quantities.

Constraint relation	Interpretation	Limits on a
n/a	constant weighting on control authority	$a \in (0, 1)$, constant
$\|\|A\delta u\|\|_p < b$	limit the weighted change in actuator speed	$a < C\left(\|A\,(u^* - u)\|_p\right)^{-1}$
$\psi^T \delta u < \Delta Q$	limit the change in pump flow	$a < \Delta Q\left(\psi^T (u^* - \bar{u})\right)^{-1}$
$\|\|A\delta\dot{x}\|\|_p < V$	limit the perceived change in task-space velocity of a point on end effector	$a < V\left(\left\|AJ\|_q \frac{\partial \Gamma}{\partial q}\Big\|_q (u^* - u)\right\|_p\right)^{-1}$
$\|\|A\delta r\|\|_p < V$	constrain the effective joystick deviation	$a < V\left(\left\|A\left(\frac{\partial R}{\partial r}\big\|_r\right)^{-1} \delta u\right\|_p\right)^{-1}$

CHAPTER XI

EXPERIMENTS: EFFECTS OF SHARED CONTROL TYPE ON MEAN TASK TIME

This chapter documents experiments undertaken to show the efficacy of the Blended Shared Control paradigm. Several tasks involving point-to-point motion were undertaken, with the primary measure being the task completion time. Four sets of experiments were performed with a total of 33 subjects divided into three skill levels. Up to four different conditions were studied in each experiment set. Each condition was unique in the type of control law that was active during the task. The apparatus used for the experiments is discussed in Section 11.3, followed by a description of the four types of experiments in Section 11.4. Section 11.5 describes the experimental procedure. An analysis of the results of all the experiments is given in Section 11.6. Finally, the results are discussed in Section 11.7.

11.1 Nomenclature

JS Joystick

4DOF Refers to the two-joystick, four degree-of-freedom set of experiments

2DOF Refers to the single-joystick, two degree-of-freedom set of experiments

P2P Point-to-point set of experiments

UGM Unconstrained Guided Motion set of experiments (*Trenching*, *Reaching*, *Four-point*, *Three-point*

MC Manual control

SC1 Shared control, Type 1; Proportional SC with constrained velocity change, $|A\delta u|_\infty < b$

SC2 Shared control, Type 2; Constant flow SC, $\psi^T \delta u = 0$

HC Haptic cue, (Indirect shared control)

Ω The direction or category of a motion primitive

$\Omega = 2021$ Denotes that the the swing is increasing, the boom is stationary, the arm is increasing, and the bucket is decreasing

ψ Vector of flow-velocity ratio of each function. Describes the share of pump flow that goes to each function

q Actuator position coordinates

11.2 Method

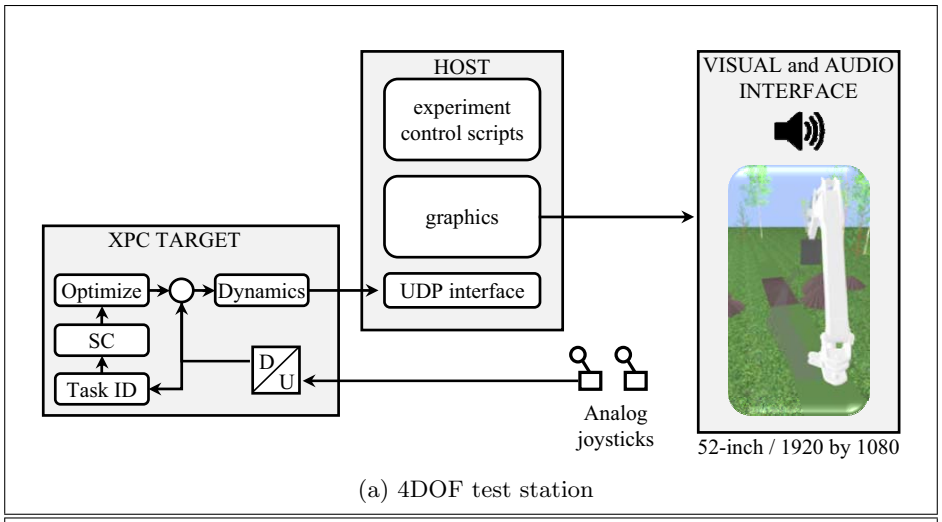
Blended SC was tested on a virtual compact excavator. The primary objective was to determine the effect of SC on task time. The study included two types of Blended SC, along with Manual Control (MC) and Haptic Cues (HC). Two major sets of experiments were performed: constrained point-to-point motion (P2P) and unconstrained guided motions (UGM). Tests were performed on two experimental platforms: a two degree-of-freedom (2DOF) platform and a four degree-of-freedom (4DOF) platform.

11.3 Experiment Setting

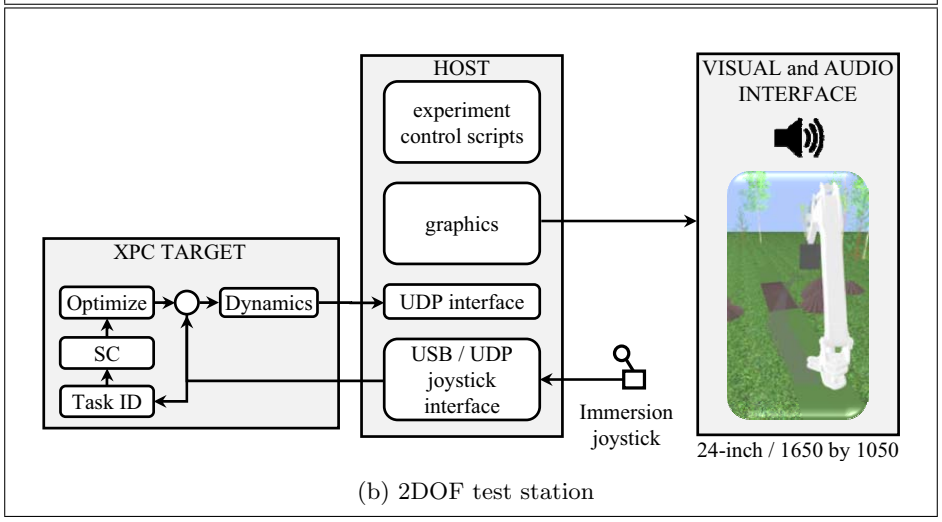
A modular software/hardware platform for studying the interface between operators and excavators has been developed at the Intelligent Machine Dynamics Laboratory by Elton and Kontz [35, 36, 49]. The software system consists of a host and a target computer, as shown in Figure 11.1. The hardware consists of an input device and a display device.

During the 4DOF experiments, the operator uses two standard excavator joysticks in the ISO configuration (left JS: boom and swing; right JS: arm and bucket) to control all four excavator functions. The operator sits within the cockpit of an actual Bobcat 435 compact excavator that has a large screen display mounted just in front of the windshield, as shown in Figure 11.3.

For the 2DOF experiments, only the boom and swing are actuated, thus requiring a single JS. The JS is a a force-feedback enabled *Impulse Engine 2000* joystick from Immersion Inc, capable of displaying up to 8.7 N in any direction within the plane of JS displacement. The operator sits at a work station, pictured in Figure 11.2, and views a 24 inch monitor. Both test stations were setup in the MaRC Highbay Laboratory at Georgia Tech.



(a) 4DOF test station



(b) 2DOF test station

Figure 11.1: Communication architecture for the two testing stations

The underlying software routines for Task ID, Optimization, Blended Shared Control, Dynamics simulation, and Graphics Display are essentially identical for both the two- and four-DOF scenarios. The Blended SC architecture and dynamic simulation are coded in C++ and MATLAB, and run in real time (at a 1 kHz sample frequency) on an xPCTarget machine. The graphics are written using OpenGL libraries and run on a standard Windows machine. The host and target communicate using UDP.

11.4 Experiment Overview

The study included two classes of experiments: constrained point-to-point motion (P2P) and unconstrained guided motion (UGM). Each class was studied on both of the interface

Figure 11.2: Operator station for the 2DOF tasks

stations. On the 4DOF platform, three different control modes were studied: Manual Control, and two types of shared control. The 2DOF tests had the same control modes, plus an additional Haptic Cue mode. The control type was the treatment variable, task completion time was the primary dependent variable. The test cases are described next.

11.4.1 Constrained Point-to-Point Motion (P2P)

The purpose of this class of tests is to measure the effect of shared control upon task completion time, for the special case where the task is precisely known by the Human Agent and the Electronic Agent. The actuator displacement between the initial and final configurations is described by a single motion primitive, $P_{(}^{\Omega}x)$. The Task Identification module of the Blended SC framework is given the motion primitive parameters, so no identification is necessary.

During the test, a sequence of targets are displayed on the screen to represent the desired final position of the manipulator, as shown in Figure 11.4. The operators are instructed to

(a) Cab of Bobcat 435

(b) View of operator inside test station

(c) Highlight showing screen display

Figure 11.3: Operator station for the 4DOF tasks

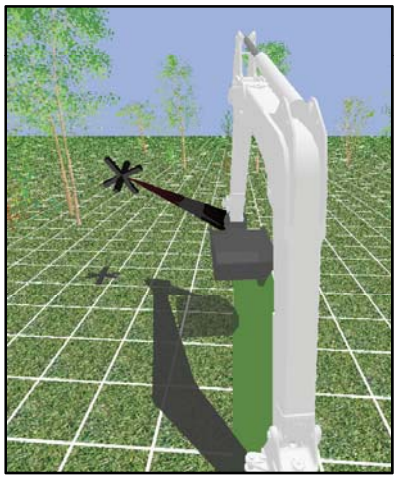


Figure 11.4: A frame from the video display during a trial of the point-to-point (P2P) tasks. The target denotes the desired final position of the excavator wrist. An arrow is always drawn between the wrist and the target to provide a sense of guidance and depth.

use the excavator bucket to quickly "capture" each of the targets as they appear on screen. An arrow drawn between the excavator wrist and the target was displayed to help overcome the lack of true depth perception associated with using a 2D display.

11.4.2 Unconstrained Guided Motion (UGM)

The manipulator motion during the UGM tests are not tightly constrained like the P2P set. Specifically, the Electronic Agent is provided no prior information about the tasks; therefore, the Task Identification module of the control architecture must learn and estimate the operator's intended motions. Much like actual digging or manipulation tasks, the human-controlled trajectory will vary between cycles, and the Task ID module must be robust to these variations.

The operators can move the excavator in any arbitrary manner. However, the scope of this research excludes non-repetitive trajectories, so giving the subjects no restrictions is not practical. To guide the operator while not imposing hard constraints, three to four static targets are displayed on screen. The idea is that the targets are not used as hard constraints as they were for the point-to-point class, but rather as reference markers for the operator to use during the motion. Each operator is instructed to control the excavator bucket through the quickest path which—*"roughly"*—passes through each target. Upon

reaching the last target, the operator retraces the path. The goal is to complete as many cycles as possible. There is no required tolerance for approaching a target, and the operator is not scored on accuracy.

The UGM tests have properties of actual, generic human-controlled manipulation tasks including high-level task specification, no hard constraints on the trajectory, task parameters that must be learned online by the Blended SC controller, and imprecise repetition from one cycle to the next.

11.4.3 Control Frameworks

One of four embodiments of the Blended Shared Control architecture is used during each trial. The goal of the experiments is to determine the effect of each control type on the task completion time. The four operation modes are listed below.

Manual Control (MC): The operator has full manual control of all degrees-of-freedom. This is accomplished by setting the Blended Shared Control parameter, a, to zero. The Manual Control (MC) mode is used as a control group for statistical analysis.

Shared Control - Type 1 (SC1): The Blended SC parameter a is chosen such that the proportional command perturbation $\delta u = a\,(u^* - u)$ satisfies the magnitude constraint

$$|A\delta u|_\infty < b$$

where A is chosen to scale δu relative to the actuator's maximum velocity for the particular operating mode

$$A = \mathrm{diag}(D_{SW}, D_{BM}, D_{AR}, D_{BK})$$

where $D_{SW}, D_{BM}, D_{AR}, D_{BK}$ are the maximum velocity of the swing, boom, arm, and bucket functions.

Shared Control - Type 2 (SC2): The constant-flow command perturbation described in Chapter 10 is used with $a = 0.5$.

Haptic Cue (HC): This mode displays haptic feedback to the operator to indirectly modify the excavator command. It is only implemented on the 2DOF test station. The force is directed towards u^* (after projection into the JS input space, r); the indirect shared control parameter a is chosen to be proportional to the distance $d = x^T x$ with

$$a = \begin{cases} 0 & \text{if } d < d_{\min}, \\ 1 & \text{if } d > d_{\max}, \\ \frac{d - d_{\min}}{d_{\max} - d_{\min}} & \text{otherwise.} \end{cases}$$

11.5 Procedure

A diverse set of subjects was recruited through various channels including professional contacts and posted advertisements. Advertisements were posted within the IMDL (targeting subjects with a novice-level understanding of excavators), the general community (targeting a group generally unfamiliar with excavators), and at a company which leases excavators and other heavy equipment (targeting experienced subjects). The subjects were compensated 20 dollars regardless of performance during the tests. A relatively large subject pool was deemed necessary since the effect of SC was known to potentially be small relative to the aggregate task time and the inherent variance associated with subjects of differing capabilities. Previous human-factors studies in robotics and manipulator control use 3 to 20 subjects [32, 55, 121, 147]. This section describes the specific testing procedure for each set of experiments.

11.5.1 Description of Testing Scenario

The experiment proceeded in phases, each of which is introduced here. Before beginning, the subject reads a description of the experiments, signs a consent waiver, and completes a brief preliminary assessment if the subject's skill level is unknown. The actual evaluations lasted approximately 70 minutes and proceeded the same order for each subject: two minute warm up, 2DOF P2P, two minute warm up, 2DOF UGM, 4DOF P2P, two minute warm

up, 4DOF UGM.[1] Each subject completed a final survey after the tests.

11.5.2 Initial Assessment

An initial survey instrument was used to classify the subjects into three groups based on the subject's experience in operating excavator-type machines. This assessment was given before the experiments begin, for the purpose of ensuring a uniform distribution of operators during the 4DOF UGM tests. Each subject was given a Subject ID number and classified based on their response to the following question.[2]

> *Relative to excavators and similar equipment, I describe myself as*
>
> - ***Unfamiliar*** - although I have seen an excavator, I have never operated a real or simulated excavator;
> - a ***Novice*** - I have a beginner-level knowledge on how excavators are controlled, and I have previously driven a real or simulated excavator for at least one hour;
> - an ***Expert*** - I have an above average level of knowledge related to excavators, and have operated an excavator, an excavator trainer/simulator, or similar equipment for at least 40 hours.

