Queuing Network Modeling of Human Multitask Performance and its Application to Usability Testing of In-Vehicle Infotainment Systems

by

Ruijia Feng

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Industrial and Operations Engineering)
in the University of Michigan
2015

Doctoral Committee:

Professor Yili Liu, Chair
Associate Professor Victoria Booth
Associate Professor Bernard J. Martin
Professor Nadine B. Sarter

To my family

ACKNOWLEDGEMENTS

This work was financially supported, in part, by the fellowship and graduate student instructor positions provided by the Department of Industrial and Operations Engineering (IOE) at the University of Michigan, M. Scheller Fellowship by the College of Engineering at the University of Michigan, and Ford-University of Michigan Alliance Project.

I would like to thank my advisor Dr. Yili Liu for his invaluable mentorship, guidance, and enormous support since I got into the program in 2009; my dissertation committee members Dr. Victoria Booth, Dr. Bernard Martin, and Dr. Nadine Sarter for their precious comments and suggestions; Dr. Yifan Chen for his professional advices and support; my lab mates Dan Nathan-Roberts, Shi Cao, and Chris Best for their help and encouragement; the faculties and staff members in the IOE department, especially Dr. Tomas Armstrong, Dr. Monroe Keyserling, Charles Woolley, Tina Blay, Elizabeth Fisher, Matt Irelan, Christopher Konrad, Wanda Dobberstein, Rod Capps, "Mint" Minto, Candy Ellis for their help and support.

I would like to thank Denny, Dabeum, Sara, Robert, Kat, Chris, Kathryn, Vernnaliz, Greg, BDC, Dandrich, Siqian, Selin, and Lauren for making this journey an unforgettable experience.

Lastly, I would like to thank my parents and family for their unconditional support and love.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Abstract

Human performance of a primary continuous task (e.g., steering a vehicle) and a secondary discrete task (e.g., tuning radio stations) simultaneously is a common scenario in many domains. It is of great importance to have a good understanding of the mechanisms of human multitasking behavior in order to design the task environments and user interfaces (UIs) that facilitate human performance and minimize potential safety hazards. In this dissertation I investigated and modeled human multitask performance with a vehicle-steering task and several typical in-vehicle secondary tasks. Two experiments were conducted to investigate how various display designs and control modules affect the driver's eye glance behavior and performance. A computational model based on the cognitive architecture of Queuing Network-Model Human Processor (QN-MHP) was built to account for the experiment findings. In contrast to most existing studies that focus on visual search in single task situations, this dissertation employed experimental work that investigates visual search in multitask situations. A modeling mechanism for flexible task activation (rather than strict serial activations) was developed to allow the activation of a task component to be based on the completion status of other task components. A task switching scheme was built to model the time-sharing nature of multitasking. These extensions offer new theoretical insights into visual search in multitask situations and enable the model to simulate parallel processing both within one task and among multiple tasks. The validation results show that the model could account for the observed performance differences from the empirical data. Based on this model, a computer-aided engineering toolkit was developed that allows the UI designers to make quantitative prediction of the usability of design concepts and prototypes. Scientifically, the results of this dissertation research offer additional insights into the mechanisms of human multitask performance. From the engineering application and practical value perspective, the new modeling mechanism and the new toolkit have advantages over the traditional usability testing methods with human subjects by enabling the UI designers to explore a larger design space and address usability issues at the early design stages with lower cost both in time and manpower.

Chapter 1.
Introduction

1.1 Computational Human Performance Modeling

Computational human performance modeling is a valuable asset for researchers to gain insights into the mechanisms of human cognition and performance and for human factors practitioners to develop and evaluate the usability of design prototypes and products. Following the pioneering work by Allen Newell on a unified theory of cognition (Newell, 1973, 1990), great efforts have been made in the past decades by researchers in the development of comprehensive human performance models based on a single cognitive architecture. From a scientific standpoint, a cognitive architecture combines numerous psychological and neuroscience findings and theories into one coherent framework. From an engineering standpoint, it provides a fixed structure with parameters for human cognition that could be used to model a wide range of tasks. It also allows modeling the interference of several tasks during multitasking.

Among the most famous computational human performance models include the Model Human Processor (MHP, Card et al., 1983), the family of Goal, Operator, Method, and Selection rules models (GOMS, John and Kieras, 1996), Soar (Laird et al., 1987), Executive-Process Interactive Control (EPIC, Meyer and Kieras, 1997), Adaptive Control of Thought-Rational (ACT-R, Anderson et al., 2004), Connectionist Learning with Adaptive Rule Induction On-line (CLARION, Sun, 2003), and Queuing Network-Model Human Processor (QN-MHP, Liu et al., 2006). These models have been successfully applied to model a large variety of tasks (Liu, 2009). Below is a brief description for each of the models, some of which have inspired or used in part for the modeling work in this dissertation.

MHP (Card et al., 1983) was developed as an engineering model of human computer interaction (HCI) to predict how long it takes for a human to perform a task. The model draws analogy of human information processing from a computer system with processors and memory storages. The model is composed of three processors (perceptual, cognitive, and motor) and some general-purpose memory storages (e.g., visual image storage, working memory, long term memory). Each of the processors is associated with a cycle time, and each of the memory storages is associated with a capacity and decay time, the values of which were derived from findings from numerous psychology studies. The time it takes to complete a task is calculated by breaking down the task into the steps in basic perceptual, cognitive, and motor levels.

The GOMS family of models (John and Kieras, 1996) is closely related to the MHP model, which decompose a task into elementary actions. Several variations of GOMS model exist including the original GOMS model (Card et al., 1983), the Keystroke-Level Model (KLM, Card et al., 1983), the Natural-GOMS-Language (NGOMSL, Kieras, 1996), and the Cognitive Perceptual Motor GOMS (CPM-GOMS, John and Gray, 1995). A detailed analysis and comparison of these GOMS models can be found in John and Kieras (1996). Most of the GOMS models are serial stage models that do not support modeling parallel activities with the exception of CPM-GOMS. For a task analysis using CPM-GOMS, a task is breakdown to perceptual, cognitive, and motor level similar to MHP, and then the task elements are examined to see whether overlapping of task elements are allowed. This enables the modeling of parallel activities, but CPM-GOMS is in a form of schedule charts instead of an executable computational implementation, thus it may not be used to generate human actions and behaviors in a simulation.

ACT-R (Anderson et al., 2004) is a cognitive architecture with a computational implementation as a production rule system. It distinguishes between declarative knowledge as chunks and procedural knowledge as production rules. ACT-R has a module structure including visual, aural, manual, vocal, procedural, goal, imaginal, and declarative modules. Each module has a buffer that can hold at most one chunk at a time. As a production system, an ACT-R simulation runs by firing production roles until a pre-defined goal state is reached.

EPIC (Meyer and Kieras, 1997) is cognitive architecture in spirit to the MHP but with more details processors and with an implementation of a production rule system similar to ACT-R. One difference between the EPIC and ACT-R is that ACT-R is essentially a serial system, which only allows one production rule to be fired at any given time, while EPIC allows parallel processing at the cognitive level. While human may be able to perform two simple cognitive tasks at once, studies (e.g., Byrne and Anderson, 1998) have shown that human may not be able to perform two complex cognitive tasks in a true parallel manner, which conflict with EPIC's configuration.

Soar (Laird et al., 1987) is a production rule based cognitive architecture designed as an artificial intelligence system that can be used to model human cognitive behaviors. Compared with other cognitive architectures that primarily aim at modeling human performance, SOAR is less constrained by the human cognition. It has a mechanism for searching through a problem space and moving the system state gradually to the goal state, which gives it strength in modeling complex cognitive behavior such as problem solving and reasoning.

CLARION (Sun, 2003) is a cognitive architecture that has been used to model cognitive psychology tasks among others. It distinguishes between implicit and explicit processes and focus on capturing the interaction between these two types of processes. It is composed of a number of subsystems including action-centered subsystem (loosely the counterpart for the procedural memory in ACT-R), non-action-centered subsystem (loosely the counterpart for the declarative memory in ACT-R), motivational subsystem, mega-cognitive subsystem. CLARION has its strength in modeling complex cognitive behaviors such as bottom-up learning and reasoning.

QN-MHP (Liu et al., 2006) is a cognitive architecture that integrates the mathematical structure of queuing network with the Model Human Processor. The procedural knowledge of how to perform a task is stored in the long-term procedural server, and the information entities traverse in the queuing network based on the server/routing settings of QN-MHP. QN-MHP can generate detailed task actions and behaviors like EPIC and ACT-R. And unlike EPIC and ACT-R, it supports instantaneous visualization of the human information processing during the simulation. More details on QN-MHP are described in the next section.

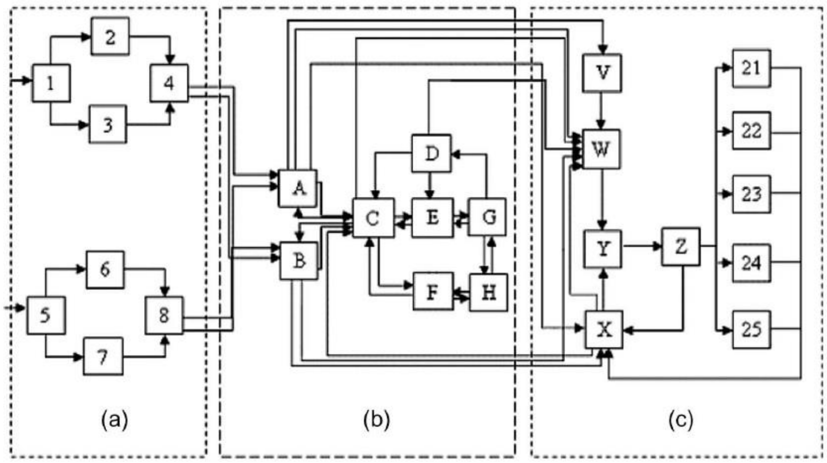
1.2 Queuing Network-Model Human Processor

1.2.1 Queuing Network

Queuing theory is the mathematical study of customers (or entities) waiting in queues before getting services or being processed at servers (or nodes). Once a customer arrives at a server, it would either get service right away, or wait in a queue if the server is at capacity. In queuing theory mathematical models are built to predict queue length, waiting time, server utilization, etc. Queueing network (QN) is a network of servers that are connected by routes. In a QN customers traverse the network from one server to another based on the routing rules until they exit the system. Since the invention of the queuing theory by Agner Krarup Erlang in the 1910s when studying the telephone exchange, QN has become a well-established mathematical discipline and has been widely used in the field of operation research, service science, manufacturing engineering, computing, traffic engineering, telecommunication, etc. Meanwhile, there is less work in using queuing network methods to study the human cognition and performance. There are studies showing evidences of the existence of queuing mechanisms in the brain. For example, Roland, et al., (1980) suggested a queue of time-ordered motor commands at the supplementary motor areas before those motor commands are executed by the primary motor area. Queuing theories were used to model spiking neurons and dynamical synapses (Annunziato and Fusi, 1998, Mattia and Del Guidice, 2000). Liu (1996a) presented a class of QN models of elementary mental processes by using reaction time to infer the plausible configuration of the human mental system. Liu (1997) used queuing networks to model human multitask performance and treated single channel theories and multiple resources theories of attention as special cases of the queueing network modeling method.

1.2.2 Queuing Network - Model Human Processor

Queuing Network - Model Human Processor (QN-MHP) is a cognitive architecture that integrates the mathematical structure of queuing network with the cognitive modeling method of Model Human Processor (MHP). It decomposes the MHP's three discrete serial stages of perceptual, cognitive, and motor processor into three subnetwork of servers (see Figure 1). Each subnetwork is composed of servers that perform distinct functions. The selection of the servers and the connection between the serves are developed on the basis of existing findings in the neuroscience and psychology. Figure 2 shows the approximate mapping of the QN-MHP servers onto the human brain areas. Natural Goal, Operators, Methods, and Selection rules Language (NGOMSL) developed by Kieras (1996) is used as the task analysis method. The detailed specification of the QN-MHP including server configurations, entity arrivals, and routing mechanism could be found in Feyen (2002), Liu, et al., (2006), and Wu (2007).



Perceptual Subnetwork

1. Common visual processing (eyes, lateral geniculate nucleus, superior colliculus, primary and secondary visual cortex)
2. Visual recognition (dorsal system)
3. Visual location (ventral system)
4. Visual recognition and location integration (distributed parallel area including the connections among V3 , V4 and V5, superior frontal sulcus, and inferior frontal gyrus)
5. Common auditory processing (middle and inner ear)
6. Auditory recognition (area from dorsal and ventral cochlear nuclei to the inferior colliculus)
7. Auditory location (area from ventral cochlear nucleus to the superior olivary complex)
8. Auditory recognition and location integration (primary auditory cortex and planum temporale)

Cognitive Subnetwork

A. Visuospatial sketchpad (right-hemisphere posterior parietal cortex)
B. Phonological loop (left-hemisphere posterior parietal cortex)
C. Central executive (dorsolateral prefrontal cortex (DLPFC), anterior-dorsal prefrontal cortex (ADPFC) and middle frontal gyrus (GFm))
D. Long-term procedural memory (striatal and cerebellar systems)
E. Performance monitor (anterior cingulate cortex)
F. Complex cognitive function: decision, calculation, anticipation of stimulus in simple reaction etc. (intraparietal sulcus (IPS), the superior frontal gyrus (SFS), the inferior frontal gyrus (GFi), the inferior parietal cortex and the ventrolateral frontal cortex, the intraparietal sulcus and the superior parietal gyrus)
G. Goal initiation (orbitofrontal region and amygdala complex)
H. Long-term declarative & spatial memory (hippocampus and diencephalons)

Motor Subnetwork

V. Sensorimotor integration (premotor cortex)
W. Motor program retrieval (basal ganglia)
X. Feedback information collection (somosensoy cortex)
Y. Motor program assembling and error detecting (supplementary motor area (SMA) and the pre-SMA)
Z. Sending information to body parts (primary motor cortex)
21-25: Body parts: eye, mouth, left hand, right hand, foot

Figure 1. General structure of the QN-MHP cognitive architecture (from Wu and Liu, 2007a)

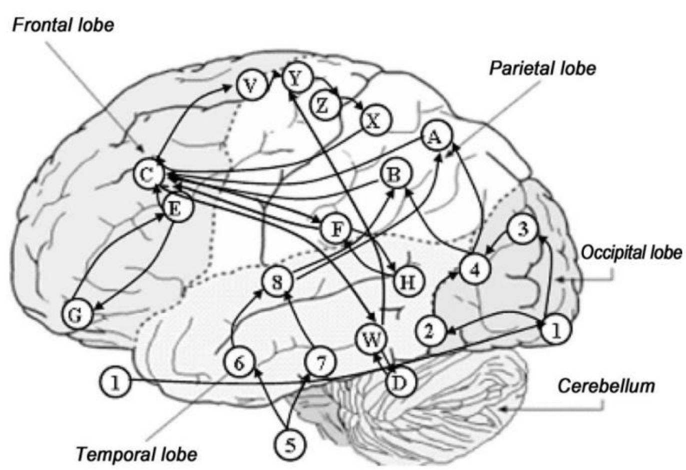


Figure 2. Approximate mapping of the QN-MHP servers onto the corresponding human brain areas (from Wu and Liu, 2007a)

The QN-MHP framework has been successfully used to simulate a wide variety of human performances including simple and choice reaction time tasks (Feyen and Liu, 2001), Psychological Refractory Period (PRP) (Wu and Liu, 2008a), visual search (Lim and Liu, 2004a, 2004b, Lim, et al., 2010, Feng and Liu 2013), map reading (Liu et al., 2006), transcription typing (Wu and Liu, 2008b), vehicle steering (Tsimhoni and Liu, 2003), driver performance and workload (Wu and Liu, 2007a, Feng and Liu, 2014), etc. Recent work has also fully implemented ACT-R as a special case of the queuing network model (Cao and Liu, 2013a). Compared with all major existing cognitive architectures, QN-MHP has its strength in modeling multitasking performance for three major reasons.

First, unlike other cognitive architectures in which the cognitive processing is either serial (e.g., ACT-R) or parallel (e.g., EPIC), QN-MHP offers a hybrid structure in its cognitive subnetwork. The central executive server (Server C in Figure 1) provides processing for low-level cognitive operations and it allows multiple operations to be processed at a time. The complex cognitive function server (Server F in Figure 1) provides processing for high-level cognitive operations such as numerical calculation, and it allows only one cognitive operation to be processed at a time. Compared with models with either serial or parallel processing, this hybrid structure incorporates a larger body of the psychological findings on multitasking. This feature is essential to model the multitasking performance involving driving, as even driving itself may not require much cognitive demand in a normal driving situation for experienced drivers, using an in-vehicle system may induce significant cognitive demand as the driver may not be familiar with the system, and have to perform tasks such as searching for particular information on a screen, etc.

Second, when modeling multitasking, most other existing cognitive architectures rely on task-dependent executive controls to strategically lock and unlock a task when switching between tasks. As being discussed by Kieras, et al., (2000), this is not a preferred method for multitask modeling as the executive controls need to be re-specified by the modeler each time new tasks are introduced. On the other hand, QN-MHP does not have a central executive control at a high level for task switching. QN-MHP uses a generic approach in which no central executive controls are specified. Instead, information entities representing different tasks could traverse the network at the same time, while they are competing for service at the local server level.

Third, a large proportion of in-vehicle non-driving tasks (e.g., setting the radio to a particular channel) are procedure-based visual-motor tasks that do not involve significant complex cognition such as problem solving, reasoning, or language comprehension. The NGOMSL technique that QN-MHP uses for task analysis is less of a barrier for modeler and model users compared with the production rule based architectures.

Additionally, QN-MHP has other distinct features that are beneficial to model the multitasking performance while driving. These features include generating instantaneous workload estimates at both the server and subnetwork level, and simulation visualization.

1.3 Scientific Merits and Broad Impact

Extensive research has been conducted to study the behavior and mechanism of human visual search task, which generally refers to the situation in which a person looks for a particular object among a number of distracting objects in a visual field (Treisman & Gelade, 1980, Wolfe, 2007). But most of the studies considered visual search as the only task performed. Less research has been done to study visual search as one of several tasks that are performed simultaneously and the interference among the tasks (Liu & Wickens, 1992; Liu, 1996b, 1996c). In the automotive and some other domains such as aviation and railway, it is a common task scenario when a human needs to interact with some device (e.g., finding an item on an electronic map or dialing a number on a phone) while simultaneously performing another continuous task (e.g., operating a vehicle). It is of great importance to gain insights into the mechanism of human multitask behavior in order to design the task environments and user interfaces (UIs) that facilitate human performance and minimize the potential safety hazards.

In the first experiment I employed experimental work that investigates how a driver's eye glance behavior and task completion time is affected during a visual search task by two basic design parameters of a touch screen device (i.e., number of buttons and their sizes). The findings from this study make contributions to the existing knowledge of how the human's eye glance behaviors are affected by increased visual search difficulty (induced by a larger amount of buttons displayed on the screen) and increased reaching difficulty (induced by a smaller size of buttons) when performed simultaneously with a continuous vehicle steering task.

A computational model based on the cognitive architecture of QN-MHP was built to account for the findings from the experiments. A modeling mechanism for flexible task activation (rather than strict serial activations) was developed to allow task component activation to be based on the status of other task components. A task switching scheme was built to allow segmentations of tasks to model time-sharing nature of multitasking as observed from the experiment. These extensions offer new theoretical insights into visual search in multitask situations and enable the model to simulate parallel processing both within one task and among multiple tasks.

In the second experiment I investigated the effect of three common control modules on the eye glance behavior, task completion time, and workload for a typical real-world radio-tuning task during simulated driving. The findings from this study make contributions to the existing knowledge of how the human's eye glance behaviors, task completion time, and workload are affected by different features of control modules (physical buttons vs. virtual buttons) and different input methods (pressing buttons vs. turning knobs). This experiment was modeled using the QN-MHP model with the multitasking features described above. The validation results show that the model could account for the observed behavior and performance differences among the control modules from the empirical data.

Based on this model, a computer-aided engineering toolkit was developed to enable the UI designers of in-vehicle infotainment systems to evaluate, predict, and benchmark the usability of design concepts and prototypes. From the engineering application and practical value perspective, the new toolkit has great advantages over the traditional usability testing methods with human subjects. It enables the UI designers to explore a larger design space and address usability issues at the early stages with lower cost both in time and manpower. This work was based on a generic cognitive architecture modeling approach that has the potential to be applicable to other multitasking domains.

1.4 **Dissertation Structure**

Chapter 2 introduces the latest development on the QN-MHP cognitive architecture based on previous work. The modeling work of a new driver steering module is described in this chapter. This lays the foundation of multitask modeling involving both steering and non-steering task, as in the following chapters.

Chapters 3 and 4 describe an experiment with a visual search task on an in-vehicle touch screen device during simulated driving and the corresponding modeling work. Chapter 3 describes the experiment for examining the effects of two basic design parameters of touch screen UIs (i.e., number of buttons and their sizes) on driver's glance behavior and task completion time while simulated driving. A simple but yet common visual-motor task of finding and pressing a particular button on a screen was developed for the examination. The participants' eye glance and performance data were collected and analyzed. Chapter 4 describes the modeling work of the tasks examined in Chapter 3 using QN-MHP. The results demonstrate that the model is able to generate eye glance behaviors and task completion time that are very similar to the human results reported in Chapter 3.

Chapters 5 and 6 describe an experiment with realistic radio-tuning tasks during simulated driving and the corresponding modeling work. Chapter 5 describes the experiment for examining the effects of typical control modules of in-vehicle infotainment systems on the driver's eye glance behavior, task completion time, and workload. Three control modules were examined including a touch screen system, a physical panel with a "direct tune" function, a physical panel with a tuning knob. The participants' eye glance behavior, task completion time, and the subjective workload were collected and analyzed. Chapter 6 describes the modeling work of the radio-tuning tasks examined in Chapter 5 using QN-MHP. The results demonstrate that the model is able to generate eye glance behaviors and performance measures that are very similar to the human results reported in Chapter 5.

Chapter 7 describes the development of a computer-aided usability testing software for in-vehicle infotainment systems. The software supports the usability testing of the UI designs created using MATLAB's GUIDE (Graphical User Interface Design Environment). It features a graphic user interface (GUI) with four steps for setting up a simulation, and reports the predicted human eye glance behaviors and other performance metrics.

Chapter 8 summarizes the results and conclusions from this dissertation work and discusses potential directions for future research.

Chapter 2.
A Driver Steering Model Using the QN-MHP

2.1 Introduction

Driving is a common yet complicated task that many people perform on daily basis. The U. S. Department of Transportation (DOT) Federal Highway Administration (FHWA) reported that on average an American driver logs 13,476 miles each year (http://www.fhwa.dot.gov/ohim/onh00/bar8.htm). And yet road accidents account for 33,561 deaths in America in 2012 according to the U.S. National Highway Traffic Safety Administration (NHTSA) (NHTSA, 2013). Among the road accidents, driver errors have been regarded as one of the leading causes. According to NHTSA, in 2010 an estimate of 899,000 or 17% of police-reported crashes involves a distracted driver, causing 3,092 fatalities or 9.4% of those killed and over 400,000 injuries. Ranney et al, (2000) found driver inattention account for approximately 25% of police reported crashes. A naturalistic 100-car driving study conducted by Virginia Tech Transportation Institute estimated that 78% of crashes and 65% of near crashes are related to driver inattention (Klauer et al., 2006).

These driver-related errors became an even more urgent issue in the past few years with the introduction of modern in-vehicle infotainment systems that allow the drivers to accomplish many non-driving tasks using multimodal interfaces such as touch screens, digital instrument clusters, and head-up-displays. Although such technologies are designed to enhance the driving experience, they may suffer from usability problems, such as driver distractions with frequent and extended eyes-off-road operations, prolonged learning curve and frustration with information overload. Computational driver models are a valuable tool set as it could help the researchers and human factor practitioners to understand the mechanism of the driving task and to test, benchmark, and make quantitative predictions on the human factor issues related to driving.