11.5.3 Final Assessment

The subjects completed an electronic survey after the experiments (see questions and responses listed in Appendix A). In sum, 33 subjects participated. The users' age ranged from 21 to 55; the average age was 28. Nine subjects were female. Three subjects were left handed. The sample space included 6 experts, 17 novices, and 10 unfamiliar subjects.

11.5.4 Experiment Procedure — Point-to-Point Motion (P2P)

A spinning, colored target was displayed on screen to show where the bucket should be at the completion of the motion. From the operator's perspective, the task is accomplished by

[1] See the Nomenclature section on page 134 for a summary of acronyms and symbols.

[2] Six subjects from the IMDL who had participated in at least two previous excavator studies were automatically classified into the Novice group.

moving the bucket to the target. A bell tone played when the operator was within a critical distance to the target, and the next target immediately appears.

The location of the targets were defined by the specific motion primitive required to reach the target. Table 11.1 lists the particular primitives used for the point-to-point tests. The motion primitive category, Ω, is defined by a four digit code; each digit denotes the *direction* a function moves in. From left to right, the digits represent Swing, Boom, Arm, and Bucket, with a 2 symbolizing *positive* motion, a 1 symbolizing *negative* motion, and a 0 representing no motion. For example, $\Omega = 1200$ denotes slewing left while extending the boom cylinder.

A trial consists of moving the actuators from an initial actuator configuration $q(0)$ to a final configuration defined by the motion primitive. The trials are paired with the motion primitive that returns the excavator to its original configuration. For example, the trial set A involves two individual trials: the first begins at with the actuators in configuration $q(0)$ and proceeds with a total displacement $x(0)$ in the direction $\Omega = 1200$ (slew to the left while extending the boom); the second also has a net displacement $x(0)$ but is in the *reverse* direction with $\Omega = 2100$ (slew to the right while lowering the boom). A *trial sequence* consisted of six repetitions of a *trial set*.

Every subject saw each trial sequence exactly once for each type of controller. The specific ordering within experiments on the 2DOF and 4DOF platforms are shown in Table 11.2 and Table 11.3, respectively. Subjects encountered the target trial sequences in the same order, but the order of the control types were arranged to balance any systematic effects.

11.5.5 Measures— Point-to-Point Motion (P2P)

A scalar measure of the remaining distance to the target is $d = \psi^T x$. The main dependent variable for each trial is the 90-percent cycle time illustrated in Figure 11.5, defined as the length of time between $d = 0.95d_0$ and $d = 0.05d_0$ where $d_0 = \psi^T x(0)$ is the initial distance to the target. Measuring cycle time in this way has several advantages; e.g., inadvertent motions are not penalized, the operator may begin when ready (the subject is not penalized for resting between targets), and the operator is not penalized by the time consuming

Table 11.1: Motion primitives used in the point-to-point motion experiments; x and q are given in inches.

Trial set	Ω	$x(0)$	$q(0)$	Platform
A	1200	$[27, 6, 0, 0]$	$[46, 161, 138, 126]$	2DOF
B	1200	$[50, 8, 0, 0]$	$[46, 161, 138, 126]$	2DOF
C	1100	$[50, 10, 0, 0]$	$[10, 164, 158, 119]$	2DOF
D	1200	$[17, 8, 0, 0]$	$[46, 161, 138, 126]$	2DOF
E	1110	$[43, 13, 20, 0]$	$[0, 164, 158, 119]$	4DOF
F	0220	$[0, 26, 26, 0]$	$[0, 137, 106, 126]$	4DOF
G	2022	$[50, 0, 65, 50]$	$[0, 153, 102, 78]$	4DOF
H	2001	$[44, 0, 0, 50]$	$[0, 161, 134, 126]$	4DOF

Table 11.2: Experiment design for 2DOF point-to-point motion tests

	Run number*															
Subj.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1, 17	0	1	2	3	1	2	3	0	2	3	0	1	3	0	1	2
2, 18	1	2	3	0	2	3	0	1	3	0	1	2	0	1	2	3
3, 19	2	3	0	1	3	0	1	2	0	1	2	3	1	2	3	0
4, 20	3	0	1	2	0	1	2	3	1	2	3	0	2	3	0	1
5, 21	3	2	1	0	2	1	0	3	1	0	3	2	0	3	2	1
6, 22	0	3	2	1	3	2	1	0	2	1	0	3	1	0	3	2
7, 23	1	0	3	2	0	3	2	1	3	2	1	0	2	1	0	3
8, 24	2	1	0	3	1	0	3	2	0	3	2	1	3	2	1	0
9, 25	0	2	1	3	2	1	3	0	1	3	0	2	3	0	2	1
10, 26	2	1	3	0	1	3	0	2	3	0	2	1	0	2	1	3
11, 27	1	3	0	2	3	0	2	1	0	2	1	3	2	1	3	0
12, 28	3	0	2	1	0	2	1	3	2	1	3	0	1	3	0	2
13, 29	3	1	2	0	1	2	0	3	2	0	3	1	0	3	1	2
14, 30	0	3	1	2	3	1	2	0	1	2	0	3	2	0	3	1
15, 31	2	0	3	1	0	3	1	2	3	1	2	0	1	2	0	3
16, 32	1	2	0	3	2	0	3	1	0	3	1	2	3	1	2	0
33	0	1	2	3	1	2	3	0	2	3	0	1	3	0	1	2

*For all subjects, the sequence of motion primitive pairs is:
A-B-C-D-A-B-C-D-A-B-C-D-A-B-C-D

Table 11.3: Experiment design for 4DOF point-to-point motion tests

	Run number*											
Subj.	1	2	3	4	5	6	7	8	9	10	11	12
1	0	1	2	0	1	2	0	1	2	0	1	2
2	1	2	0	1	2	0	1	2	0	1	2	0
3	2	0	1	2	0	1	2	0	1	2	0	1
4	0	2	1	0	2	1	0	2	1	0	2	1
5	2	1	0	2	1	0	2	1	0	2	1	0
6	1	0	2	1	0	2	1	0	2	1	0	2
7	0	1	2	0	1	2	0	1	2	0	1	2
8	1	2	0	1	2	0	1	2	0	1	2	0
9	2	0	1	2	0	1	2	0	1	2	0	1
10	0	2	1	0	2	1	0	2	1	0	2	1
11	2	1	0	2	1	0	2	1	0	2	1	0
12	1	0	2	1	0	2	1	0	2	1	0	2
13	0	1	2	0	1	2	0	1	2	0	1	2
14	1	2	0	1	2	0	1	2	0	1	2	0
15	2	0	1	2	0	1	2	0	1	2	0	1
16	0	2	1	0	2	1	0	2	1	0	2	1

	Run number											
Subj.	1	2	3	4	5	6	7	8	9	10	11	12
17	2	1	0	2	1	0	2	1	0	2	1	0
18	1	0	2	1	0	2	1	0	2	1	0	2
19	0	1	2	0	1	2	0	1	2	0	1	2
20	1	2	0	1	2	0	1	2	0	1	2	0
21	2	0	1	2	0	1	2	0	1	2	0	1
22	0	2	1	0	2	1	0	2	1	0	2	1
23	2	1	0	2	1	0	2	1	0	2	1	0
24	1	0	2	1	0	2	1	0	2	1	0	2
25	0	1	2	0	1	2	0	1	2	0	1	2
26	1	2	0	1	2	0	1	2	0	1	2	0
27	2	0	1	2	0	1	2	0	1	2	0	1
28	0	2	1	0	2	1	0	2	1	0	2	1
29	2	1	0	2	1	0	2	1	0	2	1	0
30	1	0	2	1	0	2	1	0	2	1	0	2
31	0	1	2	0	1	2	0	1	2	0	1	2
32	1	2	0	1	2	0	1	2	0	1	2	0
33	2	0	1	2	0	1	2	0	1	2	0	1

*For all subjects, the sequence of motion primitive pairs is:
E-F-G-H-E-F-G-H-E-F-G-H

"hunting" required to hit the target with the prescribed tolerance.[3]

The cycle time was automatically extracted offline by a suite of MATLAB scripts. The first trial of each primitive-controller pairing was discarded, with the remaining five trials averaged and recorded as the subject's mean task time for the particular primitive and controller.

11.5.6 Experiment Procedure— Unconstrained Guided Motion (UGM)

The UGM tasks consisted of four different scenarios named Three-Point Motion, Four-Point Motion, Trenching, and Reaching. The tasks were distinguished by the unique workspace trajectory specified by the targets which change according to the scenario, as specified in Table 11.4. The 3- and 4-Point Motions used the 2DOF simulator, while Trenching and Reaching were completed on the 4DOF platform.

The Trenching scenario required motions that mimicked a common trenching cycle, and requires actuation of all four DOFs. There were three guidance targets. One was located

[3]Clipping out the hunting time was necessary because a *post hoc* analysis revealed that an appreciable number of trials had hunting periods several times longer than the actual motion period. Further, the application domain considered here is less concerned with ultimate positioning accuracy, and more concerned with the time required to execute the the gross motion.

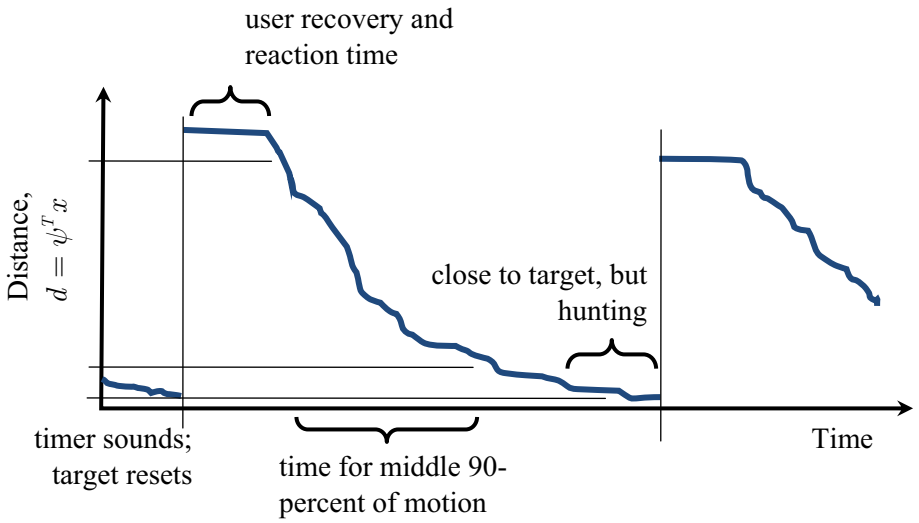


Figure 11.5: The 90-percent cycle time is defined as the time required for the weighted distance d to lose the middle 90-percent of its value. This value is used as a metric for task completion time.

Table 11.4: Location of targets in excavator workspace coordinates $[x, y, z]$ (in inches) for guided motion experiments. The target locations are sketched in Figure 11.6.

Trial Name	Target location	$[x, y, z]$	Platform
Four-point motion	Top of box	$[60, 45, 40]$	2DOF
	Over trench	$[120, 0, 50]$	2DOF
	Small box	$[30, -80, -10]]$	2DOF
	Fourth target	$[-15, -130, 0]]$	2DOF
Three-point motion	High above box	$[80, 85, 55]$	2DOF
	Over trench	$[120, 0, 20]]$	2DOF
	Small box	$[90, -110, -10]]$	2DOF
Trenching	Top of box	$[60, 45, 40]$	4DOF
	Back of trench	$[150, 0, -20]$	4DOF
	Front of trench	$[85, 0, -20]$	4DOF
Reaching	Top of box	$[60, 65, 40]$	4DOF
	Over trench	$[120, 0, 50]$	4DOF
	Bottom of small box	$[90, -110, -10]$	4DOF

atop a stack of crates; this target specified the location to unload the soil. Two additional targets were located on the ground at the start and ending locations of the trench. See Figure 11.6a. A single cycle consisted of starting above the boxes, entering the trench near the far target, digging in one motion until reaching near the forward target, and unloading the bucket above the initial target. The operator was instructed to perform as many cycles as possible. The scooping, removal, and unloading of soil was animated, but the soil reaction forces were disabled.

The Reaching scenario was not meant to be congruent with any particular real-world excavation task, apart from requiring simultaneous actuation of three functions (swing, boom, arm). Three targets were arranged as shown in Figure 11.6b: one located atop a stack of crates, another located above the trench, and a third located further away near the ground. A single cycle consisted of starting above the boxes, moving the bucket *over* the center target, reaching out with the arm to tap the far target, and returning (over the center target) to the start. The subjects were instructed to perform as many cycles as possible. Similarly, the Three- and Four-Point scenarios have target locations specified in Figure 11.6c and 11.6d.

Each operator was informed that the excavator was under their control, and that no constraints were present. Two minutes, using Manual Control, were allowed for warming up before the testing. Also, before each trial, instructions for the particular scenario were displayed on the screen.

A one-way repeated measures design was used for the 2DOF tests (swing/boom only), with comparisons made within subjects. Each value of the independent variable (the type of shared control) was seen by each subject. The practice and carryover effects associated with the trial sequences were lessened by blocking and balancing the experiment design. The orders are shown in Table 11.5. All subjects started on the 2DOF platform and alternated between the 3-Point scenario and the 4-Point scenario. The scenarios alternated to temper fatigue and learning effects. Each subject saw all scenarios four times, for a total of 8 trials per subject on the 2DOF platform. Each trial continued for 3 minutes.

A between groups experiment design was used for the 4DOF Guided Digging tests. Each

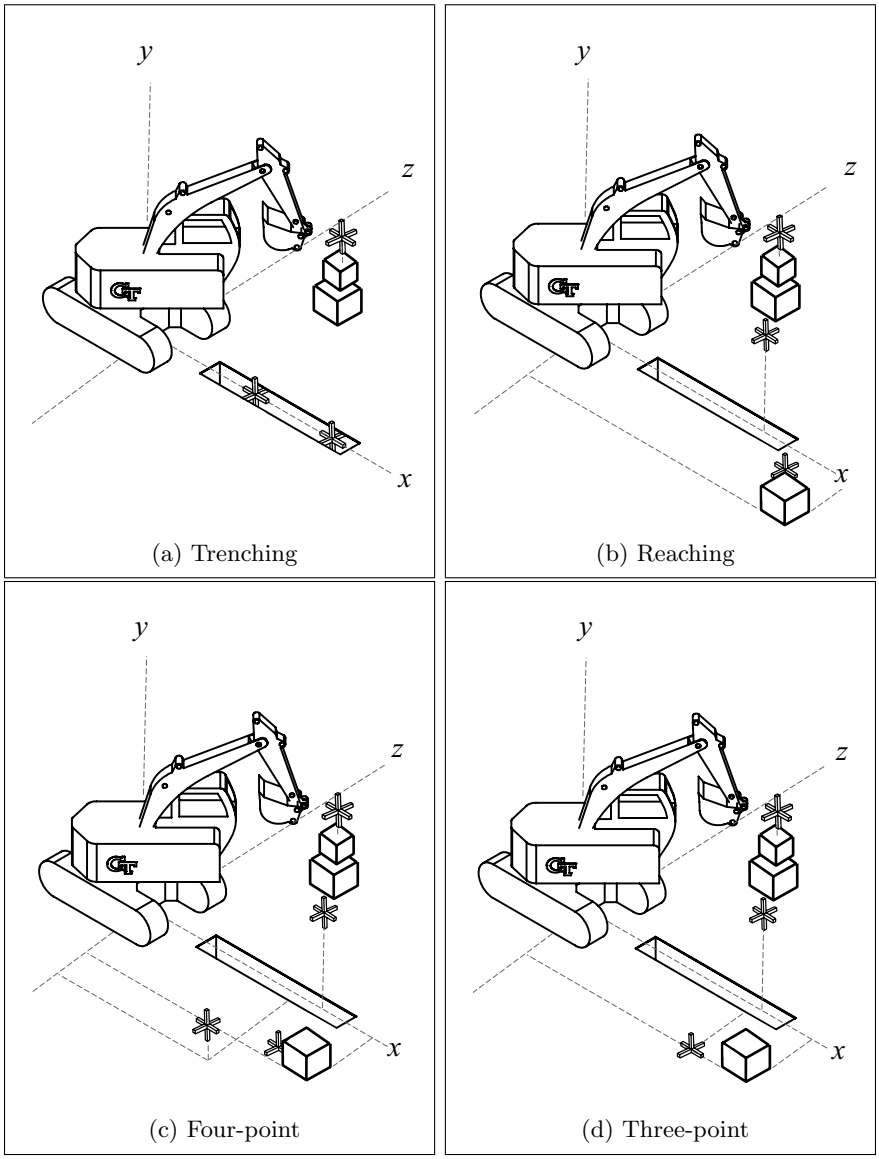


Figure 11.6: Location of targets for Unconstrained Guided Motion tasks

Table 11.5: Experiment design for 2DOF guided motion tests

	Run number									Run number							
Subj	1	2	3	4	5	6	7	8	Subj	1	2	3	4	5	6	7	8
1	A0	B0	A1	B1	A2	B2	A3	B3	17	B0	A1	B2	A3	B1	A0	B3	A2
2	A1	B1	A2	B2	A3	B3	A0	B0	18	B1	A2	B3	A0	B2	A1	B0	A3
3	A2	B2	A3	B3	A0	B0	A1	B1	19	B2	A3	B0	A1	B3	A2	B1	A0
4	A3	B3	A0	B0	A1	B1	A2	B2	20	B3	A0	B1	A2	B0	A3	B2	A1
5	A0	B3	A1	B0	A2	B1	A3	B2	21	B3	A1	B1	A3	B0	A0	B2	A2
6	A1	B2	A2	B3	A3	B0	A0	B1	22	B2	A2	B0	A0	B3	A1	B1	A3
7	A2	B1	A3	B2	A0	B3	A1	B0	23	B1	A3	B3	A1	B2	A2	B0	A0
8	A3	B0	A0	B1	A1	B2	A2	B3	24	B0	A0	B2	A2	B1	A3	B3	A1
9	A2	B2	A3	B3	A0	B0	A1	B1	25	B2	A3	B0	A1	B3	A2	B1	A0
10	A3	B3	A0	B0	A1	B1	A2	B2	26	B3	A0	B1	A2	B0	A3	B2	A1
11	A0	B0	A1	B1	A2	B2	A3	B3	27	B0	A1	B2	A3	B1	A0	B3	A2
12	A1	B1	A2	B2	A3	B3	A0	B0	28	B1	A2	B3	A0	B2	A1	B0	A3
13	A2	B1	A3	B2	A0	B3	A1	B0	29	B1	A3	B3	A1	B2	A2	B0	A0
14	A3	B0	A0	B1	A1	B2	A2	B3	30	A3	B3	A0	B0	A1	B1	A2	B2
15	A0	B3	A1	B0	A2	B1	A3	B2	31	A0	B3	A1	B0	A2	B1	A3	B2
16	A1	B2	A2	B3	A3	B0	A0	B1	32	A1	B2	A2	B3	A3	B0	A0	B1
									33	A3	B0	A0	B1	A1	B2	A2	B3

Table 11.6: Experiment design for 4DOF guided motion tests

Manual control			Shared control, type 1			Shared control, type 2		
Unfamiliar	Novice	Expert	Unfamiliar	Novice	Expert	Unfamiliar	Novice	Expert
5	1	2	7	13	22	12	3	23
6	4	10	11	21	26	14	8	33
18	9		20	24		17	28	
	15		32	25			29	
	16			27			31	
	19			30				

subject tested used only one level of the independent variable (type of shared control). The group assignments are shown in Table 11.6. The trials alternated between Trenching and Reaching. The subjects saw each scenario three times, for a total of 6 trials per subject on the 4DOF platform. Each trial continued for 3 minutes.

11.5.7 Measures— Unconstrained Guided Motion (UGM)

The cycle transition times were defined at the local minima of the distance measurement function d,

$$d(q) = \psi^T \left(|q(t) - q(0)| \right)$$

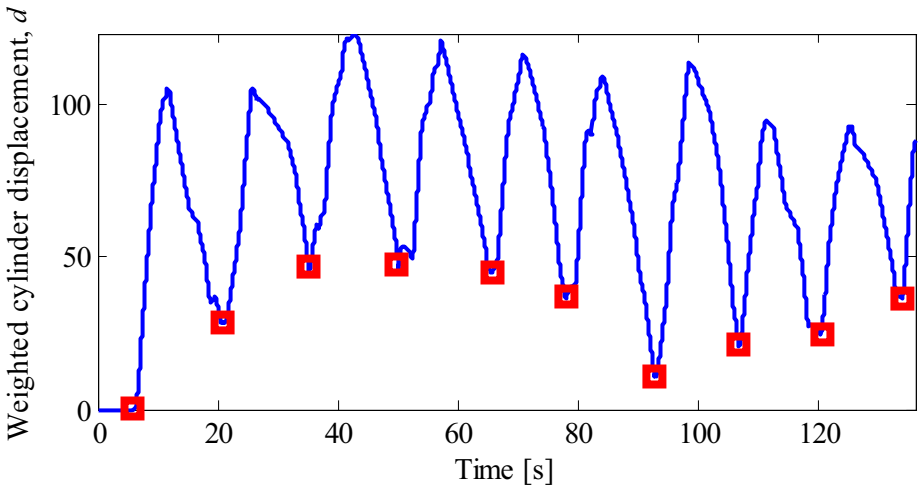


Figure 11.7: The start and end time of each cycle is determined by the minima of the weighted measurement function d.

where d may be interpreted as the relative change in volume of the fluid within the cylinders and $q(0)$ is the initial position of the cylinders. A typical plot of $d(q)$ with superimposed local minima is shown in Figure 11.7. The times of the local minima indicate the transition time between cycles. The transition time of each cycle was hand marked by the author semi-automatically after all experiment trials were complete. No knowledge of the experiment parameters were known at the time of marking, and the only variable visible at the time of marking was the weighted displacement, d.

Any trial having less than four cycles was not included in the analyzed data set. This ensures that the shared control tests are not unfairly penalized, as the RAMPE algorithm requires at least three iterations of a motion primitive before computing an estimate and initiating shared control. In sum, 21 of the 462 trials were discarded—less than 5 percent—spread roughly uniformly among the scenarios and control types. Most discards were from subjects having an *unfamiliar* skill-level and over half were from 3 subjects.

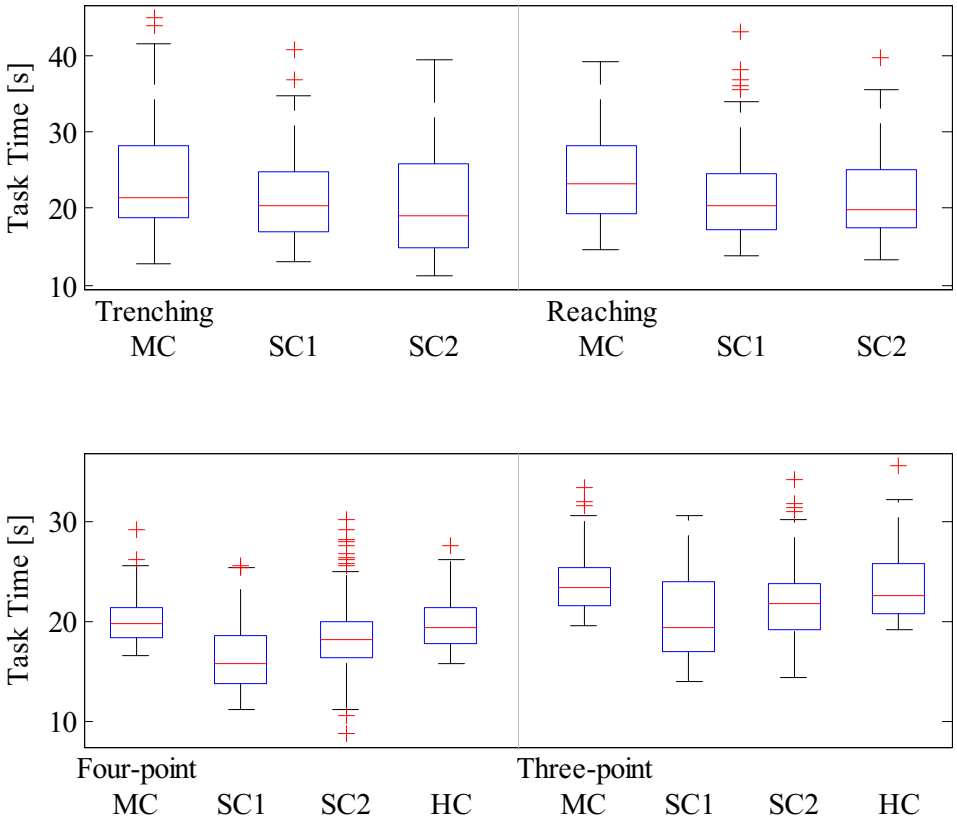


Figure 11.8: Boxplot of mean task times for Unconstrained Guided Motion (UGM) experiments

11.6 Results

The objective data analysis is based primarily on the measured task time. Figure 11.8 shows a boxplot[4] for the cycle times of each control type for each set of Unconstrained Guided Motion (UGM) experiments, grouped by control type. Figure 11.9 and Figure 11.10 shows the same type of information for each of the Point-to-Point Motion tasks. A cursory inspection reveals that the shared control groups tend to have lower cycle times, although the effect is not uniform and has a relatively large spread.

A qualitative data analysis provides information on general trends; likewise, it is beneficial to inspect the data with more rigorous statistical methods. Several assumptions must

[4]a boxplot is a useful way to display raw aggregate data. The centerline of a box is the median; the upper and lower extent of the box is the 25th and 75th percentiles; the error bars denote $1.5 \times IQR$ (interquartile range); and outliers are shown outside the error bars.

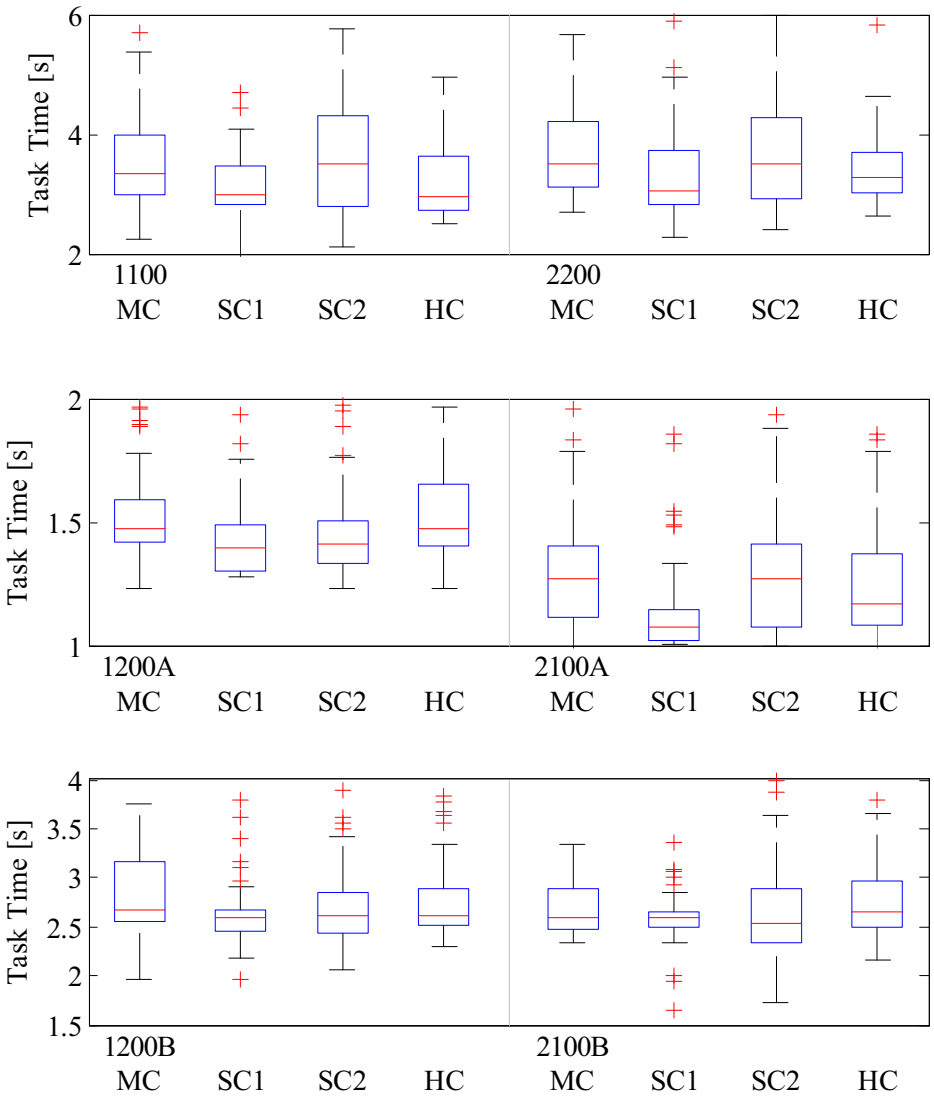


Figure 11.9: Boxplot of mean task times for 2DOF Point-to-Point Motion experiments