Driving by itself is a complicated task that is composed of "subtasks" including (1) vehicle control tasks, for example, lateral control (e.g., lane-keeping), longitudinal control (e.g., car-following), or a combination of both lateral and longitudinal control (e.g., passing a car); (2) higher level tasks, for example, looking for potential hazards (e.g., obstacles, pedestrians), recognizing road signs or traffic lights, trip planning, etc. Each of these subtasks is a complex research topic on its own. In this study we are primarily focusing on the driver's lateral (steering) control of the vehicle to maintain a proper lateral position in the lane.

Several studies have been done to model the driver's steering behavior in the past decades. One school of steering models is based on control theories which consider the driver as one of the controllers in the driver-vehicle system (Weir and McRuer, 1973; Donges, 1978, see MacAdam, 2003 for a review). In these models the driver's control behavior is typically described as a continuous transfer function in the control system. And the model usually has a seamless incorporation with rigorous vehicle dynamic models which are typically developed under the control theory. These models are often criticized as been unrealistic as little consideration was given to the limitation of the human perception, cognition and motor processing. For example, most of the inputs to the model (e.g., vehicle yaw angle, road curvature) are likely not readily perceivable by the human driver. Also since these models typically do not include the components that represent the human limitation and constraints, it's hard to expend the model capability to model the multitasking in which the driver is doing a non-driving task (e.g., making a phone call) while driving.

Another more recent school of models is based on task-independent cognitive architectures which are fixed structures that represent a generic human being with characteristics and limitations derived from numerous experimental physiology and neuroscience studies. Examples of these models include the ACT-R (Adaptive Control of Thought-Rational) based driver model (Salvucci, 2006), Soar-based driver model (Aasman, 1995), and the QN-MHP (Queuing Network-Model Human Processor) driver model (Tsimhoni and Liu, 2003; Wu and Liu, 2007a). Building these models require a thorough understanding of the corresponding cognitive architecture in order to incorporate the driving task into the task-independent framework. But they have the potential of providing a more accurate representation of multitasking performance by modeling the resource sharing among the tasks.

Among those models, ACT-R architecture has been applied in a wide range of applications. But one of the essential assumptions of ACT-R is serial processing with only one production rule being fired at any time. And task dependent central executives may be needed to model the task switching in multitasking scenarios. While QN-MHP allows parallel processing in the queuing network, the information processing flow is governed by the service time and capacity of the local servers in the queuing network. Thus information could be processed concurrently in the queuing network. At the same time, the information representing different tasks could compete for service at the local server level. It is promising to use the QN-MHP structure to establish a more realistic and physiologically plausible model of driving and multitasking that includes driving. The work described in this chapter is a continuation of the previous effort on cognitive-architecture-based driver modeling work described above.

2.2 **Methods**

2.2.1 QN-MHP Cognitive Architecture

QN-MHP simulates human cognition as a queuing network of information processing servers, derived from the psychological and neuroscience findings (Liu 1996a, 1997, Liu et al, 2006). The queuing network is composed of three subnetworks (i.e., perceptual, cognitive, and motor). Each subnetwork is composed of individual servers, each of which represents a certain function in the brain for information processing (See Figure 1). The QN-MHP version used in this study is implemented in MATLAB/Simulink software. Figure 3 is a screen shot of the model implementation in MATLAB.

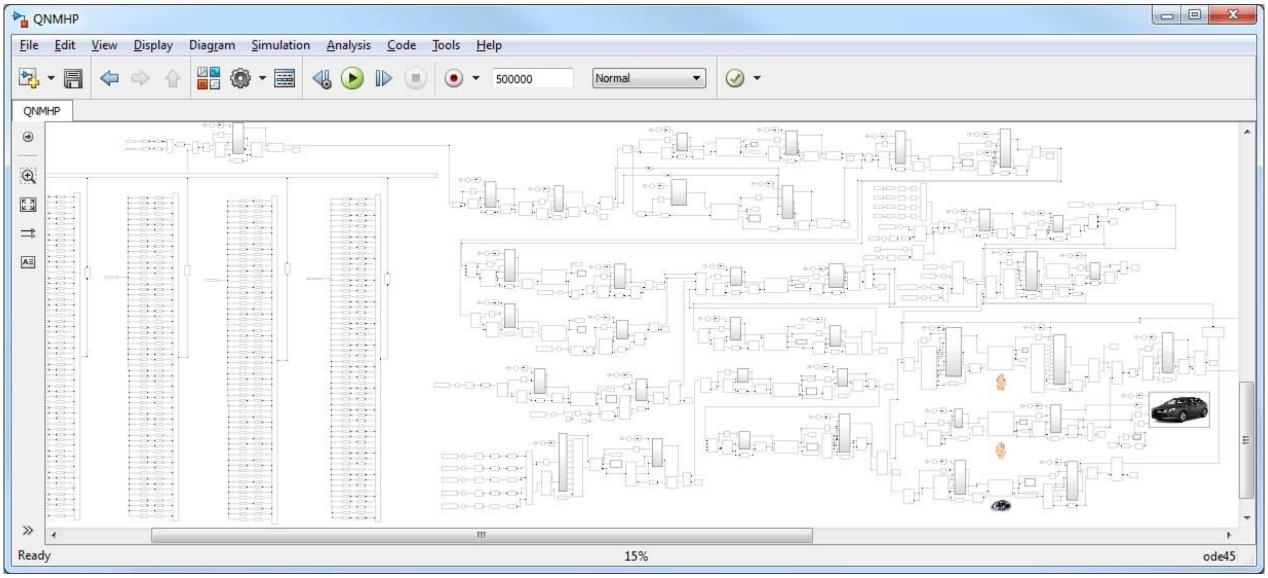


Figure 3. A screenshot of the model implementation in MATLAB

The model structure with four major components is shown in Figure 4. The task environment (box a in Figure 4) represents the environment with which the digital driver could interact with. It stores the information about the driving environments once they are specified during the simulation setup. During the simulation run it receives outputs from the QN-MHP's body part servers (e.g., the hand server to turn the steering wheel), and supplies updated input stimulus to the QN-MHP (box d)'s perceptual servers.

The vehicle dynamics (box b) is a built-in module that receives input from the QN-MHP's driving related actions (e.g., steering), and generates the vehicle responses which would be used to update the driving environment in box a. Currently a three-Degree-Of-Freedom (*longitudinal*, *lateral* and *yaw*) bicycle model is implemented for its simplicity. More details on the vehicle dynamics model could be found in Appendix A.

The QN-MHP (box d) represents the generic digital human. Its procedural long-term memory server stores the task information once it is specified in the simulation setup. During the simulation the task information is used as the instructions to the digital driver on how to perform the tasks. During the simulation, the QN-MHP is able to generate the task performance based on the information available at the queuing network.

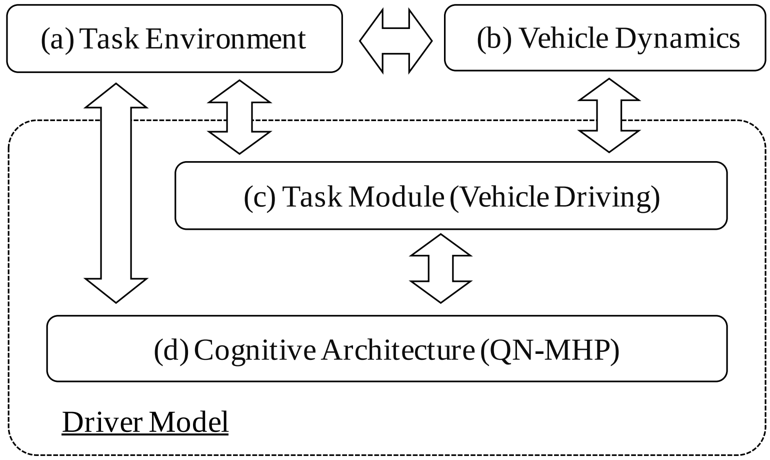
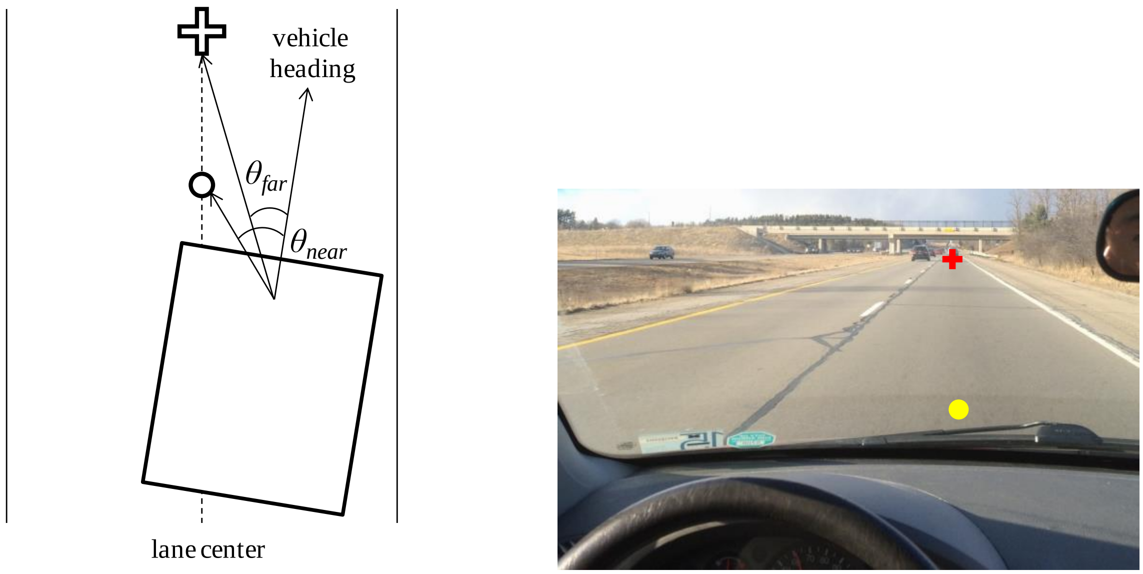


Figure 4. Major components of the model

2.2.2 Modeling the Steering Task

One of the basic assumptions of our driver model is that while driving, a driver utilizes two distinct visual cues on the road (termed near point and far point, respectively) to determine how to turn the steering wheel. This assumption is based on empirical driving studies on which parts of the road ahead supply the visual information needed by the driver (Donges, 1978; Land & Horwood, 1995). The near point represents a visible point in front of the vehicle that the driver uses to judge how close the vehicle is to the lane center. The far point represents a visible point in front of the vehicle that the driver uses to predict a near future position and apply predictive compensations. A steering control algorithm from Salvucci (2006) was used, which assumes the driver determines the steering wheel φ based on the perceived near angle θ_{near} and far angle θ_{far}. The near angle is the direction from the vehicle pointing to the near point relative to the direction of the vehicle heading. The far angle is the direction from the vehicle pointing to the far point relative to the direction of the vehicle heading, as illustrated in Figure 5.



(a) Bird's-eye view (b) Driver's point-of-view

Figure 5. Near and far point of the driving model

A control law proposed by Salvucci (2006) is used to calculate the steering wheel adjustment:

$$\Delta\varphi = k_{far}\Delta\theta_{far} + k_{near}\Delta\theta_{near} + k_{I}\min(\theta_{near}, \theta_{nmax})\Delta t$$

In which:

θ_{near} and $\Delta\theta_{far}$ are the differences of the near angle and far angle from the last cycle.

k_{near}, k_{far}, and k_{I} are the weights of the three terms

θ_{nmax} is for limiting the contribution of the θ_{near} to changes in steering angle

Δt is the time elapsed from last cycle.

A NGOMSL-style task analysis for the driving task is shown in Figure 6. A steering task cycle is composed of five Task Components (TC 1 to 5).

GOAL: Steer the vehicle in the lane

Method to accomplish goal of steering the vehicle in the lane

TC 1: Look at <far/near angle> on <road scene> at location <far point>

TC 2: Store <far/near angle value> to short-term memory

TC 3: Retrieve <far/near angle value > from short-term memory

TC 4: Determine the steering wheel angle to turn, return *AdjustmentAngle*

TC 5: Turn the steering wheel with *AdjustmentAngle*

TC 6: Go to TC 1

Figure 6. NGOMSL-style task description for the steering task

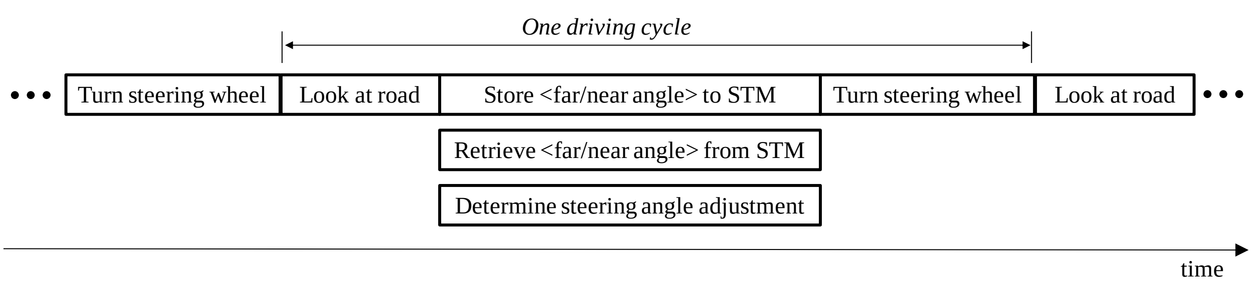


Figure 7. Sequential dependency of the task components

In the QN-MHP model the near point and far point are represented as two streams of visual stimuli that arrive at the visual perceptual server (Server 1) at a fixed inter-arrival time (currently set as 50 ms as the default value for the visual stimulus generation rate), and traverse the queuing network. The arriving entities representing the near (/far) point carry the value of the current near (/far) angle as an attribute (set by a vehicle dynamics module). At the beginning of each cycle of the steering, the model first processes the entities representing the far and near point arriving at the same time, extracts the far and near angle from the entity's attribute at the visual perceptual subnetwork. The extracted far and near angle values are stored in the Visuospatial Sketchpad server (Server A) in the cognitive subnetwork. Once both the far and near angle values are available for determining the steering angle, it takes one cognitive processing time at the Central Executive server (Server C) to determine the value of the steering angle adjustment. Once this computation is completed, the motor servers start to prepare and execute the steering action. Once the steering action is executed at the hand server, the information entity carrying the steering angle adjustment as its attribute is sent to the vehicle dynamics module (see Appendix A), which immediately generates the updated near angle and far angle based on the vehicle dynamics. The updated near angle and far angle are then used to set the attribute values for the corresponding near/far point entities in the continuous stream of near/far point entities to arrive at the perceptual servers as driving continues. The path of the information entities in the queuing network is Server 1 → Server 2/3 → Server 4 → Server A → Server C → Server W → Server Y → Server Z.

2.3 **Results**

As the first step of validating the steering model, four simulation conditions were selected with varying initial vehicle lateral positions in the lane (0.3, 0.6, 0.9, and 1.2 meters to the lane center, respectively). The vehicle speed is set at 65 miles per hour (=29 m/s). The initial vehicle heading angle and steering wheel angle are both set at 0 degree. Figure 8 shows the simulation result of the vehicle lateral position in the lane. It shows the steering model was able to steer the vehicle to the center of the lane and maintain the center position under all four conditions.

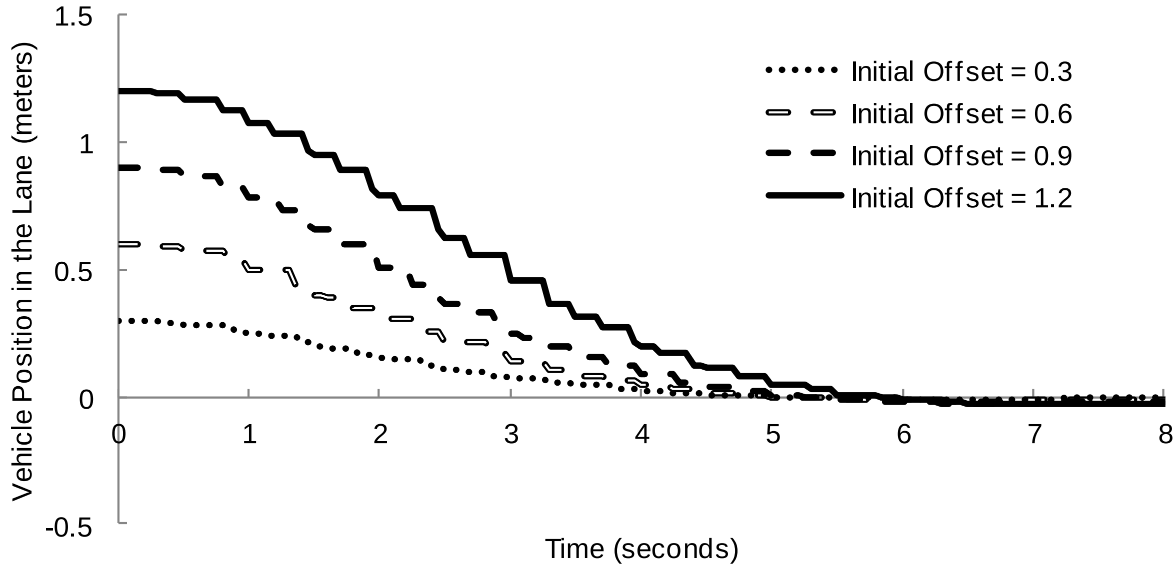


Figure 8. Simulation results of the vehicle lateral position in the lane. Positive (/negative) values means to the right (/left) of the lane center

Figure 9 shows the simulated steering wheel angle over time. The characteristics that can be observed include:

(1) For all 4 initial lateral positions, the steering adjustments are roughly composed of two phases before fading out to zero. The first phase is characterized by a curve with negative values (roughly from 0 to 3 seconds in Figure 9), which can be explained as the model is steering the vehicle to the center of the lane. It is followed by a second curve with positive values (roughly from 2 to 8 seconds in Figure 9), which can be explained as the model is straightening the vehicle with a target 0 degree heading.

(2) In the first phase larger amplitude of steering with shorter duration is applied compared with the second phase. This shows the model applies a relatively harsh adjustment to steer the vehicle to the center first, then uses a mild adjustment to straighten the vehicle.

(3) Larger initial offsets (e.g., 1.2 meters) induce larger steering adjustments in both the first and second phase compared with smaller initial offsets (e.g., 0.3 meters).

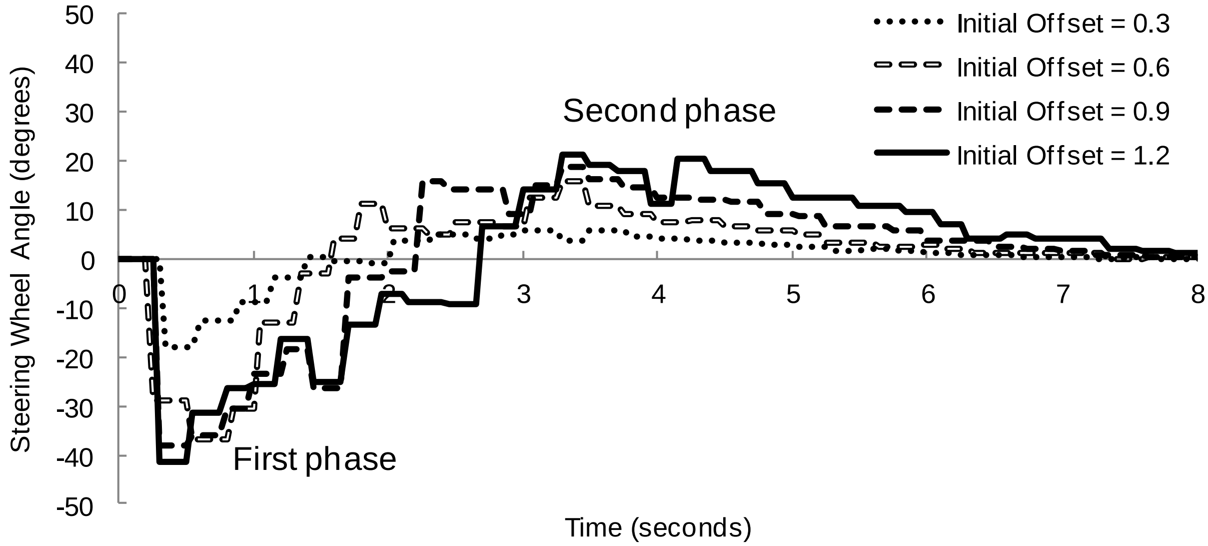


Figure 9. Simulation results of the steering wheel angle. Positive (/negative) values means steering to the right (/left) from the neutral position

To further validate the model, data from a simulated driving study (described in detail in later chapters) was used. In the experiment the subjects were asked to drive the simulated vehicle on a virtual highway. The virtual highway has two lanes in one direction, and the virtual course is a square loop with four straight sections. The driving environment is set as day time. The subjects were asked to keep the vehicle in the left lane and maintain a speed between 60-70 miles per hour. There is no other virtual vehicle in the left lane. At certain points of the driving, the subjects were asked to perform radio tuning tasks. The subjects were asked to continue with the driving task once the radio tuning task is completed. Starting from the end of each radio tuning task, a 8-second data segment were extracted to capture the subjects' behavior of how to steer the vehicle back to the center of the lane. Figure 10 shows one example of the vehicle lateral position in the 8-second segment for one of the participant and the simulation results when the same initial conditions were used (e.g., vehicle lateral position = -0.47 m, vehicle speed = 27.5 m/s). The result shows the model is able to capture some major temporal characteristics of the vehicle lateral position.

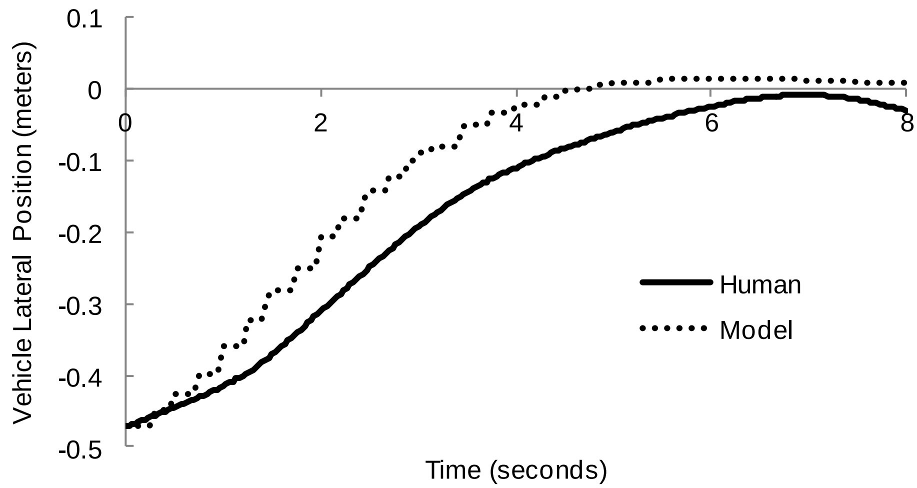


Figure 10. Model validation of the vehicle lateral position

2.4 Discussions

In this chapter we proposed a driver steering model based on the QN-MHP cognitive architecture. This model utilizes the latest findings in the driver behavior and modeling work, and incorporates them into the QN-MHP framework. The results show the model is able to capture the main steering behaviors in terms of some of the major temporal characteristics of vehicle lateral position and steering wheel angle. It could be a promising tool for modeling the driving performance, and the multitasking performance that involves driving.

There are certainly limitations to this model. Currently the model only simulates the lateral control subtask of driving. However, as we mentioned earlier, the driving task includes many other subtasks such as longitudinal control, obstacle/hazards detection and avoidance, etc. These subtasks are also essential components to get a more realistic and complete driver model. By leveraging on the inherent advantage of cognitive architecture based modeling approach, it is promising to model these subtasks and incorporate them into a more integrated driver model in future work. It should be noted that currently the model only represents a "typical" driver. There's yet parameter to account for the behavior difference between different driver populations, for example, the novice and skilled driver, the conservative and aggressive driver, or the younger and older driver, etc. More work could be done in this regard to account for the variability of the driver population. Also the model assumes only the visual information is used as model inputs, while studies have found other sensory inputs may also play a role in the steering task. For example, studies have found the vestibular and kinesthetic channels add useful information to improve the driving performance (McLane, et al., 1975, Greenberg, et al., 2003). The modeling work described in this chapter lays the foundation of multitask modeling involving both steering and non-steering task, as in the following chapters.