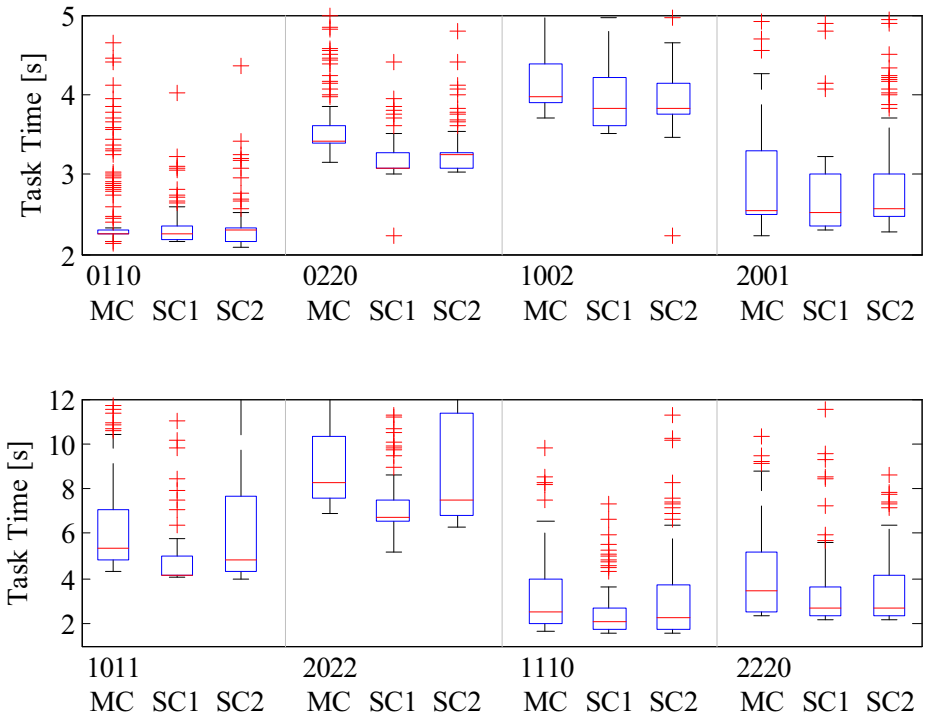


Figure 11.10: Boxplot of mean task times for 4DOF Point-to-Point Motion experiments

be satisfied before applying common statistical analyses such at ANOVA. The *validity assumptions* [148] on the data requires that negligible systematic error, or bias, is present. Primary sources of bias include selection bias and information bias. Selection bias is reduced by randomly assigning the shared control "treatments", and by ensuring a uniform representation of the operator skill-levels for each case. Information bias is reduced by automating the collection and analysis of the data.

The *distributional assumptions* are that the data is independent, normally distributed, and of uniform variance. Independence between subjects is assumed in all cases. Asserting independence between trials of the same subject is more bold since learning or fatigue effects may (intuitively) be present; the balanced experiment design helps alleviate these symptoms. A run sequence plot, of which a typical one is shown in Figure 11.11, of the task time of each trial did not indicate any gross violations of the independence assumption. A linear regression on task time did not yield a significant effect, hence strengthening the case for validity of the inter-sample independence assumption.

However, there is strong indication of non-normality for the distribution of measured

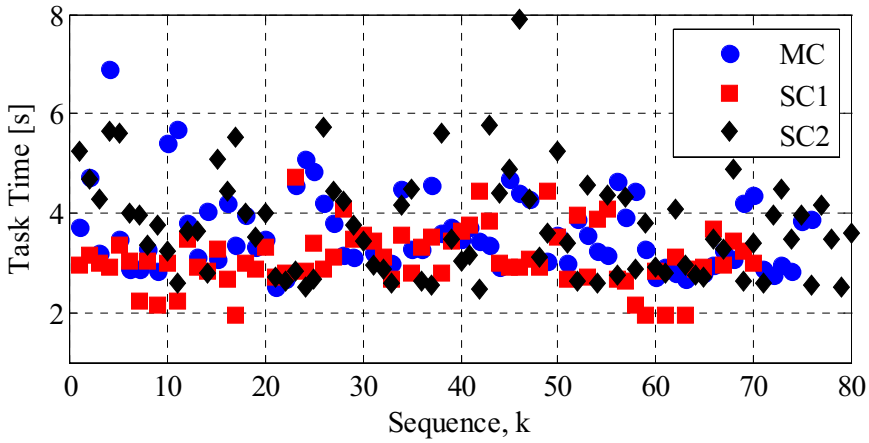


Figure 11.11: Run sequence plot for a typical subject performing the Point-to-Point task set. There appears to be no violations of the between-trials independence assumption.

task times. This is evidenced by examining histogram plots of the cycle time, for example Figure 11.12 shows the cycle times for the Trenching task. The fixed lower bound on the cycle time manifests as a distribution with positive skew. Common statistical techniques assume normally distributed measures. Thus, the units of the data should be transformed before applying these techniques; Bland provides a good layman's overview of this process [149–151]. Before analysis of the task-time data,[5] the measurements are transformed to a new variable, τ, with

$$T' = \log(T)$$

The log-transformed measurements better approximate the normal distribution, as illustrated by the histograms in Figure 11.13. A secondary advantage of using the transformed data is to lessen the influence of sampling error that is introduced by extreme observations within the skewed data. A back transform via antilog allows statistics on the transformed scale to be interpreted in the standard units (seconds). The mean task time, after back transforming into the original units, is actually the geometric mean; the geometric mean will be less than that of the raw data, and closely approximates the sample median.

[5] A cursory look at several publications within the robotics community which deal with manipulation task times reveals that many researchers neglect—either by choice or by ignorance—the skew-ness present in their measurement sets

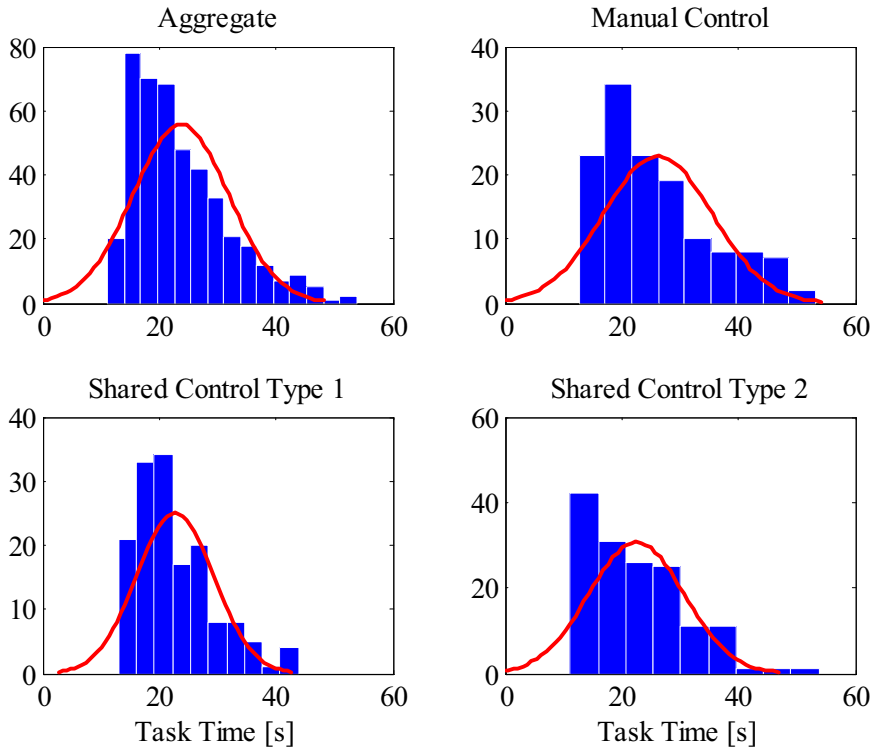


Figure 11.12: Histogram plots of task times for Trenching task, shown with an overlaid best-fit normal distribution.

11.6.1 Analysis of Results

To determine if differences among the cases are statistically significant, an analysis of variance was performed. A linear statistical model [148] is used to describe the main observations by

$$y_{ijkl} = \mu + \tau_i + \alpha_j + \beta_k + (\alpha\beta))_{jk} + \epsilon_{ijkl}$$

where y is the measured or observed cycle time, μ is the ground truth mean effect for the task, τ_i is the effect of the subject *neglecting* skill, α_j is the effect of the Shared Control type, β_k is the effect of the Operator Skill alone, $(\alpha\beta)_{jk}$ is the interaction of Control Type and Skill, and ϵ_{ijkl} is a random component describing the experimental error. The subject factor τ is treated as a nuisance factor. The effect μ is the *average* task time across all trials of the particular task for subject i, Control Type j.

The null hypothesis is that the different control types (SC1, SC2, HC, MC) have no effect on the task completion time. For each experiment scenario, a single factor ANOVA

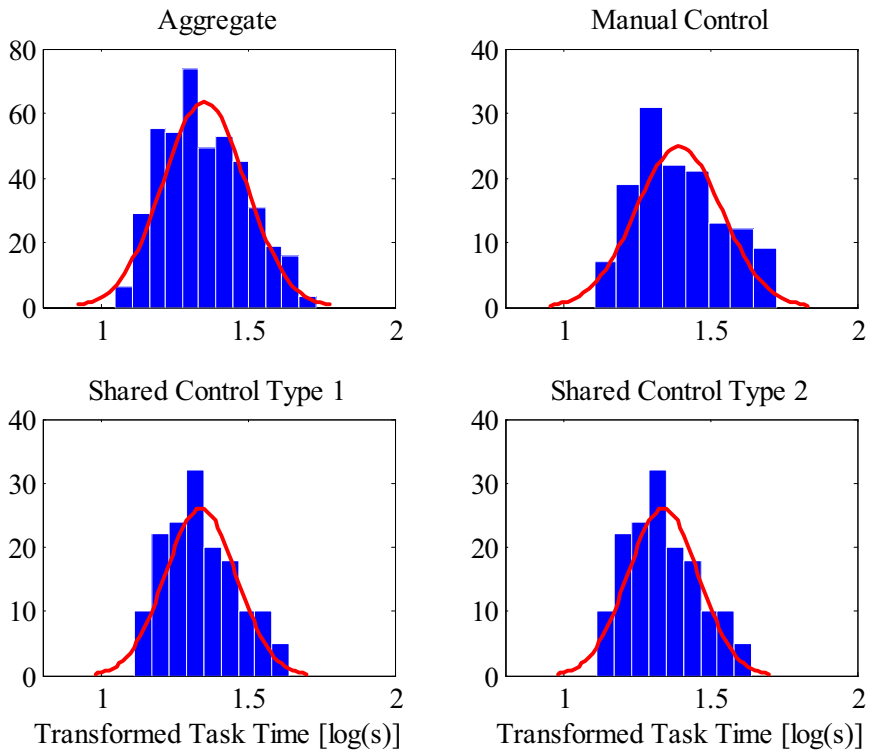


Figure 11.13: Histogram plots of log transformed task times for Trenching task, shown with an overlaid best-fit normal distribution.

test having four fixed effects is performed on the mean cycle time of each subject and each control type.

Analysis of variance results show that, with the Skill factor collapsed across all groups, the effect of Control Type does not significantly impact Trenching ($p = 0.09$) and Reaching ($p = 0.09$), but does influence 2DOF Point-to-Point ($p < 0.001$) and 4DOF Point-to-Point ($p = 0.04$) tasks. When only the effect upon the *novice* group is considered, the Control Type effects all the tasks: Trenching ($p = 0.032$), Reaching ($p = 0.045$), 2DOF P2P ($p = 0.014$) and 4DOF P2P ($p = 0.013$).

The low p-value for this test is evidence to reject the null hypothesis in favor of the alternative. From the ANOVA, it is clear that the task times are drawn from distributions having at least a pair of unequal means. Dunnett's method is used to determine specifically which Control Types are significantly different from the control group [148]. Dunnett's multiple range test is a post-hoc, two-sided multiple comparison method used to compare a set of categories to a control group. In this case, Manual Control (MC) is the baseline

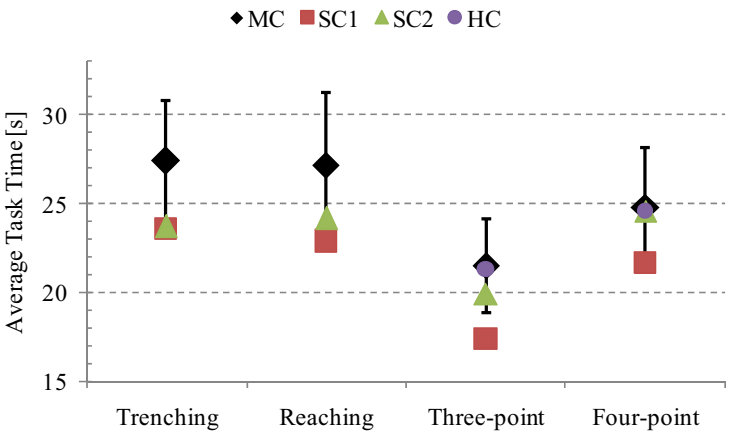


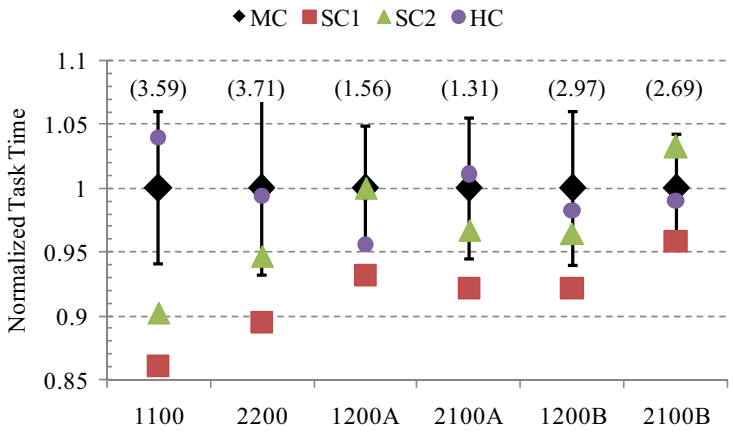
Figure 11.14: Unconstrained Guided Motion (UGM): Comparing the mean task time for different control effects, across all skill levels. The error bars correspond to the critical interval for Dunnett's multiple range test.

control group.

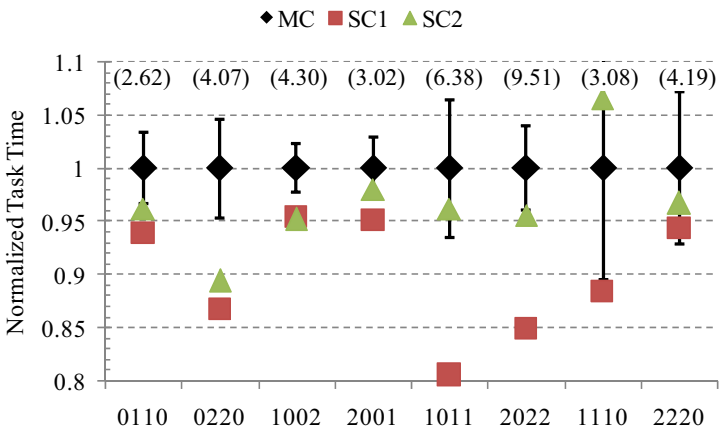
For the Unconstrained Guided Motion (UGM) tasks, the control group is a separate group of subjects. However, for the P2P tasks, the control group is a subgroup of trials *within* all trials performed by each subject. The control group must only be compared within subjects. Thus, before contrasting each Control Type for the P2P tasks, each measure (for a particular subject performing a particular task) is normalized by the mean task time under MC.

The critical interval in Dunnett's test is a function of the mean square error for data from all trials, the statistical degrees-of-freedom, and the number of samples in the two distributions being compared. For unequal sample sizes, the critical interval will vary depending on the pair being compared. The results of the Guided Motion Tasks are summarized in Figure 11.14. The P2P test results are shown in Figure 11.15. To aide interpretation of the graph, only the largest critical interval is shown. For the P2P tests, the results are shown in units of *mean manual control time*. The mean task times for manual control are shown within the parenthesis of Figure 11.15.

The operator skill level groupings are considered individually in the results shown in Figure 11.16



(a) 2DOF Point-to-Point Tasks



(b) 4DOF Point-to-Point Tasks

Figure 11.15: Point-to-point (P2P) Motion Tests: Comparing the mean task time for different control effects, across all operator skill levels. The error bars correspond to the critical interval for Dunnett's multiple range test. The task times are normalized with respect to the operator task time for MC. The numbers in parentheses are the mean task time, in seconds, for manual control.

11.6.2 Two-way Interactions

Two-way interactions were also analyzed using a two-way ANOVA formulation for experiment designs with unequal sample sizes. A significant interaction between the Control Type and Skill Level factors exists for Trenching ($p = 0.045$), Reaching ($p = 0.025$), Three-Point ($p = 0.007$), Four-Point ($p < 0.001$), and for the normalized comparison between all P2P trials ($p < 0.001$).

Interaction graphs for the Unconstrained Guided Motion tasks are shown in Figure 11.17. The significant interaction between Control Type and Operator Skill is indicated by the lack

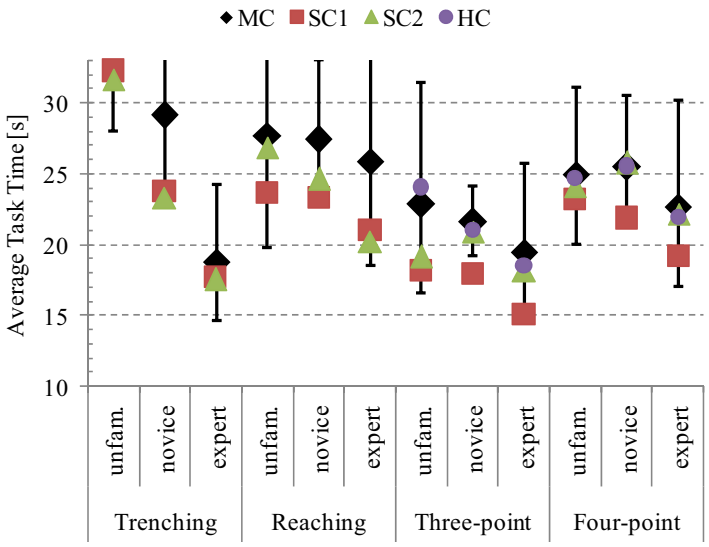


Figure 11.16: Comparing the mean task time for different control effects, segmented based on operator skill level. The error bars correspond to the critical interval for Dunnett's multiple range test.

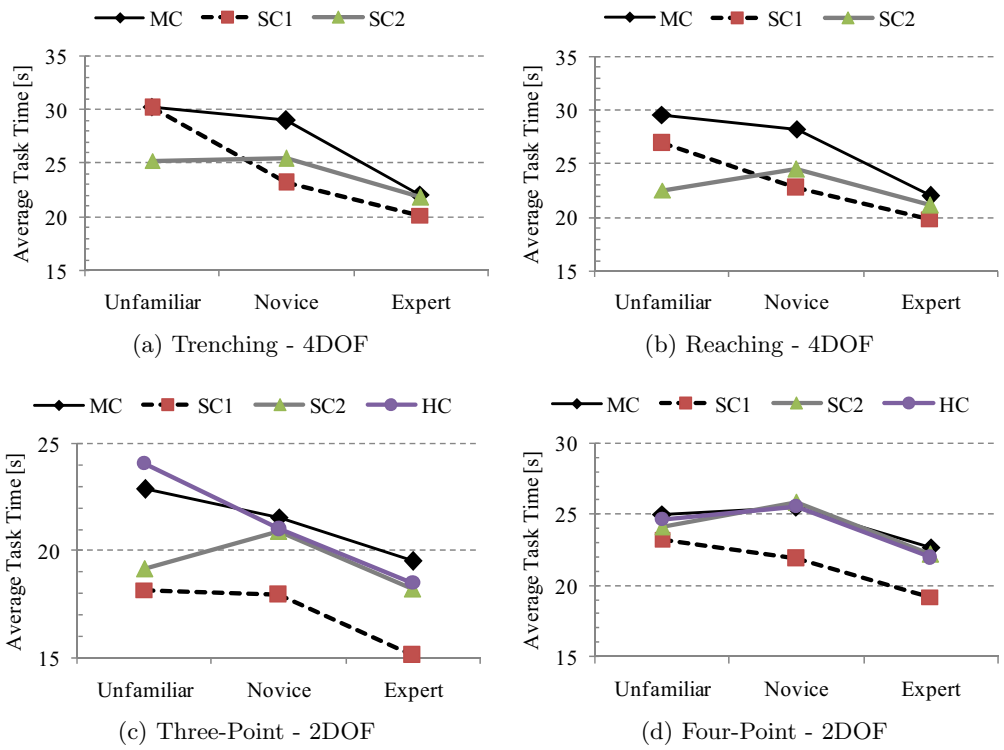
of parallelism of the lines. In general, lower task times are achieved with increasing operator skill. However, there appears to be a horizon of diminishing returns beyond which shared control does not further influence task time.

11.7 Discussion

The goal of these experiments was to examine the utility of shared control for decreasing the task completion time of representative excavator or manipulator tasks. The evidence supports the conclusion that there are scenarios for which Blended SC offers a significant time saving compared to manual control alone. Operator skill has the greatest effect on task completion time, therefore operator training is justified. However, even for very experienced operators, Blended SC has a positive effect of productivity.

The point-to-point tasks, for which the shared controller was given knowledge of the task parameters, shared control had significant effect on the task completion time compared with manual control. The proportional perturbations had a lower mean task time than the constant flow perturbations.

The mean task time for the unconstrained guided motion tasks were lower with shared



(a) Trenching - 4DOF

(b) Reaching - 4DOF

(c) Three-Point - 2DOF

(d) Four-Point - 2DOF

Figure 11.17: Interaction plots: Control type vs. Operator skill for Unconstrained Guided Motion tasks

control than with manual control; however, in most cases the outcome is not strictly statistically significant.

Haptic cues were not effective in decreasing task time for the tasks considered in this study. This contrasts with the single degree of freedom task discussed in Chapter 5 for which the effect of haptic feedback was significant. The explanation for the poorer performance of haptic feedback for the multi-DOF excavation tasks may include that the cues were not interpreted correctly, or that the cues somehow prompted lower velocity inputs.

The significant interaction between Control Type and Operator Skill implies that there may be a group of users for which the shared control is most suitable. Shared control produced greater benefit for the Novice group, whose mean task time with shared control were nearly equal to the unassisted task time of expert operators. The effect of shared control on the Unfamiliar and Expert groups was not as strong for the Novice group. For experts, there may be less opportunity for improvement, as may already be close to optimal. The task complexity relative to available operator skill must be weighed when deciding to implement the Blended SC.

CHAPTER XII

SUMMARY, CONTRIBUTIONS, AND FUTURE DIRECTIONS

This chapter presents concluding remarks. First, a summary of the contributions and accomplishments is given. Finally, suggestions and recommendations for further research are stated.

12.1 Summary and Concluding Remarks

Earthmoving machines have many interesting challenges germane to robotics, and solutions to these challenges positively impact a large industry. Presently, the human operator manually controls the manipulator motion. This research has shown that if the conventional human/machine link is severed—and the human input is augmented with assistance from an Electronic Agent—then some tasks can be completed in less time.

Specifically, a new control architecture termed Blended Shared Control was presented. The architecture allows two agents, here a Human Agent and an Electronic Agent, to simultaneously influence the machine response. This framework has an admittedly simple structure; however, *simple* does not imply ineffective.

The larger manipulator motion is decomposed into a sequence of simpler motion primitives. A motion primitive was comprised of a direction Ω and a displacement x. Every manipulator motion can be described by a sequence of these primitives. However, in general, the operator's task is not known *a priori*. Chapter 7 presented an algorithm for estimating future motions, given knowledge of the current operator input and the current actuator positions.

With the task parameters known, the motion is optimized. A simple model was developed in Chapter 8. This model retained the salient actuator constraints, while dismissing higher-order complexities common in typical models. Using this model, a time-optimal solution for completing the tasks was developed in Chapter 9.

The Blended Shared Control perturbations themselves were presented in Chapter 10.

The control perturbations were a function of the current operator input and the optimal input. A single scalar parameter enabled the level of control authority to be adjusted between full manual control and full automatic control. A series of constraints were introduced to limit the deviation from the operator command.

Chapter 10 showed analytically that the shared controller essentially "pushed" the operator commands to a lower instantaneous cost regardless of the operator dynamics. However, it remained unclear (at least from any analytical sense) what occurs globally when the operator is in the loop. Analytically proving stability and acceptability to the operator is probably impossible even for a very simple cognitive model. Thus, this particular hypothesis was tested primarily with experimental evidence. In particular, the paradigm has been tested in different situations for different operators, and performance was evaluated from an analytical and subjective point of view. Experimental results indicated a measurable effect on the task time, and are detailed in Chapter 11.

12.2 Contributions

A more intuitive method for analyzing the actuator motions and understanding the multi-function interaction of hydraulic manipulators subject to pressure, flow, and power constraints was given. This approach allows the feasible region to be visualized as a function of valve operating modes, power limits, and flow limits of the actuators and pump. Conventional approaches are very complex and require either nonlinear expressions, or difficult to train and inflexible methods that learn the interaction. In contrast, this technique provides an intuitive understanding of the way the machine response changes for different interaction cases.

This research contributed a new understanding to the time-optimal control of net velocity constrained manipulators. A particular point, u^*, is useful not only as a necessarily optimal solution (one of many) but also as a means of checking the optimality of a control input, e.g. from a human operator, which may not equal u^*.

Further, the motion primitive formulation for describing actuator motions, and the RAMPE method of task description and task parameter estimation, are helpful tools for

manipulator control. The formulation enables a compact representation of a manipulator motions. Describing the actuator motion as a combination of a *direction* and a *displacement* magnitude allows the several unique benefits for human-controlled manipulators particularly: the motion direction can be inferred based on sensed operator commands, and the future motion estimated based on previous cycles.

The perturbation-based Blended Shared Control framework is a new contribution to a family of existing shared control paradigms. The method is particularly noteworthy because only a single scalar parameter needs tuned. Moreover, the approach brings a uniquely transparent analysis of the effect that the SC perturbation has on directly observable responses such as changes in perceived end effector velocity or effective changes in joystick input.

12.3 Future Directions

It seems clear that the present study has only scratched the surface of what task-estimation based actions and increased autonomy could add to the capabilities of human-controlled manipulators performing repetitive tasks such as excavation.

Further experimental observations would be beneficial for understanding the nature of the Shared Control. Interesting areas of analysis include the following:

1. Analyze the effect of SC on the learning rate
2. Analyze the change in human behavior as a result of the perturbations
3. Analyze the performance of SC when there are conflicting objectives. Conflicts may include obstacles within the intended manipulator trajectory which only one agent is aware of. Conflict will also arise if one agent is minimizing temporal cost while the other is minimizing energetic cost.
4. Segment the subject population using other criteria, such as tendency to use multiple functions, to help explain observed effects of SC

The Shared Control parameter a tunes the level of authority held by both agents. Chapter 10 presented several alternatives for the selection of a, but there are many other possibilities. These include

1. Selecting a to be dependent on the task estimation accuracy. For example, a higher confidence in the expected task would automatically lead to an increase to the level of autonomy (i.e., a would increase with the estimation accuracy).

2. Select a based on the stage of the manipulator cycle. For example, portions of the cycle which require a high level of operator technique—such as moving through the soil—would not be modified. Portions of the cycle for which the path is less important than the endpoints would be modified by the EA.

The point u^* in Chapter 9 provides indication of optimality; namely, if the point leaves the region of ∂U in which it began then the motion is necessarily suboptimal. It may be feasible to design a feedback controller which modulates u to keep u^* within its native region. This controller would then be optimal, and may also increase the overall robustness to modeling errors.