Chapter 3.
An Experimental Investigation on the Effects of Number of Buttons and Button Size on Visual Search Task Performance while Driving

3.1 Introduction

In recent years electronic infotainment systems have been brought into many vehicles, allowing drivers to accomplish various non-driving tasks (e.g., listening to music, adjust temperatures) while driving. These systems are able to integrate a large number of functionalities (e.g., audio, communication, navigation) into one single device. Studies have suggested that interaction with some in-vehicle systems may have a detrimental effect on driving performance (Lansdown, 2004). Wierwille and Tijerina (1998) have previously demonstrated a relationship between the visual demands of in-vehicle systems and traffic accident rate. Accident data from 2010, as compiled by the NHTSA, indicated that 17% (an estimated 899,000) of all police-reported crashes involved some type of driver distraction. Of those 899,000 crashes, distraction by a device/control integral to the vehicle was reported in 26,000 crashes (3% of the distraction related police-reported crashes).

A number of studies have been done to investigate the impact of non-driving tasks with visual displays or touch screens on driving performance. Tsimhoni, et al, (2004) found the driving control was significantly degraded when typing an address on a touch screen keyboard. Horrey and Wickens (2004) found that using a head-down display degrades task performance compared with using a display which is more proximally located to the road scene.

However, less research was done to study how the different design parameters of a touch screen device would affect the driver's visual search task performance. Boyle, et al, (2013) studied how different text length (short medium, long) affects the driver's visual behavior for a text entry and text reading task. Mehler, et al (2012) examined an address entry task using portable telematic devices including a touch screen. Jin, et al (2007) studied the effect of touch screen button sizes and spacing on older adults as a single task (i.e., without driving involved).

The number of objects on a screen and their sizes are two basic design parameters for any UIs. Wolfe (2006) summarized the findings of visual search, including that the efficiency of visual search usually decreases with the increasing number of distractors. Visual search efficiency is also affected by the spatial distribution of the items in the visual field. As the density of the items increases, visual search usually becomes faster (Nothdurft, 2000), as it is less likely to require eye saccades and/or head movements in order to move the new items into the fovea vision. On the other hand, the search time increases when the items are getting too close to each other, so that it prevents the identification of the individual items (Vlaskamp & Hooge, 2006). While visual search has been extensively studied in the past, most of the research work focused on studying visual search as a single task. Little research has been done to study visual search during multitasking scenarios, for example, how other ongoing tasks may affect the strategy and execution of the visual search task. While in the real world conditions, visual search is often performed as only one of several active tasks.

This chapter describes an experiment to examine the effects of two basic design parameters of touch screen user interfaces (UIs) with virtual buttons (i.e., number of buttons and button sizes) on driver's glance behavior while simulated driving. A simple visual search and button pressing task was designed, in which the drivers were asked to find a specific target button on the screen among other buttons, and press on the target button once they found it. This task would be performed as a single task (i.e., without driving involved) and a dual task (i.e., while driving a simulator). The driver's glance behavior were recorded using video cameras, and various glance metrics were used to investigate how different design parameters would affect the driver's glance behavior, and the implications to road safety.

3.2 **Methods**

3.2.1 Participants

Twenty participants (10 male and 10 female), all of whom were employees of a company in the United States, were recruited to participate in the experiment. For the age distribution, 9 (45%) participants were 20-29 years old, 1 (5%) participants were 30-39 years old, 4 (20%)

participants were 40-49 years old, 5 (25%) participants were 50-59 years old, 1 (5%) participants were 60-69 years old. All participants have obtained valid driver license for at least one year.

3.2.2 Experiment setup and procedure

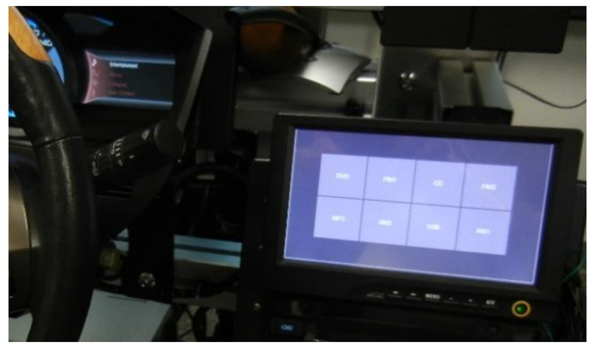
The experiment was conducted in a fixed-base driving simulator in a laboratory environment as shown in Figure 11. The front virtual road scene was projected on a flat projection screen in front of the driving simulator. A resistive touch screen was mounted in the center console area of the driving simulator. The touch screen was 31 degree below the horizontal line of sight, and 51 degree to the right of the center. The touch screen has an 8-inch (203 mm) diagonal size with a screen resolution of 800-by-480 pixels. Two video cameras with a 30 Hz frame rate were mounted on the driving simulator. One camera was directed at the participant’s eyes. After the experiment, participant’s gaze direction was manually coded from the video by a human data reducer. Another camera captures a close-up view of the area from the steering wheel to the touch screen.

Figure 11. Experiment on a driving simulator

For the visual search task, multiple buttons with different labels were displayed on the touch screen (see Figure 12b). The goal of the task was to find the target button which was labeled with “USB”, and press the target button with the right hand. The participants were asked to put their hands on the steering wheel before and after the search task. The target button is always the “USB” button for the entire experiment.

The participants were asked to drive the simulator in a virtual highway (see Figure 12a). The virtual highway has two lanes both in the same direction, and the virtual course is a square loop with four straight sections connected by four curved corners. The driving environment was set as day time. The participants were asked to keep the vehicle in the left lane and maintain a speed of between 60-70 miles per hour. There was no other virtual vehicle in the left lane. The participants were asked to always put both hands on the steering wheel except when doing the visual search task.

(a) Driving simulator virtual environment



(b) The touch screen for the non-driving task

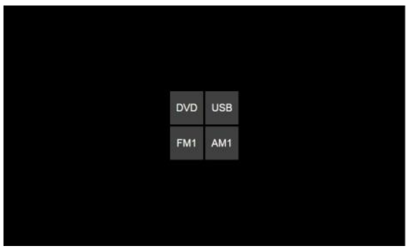
Figure 12. Driving and non-driving tasks

Once the participants arrived at the laboratory, they were firstly asked to complete a consent form at a desk. Then they were asked to sit in the driving simulator. They were asked to adjust the seat position (forward and backward, up and down) to make sure they can reach the touch screen with the arm only (i.e., without whole body movements). Then an introduction session was given to the participants to ensure that they understood the tasks they were about to perform. This was followed by a practice session for both the button searching task and the driving task. Then the experiment proceeded to the data collection part, which was divided into a single task session and a dual task session. Half of the participants started with the single task session, while the other half started with the dual task session. The order of the trials was randomly placed in each session, but the same order was used for all participants.

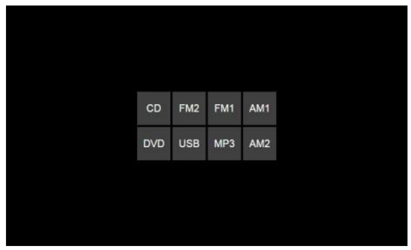
The experimenter only instructs the participants to start the button search task only when the vehicle is in the straight section of the road (i.e., not when the vehicle is entering, negotiating, or leaving a curve). The participants were asked to perform the button search task only when they believe it is safe to do so.

3.2.3 Experimental Design

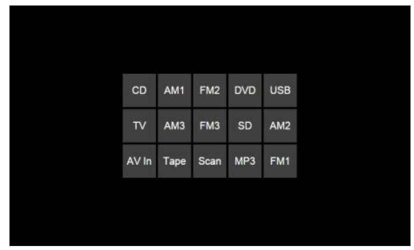
There are three independent variables in this experiment: (1) Task condition (two levels: single task or dual task), (2) Button sizes (three levels: small, medium, or large), (3) Number of buttons (three levels: 4, 8, or 15). A full-factorial within-subject design was used for the experiment. The nine combinations of button sizes and number of buttons are illustrated in Figure 13.



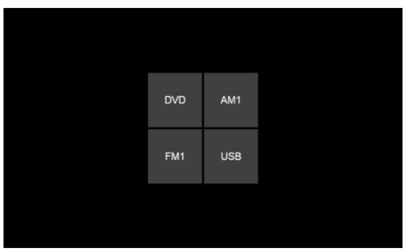
(a) 2x2 Small Buttons



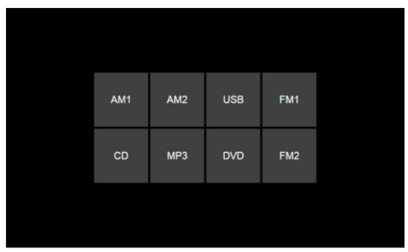
(b) 2x4 Small Buttons



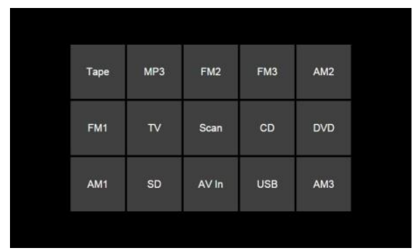
(c) 3x5 Small Buttons



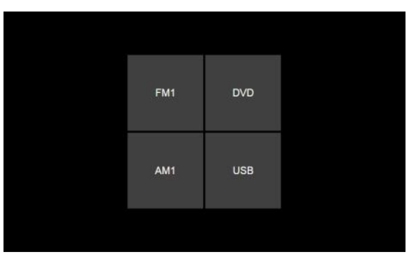
(d) 2x2 Medium Buttons



(e) 2x4 Medium Buttons

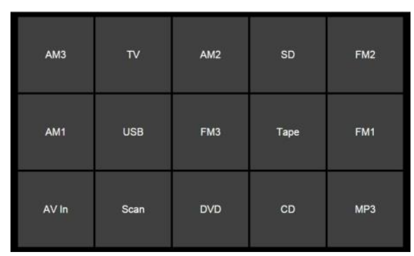


(f) 3x5 Medium Buttons



(g) 2x2 Large Buttons

(h) 2x4 Large Buttons



(i) 3x5 Large Buttons

Figure 13. Nine Test Conditions (3 levels for button sizes and 3 levels for number of buttons)

For all the combinations, the font size of the labels remains the same (20 pixels). The horizontal and vertical spacing between buttons also remains the same (4 pixels, or 0.9 mm). The default background color of the buttons is grey (HTML color code: #404040). Once a button is pressed, the button's background color changes to dark green (HTML color code: #1B402C). The touch panel designs of each combination were created using HTML and presented in a web browser in full screen mode.

3.2.4 Independent Variables

(1) Button Sizes

Three button sizes were examined in the experiment: (1) small (66 pixels or 14 mm), (2) medium (108 pixels or 24 mm), and (3) large (150 pixels or 33 mm).

(2) Number of Buttons

Three layouts of the buttons were examined in the experiment: (1) 4 buttons in a 2-by-2 layout, (2) 8 buttons in a 2-by-4 layout, and (3) 15 buttons in a 3-by-5 layout.

The labels for the buttons are selected from common acronyms for media sources. For the 2x2 layout, the 4 labels are “AM1”, “FM1”, “USB”, and “DVD”. For the 2x4 layout, the same labels from the 2x2 layout were used with the addition of “AM2”, “FM2”, “MP3”, and “CD”. For the 3x5 layout, the same labels from the 2x4 layout were used with the addition of “AM3”, “FM3”, “TV”, “SD”, “Tape”, “AV In”, and “Scan”.

(3) Single or dual task

In the single task condition, the simulated vehicle is stopped on the side of the road. The participants were instructed to conduct the visual search task without driving the simulator. In the dual task condition, the participant was asked to drive the simulator in a highway scenario, and at given points, they were verbally instructed by the experimenter to do the button search task. The experimenter only instructed the participants to start the button search task when the vehicle was in the straight section of the road (i.e., not when the vehicle is entering, or negotiating, or leaving a curve). The participants were asked to perform the button search task only at a time when they believed it was safe to do so.

3.2.5 Dependent Variables

The task completion time was selected as a performance measure of the button-pressing task. Several glance behavior measures were selected, including total eyes off road time (TEORT), number of eyes-off-road glances, and proportion of long eyes-off-road. TEORT is the cumulative time when the driver is looking away from the road when performing the non-driving task. Studies have shown that besides the TEORT, long glances off the road are particularly related to road crashes and near crashes (NHTSA, 2010). Horrey and Wickens (2007) have shown that compared with the average glance durations, the tail end (i.e., larger values) of the glance duration distribution is more related to crash risks. A single glance is considered as “long” if its duration is longer than 2.0 seconds. This threshold was widely used in existing guidelines and standards (e.g., AAM 2006, NHTSA 2013). Note there are different threshold values been used in some other studies (1.6 seconds used by Wierwille 1993b, Horrey and Wickens, 2007).

A total of 4 trials were used. That gives a total of 72 trials for each participant.

3.3 **Results**

3.3.1 Task Completion Time

For the task when the simulator is parked, the task completion time is the duration from the onset of the buttons on the touch screen to the time when a button is pressed. Note the participants look directly at the touch screen at the beginning of each trial. For the task performed during driving, the definition of task completion time from Tsimhoni and Green (2001) was used, which defines it as the duration from the beginning of the first glance at the device to the end of the last glance during the trial.

Figure 14 shows the distribution of the task completion time under the 9 test conditions when the simulator is parked. Figure 15 shows the distribution when the participants were driving the simulator. It seems that in both the parked and driving conditions, the tasks with larger number of buttons are associated with more tasks with longer completion time. Shapiro-Wilk tests were conducted to test the normality of the distributions. The results show that when the simulator is parked, the data violate the normality in all conditions ($p < 0.013$) with the exceptions of 2x2 small (W = 0.986, $p = 0.509$), 2x2 medium (W = 0.987, $p = 0.621$), 2x2 large (W = 0.977, $p = 0.168$), and 2x4 large (W = 0.979, $p = 0.206$). When the participants were driving the simulator, the normality is violated in all 9 conditions (all $p < 0.05$).

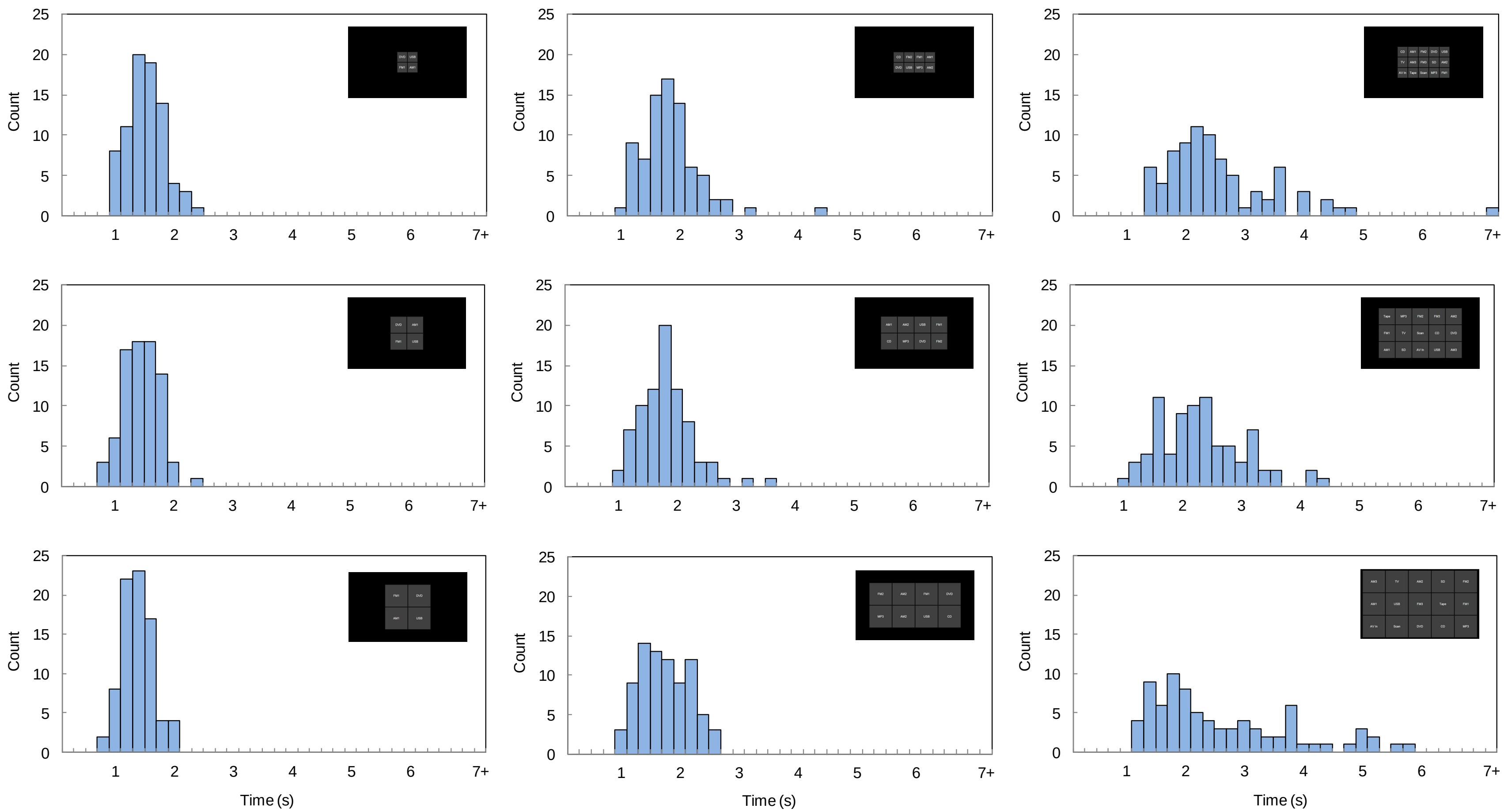


Figure 14. Distribution of task completion time by button number and size in the PARKED condition

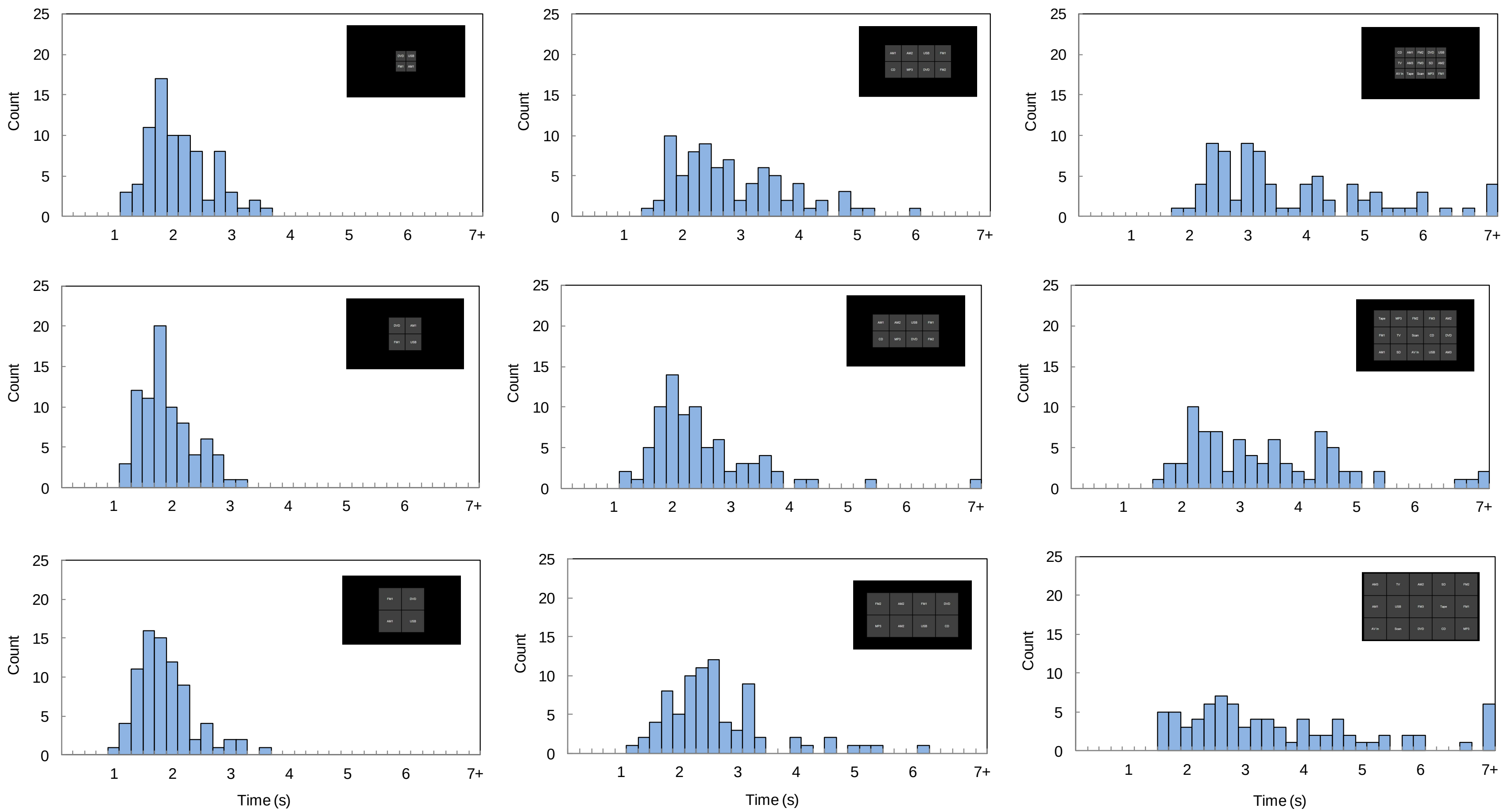


Figure 15. Distribution of task completion time by button number and size in the DRIVING condition

Table 1 shows the statistics of the task completion time for the nine task conditions. Since the normality was violated in some of the conditions, the nonparametric Friedman Test was used to examine whether there are significant differences of the task completion time among different levels of the independent variables.

Table 1. Statistics of task completion time in experiment 1 (all units are in seconds)

Number of Buttons	Button Size	Parked				Driving			
		M	SD	50th Percentile	95th Percentile	M	SD	50th Percentile	95th Percentile
2x2	Small	1.42	0.31	1.40	1.91	2.00	0.54	1.90	2.94
	Medium	1.35	0.29	1.37	1.77	1.82	0.45	1.74	2.67
	Large	1.27	0.26	1.24	1.71	1.80	0.52	1.74	2.81
2x4	Small	1.75	0.51	1.72	2.60	2.79	0.98	2.55	4.74
	Medium	1.72	0.45	1.66	2.57	2.39	0.94	2.15	3.78
	Large	1.65	0.41	1.62	2.31	2.57	0.94	2.35	4.59
3x5	Small	2.49	0.98	2.24	4.30	3.73	1.68	3.10	6.76
	Medium	2.23	0.73	2.14	3.45	3.36	1.43	3.07	5.38
	Large	2.53	1.19	2.17	4.91	3.61	1.86	3.14	7.47

Effects of the number of buttons. For small, medium, or large buttons, the task completion time was significantly longer for the 3x5 layout than the 2x4 layout, and was significantly longer for the 2x4 layout than the 2x2 layout (all $p < 0.001$).

Effects of button sizes. (1). With the 2x2 layout, there was a significant difference in task completion time depending on the size of the buttons, $\chi^2(2) = 21.511$, $p < 0.001$. Post hoc analysis with Wilcoxon signed-rank tests was conducted with a Bonferroni correction applied, resulting in a significance level set at $p < 0.017$ (= 0.05/3 comparisons). The task completion time is significantly longer for the small buttons than both the medium buttons (Z = -3.507, $p < 0.001$) and large buttons (Z = -4.282, $p < 0.0005$). There was no significant differences of task completion time between the medium and large buttons (Z = -0.569, $p = 0.569$). (2). With the 2x4 layout, there was a significant difference in task completion time depending on the size of the buttons, $\chi^2(2) = 9.849$, $p = 0.007$. Same post hoc analysis shows that the task completion time was significantly longer for the small buttons than the medium buttons (Z = -3.651, $p < 0.001$).