The Blended SC paradigm presented here places the Human Agent on the highest level of control. The human's task is estimated, and the human's commands are perturbed. There is a symmetric embodiment not discussed in this thesis, in which the EA would complete some autonomous task, and the operator would only intervene according to certain criteria.

Regardless of the value of a, the architecture only acts in response to an operator input. Thus, even when $a = 1$ and the EA has full control of the manipulator, the commands which the EA issues are based upon the commands of the HA. Thus, if the HA is commanding a null motion (by not moving the joysticks), the EA will issue a null command. Similarly, if the HA commands the boom and arm cylinders to retract, then the EA must also only give commands that cause the boom and arm cylinders to retract. These convenient constraints are a result of defining the manipulator task using the motion primitive formulation discussed in Chapter 7. Therefore, when the EA is given full authority, that authority is inherently restricted to a u-plane which is defined by the operator's present inputs. This has interesting implications, in that now the HA can use the displacements of the joysticks to "conduct" a large variety of autonomous manipulator motions, in a manner reminiscent of the way a maestro conducts the flow of a melody. Exploiting this feature, one may be able

to pre-program a wide variety of fundamental motions. The motion ultimately executed would be a function of the joystick displacements and the machine state at the start of the motion. An extension of the research in this direction would have interesting application to the field of collaborative control and behavior modeling.

APPENDIX A

POST-EXPERIMENT QUESTIONNAIRE AND RESULTS

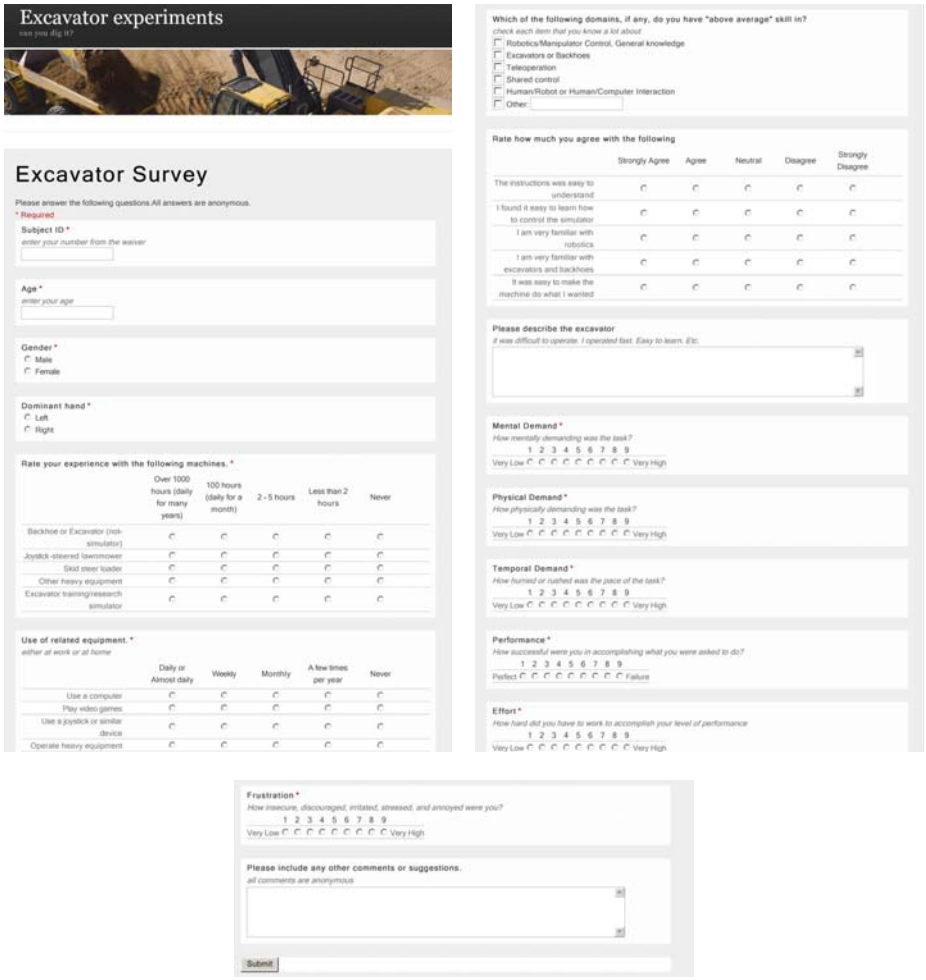
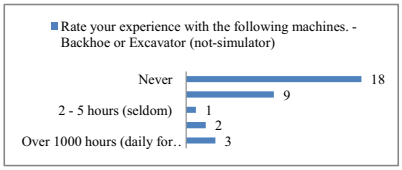


Figure A.1: Survey questions. The survey was administered using a web-based form.

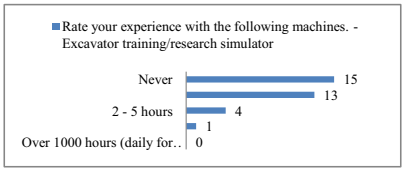
Rate your experience with the following machines. - Backhoe or Excavator (not-simulator)

Over 1000 hours (daily for many years)	**3**	9%
100 hours (daily for many months)	**2**	6%
2 - 5 hours (seldom)	**1**	3%
Less than 2 hours	**9**	27%
Never	**18**	55%



Rate your experience with the following machines. - Excavator training/research simulator

Over 1000 hours (daily for many years)	**0**	0%
100 hours (daily for a month)	**1**	3%
2 - 5 hours	**4**	12%
Less than 2 hours	**13**	39%
Never	**15**	45%



Rate your experience with the following machines. - Any related machine

Over 1000 hours (daily for many years)	**4**	12%
100 hours (daily for a month)	**4**	12%
2 - 5 hours	**4**	12%
Less than 2 hours	**12**	36%
Never	**9**	27%

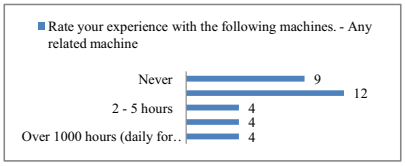
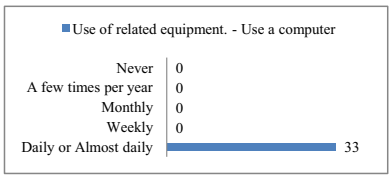


Figure A.2: Summary of survey responses: Use of equipment related to excavators.

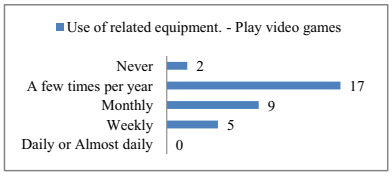
Use of related equipment. - Use a computer

Daily or Almost daily	33	100%
Weekly	0	0%
Monthly	0	0%
A few times per year	0	0%
Never	0	0%



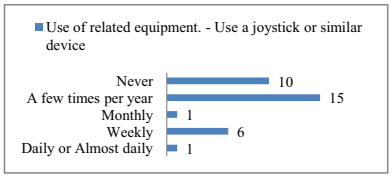
Use of related equipment. - Play video games

Daily or Almost daily	0	0%
Weekly	5	15%
Monthly	9	27%
A few times per year	17	52%
Never	2	6%



Use of related equipment. - Use a joystick or similar device

Daily or Almost daily	1	3%
Weekly	6	18%
Monthly	1	3%
A few times per year	15	45%
Never	10	30%



Use of related equipment. - Operate heavy equipment

Daily or Almost daily	3	9%
Weekly	1	0%
Monthly	3	9%
A few times per year	6	18%
Never	20	64%

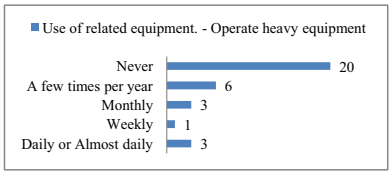
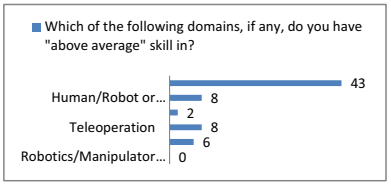


Figure A.3: Summary of survey responses: Use of related equipment.

Which of the following domains, if any, do you have "above average" skill in?

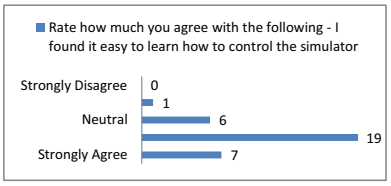
Robotics/Manipulator Co	0	0%
Excavators or Backhoes	6	18%
Teleoperation	8	24%
Shared control	2	6%
Human/Robot or Human/	8	24%
Other	43	130%

People may select more than one checkbox, so percentages may add up to more than 100%.



Rate how much you agree with the following - I found it easy to learn how to control the simulator

Strongly Agree	7	21%
Agree	19	58%
Neutral	6	18%
Disagree	1	3%
Strongly Disagree	0	0%



Rate how much you agree with the following - I am very familiar with robotics

Strongly Agree	8	24%
Agree	4	12%
Neutral	9	27%
Disagree	7	21%
Strongly Disagree	5	15%

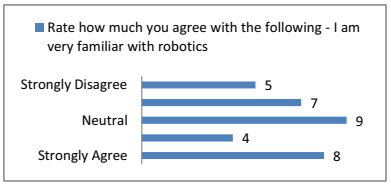
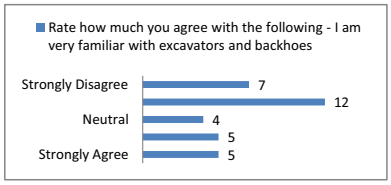


Figure A.4: Summary of survey responses: Familiarity with task domain.

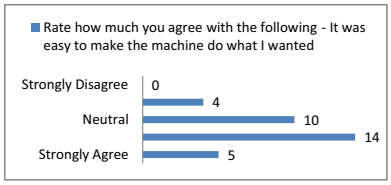
Rate how much you agree with the following - I am very familiar with excavators and backhoes

Strongly Agree	**5**	15%
Agree	**5**	15%
Neutral	**4**	12%
Disagree	**12**	36%
Strongly Disagree	**7**	21%



Rate how much you agree with the following - It was easy to make the machine do what I wanted

Strongly Agree	**5**	15%
Agree	**14**	42%
Neutral	**10**	30%
Disagree	**4**	12%
Strongly Disagree	**0**	0%



Rate how much you agree with the following - The instructions was easy to understand

Strongly Agree	**12**	36%
Agree	**18**	55%
Neutral	**3**	9%
Disagree	**0**	0%
Strongly Disagree	**0**	0%

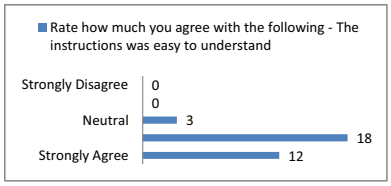


Figure A.5: Summary of survey responses: Other queries.

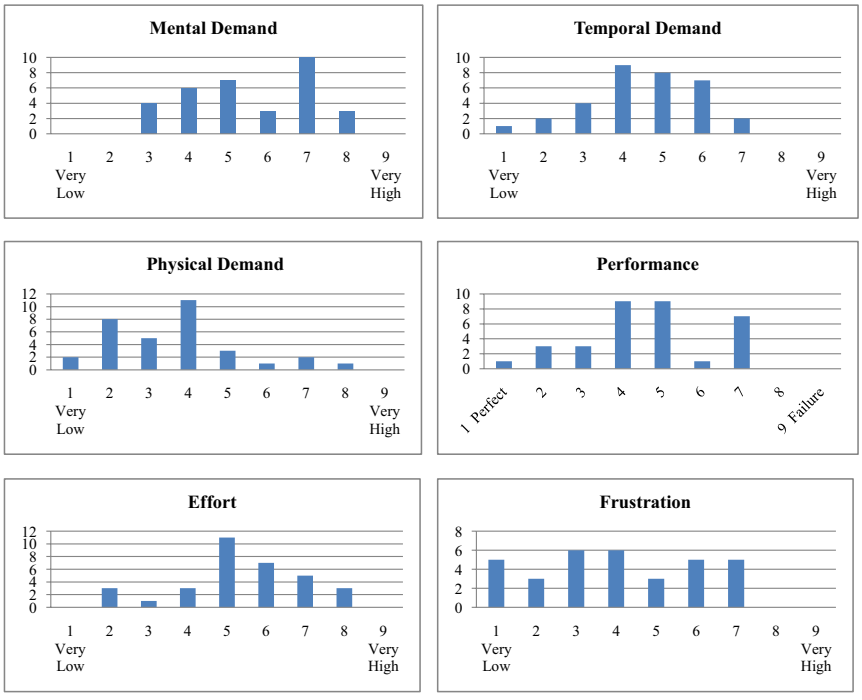


Figure A.6: Summary of survey responses: Operator workload self-assessment.

REFERENCES

[1] R. W. Nielsen, "Construction field operations and automated equipment," *Automation in Construction*, vol. 1, no. 1, pp. 35–46, 1992.

[2] W. Poppy, "Driving forces and status of automation and robotics in construction in europe," *Automation in Construction*, vol. 2, no. 4, pp. 281–289, 1994.

[3] T. Alekseeva, K. Artem'ev, A. Bromberg, R. Voitsekhouskii, and N. Ul'yanov, *Machines for Earthmoving Work, Theory and Calculations*. New Delhi, India: Amerind Publishing Co. Pvt Ltd, 1992.

[4] A. B. Carson, *General Excavation Methods*. New York: F. W. Dodge Corp, 1961.

[5] N. D. Manring and R. E. Johnson, "Modeling and designing a variable-displacement open-loop pump," *Journal of dynamic systems, measurement, and control*, vol. 118, no. 2, 1996.

[6] R. Rahmfeld and M. Ivantysynova, "Displacement controlled linear actuator with differential cylinder-a way to save primary energy in mobile machines," in *Proc. of 5th International Conference on Fluid Power Transmission and Control (ICFP'2001)*, Hangzhou, China, 2001, pp. 316–322.

[7] P. Opdenbosch, "Auto calibration and control applied to electro hydraulic valves," Doctoral Thesis, Georgia Instutute of Technology, 2007.

[8] X. Yang, D. B. Stephenson, and M. J. Paik, "Bidirectional pilot operated control valve," U.S. Patent 6 328 275, 2001.

[9] J. A. Aardema and D. W. Koehler, "System and method for controlling an independent metering valve," U.S. Patent 5 947 140, 1999.

[10] A. Jansson and J. O. Palmberg, "Separate controls of meter-in and meter-out orifices in mobile hydraulic systems," *SAE Transactions*, vol. 99, pp. 377–383, 1990.

[11] S. Liu and Y. Bin, "Energy-saving control of single-rod hydraulic cylinders with programmable valves and improved working mode selection," *SAE Transactions, Journal of Commercial Vehicle*, vol. 2002-01-1343, pp. 51–61, 2002.

[12] A. Shenouda, "Quasi-static hydraulic control systems and energy savings potential using independent metering four-valve assembly configuration," Doctoral Thesis, Georgia Instutute of Technology, 2006.

[13] P. Y. Li, C. Y. Li, and T. R. Chase, "Software enabled variable displacement pumps," in *ASME International Mechanical Engineering Congress and Exposition (IMECE 05); The Fluid Power and Systems Technology Division (FPST)*, vol. 12, Orlando, FL, United States, 2005, pp. 63–72.

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

[15] K. D. Garnjost, "Energy-conserving regeneratice-flow valves for hydraulic servomotors," U.S. Patent 4 840 111, 1989.

[16] X. Liang and T. Virvalo, "Accumulator-charged drive for a hydraulic boom to save energy," in *ASME International Mechanical Engineering Congress and Exposition, (IMECE 02); The Fluid Power and Systems Technology Division, (FPST)*, New Orleans, LA, United States, 2002, pp. 117–122.

[17] K.-E. Rydberg, "Hydraulic accumulators as key components in energy efficient mobile systems," in *The Sixth International Conference on Fluid Power Transmission and Control (ICFP 2005)*, Hangzhou, P. R. China, 2005.

[18] G. Zarotti and R. Paoluzzi, "Control valves for cylinder regenerative circuits," in *Innovations in Fluid Power: Seventh Bath International Fluid Power Workshop*, C. Burrows and K. Edge, Eds. Taunton, Somerset, England: Research Studies Press LTD., 1995, pp. 71–88.

[19] Y. Zhang, Q. Wang, and Q. Xiao, "Simulation and experimental research on energy regeneration with hydraulic motor for hybrid drive excavator," *Chinese Journal of Mechanical Engineering*, vol. 43, no. 8, pp. 218–23, 2007.

[20] P. Beater and M. Otter, "Multi-domain simulation: Mechancis and hydraulics of an excavator," in *3rd International Modelica Conference*, P. Fritzson, Ed., Linkoping, 2003, pp. 331–340.

[21] H. Merrit, *Hydraulic Control Systems*. New York: Wiley, 1967.

[22] C. J. J. Paredis, "An open-source modelica library of fluid power models," in *Bath/ASME Symposium on Fluid Power and Motion Control (FPMC 2008)*, Bath, UK, 2008.

[23] D. M. Bullock, S. M. Apte, and I. J. Oppenheim, "Force and geometry constraints in robot excavation," in *Proc. of Space 90: Engineering, Construction and Operations in Space*. Albuquerque, NM, USA: Publ by ASCE, Boston, MA, USA, 1990, pp. 960–969.

[24] A. J. Koivo, "Kinematics of excavators (backhoes) for transferring surface material," *Journal of Aerospace Engineering*, vol. 7, no. 1, pp. 17–32, 1994.

[25] A. J. Koivo, M. Thoma, E. Kocaoglan, and J. Andrade-Cetto, "Modeling and control of excavator dynamics during digging operation," *Journal of Aerospace Engineering*, vol. 9, no. 1, p. 10, 1996.

[26] Y. H. Zweiri, L. D. Seneviratne, and K. Althoefer, "Model-based automation for heavy duty mobile excavator," in *Intelligent Robots and System, 2002. IEEE/RSJ International Conference on*, vol. 3, 2002, pp. 2967–2972 vol.3.

[27] ——, "Modeling and control of heavy duty vehicles for excavation tasks," in *Proceedings of the IASTED International Conf. for Control and Applications*, Mexico, 2002.

[28] M. Krishna, "Optimal motion generation for hydraulic robots," Doctoral Thesis, Robotics Institute, Carnegie Mellon University, 1999.

[29] M. Krishna, M. Krishna, and J. Bares, "Hydraulic system modeling through memory-based learning," in *Intelligent Robots and Systems, 1998. Proceedings., 1998 IEEE/RSJ International Conference on*, J. Bares, Ed., vol. 3, 1998, pp. 1733–1738 vol.3.

[30] R. Filla, "An event-driven operator model for dynamic simulation of construction machinery," in *The Ninth Scandinavian International Conference on Fluid Power*, Linkoping, Sweden, 2005.

[31] R. Filla, A. Ericsson, and J.-O. Palmberg, "Dynamic simulation of construction machinery: Towards an operator model," in *IFPE 2005 Technical Conference*, Las Vegas, 2005, pp. 429–438.

[32] L. E. Bernold, "Quantitative assessment of backhoe operator skill," *Journal of Construction Engineering and Management*, vol. 133, no. 11, pp. 889–899, 2007.

[33] S. P. DiMaio, S. P. DiMaio, S. E. Salcudean, C. Reboulet, S. A. T. S. Tafazoli, and K. A. H.-Z. K. Hashtrudi-Zaad, "A virtual excavator for controller development and evaluation," in *Proc. of IEEE International Conference on Robotics and Automation*, S. E. Salcudean, Ed., vol. 1, 1998, pp. 52–58 vol.1.

[34] R. Fales, E. Spencer, K. Chipperfield, F. Wagner, and A. Kelkar, "Modeling and control of a wheel loader with a human-in-the-loop assessment using virtual reality," *Transactions of the ASME. Journal of Dynamic Systems, Measurement and Control*, vol. 127, no. 3, pp. 415–23, 2005.

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

[36] A. Enes, M. Elton, and W. Book, "A virtual reality operator interface station with hydraulic hardware-in-the-loop simulation for prototyping excavator control systems," in *Proc. of IEEE AIM2009: Advanced Intelligent Mechatronics*, Singapore, 2009.

[37] T. Dill, "Excavating machine," U.S. Patent 244 718, 1881.

[38] R. Book, "Programmable electrohydraulic valve," Ph.D. dissertation, University of Illionois at Urbana-Champaign, 1998.

[39] W. G. Holzbock, "Energy-saving electrohydraulic systems," in *Proceedings of The National Conferences on Fluid Power*, 1977, pp. 324–331.

[40] K. D. Kramer and E. H. Fletcher, "Electrohydraulic valve system," Unites States Patent Re. 33,846, 1992.

[41] J. L. Pfaff and K. A. Tabor, "Velocity based electronic control system for operating hydraulic equipment," U.S. Patent 6 732 512, 2004.

[42] ——, "Method of selecting a hydraulic metering mode for a function of a velocity based control system," U.S. Patent 6 880 332, 2005.

[43] K. A. Tabor, "A novel method of controlling a hydraulic actuator with four valve independent metering using load feedback," in *2005 SAE Commercial Vehicle Engineering Conference*, Rosemont, IL, 2005.

[44] Y. Huayong, S. Wei, and X. Bing, "New investigation in energy regeneration of hydraulic elevators," *IEEE/ASME Transactions on Mechatronics*, vol. 12, no. 5, pp. 519–26, 2007.

[45] X. Liang and T. Virvalo, "An energy recovery system for a hydraulic crane," *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, vol. 215, no. 6, pp. 737–744, 2001.

[46] A. Pourmovahed, N. H. Beachley, and F. J. Fronczak, "Analytical and experimental modeling of a hydraulic energy regeneration system," in *Proc. of ASME Fluid Power and Systems Technology*, Dallas, 1990, p. 12.

[47] A. Barrientos, O. Luengo, and A. Mora, "Teleoperated backhoe excavator with haptic control," in *Proceedings of the 16th IAARC/IFAC/IEEE International Symposium on Automation and Robotics in Construction (ISARC 99)*, Madrid, Spain, 1999, pp. 491–6.

[48] H. Ching and W. J. Book, "Internet-based bilateral teleoperation based on wave variable with adaptive predictor and direct drift control," *Transactions of the ASME, Journal of Dynamic Systems, Measurement and Control*, vol. 128, no. 1, pp. 86–93, 2006.

[49] M. E. Kontz, "Haptic control of hydraulic machinery using proportional valves," Ph.D. dissertation, Georgia Instutute of Technology, 2007.

[50] P. Lawrence, S. Salcudean, N. Sepehri, D. Chan, S. Bachmann, N. Parker, M. Zhu, and R. Frenette, "Coordinated and force-feedback control of hydraulic excavators," in *International Symposium on Experimental Robotics (ISER 95)*, Stanford, CA, 1995.

[51] L. Sim, M. L. Cummings, and C. A. Smith, "Past, present and future implications of human supervisory control in space missions," *Acta Astronautica*, vol. 62, no. 10-11, pp. 648–655, 2008.

[52] S. Tafazoli, S. E. Salcudean, K. Hashtrudi-Zaad, and P. D. Lawrence, "Impedance control of a teleoperated excavator," *IEEE Transactions on Control Systems Technology*, vol. 10, no. 3, pp. 355–367, 2002.

[53] H. Araya and M. Kagoshima, "Semi-automatic control system for hydraulic shovel," *Automation in Construction*, vol. 10, no. 4, pp. 477–486, 2001.

[54] Y. Wai and J. Pretlove, "Semi-autonomous teleoperation system with vision guidance," in *Proc. SPIE - Int. Soc. Opt. Eng.*, vol. 3524, Boston, 1998, pp. 71–9.

[55] M. E. Kontz, M. C. Herrera, J. D. Huggins, and W. J. Book, "Impedance shaping for improved feel in hydraulic systems," in *Proc. of ASME International Mechanical Engineering Congress and Exposition (IMECE 07)*, vol. 4, Seattle, WA, 2007, pp. 185–194.

[56] M. Haga, W. Hiroshi, and K. Fujishima, "Digging control system for hydraulic excavator," *Mechatronics*, vol. 11, no. 6, pp. 665–676, 2001.

[57] Audemar, "Hydraulic transmission for reproducing mechanical motions at remote points," U.S. Patent 2 597 050, 1947.

[58] J. Chytil, "Analog-programmed control system for excavators having jibs," U.S. Patent 3 636 325, 1968.

[59] D. J. Rocke, "Automatic excavation control system and method," U.S. Patent 5 446 980, 1995.

[60] T. M. Sagaser, "Electronic bucket positioning and control system," U.S. Patent 4 844 685, 1989.

[61] C. D. Wickens, "Processing resources and attention," in *Multiple-task performance*, D. L. Damos, Ed. Bristol, PA: Taylor & Francis, 1991.

[62] C. Wickens and J. Hollands, *Engineering psychology and human performance*, 3rd ed. Upper Saddle River, New Jersey: Prentice-Hall Inc., 1999.

[63] G. Field and Y. Stepanenko, "Iterative dynamic programming: an approach to minimum energy trajectory planning for robotic manipulators," in *Proc. of IEEE International Conference on Robotics and Automation (ICRA96)*, vol. 3, Minneapolis, MN, USA, 1996, pp. 2755–60.

[64] M. Vukobratovic and M. Kircanski, "A method for optimal synthesis of manipulation robot trajectories," *Journal of Dynamic Systems, Measurement, and Control*, vol. 104, no. 2, pp. 188–193, 1982.

[65] S. F. P. Saramago and V. Steffen, "Optimization of the trajectory planning of robot manipulators taking into account the dynamics of the system," *Mechanism and Machine Theory*, vol. 33, no. 7, pp. 883–894, 1998.

[66] S. Ma, "Real-time algorithm for quasi-minimum energy control of robotic manipulators," in *Proceedings of the IEEE IECON 21st International Conference on Industrial Electronics, Control, and Instrumentation*, vol. 2, 1995, pp. 1324–1329.

[67] P. Rowe, "Adaptive motion planning for autonomous mass excavation," Doctoral Thesis, Robotics Institute, Carnegie Mellon University, 1999.

[68] P. Rowe and A. Stentz, "Parameterized scripts for motion planning," in *Proc. of International Conference on Intelligent Robots and Systems (IROS)*, Grenoble, France, 1997.

[69] P. S. Rowe, "Learning system and method for optimizing control of autonomous earthmoving machinery," U.S. Patent 6 076 030, 2000.

[70] P. S. Rowe and A. J. Stentz, "Automated system and method for control of movement using parameterized scripts," U.S. Patent 5 908 458, 1999.

[71] P. J. A. Lever and X. Shi, "Experimental autonomous excavation using fuzzy behavior control," in *1996 IEEE International Conference on Systems, Man and Cybernetics. Information Intelligence and Systems*, vol. 4, Beijing, China, 1996, pp. 2631–6.

[72] X. Shi, W. Fei-Yue, and P. J. A. Lever, "Robotic excavation: experimental results using fuzzy behavior control," in *Proceedings of the 13th World Congress, International Federation of Automatic Control. Vol. M. Chemical Process Control, Mineral, Mining, Metals (IFAC97)*, San Francisco, 1997, pp. 499–504.

[73] A. Hemami, "Fundamental analysis of automatic excavation," *Journal of Aerospace Engineering*, vol. 8, no. 4, pp. 175–179, 1995.

[74] S. Blouin, A. Hemami, and M. Lipsett, "Review of resistive force models for earth-moving processes," *Journal of Aerospace Engineering*, vol. 14, no. 3, p. 102, 2001.

[75] A. Hemami, "Motion trajectory study in the scooping operation of an lhd-loader," *IEEE Transactions on Industry Applications*, vol. 30, no. 5, pp. 1333–8, 1994.

[76] L. Plonecki, W. Trampczynski, and J. Cendrowicz, "A concept of digital control system to assist the operator of hydraulic excavators," *Automation in Construction*, vol. 7, no. 5, pp. 401–411, 1998.

[77] L. E. Bernold, "Experimental studies on mechanics of lunar excavation," *Journal of Aerospace Engineering*, vol. 4, no. 1, pp. 9–22, 1991.

[78] ——, "Motion and path control for robotic excavation," *Journal of Aerospace Engineering*, vol. 6, no. 1, pp. 1–18, 1993.

[79] C. P. Tan, Y. H. Zweiri, K. Althoefer, and L. D. Seneviratne, "Online soil parameter estimation scheme based on newton-raphson method for autonomous excavation," *IEEE/ASME Transactions on Mechatronics*, vol. 10, no. 2, pp. 221–229, 2005.