There was no significant differences in task completion time between the small and large buttons ($Z = -1.827$, $p = 0.068$), or between the medium and large buttons ($Z = -1.730$, $p = 0.084$). (3). With the 3x5 layout, there was no significant difference in task completion time depending on the size of the buttons, $\chi^2(2) = 4.821$, $p = 0.09$.

3.3.2 Total Eyes-Off-Road Time

The total eyes-off-road time (TEORT) is the cumulative time when the driver is looking away from the road when performing the non-driving task.

Figure 16 shows the distribution of the task completion time under the 9 test conditions. Shapiro-Wilk tests were conducted to test the normality of the distributions. The results show that the data violate the normality in all conditions ($p < 0.05$) with two exceptions of 2x2 small ($W = 0.986$, $p = 0.523$) and 2x2 medium ($W = 0.975$, $p = 0.121$).

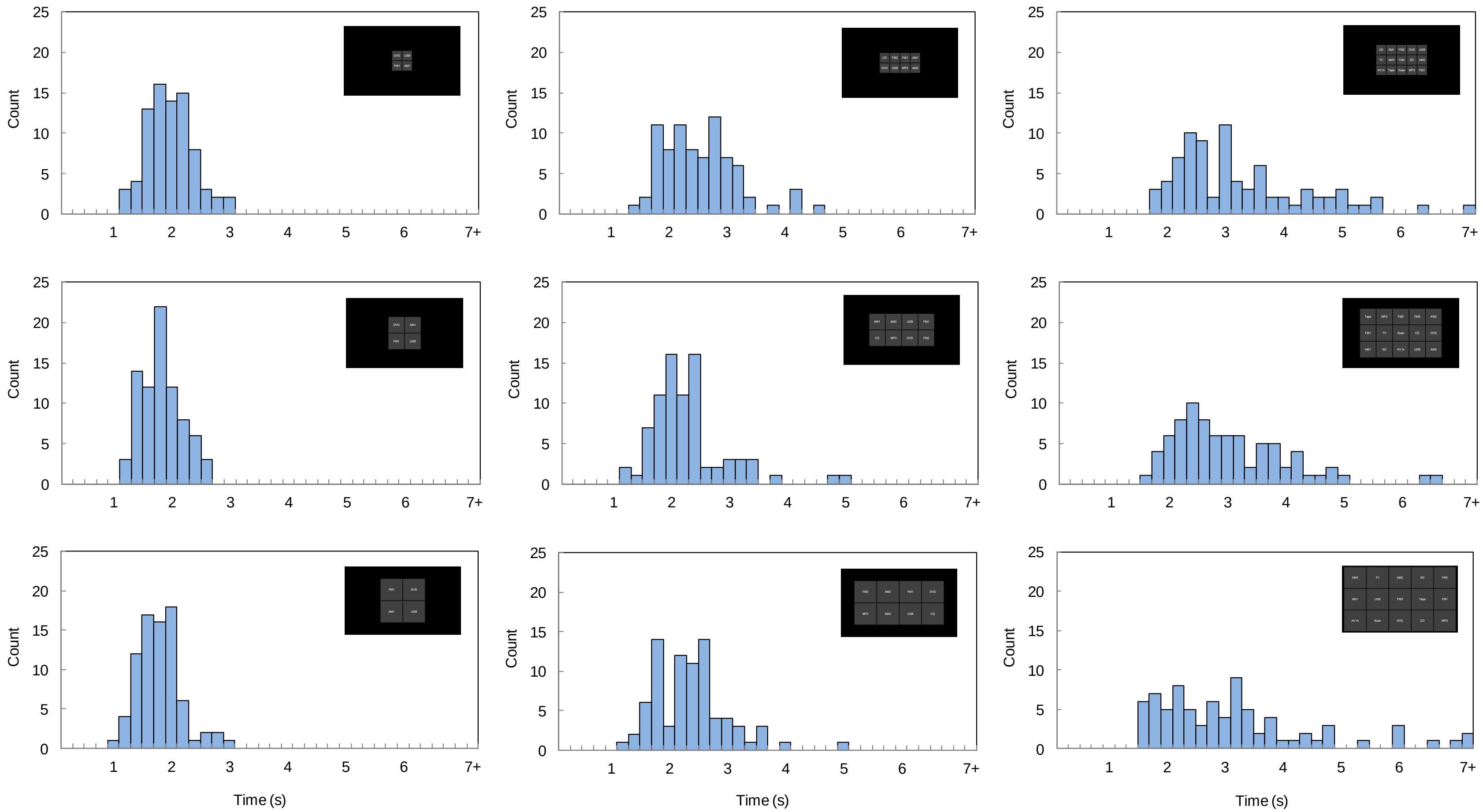


Figure 16. Distribution of TEORT by button number and size

Table 2 shows the statistics of the TEORT for the 9 task conditions. Since the normality was violated in some of the conditions, the nonparametric Friedman Test was used to examine whether there are significant differences in the TEORT among different levels of the independent variables.

Table 2. Statistics of TEORT in experiment 1 (all units are in seconds)

Number of Buttons	Button Size	M	SD	50th Percentile	95th Percentile
2x2	Small	1.87	0.39	1.84	2.54
	Medium	1.71	0.33	1.68	2.24
	Large	1.70	0.39	1.67	2.50
2x4	Small	2.43	0.65	2.37	3.75
	Medium	2.16	0.68	2.04	3.24
	Large	2.28	0.65	2.20	3.47
3x5	Small	3.20	1.26	2.89	5.34
	Medium	2.92	0.99	2.72	4.64
	Large	3.11	1.48	2.80	5.97

Effects of button sizes. (1). With the 2x2 layout, there was a significant difference in TEORT depending on the size of the buttons, $\chi^2(2) = 28.682$, $p < 0.0005$. Post hoc analysis with Wilcoxon signed-rank tests show that the TEORT was significantly longer for small buttons than both medium buttons ($Z = -4.008$, $p < 0.0005$) and large buttons ($Z = -4.678$, $p < 0.0005$). There was no significant differences in TEORT between the medium and large buttons ($Z = -0.452$, $p = 0.651$). (2). With the 2x4 layout, there was a significant difference in TEORT depending on the size of the buttons, $\chi^2(2) = 8.884$, $p = 0.012$. Same post hoc analysis shows that the TEORT was significantly longer for small buttons than medium buttons ($Z = -3.016$, $p = 0.003$), while there was no significant differences in TEORT between the small and large buttons ($Z = -1.446$, $p = 0.148$), or between the medium and large buttons ($Z = -1.672$, $p = 0.095$). (3). With the 3x5 layout, there was no significant difference in task completion time depending on the size of the buttons, $\chi^2(2) = 5.003$, $p = 0.082$.

Effects of the number of buttons. For small, medium, or large buttons, the TEORT was significantly longer for the 3x5 layout than the 2x4 layout, and was significantly longer for the 2x4 layout than the 2x2 layout (all $p < 0.001$).

3.3.3 Number of eyes-off-road glances

The total number of eyes-off-road glances that were used by the participants in each task trial was extracted from the data. In a total of 720 task trials, task trials with a single glance account for 56% (402), task trials with 2 glances account for 33% (236), and task trials with more than 2 glances account for 11% (85). Figure 17 shows the percentage of the number of glances (1 glance, 2 glances, or more than 2 glances) in each test condition. For the 2x2 designs, about 70-80% of the tasks were completed with a single glance, and the rest of the tasks were completed with 2 glances. For the 2x4 designs, about 50-60% of the tasks were completed with a single glance, and about 30-40% were completed with 2 glances. For the 3x5 designs, about 30-40% of the tasks were completed with a single glance, about 30-50% were completed with 2 glances, and about 20-30% were completed with more than 2 glances.

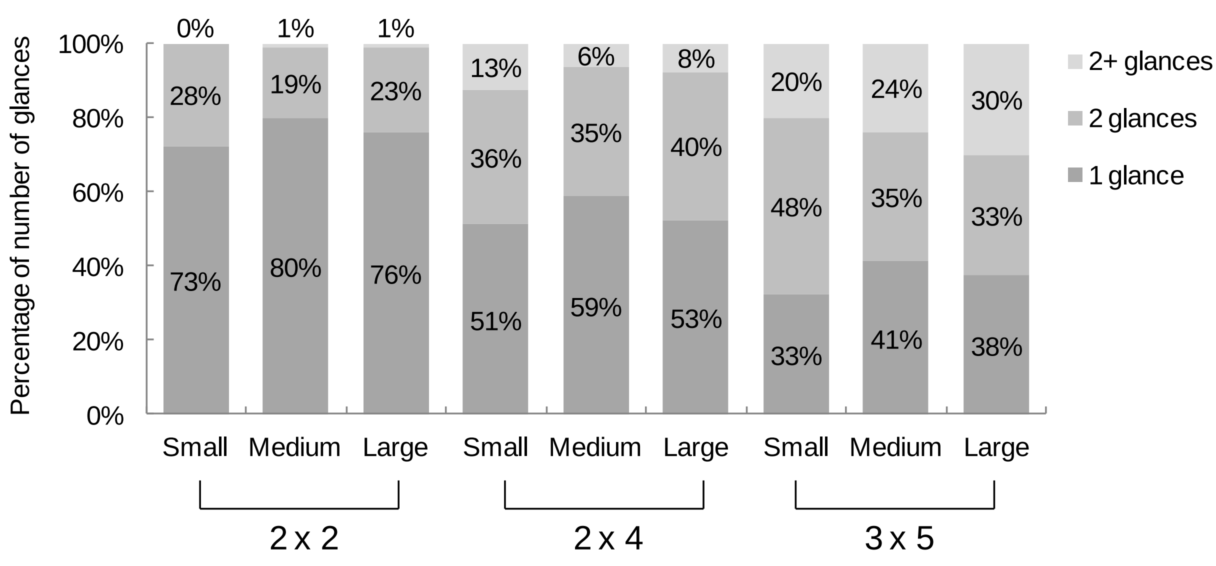


Figure 17. Percentage of number of eyes-off-road glances by button sizes and number of buttons

3.3.4 Number of long (eyes-off-road) glances

Figure 18 shows the distributions of the duration of the individual glances in each of the 9 test conditions. A single glance is considered as “long” if its duration is longer than 2.0 seconds.

The result shows that in the 720 task trials (= 9 test conditions x 4 replications x 20 participants), there are a total of 223 glances that lasted longer than 2.0 seconds. Figure 18 shows the distributions of the individual glances in the nine test conditions. The bins that represent long glances are in red color. It could be observed that with the increased number of buttons, there is an increased number of long glances. Table 3 shows the statistics of the number of long glances in the 9 task conditions.

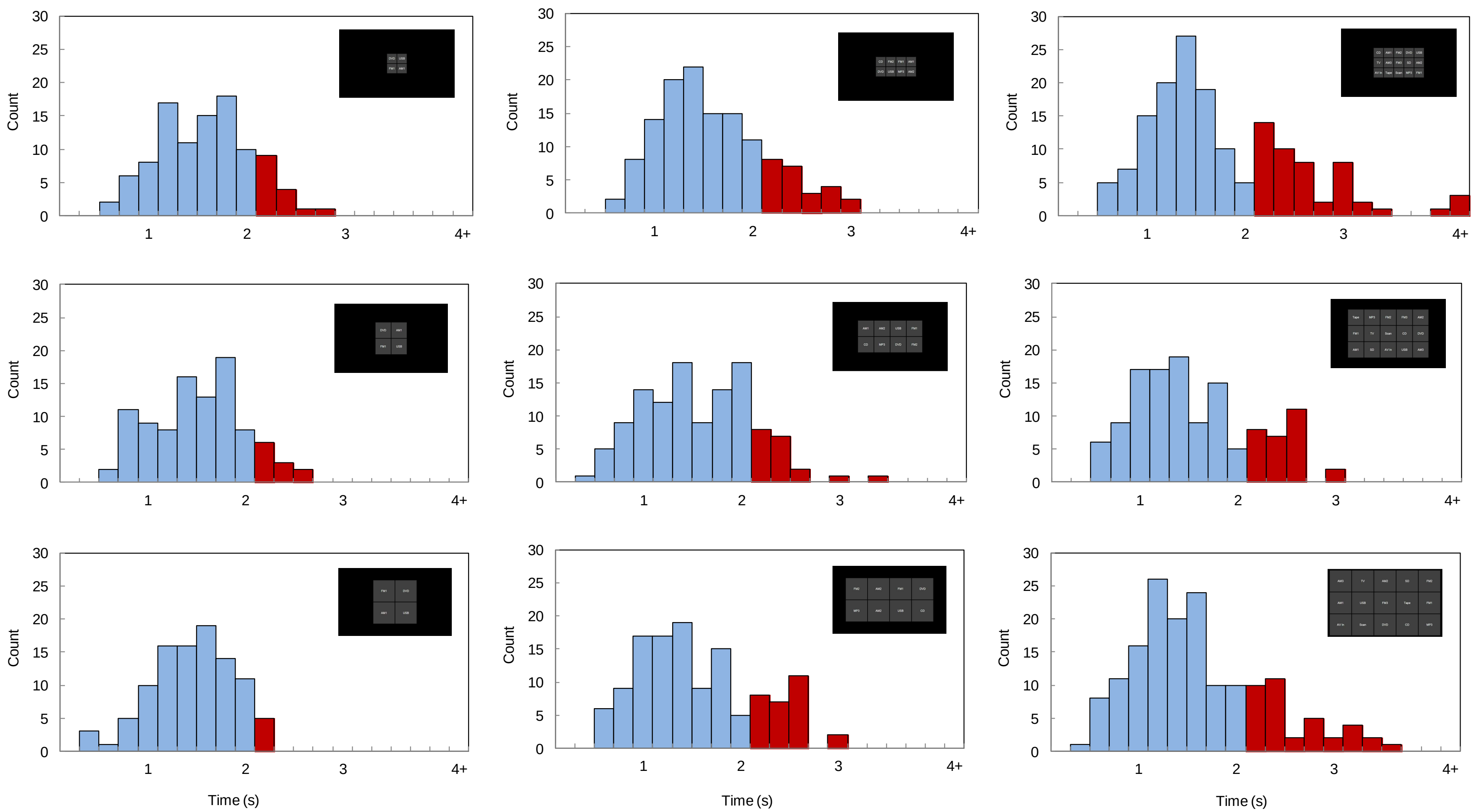


Figure 18. Distribution of individual glance duration by button number and size. The bins that represent glances longer than 2 seconds are in red color.

Table 3. Statistics of number of long glances in experiment 1

Number of Buttons	Button Size	M	SD	50th Percentile	95th Percentile
2x2	Small	0.19	0.39	0.00	1.00
	Medium	0.14	0.35	0.00	1.00
	Large	0.06	0.24	0.00	1.00
2x4	Small	0.30	0.46	0.00	1.00
	Medium	0.24	0.43	0.00	1.00
	Large	0.35	0.48	0.00	1.00
3x5	Small	0.59	0.59	1.00	1.05
	Medium	0.46	0.53	0.00	1.00
	Large	0.46	0.64	0.00	1.05

3.3.5 Relationship between the number of glances and glace duration

In this section we aimed to examine whether the occurrence of the long glances is related to the number of glances used in a task trial. Figure 19 shows the distributions of the glance duration for the task trials (1) when a single glance was used (top figure), and (2) when more than one glance were used (bottom figure). Shapiro-Wilk tests were conducted to test the normality of the two distributions. The result shows that the data violate the normality in both the task trials with a single glance (W = 0.943, $p < 0.001$) and the task trials with more than one glance (W = 0.919, $p < 0.001$). Given the violation of the normality and the unequal sample size, a Kruskal-Wallis test was conducted to examine whether there is a significant difference in glance duration between the task trials with a single glance and task trials with more than one glance. The result shows that the glance duration in the task trials with a single glance is significantly longer than the glance duration in the task trials with more than one glance ($\chi2(1) = 413.058$, $p < 0.001$).

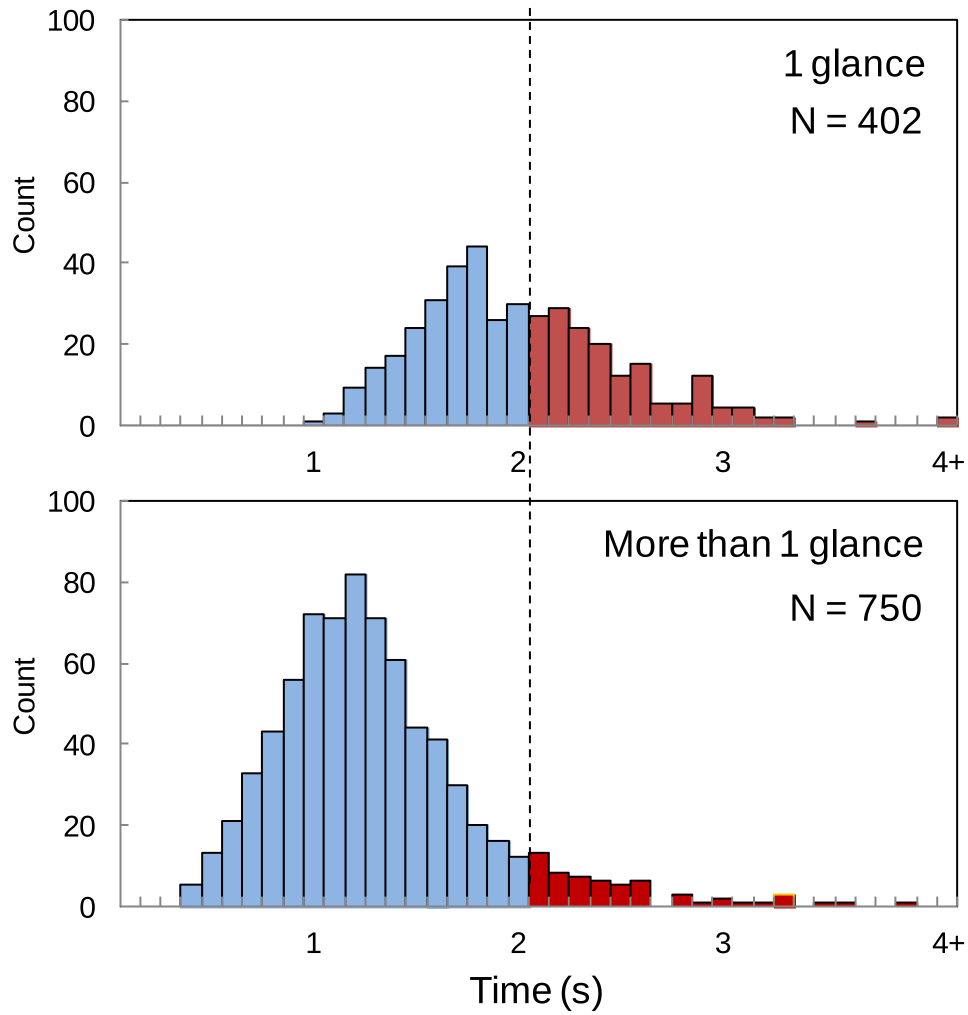


Figure 19. Distributions of glance duration when single glance was used (top figure) and when more than one glances were used (bottom figure). The dash line is the 2-second threshold for long glances (shown in red color).

As shown in section 3.3.3, a single glance was used in 56% (402) of a total of 720 task trials. Figure 19 shows the long glances from a single glance (the red bins on the top figure) account for 74% (164) of a total of 223 long glances. On the other hand, task trials with more than one glances account for 44% (318) of the total of 720 task trials, but the long glances from these task trials (the red bins on the bottom figure) account for 59 (26%) of the long glances. This seems to suggest that long glances are overrepresented in the task trials with a single glance.

Figure 20 shows the distributions of glance duration for the task trials with a single glance in the nine test conditions. It could be observed that the proportion of long glances increases with the increased number of buttons.

Figure 21 shows the distributions of glance duration for the task trials with more than one glance in the nine test conditions. It could be observed that very few long glances occurred with both the 2x2 designs and 2x4 designs. More long glances occurred with the 3x5 test conditions.

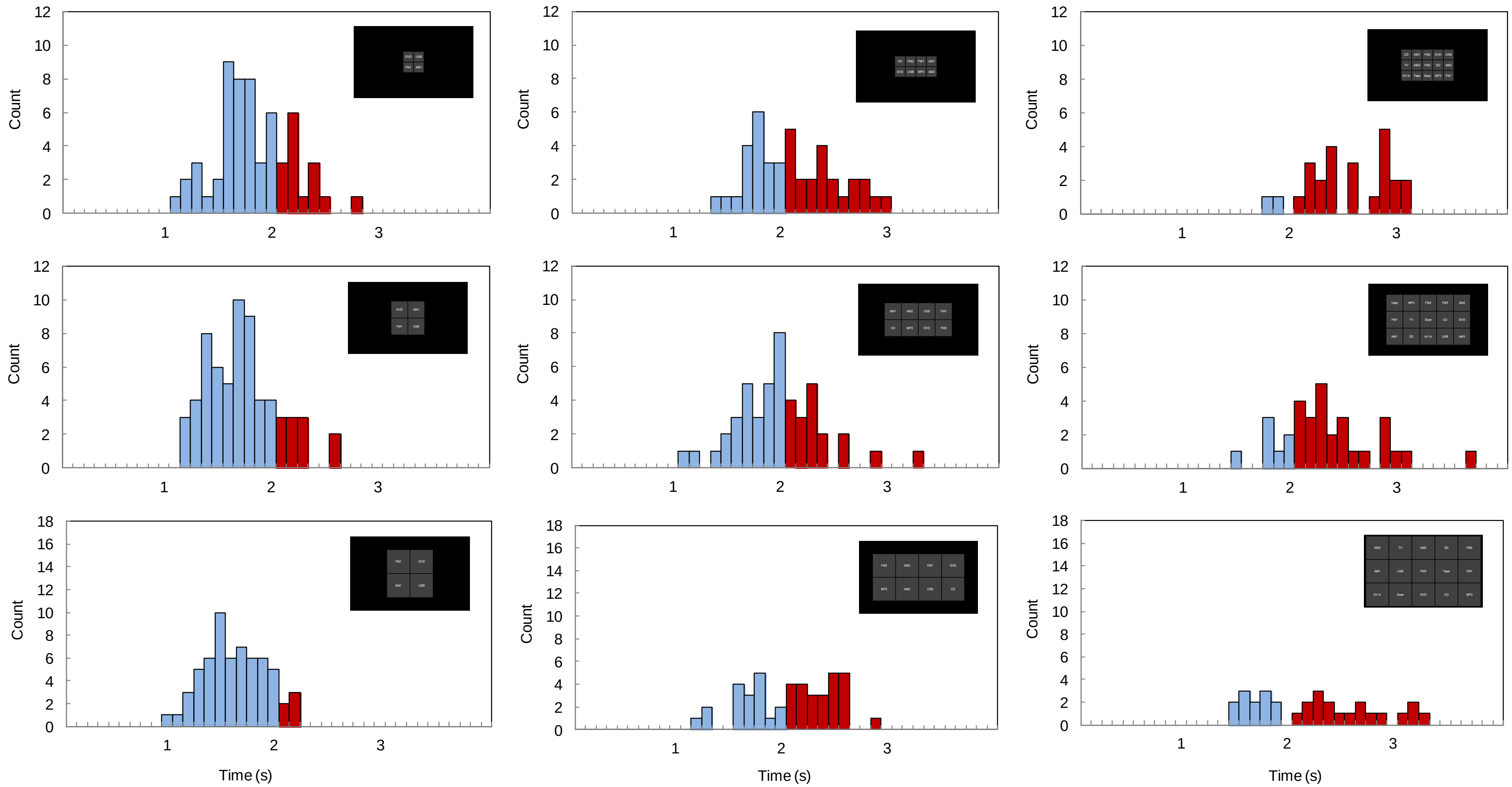


Figure 20. Distribution of glance duration for task trials with a single glance. The bins that represent glances longer than 2 seconds are in red

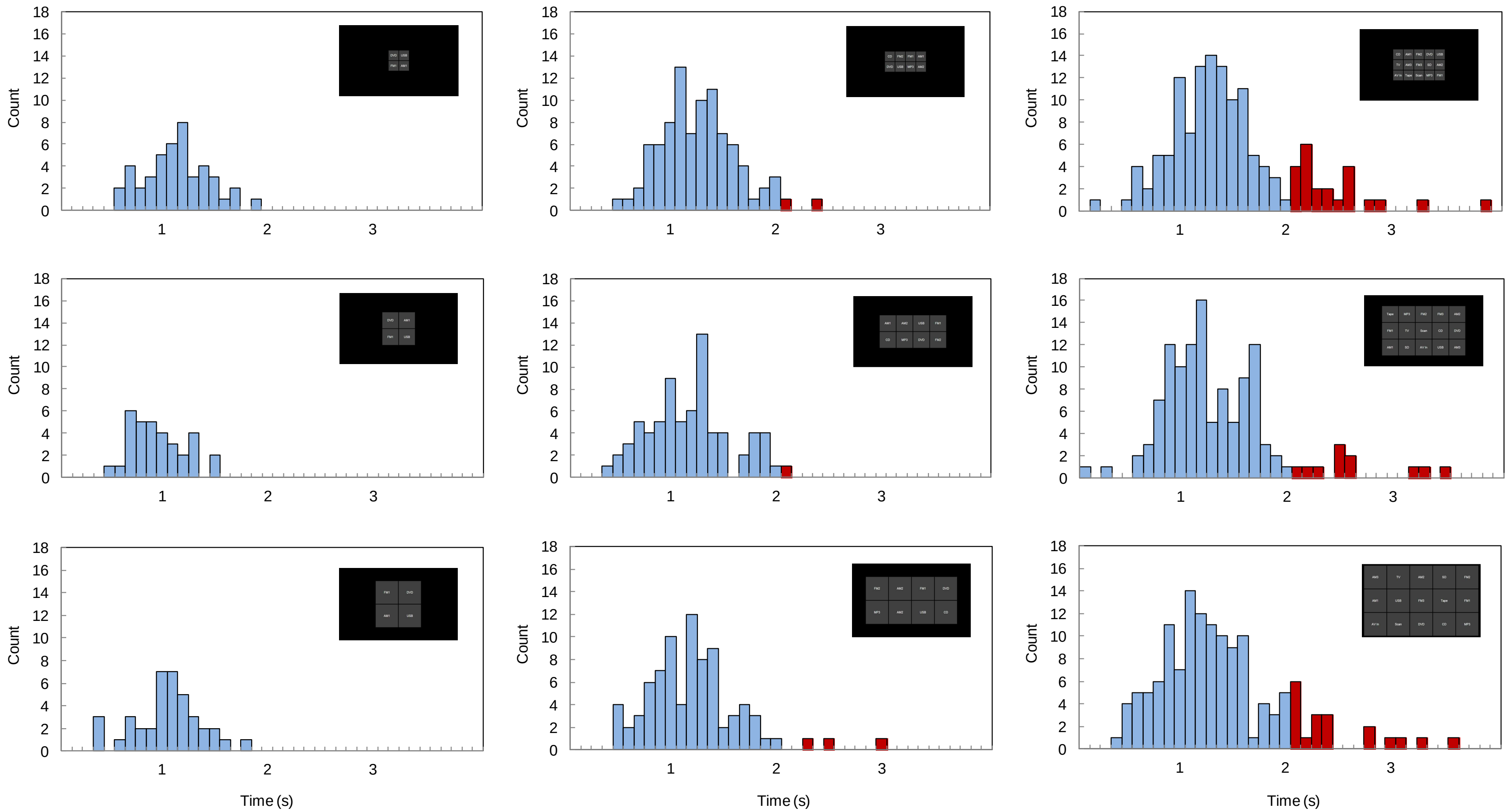


Figure 21. Distribution of glance duration for task trials with <u>more than one glance</u>. The bins that represent glances longer than 2 seconds are in red

These results shows that for the 2x2 and 2x4 test conditions, almost all the long glances came from the task trials when a single glance was used. For the 3x5 design, long glances occurred in task trials with both a single glance and multiple glances. But when only a single glance was used, more than half of the glances were longer than the 2-second threshold.

3.4 **Conclusions**

Most modern user interfaces integrate many functions and features into one digital screen. The number of buttons in one screen and their sizes are two basic design parameters that must be dealt with by UI designers. The users of such systems often need to conduct visual search for specific buttons on a given screen. This is especially true at the learning stage of a new system or after major software UI updates. Searching for a particular button may induce safety hazards when it is conducted concurrently with the driving task, which primarily relies on visual perception for successful performance.

An experiment was designed in which participants were asked to drive a driving simulator, and at certain points of the driving, they were asked to look for a particular button among a series of distracting buttons and then press on it. From the experiment we found that regardless of the button sizes, the task completion time and driver's total eyes-off-road time (TEORT) increased significantly with increased number of buttons on the screen. With smaller number of buttons (i.e., 4 or 8), small button size seems to be associated with longer task completion time and TEORT compared with medium or large buttons.

The results also show that while the majority of the tasks with the 4 buttons were completed with a single glance off the road, as the number of buttons on the screen increase, it becomes more likely for the driver to use multiple glances in order to complete the task. In addition, the number of long glances (defined by duration longer than 2 seconds) also increased with the increased number of buttons. A further analysis of the relationship between the number of glances and glance duration revealed that long glances may be overrepresented in the task trials with a single glance. With a smaller number of buttons (4 or 8), almost all long glances occurred when the participant completed the task trial using a single glance. With the largest number of buttons (15), long glances occurred in both the task trials when a single glance was used and the task trials when multiple glances were used. These findings may suggest that for a simple visual search and button-pressing task, drivers tend to use a single glance to complete the task even though this may induce long glance duration.

It is necessary to note that since the location of the target button is randomly placed with equal probability in each task trial, there's no other information about the possible locations of the target button available to the participant to facilitate the task. From the application standpoint, this task scenario represents a novice user who does not have any prior knowledge about the button locations of a system. But in reality during the usage of a system over time, users may gain some knowledge about the system layout which could be used to narrow down the search area to a smaller region rather than the entire screen.

Chapter 4.
Queuing Network Modeling of Visual Search Task Performance while Driving

4.1 Introduction

Great efforts have been made to model human visual search performance in the past decades. One of the simplest models is the serial search model (Neisser, 1964), which assumes that the items are inspected one at a time, and each item takes a constant time to inspect. Thus the time it takes to find a target is the multiple of the inspection time per item and the number of items over 2. The division by 2 is based on the assumption that the target has equal probability to appear at any step of the serial inspections. This model works well if the search space is organized coherently in structure, for example when finding a number from a yellow book. But it does not work so well when the search space is less organized, for example, finding a street name on a map. One of the earliest and most influential theories on visual search is the Feature Integration Theory (FIT; Treisman & Gelade, 1980). According to this theory, visual search proceeds in several stages, and in its first stage, a set of low-level features (color, motion, orientation, etc) are processed in parallel feature channels of the human visual system. In the second stage, these features are integrated to form a global salience map that can be accessed to direct attention to the most conspicuous location. Another visual search theory is the N-SEEV cognitive model of visual attention recently proposed by Steelman et al., (2009). It incorporates both bottom-up (salience, effort) and top-down (expectancy, value) factors that move attention to selectively attend various sources of information. The major limitation of the N-SEEV model is that it has quite a number of free parameters as a computational model. It would not be easy to quantify these parameters when applying it to specific attention tasks in the applied domain. Guided Search (Wolfe, 1994, 2007) is another important visual search model, which assumes that visual search is guided by a combination of influence from the bottom-up and top-down factors (termed ‘activation’). The human perceiver would first look at the location with the highest activation, then look at the location with the second highest activation, etc.

Besides the visual search models discussed above, efforts have also been made to incorporate visual search mechanisms to the comprehensive cognitive architectures. The work includes the reinforcement learning based model of eye movements accounting for both the top-down and the bottom-up processes (Lim & Liu, 2009), and the ACT-R-based Eye Movements and Movement of Attention (EMMA; Salvucci, 2001) model and Pre-attention and Attentive Vision (PAAV; Nyamsuren & Taatgen, 2013) model.

The following section describes the modeling work using QN-MHP to simulate a visual search task in both single and dual task situations (i.e., visual search alone or with a concurrent steering task).

4.2 **Methods**

4.2.1 Task Analysis

To model any task using the QN-MHP architecture, a task analysis is required in the format of Natural Goals, Operators, Methods, and Selection rules Language (NGOMSL; Kieras, 1999). The NGOMSL task analysis breaks down the task in a “top-down, breadth-first” manner into “atomic-level” Task Components (TC). Each TC is associated with a task-independent context-free QN-MHP operator from the QN-MHP operator library. The operators have parameters which could be set either explicitly by the modeler when specifying the task before the simulation, or implicitly by the model during the simulation. The sequential dependency of the task components are set by the modeler before the simulation. This task specification is loaded into the QN-MHP model and the task is performed as the model runs in response to its associated environment stimuli.

4.2.2 Task Components

A NGOMSL task analysis for the visual search task was conducted. The result is shown in Figure 22 with 9 TCs.

GOAL: Find and press the USB button on the display

Method to accomplish goal of finding and pressing the USB button on the display

TC 1: Look at <feature> on <device> at location <x, y>

TC 2: Store <feature value> to short-term memory

TC 3: Retrieve <feature value> from short-term memory

TC 4: Compare <feature value> to <target value (i.e., "USB")>

If match, return *result* = 1, else return *result* = 0

TC 5: If *result* = 1, go to TC 7, else go to TC 6

TC 6: Visual Search with <search pattern>

TC 7: Reach <location> with <which hand> <with or without visual guidance>

TC 8: Click with <which hand> <which finger>

TC 9: Return with goal accomplished

*TC: Task Component

Figure 22. NGOMSL-style task analysis of the visual search task

Below is a description of each of the task components:

(1) TC 1 is associated with the perceptual operator LOOK_AT. This operator directs the model to "look at" a specific location in the environment for the specific visual feature (e.g., color, orientation, text). In the visual search task, the visual feature is set to "text", as to accomplish the goal of finding the "USB" button, the model needs to extract the labels of the buttons and check whether is the target button or not. The location to look at is generated as the output of TC 6 (visual search).

(2) TC 2 is associated with the memory operator STORE_TO_STM (Short Term Memory). This operator stores information entities to the working memory servers (i.e., Visualspatial Sketchpad or Phonological Loop) depending on the type of information they carry (i.e., visual-spatial or verbal). In this case, the text information is stored at the Phonological Loop.

(3) TC 3 is associated with the memory operator RETRIEVE_FROM_STM. This operator retrieves information from the short-term memory, and assigns the retrieved value to a global variable that is accessible by other operators.

(4) TC 4 is associated with the cognitive operator COMPARE. It compares two values and returns the result of 1 if a match is found and 0 if a mismatch is found. In this case, it compares a text perceived from a button with the target text "USB".

(5) TC 5 is associated with the procedural flow operator GOTO. It specifies the procedural sequence of the steps in a task list. Specifically, it determines which step or steps in the task list should be activated based on the output of the cognitive entity upon its completion of processing. In this case, if a match is found (i.e., a button labeled "USB" is perceived and recognized), it activates TC 7 to start the reaching action. Otherwise, it activates TC 6 to decide whether to continue checking at the location of the current visual attention, or move the visual attention to a new location to check. Note here for simplicity we assume the reach is initiated after the identification of the target label. It is observed during the experiment that sometimes the participants started the reach before the identification of the target and adjusted the final direction towards the target after the identification. This strategy is not currently modeled.

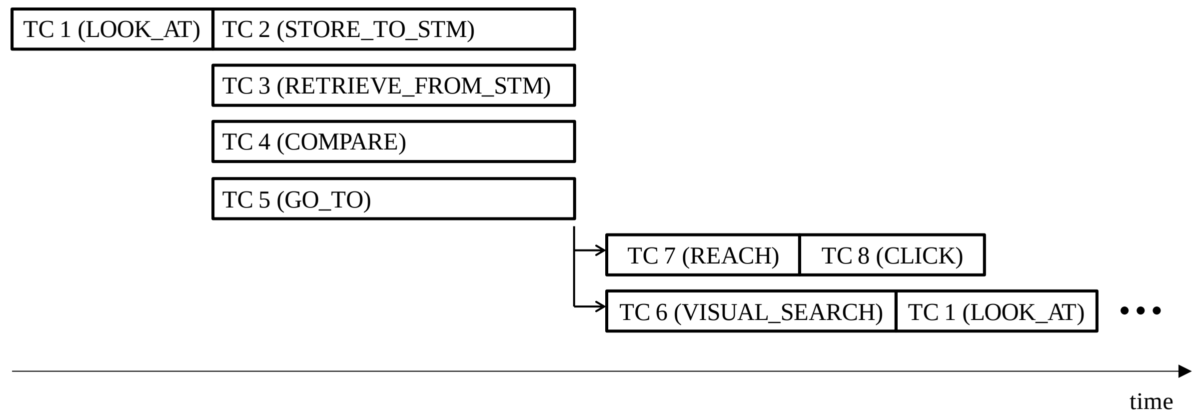
(6) TC 6 is associated with the cognitive operator VISUAL_SEARCH. This operator first checks if a visual search is needed based on the search progress (i.e., how many objects under the current visual attention have already been checked). If this number is smaller than the total number of items under the current visual attention, the processing stops. Otherwise it indicates that all the items in the current visual attention have been examined, and no target has been found. In this case it determines the new visual attention location by randomly selects (with equal probability) a new item that has not been previously examined, and activates the LOOK_AT operator to move the visual attention to the new location.

(7) TC 7 is associated with the motor operator REACH_WITH_HAND. This operator initiates a reaching action using the model's hand servers. The modeler specifies which hand (left or right) for the reaching, and whether the reach is with or without visual guidance. The target location for the reaching is implicitly specified by the cognitive entity that activates this operator. In this visual search task, it is assumed that this task component is activated when the target button is found in TC 4. And the reaching is executed with the right hand and with visual guidance (i.e., no touch-typing). In the current version of QN-MHP, if the reaching is with visual guidance, the time it takes to reach the target is determined based on Fitts's Law and its extensions (MacKenzie and Buxton, 1992, Bi et al., 2013).

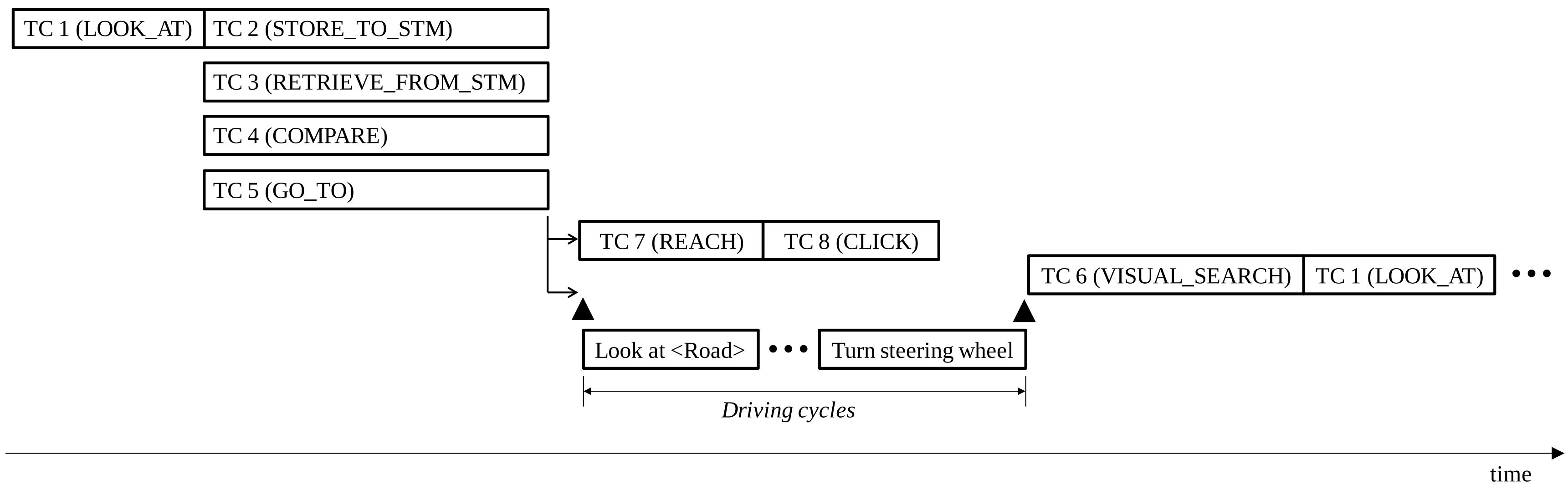
(8) TC 8 is associated with the motor operator CLICK_WITH_FINGER. This operator initiates a clicking action using the model's hand servers. The modeler specifies which hand and which finger is to be used for the clicking. This task component is activated when TC 7 is completed.

(9) TC 9 is associated with the procedural flow operator GOAL_ACCOMPLISHED. It indicates the completion of a task. Once this operator is activated, it reports the completion of the task, and checks if there are any other pending tasks, and switches to that task if there is any. If there is no other pending task, it terminates the simulation. In this visual search task, the task goal is accomplished when the target button is pressed in TC 8.

Figure 23 demonstrates the multitasking scheme of the visual search and the driving task. Figure 23 (a) shows the sequential dependency of the task components of the visual search task in a single task condition. Figure 23 (b) shown the task components of the visual search task are segmented by the driving task in a dual task condition, in which the task components representing the driving cycle is inserted into the task components of the visual search task. During the visual search task each time when the target object is not found and a new visual search is needed, instead of immediately activating the visual search task component (TC 6), the model checks an internal clock of the QN-MHP which record how much time the human has already switched away from the primary task of driving, and then decides whether it should switch to the driving task by activating the task components of the driving task or continue with the visual search task.



(a) Sequential dependency of the task components (TCs) for the search task in a single task condition

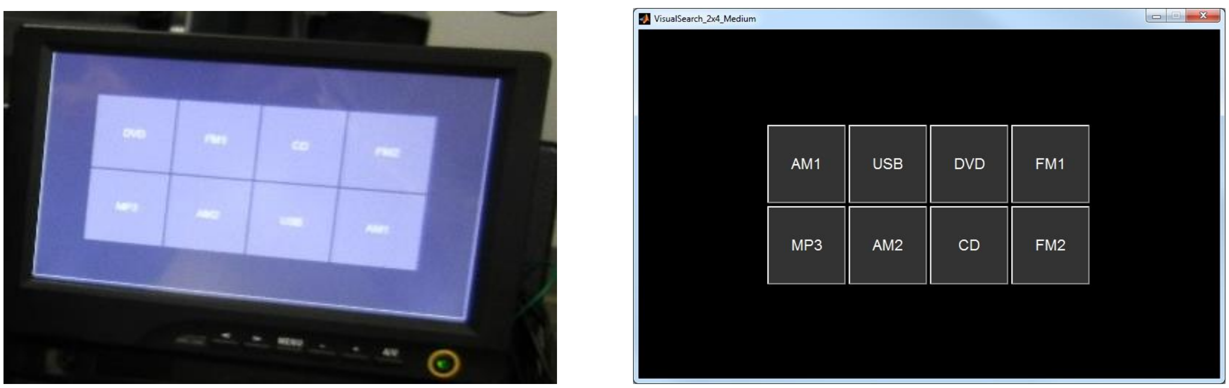


(b) Insert driving task in front of the visual search TC

Figure 23. Multitasking of the search task and driving

4.2.3 Device Mockups

Aside from the cognitive model, which represents the human, digital mockups of the devices are needed to simulate the interaction between the human and the environment. The digital mockups provide stimuli as inputs to the cognitive model, and could be acted upon by the actions generated by the cognitive model (e.g., pressing a button). Digital mockups for this visual search task were created using MATLAB GUIDE (Graphical User Interface Design Environment). Figure 24 shows both the physical device used in the human experiment and the digital mockups created for the simulation.



(a) Real Device used by the human participants (b) Digital mockup used by the model

Figure 24. A comparison of the real device used in the human experiment and the digital mockup

4.2.4 Specifications of the Task Environment and Initial Conditions

The task environment specifications include (1) the physical location of the touch screen device relative to the steering wheel, (2) the visual angle between the road scene and the device, and (3) the viewing distance of the device from the driver. These specifications were set as the same value as those measured from the experiment with human subjects. For the single task condition, it is assumed that the driver starts the task with both hands on the steering wheel, and eyes looking at the center of the touch screen. For the dual task, it is assumed that the driver starts the driving task first with both hands on the steering wheel, and eyes looking at the front road.

4.3 **Results**

The human results from Chapter 3 show that with increased number of buttons on the screen, the task completion time increases in both parked and driving conditions, the driver's Total Eyes Off Road Time (TEORT) increases, the drivers needed more glances to complete the task, and the number of long glances (defined using a threshold of 2.0 seconds) also increases. The button sizes seem to affect the task completion time and TEORT only when there was a small amount of buttons (2x2 or 2x4).

Simulations were conducted for the search task with ten replications. As shown in Figure 26, the simulation result is able to capture the increased task completion time with the increased number of buttons. The mean absolute percentage error (MAPE) is 9.9%. The root-mean-square error (RMSE) is 0.34 seconds. As shown in Figure 27, the simulation result is able to capture the increased TEORT with the increased number of buttons. The MAPE is 8.8%. The RMSE is 0.29 seconds.