[80] E. Budny and W. Gutkowski, "Controlled excavation along a prescribed path," in *Proceedings of ASCE Specialty Conference: ROBOTICS for Challenging Environments (RCE II)*, Albuquerque, NM, 1996, pp. 227–234.

[81] M. Bodur, H. Zontul, A. Ersak, A. J. Koivo, H. O. Yurtseven, E. Kocaoglan, and G. Pasamehmetoglu, "Dynamic cognitive force control for an automatic land excavation robot," in *7th Mediterranean Electrotechnical Conference*, vol. 2, Antalya, Turkey, 1994, pp. 703–6.

[82] P. K. Vaha and M. J. Skibniewski, "Cognitive force control of excavators," *Journal of Aerospace Engineering*, vol. 6, no. 2, pp. 159–166, 1993.

[83] M. H. Chiang and C. C. Huang, "Experimental implementation of complex path tracking control for large robotic hydraulic excavators," *International Journal of Advanced Manufacturing Technology*, vol. 23, no. 1-2, pp. 126–32, 2004.

[84] T. Makkonen, K. Nevala, and R. Heikkil, "A 3d model based control of an excavator," *Automation in Construction*, vol. 15, no. 5, pp. 571–577, 2006.

[85] M. Kahn and B. Roth, "The near-minimum-time control of open-loop articulated kinematic chains," *Transactions of the ASME. Journal of Dynamic Systems, Measurement and Control*, vol. 4, no. 9, p. 164, 1971.

[86] E. Budny, M. Chlosta, and W. Gutkowski, "Optimal control of an excavator bucket positioning," in *19th International Symposium on Automation and Robotics in Construction (ISARC)*, Gaithersburg, MD, 2002, pp. 481–488.

[87] J. E. Bobrow, S. Dubowsky, and J. S. Gibson, "Time-optimal control of robotic manipulators along specified paths," *International Journal of Robotics Research*, vol. 4, no. 3, pp. 3–17, 1985.

[88] V. Braibant and M. Geradin, "Optimum path planning of robot arms," *Robotica*, vol. 5, pp. 323–31, 1987.

[89] B. K. Kim and K. G. Shin, "Minimum-time path planning for robot arms and their dynamics," *IEEE Transactions on Systems, Man and Cybernetics*, vol. SMC-15, no. 2, pp. 213–23, 1985.

[90] F. Pfeiffer and R. Johanni, "A concept for manipulator trajectory planning," *IEEE Journal of Robotics and Automation*, vol. 3, no. 2, pp. 115–123, 1987.

[91] K. G. Shin and N. D. McKay, "Minimum-time control of robotic manipulators with geometric path constraints," *IEEE Transactions on Automatic Control*, vol. AC-30, no. 6, pp. 531–41, 1985.

[92] A. Weinreb and A. E. Bryson, "Optimal control of systems with hard control bounds," *IEEE Transactions on Automatic Control*, vol. AC-30, no. 11, pp. 1135–8, 1985.

[93] E. Budny, M. Chlosta, and W. Gutkowski, "Load-independent control of a hydraulic excavator," *Automation in Construction*, vol. 12, no. 3, pp. 245–254, 2003.

[94] W. Richardson-Little and C. Damaren, "Position accommodation and compliance control for robotic excavation," *Journal of Aerospace Engineering*, vol. 21, no. 1, pp. 27–34, 2008.

[95] Q. P. Ha, Q. H. Nguyen, D. C. Rye, and H. F. Durrant-Whyte, "Impedance control of a hydraulically actuated robotic excavator," *Automation in Construction*, vol. 9, no. 5-6, pp. 421–435, 2000.

[96] N. Hogan, "Impedance control: an approach to manipulation. i. theory," *Transactions of the ASME. Journal of Dynamic Systems, Measurement and Control*, vol. 107, no. 1, pp. 1–7, 1985.

[97] ——, "Impedance control: an approach to manipulation. ii. implementation," *Transactions of the ASME. Journal of Dynamic Systems, Measurement and Control*, vol. 107, no. 1, pp. 8–16, 1985.

[98] ——, "Impedance control: an approach to manipulation. iii. applications," *Transactions of the ASME. Journal of Dynamic Systems, Measurement and Control*, vol. 107, no. 1, pp. 17–24, 1985.

[99] T. E. Steenwyk and E. J. Walstra, "Control for hydraulically operated construction machine having multiple tandem articulated members," U.S. Patent 5 572 809, 1996.

[100] X. Shi, P. Lever, and F.-Y. Wang, *Autonomous rock excavation: intelligent control techniques and experimentation*, 1st ed., ser. Intelligent Control and Intelligent Automation. World Scientific, 1998, vol. 10.

[101] S. Singh, "State of the art in automation of earthmoving," *Journal of Aerospace Engineering*, vol. 10, no. 4, pp. 179–188, 1997.

[102] ——, "State of the art in automation of earthmoving," in *Workshop on Advanced Geomechatronics*, Sendai University, Japan, 2002.

[103] E. S. Slaughter, "Characteristics of existing construction automation and robotics technologies," *Automation in Construction*, vol. 6, no. 2, pp. 109–120, 1997.

[104] D. A. Bradley and D. W. Seward, "Developing real-time autonomous excavation-the lucie story," in *34th IEEE Conference on Decision and Control*, vol. 3, 1995, pp. 3028–3033.

[105] ——, "Development, control and operation of an autonomous robotic excavator," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 21, no. 1, pp. 73–97, 1998.

[106] S. Singh and H. Cannon, "Method and apparatus for determining an excavation strategy," U.S. Patent 6 108 949, 2000.

[107] H. Cannon, "Extended earthmoving with an autonomous excavator," Masters Thesis, Robotics Institute, Carnegie Mellon University, 1999.

[108] H. Cannon and S. Singh, "Models for automated earthmoving," in *Experimental Robotics VI. Lecture Notes in Control and Information Sciences*, vol. 250, Sydney, NSW, Australia, 2000, pp. 163–72.

[109] S. Singh, "Synthesis of tactical plans for robotic excavation," Doctoral Thesis, Carnegie Mellon University, 1995.

[110] S. Singh and M. C. Leu, "Optimal trajectory generation for robotic manipulators using dynamic programming," *Transactions of the ASME Journal of Dynamic Systems, Measurement and Control*, vol. 109, no. 2, pp. 88–96, 1987.

[111] A. Stentz, J. Bares, and P. Rowe, "A robotic excavator for autonomous truck loading," *Autonomous Robots*, vol. 7, no. 2, pp. 175–186, 1999.

[112] T. B. Sheridan, *Telerobotics, automation, and human supervisory control*. Cambridge, MA: The MIT Press, 1992.

[113] E. C. Poulton, *Tracking skill and manual control*. New York: Academic Press, Inc., 1974.

[114] D. J. Bruemmer, D. A. Few, R. L. Boring, J. L. Marble, M. C. Walton, and C. W. Nielson, "Shared understanding for collaborative control," *IEEE Transactions on Systems, Man and Cybernetics, Part A (Systems and Humans)*, vol. 35, no. 4, pp. 494–504, 2005.

[115] J. A. Caywood, D. M. De Moss, E. C. Fitch, R. F. Osborn, and J. O. Matous, "Automatic back hoe control system," U.S. Patent 3 339 763, 1966.

[116] T. Inui, K. Ootsuka, S. Nogami, M. Igarashi, T. Horikoshi, and K. Izumi, "Semi-automatic hydraulic excavator," U.S. Patent 4 377 043, 1983.

[117] D. J. Rocke, L. J. Devier, and D. B. Herget, "Teaching automatic excavation control system and method," U.S. Patent 5 493 798, 1996.

[118] C. Urdiales, A. Poncela, I. Sanchez-Tato, F. Galluppi, M. Olivetti, and F. Sandoval, "Efficiency based reactive shared control for collaborative human/robot navigation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2007)*, San Diego, CA, 2007, pp. 3586–3591.

[119] H. K. Kim, J. Biggs, W. Schloerb, M. Carmena, M. A. Lebedev, M. A. L. Nicolelis, and M. A. Srinivasan, "Continuous shared control for stabilizing reaching and grasping with brain-machine interfaces," *IEEE Transactions on Biomedical Engineering*, vol. 53, no. 6, pp. 1164–1173, 2006.

[120] S. S. Nudehi, R. Mukherjee, and M. Ghodoussi, "A shared-control approach to haptic interface design for minimally invasive telesurgical training," *IEEE Transactions on Control Systems Technology*, vol. 13, no. 4, pp. 588–592, 2005.

[121] P. G. Griffiths and R. B. Gillespie, "Sharing control between humans and automation using haptic interface: primary and secondary task performance benefits," *Human Factors*, vol. 47, no. 3, pp. 574–90, 2005.

[122] M. E. Kontz and W. J. Book, "Flow control for coordinated motion and haptic feedback," *International Journal of Fluid Power*, vol. 8, no. 3, pp. 13–23, 2007.

[123] W. A. Allen, P. D. Anderson, and W. J. Bradbury, "Coordinated control for a work implement," U.S. Patent 5 424 623, 1995.

[124] W. T. Good and J. E. Vilhauer Jr, "Automatic parallel parking system," U.S. Patent United States Patent 4 735 274, 1988.

[125] G. Pires and U. Nunes, "A wheelchair steered through voice commands and assisted by a reactive fuzzy-logic controller," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 34, no. 3, pp. 301–14, 2002.

[126] S. Hayati and S. T. Venkataraman, "Design and implementation of a robot control system with traded and shared control capability," in *Proc. of IEEE International Conference on Robotics and Automation (ICRA89)*, 1989, pp. 1310–1315 vol.3.

[127] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *The International Journal of Robotics Research*, vol. 5, no. 1, pp. 90–98, 1986.

[128] A. Poncela, C. Urdiales, E. J. Perez, and F. Sandoval, "A new efficiency-weighted strategy for continuous human/robot cooperation in navigation," *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, vol. 39, no. 3, pp. 486–500, 2009.

[129] G. Chuanfan, T. Tzyh-Jong, X. Ning, and A. K. Bejczy, "Fusion of human and machine intelligence for telerobotic systems," in *Proc. of IEEE International Conference on Robotics and Automation (RA95)*, vol. 3, 1995, pp. 3110–3115.

[130] A. E. Bryson and Y. Ho, *Applied Optimal Control*. Watham, MA: Blaisdell Pub. Co., 1969.

[131] J. Stephens and L. Mast, "Adjustable hydraulic metering system," U.S. Patent 7 401 542, 2008.

[132] K. V. Mardia, *Multivariate analysis*, ser. Probability and mathematical statistics. London: Academic Press, 1979.

[133] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.

[134] O. C. Jenkins and M. J. Mataric, "Deriving action and behavior primitives from human motion data," in *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 3. Piscataway, NJ, USA: IEEE, 2002, pp. 2551–6.

[135] O. C. Jenkins, M. J. Mataric, and S. Weber, "Primitive-based movement classification for humanoid imitation," in *IEEE International Conference on Humanoid Robots (Humanoids 2000)*, 2000.

[136] J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

[137] C. Bishop, *Neural networks for pattern recognition*. New York: Oxford University Press, 1995.

[138] J. Hertz, A. Krogh, and R. Palmer, *Introduction to the theory of neural computation*. Redwood City, CA: Addison-Wesley, 1991.

[139] A. H. Jazwinski, *Stochastic processes and filtering theory*. New York: Academic Press, 1970.

[140] P. Opdenbosch and N. Sadegh, "Control of discrete-time systems via online learning and estimation," in *2005 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM2005*, vol. 2. Monterey, CA, USA: IEEE, 2005, pp. 975–80.

[141] X. Yang, M. J. Paik, and J. L. Pfaff, "Pilot operated control valve having a poppet with integral pressure compensating mechanism," U.S. Patent 6 745 992, 2004.

[142] K. A. Tabor, "Optimal velocity control and cavitation prevention of a hydraulic actuator using four valve independent metering," in *2005 SAE Commercial Vehicle Engineering Conference*, Rosemont, IL, 2005.

[143] J. L. Pfaff and K. A. Tabor, "Method of sharing flow of fluid among multiple hydraulic functions in a velocity based control system," U.S. Patent 6 779 340, 2004.

[144] M. Krishna and J. Bares, "Constructing hydraulic robot models using memory-based learning," *Journal of Aerospace Engineering*, vol. 12, no. 2, p. 34, 1999.

[145] M. Cobo, R. Ingram, and S. Cetinkunt, "Modeling, identification, and real-time control of bucket hydraulic system for a wheel type loader earth moving equipment," *Mechatronics*, vol. 8, no. 8, pp. 863–885, 1998.

[146] P. Holobut, "Time-optimal control of hydraulic manipulators with path constraints," *Journal of Theoretical and Applied Mechanics*, vol. 43, no. 3, pp. 523–38, 2005.

[147] M. K. O'Malley, A. Gupta, M. Gen, and Y. Li, "Shared control in haptic systems for performance enhancement and training," *Journal of Dynamic Systems, Measurement and Control, Transactions of the ASME*, vol. 128, no. 1, pp. 75–85, 2006.

[148] D. C. Montgomery, *Design and analysis of experiments*, 4th ed. New York: John Wiley and Sons, 1997.

[149] J. M. Bland and D. G. Altman, "Statistics notes. logarithms," *BMJ (Clinical Research Ed.)*, vol. 312, no. 7032, pp. 700–700, 1996.

[150] ——, "Statistics notes. transforming data," *BMJ (Clinical Research Ed.)*, vol. 312, no. 7032, p. 770, 1996.

[151] ——, "Statistics notes. the use of transformation when comparing two means," *BMJ (Clinical Research Ed.)*, vol. 312, no. 7032, p. 1153, 1996.

VITA

Aaron was born in Wyoming and raised in Idaho. He graduated in 2005 with a Bachelor of Science degree, Summa Cum Laude, in Engineering Physics, earning Science Honors and the Dean's Award, from Northwest Nazarene University in Nampa, Idaho. He earned a Masters of Science in Mechanical Engineering in 2007 and a Doctoral degree in 2010 from the Georgia Institute of Technology in Atlanta, Georgia. His primary academic interests are in System Dynamics, Robotics, Applied Control Theory, and Management of Technology. At Georgia Tech, Aaron was a Presidential Fellow, a Woodruff Fellow, and an ARCS Scholar. After graduation, Aaron will join the technical staff at the Massachusetts Institute of Technology Lincoln Laboratory.