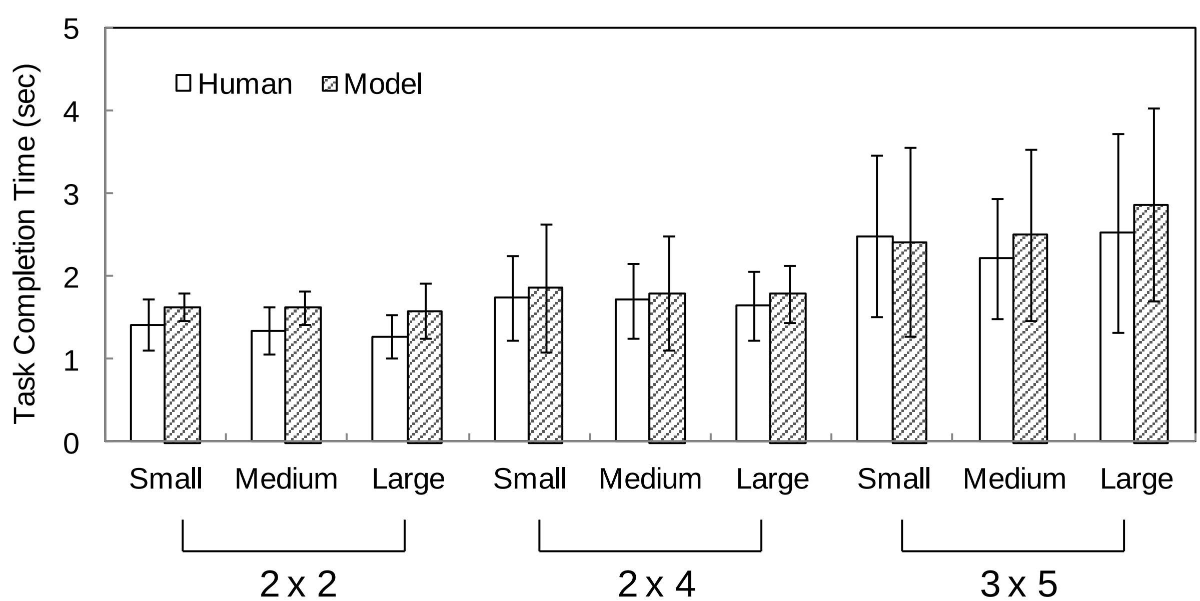


Figure 25. Effects of button number and size on task completion time in the parked condition for both human and model results. Error bars represent standard deviation

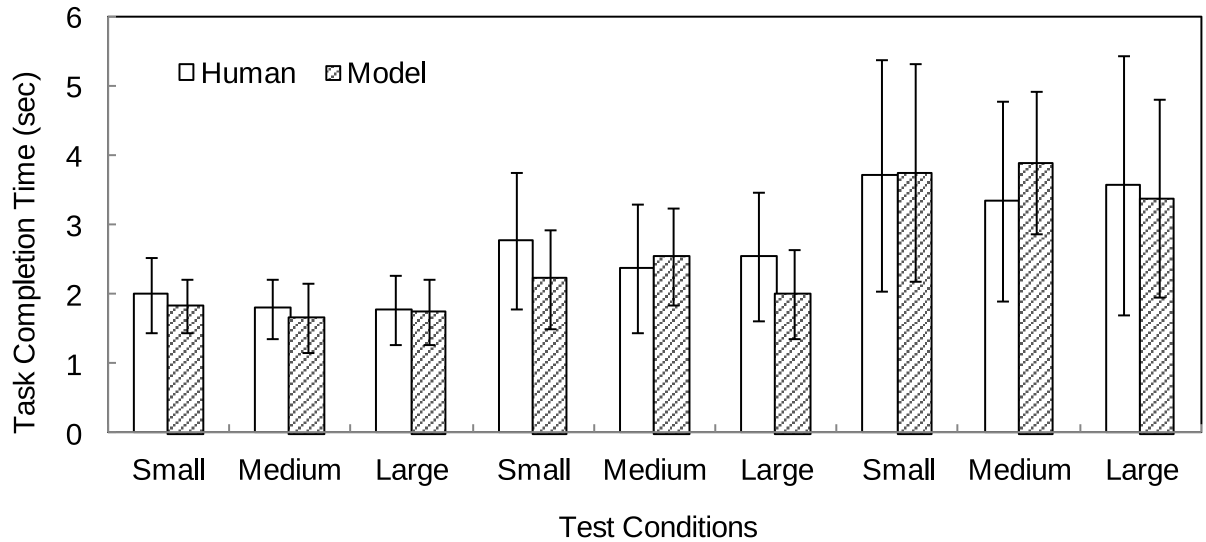


Figure 26. Effects of button number and size on task completion time in the driving condition for both human and model results. Error bars represent standard deviation

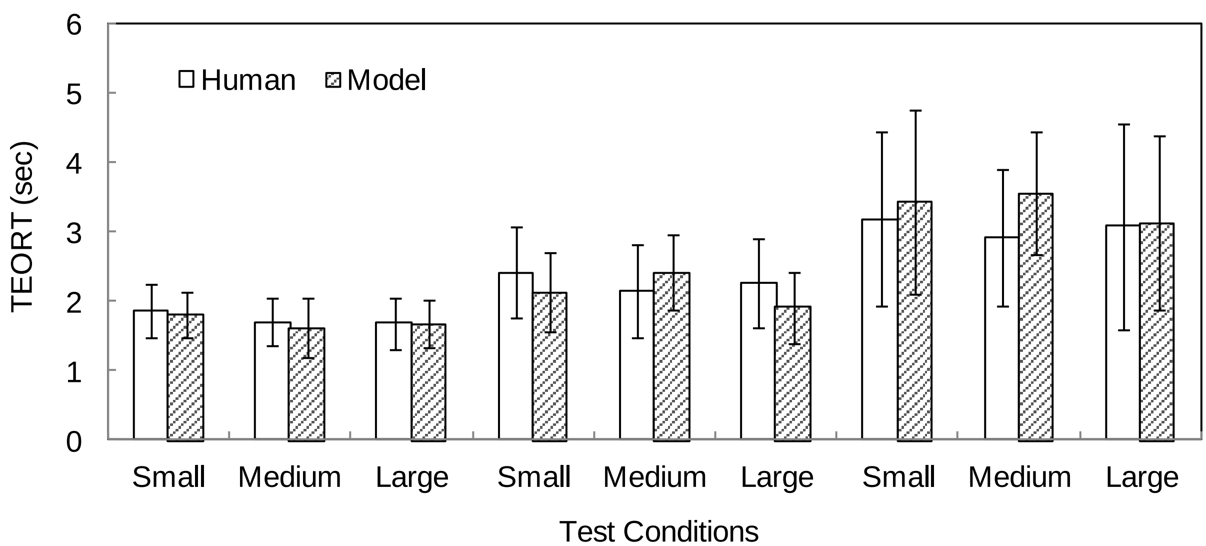


Figure 27. Effects of button number and size on TEORT for both human and model results. Error bars represent standard deviation

As shown in Figure 28 and Figure 29, the model is also able to produce similar individual glance duration data that could account for the differences of the number of glances and long glances in the task conditions.

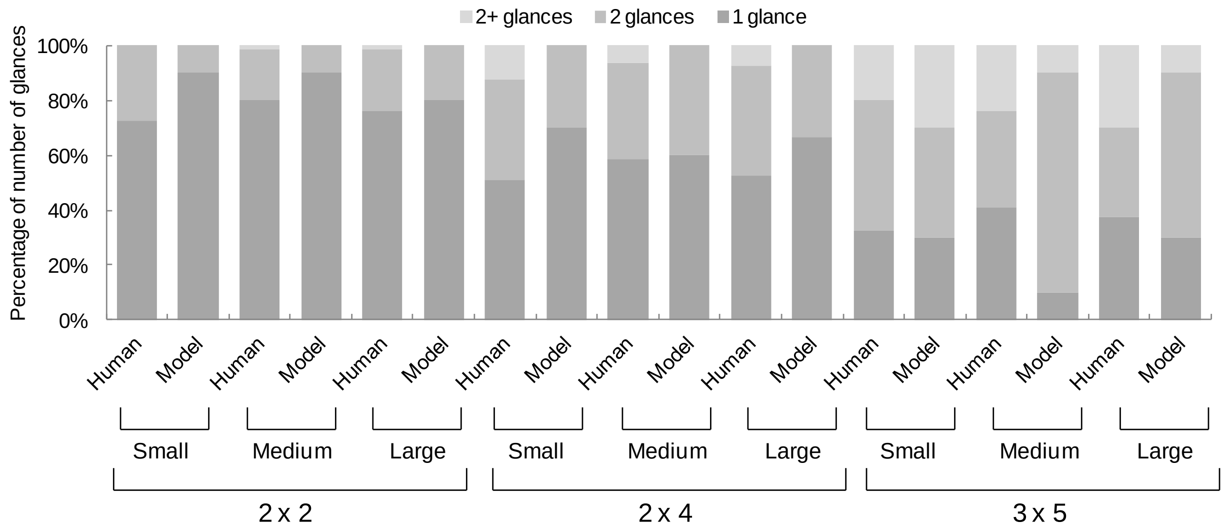


Figure 28. Effects of button number and size on the number of glances for both human and model results

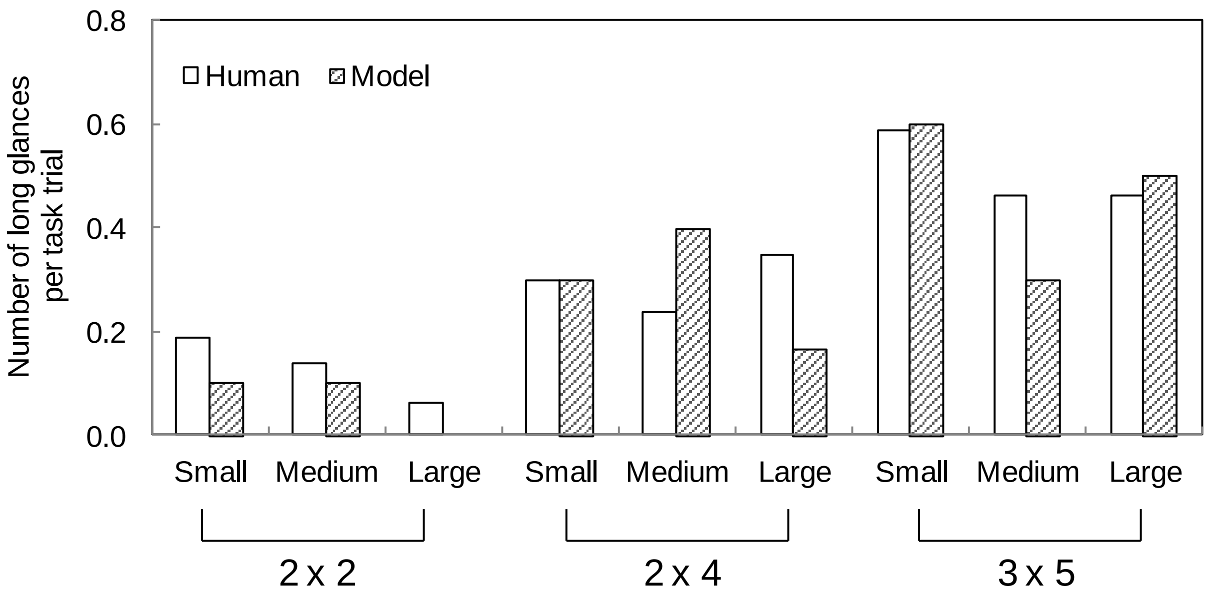


Figure 29. Effects of button number and size on number of long glances per task trial for both human and model results

4.4 **Discussions**

As the result shows, the model is able to generate similar task completion time as the empirical data from the human subjects for both the single and dual task conditions. As the number of button increases, the model prediction of the task completion time also increases. This could be accounted for by two mechanisms of the model. The first mechanism is the serial nature of eye saccade when searching for the target button among distracting buttons. The model could only move the visual attention to one location at a time. With more buttons spatially distributed on the display, it would take longer time on average to find the target button. The second mechanism is the limitation on the number of objects that could be recognized simultaneously. It is commonly believed that there is a bottle neck in the human visual perception that limits the number of objects that could be recognized at the same time. Some studies have identified the limit to be 4 (Atkinson, 1976). In the current QN-MHP model the capacity of the visual object recognition server is set as 4. Thus with more buttons on the display, it may take longer for the server to process the visual stimuli entities which represent the buttons.

Another observation from the visual search task is that for the same number of buttons, the button size does not seem to affect the task completion time. There are two factors that may play a role. Firstly, since the same width of spacing between buttons were used for all three different button sizes, and the labels were always placed in the center of the button, the labels in the larger button sizes are more spatially apart from each other. Thus for larger button sizes, it takes longer time to execute the eye saccade from one button to another. Secondly, according to Fitts's law which was implemented in the REACH-WITH-HAND operator, for the same reaching distance, it takes shorter time to reach to a larger button than a smaller button. So the affects of the two factors above may cancel each other, which lead to the non-significance of the button sizes on the task completion time.

The longer task completion time for the dual task condition for most designs except the 2x2 small and medium size buttons could be explained. With very few buttons (4 buttons in this case), most drivers were able to complete the task with a single shot of eyes-off-road. Thus there is no difference of task completion time from the single task condition. While when there are more buttons on the device, it would take longer time to complete the task than the driver is willing to keep the eyes off the road. Thus the visual search task is segmented and the driver is shifting the visual attention back on the road to check the road, and if the driving safety is confirmed, he/she shift the visual attention back on the road to continue the visual search task. By implementing this mechanism using an internal clock at the performance monitor server, the model is able to imitate this behavior in the dual task condition, and generate similar results as from the empirical data.

There are several limitations to this study. First, since no eye tracking devices were used in the experiment, we were not able to look into the details of the driver's eye movement behavior during the visual search task. An equal-probability random special search over the entire device was used in the visual search algorithm. While in reality the driver may use other search patterns. For example, since the device is mounted on the center console which is on the driver's right side, drivers may start search from the left size on the device (near side) to the right size on the device (far side). This other visual search patterns were yet to be examined. Secondly, for simplicity the model assumes a perfect visual search memory, so that a button that was examined before would never been reselected in the search process even if the search progress is segmented by the driving task. While studies have found evidences that the visual search memory does exist (Kristjansson, 2000) but it is not perfect either (Horowitz and Wolfe, 2005). Some researchers were building models to settle somewhere in between. For example, in Wolfe's Guided Search model (Wolfe, 2007), an arbitrary probability of 0.75 of reselecting a previously examined item was implemented. And this probability does not change over the search progress, meaning a recently examined item has the same probability to be reselected as an item that has been examined a while ago. More research advancement in the visual search memory is needed for a more accurate model.

Chapter 5.

An Experimental Study on Task Performance and Mental Workload in Using Three Typical In-Vehicle Infotainment Systems while Driving

5.1 **Introduction**

Driver distraction has been regarded as one of the leading causes of road accidents. And it is becoming a growing concern in the past decade with the introduction of various electronic technologies into the vehicle (e.g., touch screens, digital instrument clusters, and head-up-displays). These in-vehicle technologies allow the drivers to accomplish many “secondary tasks”, which is defined by the U.S. National Highway Traffic Safety Administration (NHTSA) as “any interaction a driver has with an in-vehicle device that is not directly related to the primary task of the safe operation and control of a vehicle”. Examples of secondary tasks include adjusting cabin temperatures, tuning radios, finding the nearest gas station, etc. Previous studies on naturalistic driving have found that performing secondary task is fairly common while driving. A naturalistic 100-car driving study performed by the Virginia Tech Transportation Institute (VTTI) found that performing secondary task was observed in 54% of the randomly selected baseline time segments from their data.

Although such in-vehicle technologies are designed to enhance the driving experience, they may have a potential negative impact on the driving safety by inducing driver distraction and increased workload while driving. According to NHTSA, in 2010 an estimate of 899,000 or 17% of police-reported crashes involves a distracted driver, causing 3,092 fatalities or 9.4% of those killed and over 400,000 injuries. According to the police reports, in 26,000 of those 899,000 crashes the driver was using an integrated control device. Furthermore, these numbers are likely underreported given the difficulty in identifying the use of these control devices during the accident investigation. Ranny et al, (2000) found driver inattention account for approximately

25% of police reported crashes. The naturalistic 100-car driving study estimated that 78% of crashes and 65% of near crashes were related to driver inattention (Klauer et al., 2006)

Given the importance of this issue, many researches had been done to study how secondary tasks are related to driving distraction and safety. Along the lines of the Model Human Processor (MHP), a secondary task could interfere with the driving task during the three processing stages of a task (i.e., perceptual, cognitive, and motor). At the perceptual stage, a visual-manual secondary task may require the driver to move the eyes away from the road in order to perceive the visual information on the devices ("eyes off the road"). At the cognitive stage, it may require the driver to reallocate his/her mental resources (e.g., attention, working memory) from the driving task ("mind off the road"). At the motor stage, it may require the driver to move his/her hands away from the steering wheel to manipulate a device, for example, clicking a button or turning a knob ("hands off the wheel").

Extensive studies have found that driver distraction degrades the driving performance in multiple aspects, including degraded lane keeping performance (e.g., increased standard deviation of lane positions (SDLP), increased lateral acceleration on curves, and more lane deviations), degraded longitudinal control, and longer response time to object and events (Horrey and Wickens, 2004, Reed-Jones, et al. 2008, Peng, et al, 2013). According Wickens's Multiple Resource Theory (MRT), interference to driving performance would occur when the secondary task is competing for the same mental resources (e.g., visual or auditory modalities). As the driving task relies heavily on visual information (Sivak, 1998), secondary tasks that require substantial usage of visual information would more likely to interfere with the driving task.

A variety of metrics have been developed based on the driver's visual behavior to measure the driver distraction, including number of glances (off the road), duration of glances, and total eyes-off-road time (TEORT). Guidelines and standards have been proposed to evaluate various secondary tasks and their associated user interface designs. The Society of Automotive Engineering (SAE) Standard J2364 proposes that the maximum time for drivers to complete navigation-related tasks involving visual displays and manual controls should be less than 15 seconds (referred to as the 15-Second Rule) (Green, 1999). In 2013, NHTSA published a guideline for in-vehicle electronic devices which recommend that tasks should be completed by the driver with glances away from the road of 2 seconds or less and a TEORT of 12 seconds or less (referred to as the 2/12 rule). Many visual-manual tasks are composed of multiple steps that altogether take more than a few seconds to complete even as a single task (i.e., without driving). In order to complete these tasks while driving, a time-sharing strategy is needed to shift the visual attention back and forth between the road scene and the devices (Tivesten and Dozza, 2014).

Most previous work focused on secondary tasks using one specific type of user interface (e.g., a navigation screen with a virtual keypad for address entry). Little research is done to compare different types of user interfaces (UIs) or UI elements on driver distraction for identical or similar secondary tasks, and how the inherently characteristics would affect task performance. While this is one of the important design decisions that HMI practitioners has to make when multiple design options are available. In addition, some studies have proposed that drivers avoid eyes-off-road time longer than a specific time, and drivers would divert the eyes back on the road right before this critical time is reached (e.g., 1.5 seconds by Wierwille, 1993). It is unclear if this critical time is the only constrain that dictates visual behavior (specifically, the timing for the start of the on-road glances). It is possible that there are other constrains imposed from the user interface that may also affect visual behavior. Typical in-vehicle control methods include the traditional physical push buttons (e.g., to turn the AC on or off), knobs (e.g., to increase or decrease the radio volume), resistive virtual buttons (e.g., to type in address on a touch screen navigation system), 2D gesture controls (e.g., swipe, pinch to zoom on a multi- touch screen). These control methods have their distinct interaction characteristics. For example, physical push buttons provide tactile feedback after the button actuation, while virtual buttons may only provide visual feedback after the button actuation. One of the fundamental principles in the NHTSA guidelines states that "*Any task performed by a driver should be interruptible at any time*." But there is a lack of studies on how a specific user interface and/or task specification would affect whether the task could be interrupted or not at a given moment.

This chapter describes an experiment in which we aimed at examining the effects of different control methods of in-vehicle technologies on the driver's visual behavior, task performance, and workload. Three control methods were selected for the investigation: (1) physical push buttons, (b) physical knobs, and (3) visual buttons without tactile feedback. We hypothesized that these three control methods would induce various levels of driver visual behavior, task performance, and workload.

5.2 **Methods**

5.2.1 Participants

Twenty participants (10 male and 10 female), all of whom were employees from a company in the U.S., were recruited for the experiment. In terms of age distribution, 9 (45%) participants were 20-29 years old, 1 (5%) participants were 30-39 years old, 4 (20%) participants were 40-49 years old, 5 (25%) participants were 50-59 years old, 1 (5%) participants were 60-69 years old. All participants had got their driver license for at least one year. The participants contributed to the experiment on their work time.

5.2.2 Tasks

The experiment was conducted in a stationary driving simulator in a laboratory environment as shown in Figure 30(a). The front virtual road scene was projected on a flat projection screen in front of the driving simulator. The tasks for the participants are described below:

Radio tuning task: The participants were asked to perform a radio tuning task using three methods as shown in Figures 2, 3, and 4.

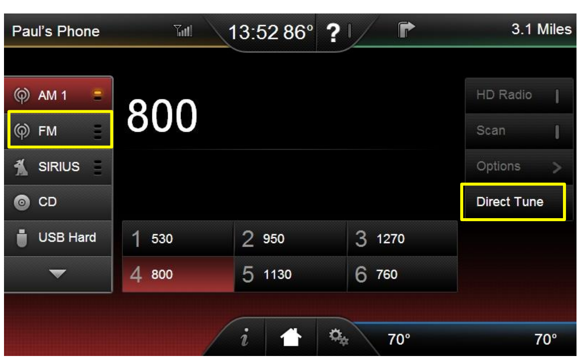
(a) Tuning radio using the "direct tune" method (b) A zoom-in view of the physical panel

Figure 30. Task Procedure using the physical buttons: (1) press the power button, (2) press the AM/FM button, (3) press the "DIRECT" button, (4) enter the radio frequency on the keypad.

(a) Tuning radio using the knob (b) A zoom-in view of the physical panel

Figure 31. Task Procedure using the knob: (1) press the power button, (2) press the AM/FM button, (3) turn the knob to decrease or increase the frequency shown on the display (“590” as in the picture)

(a) Tuning radio on the touch screen (b) Click the “Entertainment” button



(c) Click the “FM” then “Direct Tune” button (d) Enter the radio frequency

Figure 32. Task procedure using the virtual buttons

Table 4. Task procedures of tuning the radio to FM 98.7 using the three methods

Device	**Physical Buttons**	**Physical Knobs**	**Virtual Buttons**
Initial state	System OFF	System OFF	Home Screen
Step 1	Press ‘Power’	Press ‘Power’	Press ‘Entertainment’
Step 2	Press ‘AM/FM’	Press ‘AM/FM’	Press ‘FM1’
Step 3	Press ‘DIRECT’	Turn the knob to 98.7	Press ‘Direct Tune’
Step 4	Press ‘9’	*Done*	Press ‘9’
Step 5	Press ‘8’		Press ‘8’
Step 6	Press ‘7’		Press ‘7’
Step 7	*Done*		Press “Enter”
Step 8			*Done*

Driving task: The participants were asked to drive the simulated vehicle on a virtual highway. The virtual highway has two lanes in one direction, and the virtual course is a square loop with four straight sections connected by four curved corners. The driving environment is set as day time. The participants were asked to keep the vehicle in the left lane and maintain a speed of between 60-70 miles per hour. There is no other virtual vehicle in the left lane. The participants were asked to put both hands always on the steering wheel except when doing the radio-tuning task.

5.2.3 Experimental Design

There are two independent variables in this experiment: (1) Task condition (two levels: single or dual task), (2) Control modules (three levels: physical buttons, knobs, or virtual buttons).

Independent variables: Two task conditions were examined in the experiment: (1) single task, and (2) dual task. In the single task condition, the simulated vehicle is stopped on the side of the road. The participants were instructed to perform the radio-tuning task without driving the simulator. In the dual task condition, the participant was asked to drive the simulator in a highway scenario, and at given points, they were verbally instructed by the experimenter to start the radio-tuning task. The experimenter only instructs the participants to start the task when the vehicle is in the straight section of the road (i.e., not when the vehicle is entering, negotiating, or leaving a curve). The participants were asked to perform the radio-tuning task only at a time when they believe it is safe to do so. Three control modules (physical buttons, knob, and virtual buttons) were investigated in the experiment. The details of these three modules were described above.

Dependent variables: Two video cameras were used for collecting the task performance data. One camera was used to capture the participant's front face, which could be used to identify whether the participant was looking at the road or the device. Another camera was used to capture a close-up view of the area from the steering wheel to the device. This information could be used to measure the performance of the radio-tuning task. The driving related data were recorded by the driving simulator.

Task completion time is used as one of the performance measures. For the single task condition, task completion time is measured from the time when the driver starts to move his/her right hand away from the steering wheel to the time when the target radio frequency is reached. For the dual task condition, the task completion time is measured from the time when the participant started to move his/her eyes from the road to the device, to the time when the target radio frequency is reached. Other performance measures include the total eyes-off-road time (TEORT), which can be extracted from the video recording, and the number of glances (to the device from the road).

After each session the participants were asked to rate their subjective workload using the standard NASA-Task Load Index (NASA-TLX, Hart & Staveland, 1988, Hart, 2006).

A full factorial design was used in this experiment. Two independent variables gives 6 (= 2x3) combinations in total. For each combination, 3 trials were used. The experiment used a within-participant design. Each participant went through all the combinations. That gives a total of 18 (= 6 combinations x 3 trails per combination) trials for each participant.

5.2.4 Procedure

Once the participants arrived at the laboratory, they were firstly asked to complete a consent form. Then an introduction session was given to the participants to ensure that they understood the tasks they were about to perform. This was followed by a practice session for both the radio-tuning task and the driving task. Then the data collection part started, which was divided into one single task session and one dual task session. Half (10) of the participants started with the single task session, while the other half (10) started with the dual task. Half (10) of the participants started with the virtual buttons, a quarter (5) of them started with the physical buttons, and a quarter (5) of them started with the knob.

5.3 **Results**

5.3.1 Mental Workload

The mean and standard deviation of the subjective mental workload were shown in Figure 33. Statistical analyses of paired *t*-test were conducted to examine whether there is significance difference of workload between the two task conditions and among the three control modules. With alpha level of 0.05, it is found that the mental workload was significantly higher (all $p < 0.001$) in the driving condition than the parked conditions for all three control modules. In the parked condition, the mental workload when using the knob is significantly lower than using either physical buttons ($p = 0.031$) or virtual buttons ($p = 0.019$), while there is no significant difference ($p = 0.088$) in mental workload between using physical button and virtual button. In the driving condition, significant differences of mental workload were found in all three comparisons. Specifically, the mental workload when using the knob is significantly lower ($p = 0.0027$) than using the physical buttons, and the mental workload when using the physical buttons is significantly lower ($p = 0.0011$) than using the virtual buttons.

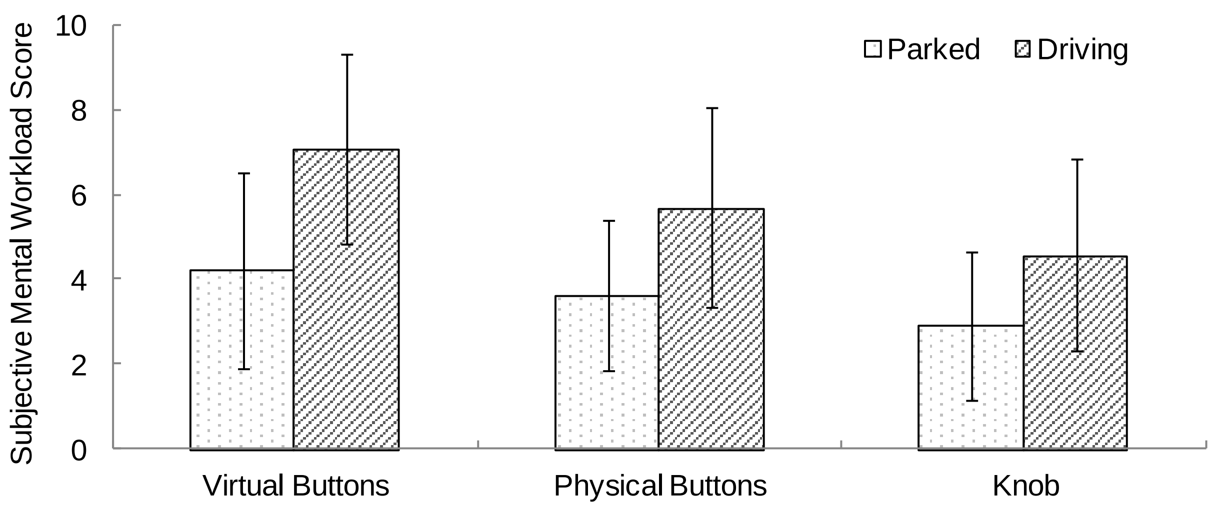


Figure 33. Subjective Mental Workload

5.3.2 Task Completion Time

For the task when the simulator is parked, the task completion time is the duration from the time when the participant started to perform the radio-tuning task to the time when the target radio frequency is reached. For the task performed during driving, the definition of task completion time from Tsimhoni and Green (2001) was used, which defines it as the duration from the beginning of the first glance at the device to the end of the last glance during the trial.

The statistics of the task completion time is shown in Table 5. The mean and standard deviation of the task completion time is shown in Figure 34. It is noticeable that there are larger variances of the task completion time in the driving condition than the parked condition.

Table 5. Statistics of task completion time in experiment 2 (all units are in seconds)

	Parked				Driving			
Control Module	M	SD	50th Percentile	95th Percentile	M	SD	50th Percentile	95th Percentile
Virtual buttons	7.46	1.08	7.62	9.03	10.81	3.01	10.14	16.01
Physical buttons	5.72	1.32	5.51	7.66	7.02	2.03	6.71	10.90
Knob	5.52	1.45	5.50	8.22	7.16	2.44	7.04	11.95

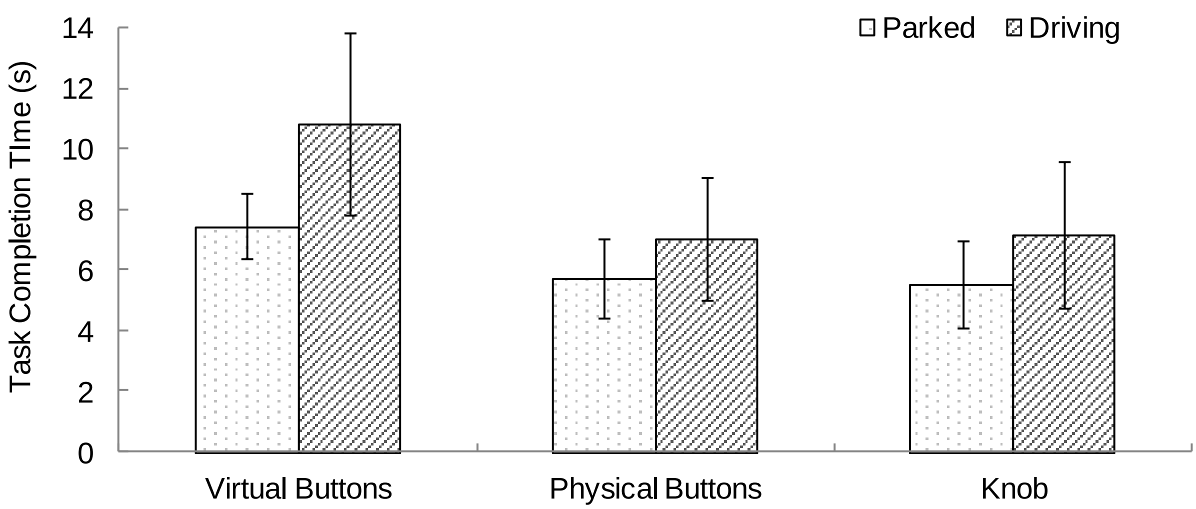


Figure 34. Task completion time

Wilcoxon signed-rank tests was conducted to examine whether there is a significant difference of task completion time between the parked and driving condition. The result shown the task completion time was significantly longer when using visual buttons ($Z = -3.680$, $p < 0.001$), physical buttons ($Z = -2.352$, $p = 0.019$), and knob ($Z = -3.260$, $p = 0.01$).

The nonparametric Friedman Test was used to examine whether there are significant differences of the task completion time among the three control modules in both the parked and driving conditions. When the driving simulator is parked, the result shows there was a significant difference in task completion time among the three control modules, $\chi^2(2) = 24.333$, $p < 0.001$. Post hoc analysis with Wilcoxon signed-rank tests was conducted with a Bonferroni correction applied, resulting in a significance level set at $p < 0.017$ (= 0.05/3 comparisons). The result shows the task completion time with the virtual buttons was significantly longer than the task completion time with both the physical buttons ($Z = -3.550$, $p < 0.001$) and the knob ($Z = -3.724$, $p < 0.001$), while there was no significant difference between the physical buttons and knob ($Z = -0.675$, $p = 0.500$).

When the participants were driving the virtual car in the simulator, the result shows a similar pattern. There was a significant difference in task completion time among the three control modules, $\chi^2(2) = 21.895$, $p < 0.001$. The same post hoc analysis with Wilcoxon signed-rank tests was conducted. And the result shows the task completion time with the virtual buttons was significantly longer than the task completion time with both the physical buttons ($Z = -3.783$, $p < 0.001$) and the knob ($Z = -3.702$, $p < 0.001$), while there was no significant difference between the physical buttons and knob ($Z = -1.087$, $p = 0.277$).

5.3.3 Total Eyes-Off-Road Time

The statistics of the total eyes-off-road time (TEORT) is shown in Table 6. The mean and standard deviation of the task completion time is shown in Figure 35. The mean value of TEORT when using the virtual buttons is roughly 3 seconds longer than the mean of TEORT when using either the physical buttons or the knob.

Table 6. Statistics of TEORT in experiment 2 (all units are in seconds)

Control Module	M	SD	50th Percentile	95th Percentile
Virtual buttons	8.27	1.95	8.07	11.23
Physical buttons	5.28	2.14	4.84	7.68
Knob	4.93	1.66	4.60	8.30

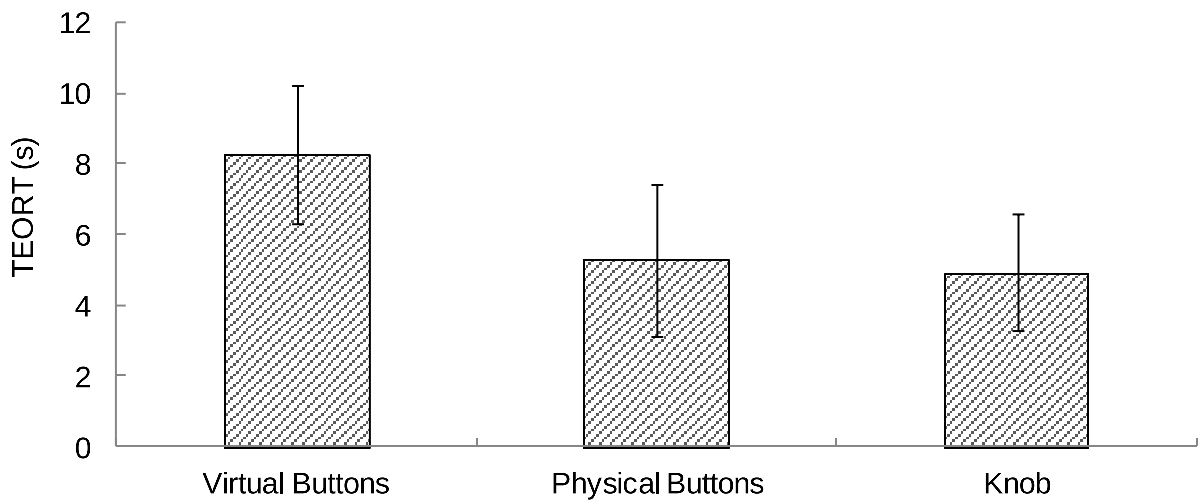


Figure 35. Total eyes-off-road time (TEORT)

The nonparametric Friedman Test was used to examine whether there are significant differences of the TEORT the three control modules. The result shows there was a significant difference in task completion time among the three control modules, $\chi^2(2) = 25.579$, $p < 0.001$. Post hoc analysis with Wilcoxon signed-rank tests was conducted with a Bonferroni correction applied, resulting in a significance level set at $p < 0.017$ (= 0.05/3 comparisons). The result shows the TEORT with the virtual buttons was significantly longer than the TEORT with both the physical buttons (Z = -3.582, $p < 0.001$) and the knob (Z = -3.823, $p < 0.001$), while there was no significant difference between the physical buttons and knob (Z = -0.373, $p = 0.709$).

5.4 **Discussions**

This experiment examined three typical and inherently different in-vehicle control modules, namely physical buttons, virtual buttons, and knobs, for their effect on the driver's visual behavior, task performance, and workload. A visual-manual task of radio-tuning was performed on each of the control modules in both single and dual task conditions. As the primary information for driving is from the visual channel, metrics based on eyes-off-road were used to measure the level of driver distraction. And subjective ratings based on NASA-TLX were used to measure the driver's mental workload. For the workload, the experiment results show that for all three control modules, the mental workload increases significantly when driving is involved. An explanation for the increased workload when driving is that, while driving, the radio tuning task is partitioned as driver needs to switch the visual attention back on the road to check the vehicle stability, and make corrective maneuvers if needed until the stability is regained, and then switch the visual attention back to the radio tuning device. This essentially increases the utilizations of the mental resources for different purposes (e.g., switching visual attention between the road scene and the device, using an internal clock to make sure the eyes-off-road does not exceed a threshold, etc.). For both single and dual task conditions, using the knob seems to induce the lowest workload among the three. In the single task condition, no significant difference of workload was found between the physical and virtual buttons, but once driving is involved, the visual buttons induces more workload than the physical buttons.

The experiment results show that the visual buttons induces the longest task completion time and total eyes-off-road time among the three. No significant different differences were found between the physical button and the knob. This may be due to the inherent characteristics of the control modules. For example, clicking a virtual button on a touch screen without tactile feedback requires visual attention to confirm that the correct button is successfully clicked (by using the visual cues of, for example, a change of the button's background color). And in some cases, the next button to click would not appear until the previous button has been pressed (e.g., The 'FM' button on the entertainment screen (Figure 32 (c)) appears only after the 'Entertainment' button on the home screen (Figure 32 (b)) is pressed). Thus the step of searching for the next button could start only after the pressing of the previous button finishes. While for the physical buttons on the other hand, the searching for the next button may start simultaneously as the start of pressing the previous button, as we assume that a successful pressing on a physical button does not require visual attention as it could be confirmed by the tactile feedback.

Even though the result shows the TEORT with the virtual buttons (about 8 seconds on average) is mostly shorter than the NHTSA guideline of 12 seconds. The distribution of the eyes-off-road time duration shows that 28% of the eyes-off-road are longer than 2 seconds, and 11% of the eyes-off-road are longer than 3 seconds. At the experiment driving speed of 60-70 miles per hour, this translates to roughly 60 or 90 meters driving without looking at the road. It is also interesting to notice that even it seems there’s no significant difference in the TEORT between the physical buttons and the knob (both with an average ETORT of about 5 seconds), the distribution of the eyes-off-road duration seems to suggest that when using the knob, looking at the device were performed in a more short-and-rapid fashion without longer eyes-off-road (e.g., only 6% of the eyes-off-road were longer than 2 seconds compared with 16% when using the physical buttons).

Chapter 6.

Queuing Network Modeling of Visual-Manual Secondary Tasks while Driving

6.1 Introduction

Today in-vehicle infotainment systems allow the driver to accomplish many non-driving tasks (e.g., listening to radios, adjusting cabin temperatures, finding the nearest gas station) using multimodal interfaces, such as touch screens, digital instrument clusters, head-up-displays, etc. Although such systems are designed to enhance the driving experience, they often suffer from serious usability problems, such as driver distractions with frequent and extended eyes-off-road operations, prolonged learning curve with information overload and unintuitive designs.

Compared with the traditional usability testing methods using real human subjects, digital human models can be used to test design concepts and prototypes with low costs in both time and manpower. This allows the HMI designers to explore a larger design space and address usability issues at the early stages of the system design process. Digital human models have been a valuable asset in many industries to analyze the human factors and ergonomics problems of products (e.g., Feyen, et al., 2000). In the past decades many driver models have been developed for particular aspects of the driving task, such as lane-keeping, car-following, and road signs searching. Several models have been developed based on task-independent cognitive architectures, including the ACT-R based driver models (Salvucci, 2006) and the QN-MHP (Queuing Network-Model Human Processor) driver model (Liu et al., 2006; Wu and Liu, 2007a). ACT-R assumes serial processing of production rules (i.e., only one production rules can be fired at any time). To model multitasking, each individual model needs to pass control to the other, and thus requires modifications of the single task models in order to model the multitasking scenarios. Salvucci (2002) compared two version of the model termed Single-Step and Group-Step. For the Single-Step model, the driving task intervenes after each firing of the rules of the secondary task, while for the Group-Step model, the driving task intervenes after a predetermined group of production rules got fired. But the way to group the production rules are largely arbitrary. There are no rigorous criteria or guidelines for this approach.

QN-MHP simulates the human cognition as a queuing network of information processing servers derived from the psychological and neuroscience fields (Liu 1996, 1997, Liu, Feyen, & Tsimhoni, 2006). This modeling framework has been successfully used to simulate a wide variety of human performances including transcription typing (Wu & Liu, 2008a), visual search (Feng & Liu, 2013), vehicle steering (Tsimhoni & Liu, 2003), driver performance and workload (Wu & Liu, 2007), etc. Recent work has also fully implemented ACT-R as a special case of the queuing network model (Cao & Liu, 2013).

Along the lines of the Model Human Processor (MHP), a secondary task could interfere with the driving task during the three processing stages of a task (i.e., perceptual, cognitive, and motor). At the perceptual stage, a visual-manual secondary task may require the driver to move the eyes away from the road in order to perceive the visual information on the devices ("eyes off the road"). At the cognitive stage, it may require the driver to reallocate his/her mental resources (e.g., attention, working memory) from the driving task ("mind off the road"). At the motor stage, it may require the driver to move his/her hands away from the steering wheel to manipulate a device, for example, clicking a button or turning a knob ("hands off the wheel").

Nonetheless even for the generic cognitive architecture-based models, there is still a gap that prevents them from being deployed in practice, as using them usually requires the designers to have a fairly deep understanding of the theoretical foundation of the model as well as programming skills in order to setup a simulation to run any particular task with HMI designs.

6.2 **Methods**

6.2.1 Task Analysis

To model any task using the QN-MHP architecture, a task analysis is required in the format of NGOMSL (Natural Goals, Operators, Methods, and Selection rules Language, Kieras, 1999). The NGOMSL task analysis breaks down the task in a "top-down, breadth-first" manner into "atomic-level" Task Components (TC). Each task component is associated with a task independent "context-free" QN-MHP operator from the QN-MHP operator library. The operators are set with parameters either explicitly by the modeler when specifying the task before the simulation, or implicitly by the model during the simulation. The sequential dependency of the task components are set by the modeler before the simulation. This task specification was loaded to the QN-MHP model and the task was performed in response to its associated environment stimuli. More details on the QN-MHP task analysis could be found in Chapter 2.

A NGOMSL-style task analysis of tuning the radio station using the direct tune was conducted as shown in Figure 36.

GOAL: Tune the radio to FM 98.7 using the direct tune on the physical panel

Method to accomplish goal of tuning the radio to FM 98.7 using the direct tune

TC 1: Look at <text feature> on <physical panel> at location <190, 46>

TC 2: Store <feature value> to short-term memory

TC 3: Retrieve <feature value> from short-term memory

TC 4: Compare <feature value> to <“Power”>

If match, return *result* = 1, else return *result* = 0

TC 5: If *result* = 1, go to TC 6, else go to TC 999 // 999 is a dummy TC

TC 6: Reach <location> with <right hand> <with visual guidance>

TC 7: Click with <right hand> <index finger>

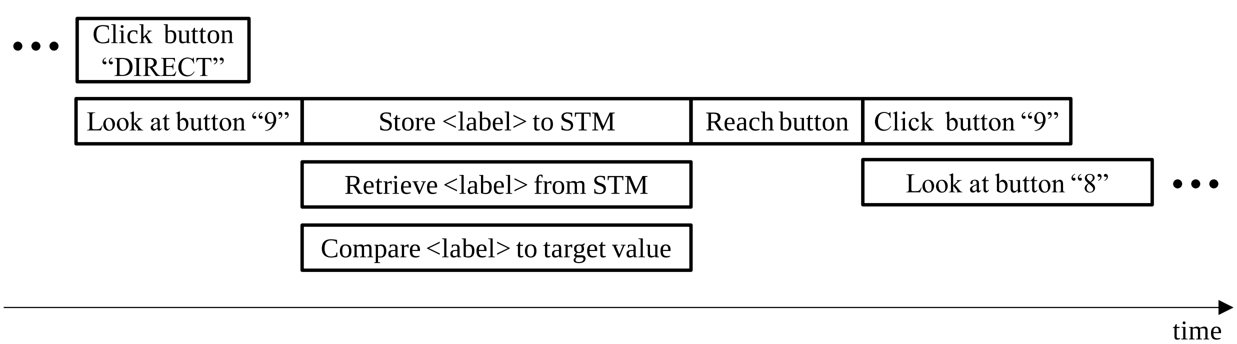
TC 8: Look at <text feature> on <physical panel> at location <224, 88>

... : ...

TC N-1: Click with <right hand> <index finger> // click the last button (“7”)

TC N: Return with goal accomplished

(a) Task Description



(b) Sequential Dependency of the Task Components

Figure 36. NGOMSL-style task analysis of the direct tune using the physical panel

GOAL: Tune the radio to FM 98.7 using the <u>direct tune</u> on the <u>touch screen</u>

Method to accomplish goal of tuning the radio to FM 98.7 using the direct tune

TC 1: Look at <text feature> on <physical panel> at location <457, 130>

TC 2: Store <feature value> to short-term memory

TC 3: Retrieve <feature value> from short-term memory

TC 4: Compare <feature value> to <"Entertainment">

If match, return *result* = 1, else return *result* = 0

TC 5: If *result* = 1, go to TC 6, else go to TC 999 // 999 is a dummy TC

TC 6: Reach <location> with <right hand> <with visual guidance>

TC 7: Click with <right hand> <index finger>

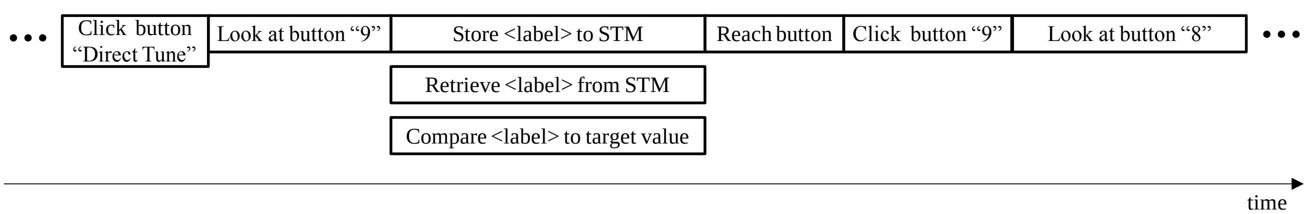
TC 8: Look at <text feature> on <physical panel> at location <167, 75>

... : ...

TC N-1: Click with <right hand> <index finger> // click the last button ("Enter")

TC N: Return with goal accomplished

(a) Task Description



(b) Sequential Dependency of the Task Components

Figure 37. NGOMSL-style task analysis of the direct tune using the touch screen

GOAL: Tune the radio to FM 107.1 using the knob on the physical panel

Method to accomplish goal of tuning the radio to FM 98.7 using the direct tune

TC 1: Look at <text feature> on <physical panel> at location <190, 46>

… : … // TC 1-14 are identical to the TC 1-14 for the direct tune on the physical panel

TC 15: Look at <text feature> on <physical panel> at location <190, 204>

TC 16: Store <feature value> to short-term memory

TC 17: Retrieve <feature value> from short-term memory

TC 18: Compare <feature value> to <“TUNE”>

If match, return *result* = 1, else return *result* = 0

TC 19: If *result* = 1, go to TC 20, else go to TC 999 // 999 is a dummy TC

TC 20: Reach <location> with <right hand> <with visual guidance>

TC 21: Look at <text feature> on <physical panel> at location <32, 96>

TC 22: Store <feature value> to short-term memory

TC 23: Retrieve <feature value> from short-term memory

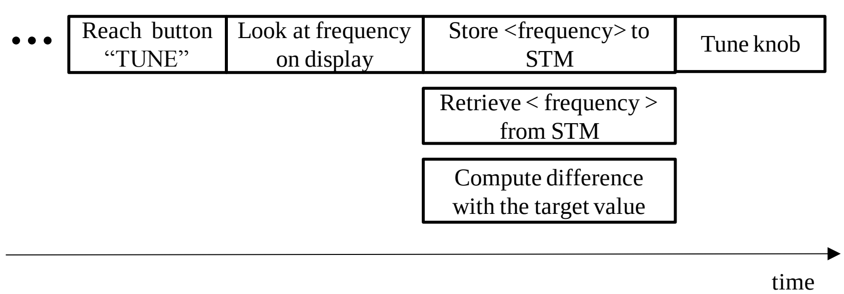
TC 24: Compute <subtract from> <107.1> // 107.1 is the target frequency

TC 25: If *result* = 0, go to TC 27, else go to TC 26

TC 26: Turn knob with <right hand>

TC 27: Return with goal accomplished

(a) Task Description



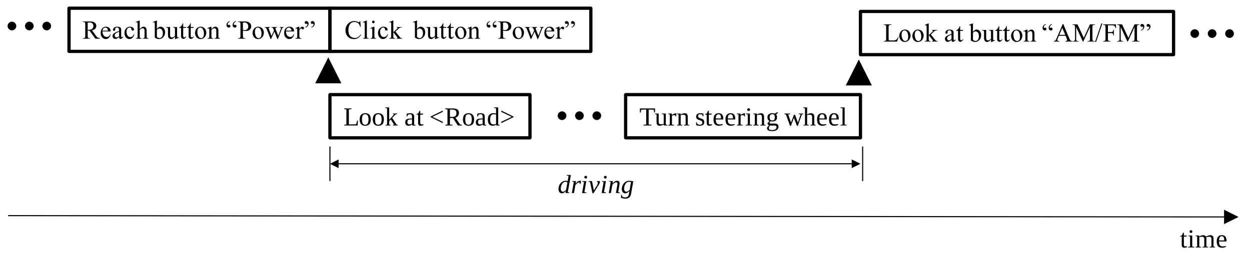
(b) Sequential Dependency of the Task Components

Figure 38. NGOMSL-style task analysis of the radio tuning using the knob

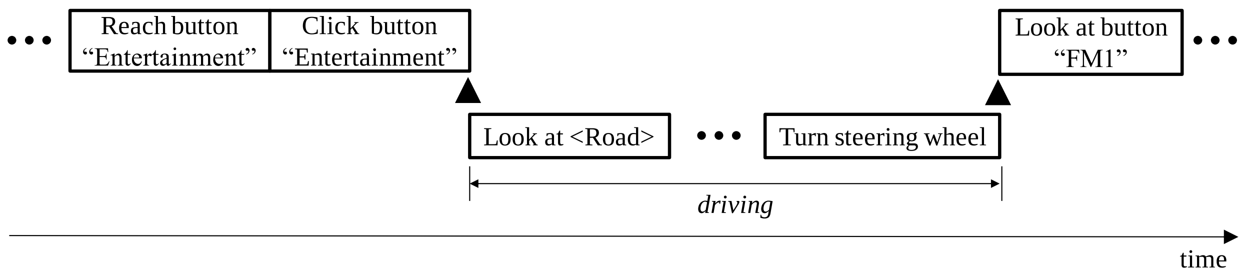
The description for the task of using the touch screen follows a similar pattern, while there are differences on the sequential dependency of the steps (i.e., at what time a step starts being executed) due to the inherent characteristics of the devices. For example, clicking a virtual button on a touch screen without tactile feedback requires visual attention to confirm that the correct button is successfully clicked (by using the visual cues of, for example, a change of the button's background color). And in some cases, the next button to click would not appear until the previous button has been clicked (e.g., The 'FM' button in the entertainment screen appears only after the 'Entertainment' button is clicked). Thus the step of searching for the next button could start only after the clicking of the previous button finishes. For the physical panel on the other hand, search for the next button could start simultaneously as the start of clicking the previous button, as it is assumed that successful clicking on a physical button could be confirmed sufficiently by the tactile feedback (currently not explicitly modeled) with no need for visual attention. We do assume that the driver needs the visual guidance for reaching the button (i.e., no 'touch typing'), so the search for the next button could start as early as the button is reached by hand, but not earlier.

6.2.2 Multitasking Modeling

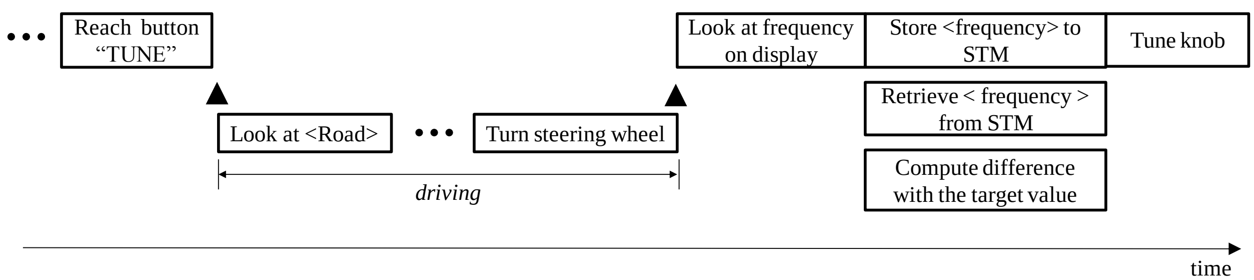
Given that the secondary tasks usually take at least a few seconds to complete even as a single task, when they are performed with driving, the driver is likely not able to complete the entire secondary task with a single glance of eyes off the road. Thus the secondary task needs to be segmented by the driving task so the driver could look back to the road to check the safety and vehicle stability, and make potential necessary maneuvers before continuing with the secondary task. Figure 39 shows how the three secondary task of tuning radio are segmented by the driving task. For the physical buttons, we assume that the driver is able to use the haptic feedback instead of visual feedback to confirm a successful clicking of a button. Thus he could start looking back to the road as soon as he starts to click the button. But this is not possible when using the touch screen, where the driver could only use the visual feedback to confirm if a button is successfully clicked. In this case, the driver can only start looking at the road when the clicking action is completed.



(a) Using physical buttons



(b) Using virtual buttons



(c) Using knob

Figure 39. Multitasking of radio tuning and driving

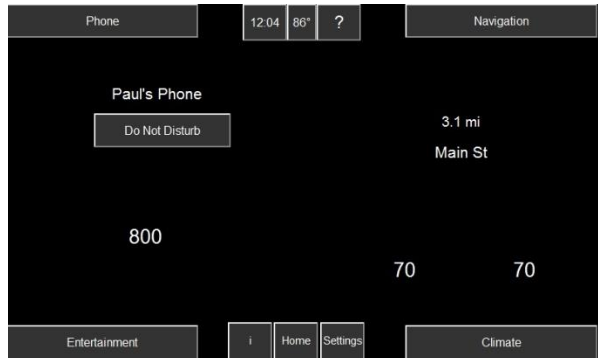
6.2.3 Device Mockups

Aside from the cognitive model which represents the human, digital mockups of the devices are needed to simulate the interaction between the human and the environment. The digital mockups provide stimuli as inputs to the cognitive model, and could be acted upon by the actions generated by the cognitive model (e.g., clicking a button). Digital mockups for this visual search task were created using MATLAB GUIDE (Graphical User Interface Design Environment). Figure 40 shows both the physical device using in the human experiment and the digital mockups created for the simulation.

(a) Real device used by the human participants (b) Digital mockup used by the model

Figure 40. Real device used in the human experiment and its digital mockup

(a) Home screen – real device

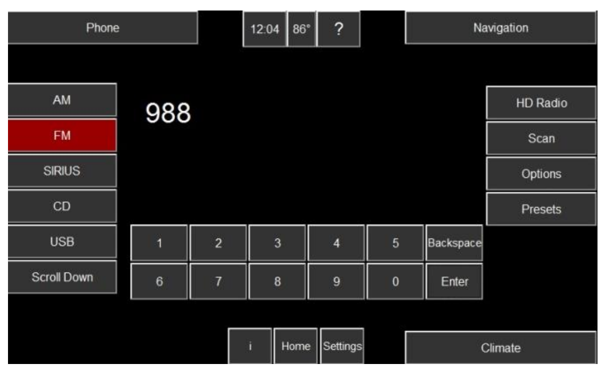


(b) Home screen – digital mockup

(c) FM preset screen – real device

(d) FM present screen – digital mockup

(e) Direct tune screen – real device



(f) Direct tune screen – digital mockup

Figure 41. Real device used in the human experiment and its digital mockup

6.2.4 Specifications of the Task Environment and Initial Conditions

The task environment specifications include (1) the physical location of the devices (i.e., the physical panel and the touch screen) relative to the steering wheel, (2) the visual angle between the road scene and the devices, and (3) the viewing distance of the devices from the driver. These specifications were set as the same value as measured from the experiment with human subjects in Chapter 5. For the single task condition, it is assumed that the driver starts the task with both hands on the steering wheel, and eyes looking at the center of the touch screen. For the dual task, it is assumed that the driver starts the driving task first with both hands on the steering wheel, and eyes on the road scene. After setting up the simulation, simulation was conducted for the radio tuning task.

6.3 **Results**

6.3.1 Mental Workload

The experiment with human subjects showed that the mental workload was significantly higher in the driving condition than the parked conditions for all three control modules. The simulation results are shown in Figure 42. The model results are similar to the empirical data from the human participants for both the parked and driving conditions. For the parked condition, the mean absolute percentage error (MAPE) is 14.1%. The root-mean-square error (RMSE) is 0.50. For the driving condition, the MAPE is 18.8%. The RMSE is 1.13.

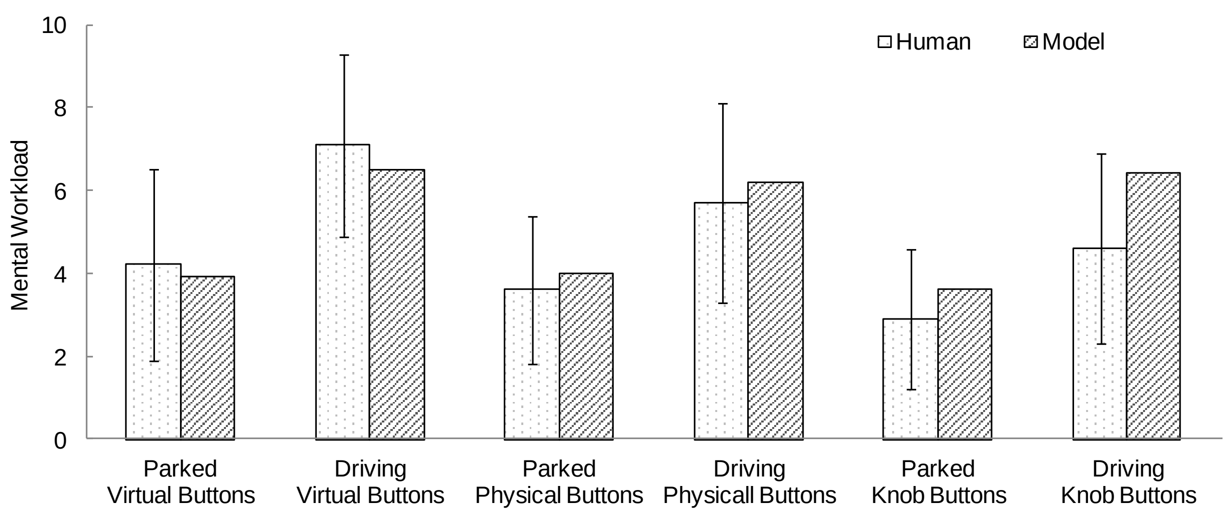


Figure 42. Simulation results of mental workload in comparison to human results

6.3.2 Task Completion Time

The experiment with human subjects showed that the dual task has significantly longer reaction time than single task for all three control modules. In the single task condition, the task completion time is significantly longer when using the virtual buttons than using either the physical buttons or the knob. There is no significant difference in task completion time between the physical buttons and knob. In the dual task condition, the task completion time is significantly longer when using the virtual buttons than using either the physical buttons or the knob. There is no significant difference in task completion time between the physical buttons and knob. The model predictions of the task completion time are shown in Figure 43. The model predictions are similar to the empirical human data. For the parked condition, the MAPE is 5.7%. The RMSE is 0.38 second. For the driving condition, the MAPE is 9.0%. The RMSE is 0.81 second.

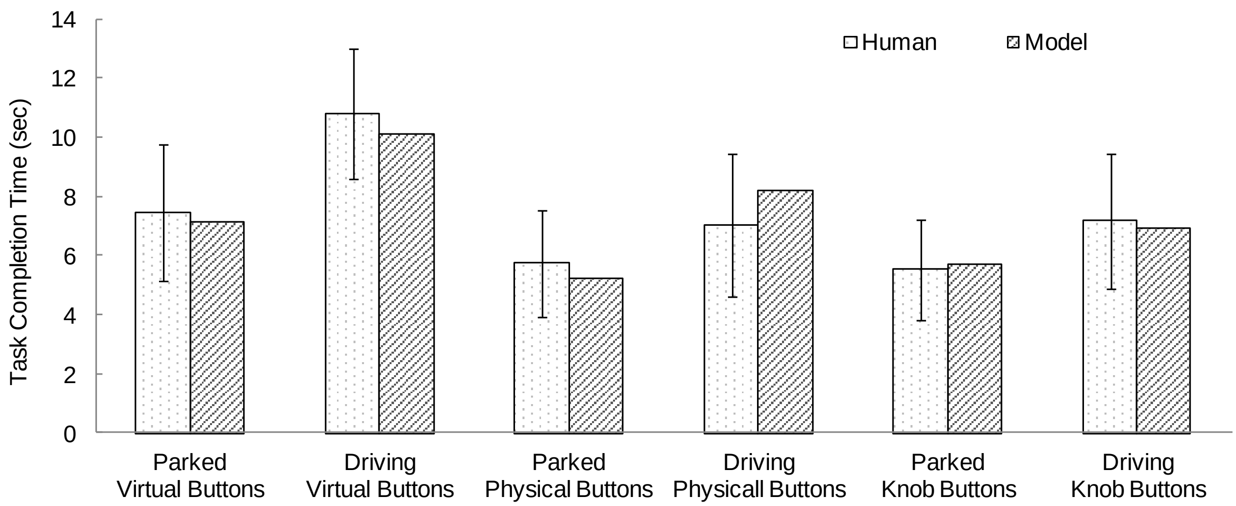


Figure 43. Simulation results of task completion time in comparison to human results

6.3.3 Total Eyes Off Road Time (TEORT)

The experiment with human participants shows that the TEORT was significantly longer when using the virtual buttons than either the physical buttons or the knob, while there is no significant difference of TEORT between the physical buttons and the knob. The model simulation results of the TEORT are shown in Figure 44. As can be seen the model predictions are similar to the human data. The MAPE is 12.1%. The RMSE is 0.71 second.

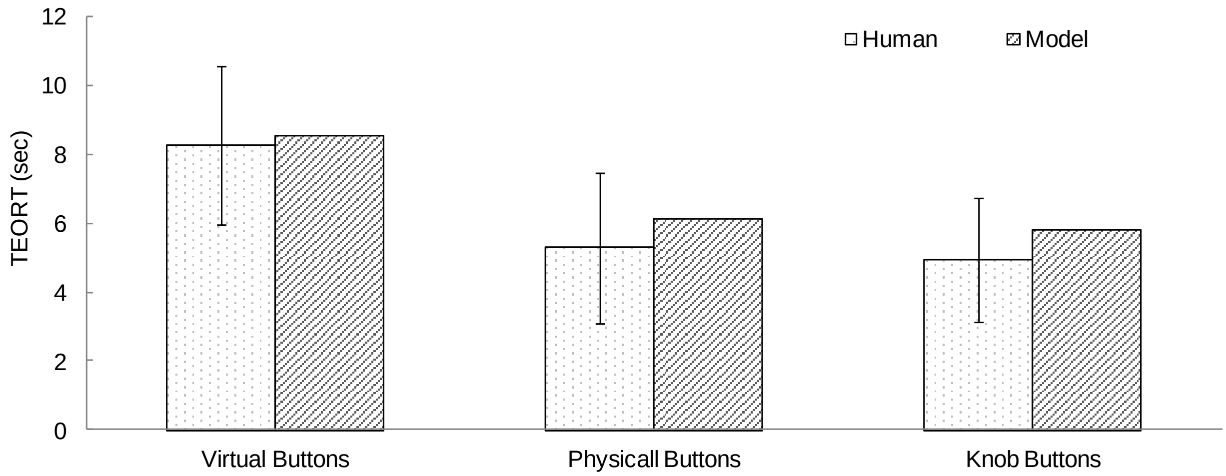


Figure 44. Simulation results of TEORT in comparison to human results

6.4 **Discussions**

The observed performance differences among the three control modules and the two task scenarios may be accounted for by the digital driver model in several aspects. As discussed in the previous section, the tasks on the physical panel take shorter time to complete, because the driver is able to click one button while simultaneously direct the visual attention to the next button, but this is not possible for the touch screen without tactile feedback, because the user could only relies on the visual feedback to confirm the button is successfully clicked. In addition for the touch screen, some virtual buttons may not appear (e.g., the 'FM' button in the above case) until the activation of previous buttons prevents the driver from performing the task in a more parallel fashion. These differences were naturally captured during the task and device setup of the QN-MHP simulation. The predicted task completion time when using the touch screen in the dual task condition is about 1 second longer than the human data (mean value). This could be due to the time pressure in the dual task condition human subjects sometimes skip the confirmation of button clicking, and start to look at the next button location before a clicking action is completed. This essentially makes the task faster to complete, while it risks of making errors of unsuccessful clicking of a virtual button.

An explanation for the increased task completion time and mental workload when driving is involved, is that, while driving, the radio tuning task is partitioned as driver needs to switch the visual attention back on the road to check the vehicle stability, and make corrective maneuvers if needed until the stability is regained, and then switch the visual attention back to the radio tuning device. This essentially makes the task longer to complete and increases the server utilizations which are used to estimate the mental workload.

It should be noted that each of the task analysis presented in this chapter only represents one typical case for how the task is performed. It is assumed that the driver has perfect and complete memories of the locations of the target buttons, so the visual attention could be directed by the top-down knowledge, rather than having to perform a thorough visual search over the whole device for the target button. This is essentially only modeling the user population who are familiar with the devices (which is assumed to be the case of the subjects in the validation experiment). If it is of interest to model the novice users who may not always remember the target button locations, or to model the extremely proficient user who may be able to ‘touch type’ without the use of visual attention, different task analyses have to be conducted to represent the differences in the task strategies for the intended user population.

Chapter 7.

A Computer-Aided Usability Testing Tool for In-Vehicle Infotainment Systems

7.1 Introduction

Usability evaluation is one of the important processes of user-centered design for interactive systems. The traditional methods for usability testing typically rely on human test subjects, who are asked to perform specified tasks using the systems being tested. Their performances are recorded and then analyzed. Compared with the traditional methods, digital human models can be used to test design concepts and prototypes with low costs in both time and manpower. They allow the user interface (UI) designers to compare multiple design concepts, explore a larger design space, and address usability issues at the early stages of the design process.

Digital human models have been a valuable asset in many industries to analyze the physical ergonomics problems of product or biomechanical injury risk of workplace designs (Feyen, et al., 2000, Chaffin, 2007). One of the most successful digital human models is Jack which was originally developed by the University of Pennsylvania and later became commercial software (SIEMENS). The model could be used to address the ergonomic aspects of manual operations. The University of Michigan developed Human Motion Simulation (HUMOSIM) framework that could simulate realistic human movements and be used for ergonomic analysis of products and workplaces (Reed, et al., 2006). Despite the success of these digital models, they are largely focused on the physical ergonomics (e.g., involving human posture, reach, movements) based mostly on human biomechanics. Given the complexity of human cognition, fewer applications are found in the industry with human cognitive models.

Many driver models have been developed for particular aspects of the driving task, such as lane-keeping, car-following, and road signs searching. Several models have been developed based on task-independent cognitive architectures, including the ACT-R based driver models

(Salvucci, 2006) and the QN-MHP (Queuing Network-Model Human Processor) driver model (Liu, Feyen, & Tsimhoni, 2006; Wu & Liu, 2007a). QN-MHP simulates the human cognition as a queuing network of information processing servers derived from the psychological and neuroscience fields (Liu 1996, 1997, Liu, Feyen, & Tsimhoni, 2006). This modeling framework has been successfully used to simulate a wide variety of human performances including transcription typing (Wu & Liu, 2008a), visual search (Feng & Liu, 2013), vehicle steering (Tsimhoni & Liu, 2003), driver performance and workload (Wu & Liu, 2007), etc. Recent work has also fully implemented ACT-R as a special case of the queuing network model (Cao & Liu, 2013). Nonetheless even for these generic cognitive architecture-based models, there is still a gap that prevents them from being deployed in practice, as using them usually requires the designers to have a fairly deep understanding of the theoretical foundation of the model as well as programming skills in order to setup a simulation to run any particular task with HMI designs.

Pew (2008) listed three major challenges for a successful digital human model: (1). Simplified model development; (2) better capabilities for articulating and visualizing how the models work, and (3) model validation. Several digital cognitive models were developed in the past decades with the aim of predicting human performance. A summary of these existing model s are listed in Table 1.

The work described in this chapter aims at developing a CAE software tool that could make quantitative predictions of the usability of in-vehicle infotainment systems in terms of the task performance and workload. To achieve this goal, the core of the software needs to be built upon a model that is capable of simulating human multi-task performance including driving and a wide variety of in-vehicle tasks in an accurate and reliable manner. In addition, the software needs to allow the UI designers to set up the simulation with an easy-to-follow and user-friendly interface.

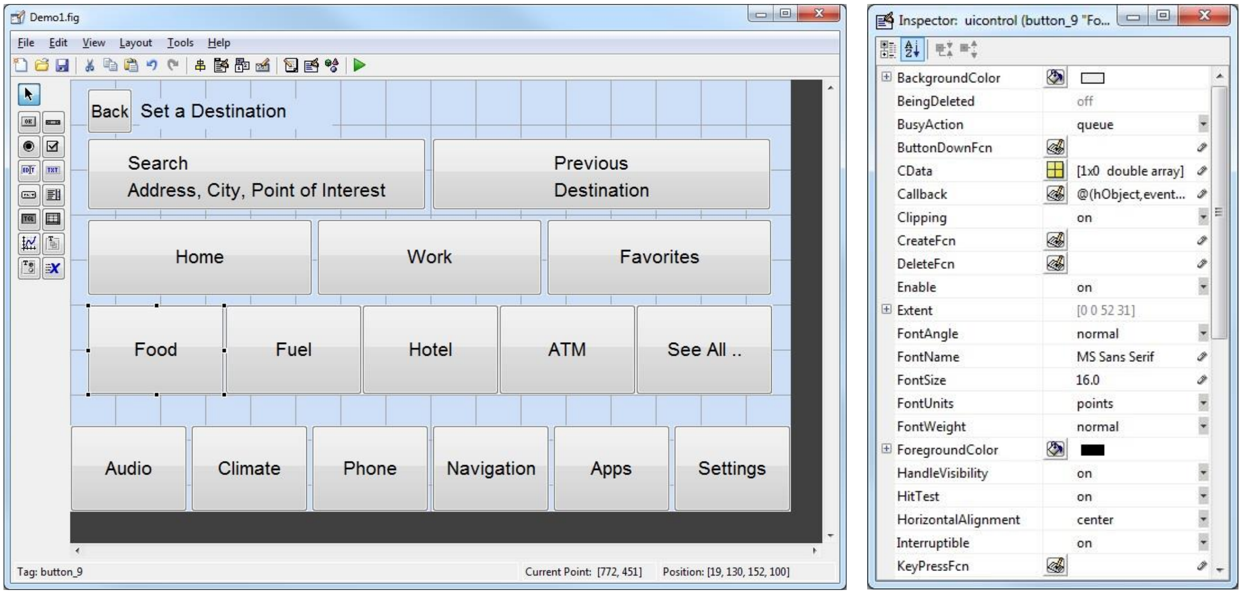
Table 7. Summary of some existing cognitive modeling tools

Tools	Theoretical framework	Modeling multitask	Software GUI	UI prototyping	Software requirements	References
QN-MHP MATLAB	QN-MHP	Yes	Yes	Yes (MATLAB GUIDE)	MATLAB/Simulink	Feng, et al, 2014
QN-MHP with VBA	QN-MHP	Yes	Yes	Rudimentary with static images only	ProModel + VBA in Excel	Wu & Liu, 2009
QN-ACTR	ACT-R, QN	Yes	Yes	Yes	Micro Saint	Cao & Liu, 2012
ADAT	SEEV and N-SEEV attention models	Yes	Yes	No	-	Sebok, et al., 2012
GLEAN	EPIC	No	Rudimentary menu interface	No	Macintosh Common Lisp	Kieras, et al., 1995
CRITIQUE	KLM	No	Yes	Yes (subArctic toolkit)	-	Hudson, et al., 1999
APEX	GOMS	Yes	-	-	Standalone software	Freed, et al, 2003
ACT-Simple	ACT-R	No	No	No	-	Salvucci & Lee, 2003
Distract-R	ACT-R	Yes	Yes	Yes	Standalone software	Salvucci, et al, 2005
CogTool	KLM + ACT-R	No	Yes	Import HTML	Standalone software	John, et al., 2004
E-GOMS	GOMS	No	Yes	-	-	Gil, 2010

7.2 **Software Development**

7.2.1 UI Design Prototyping

To evaluate a UI design, first a prototyping tool is needed to create a digital mockup of the design that the cognitive model could interact with. The interaction would include the cognitive model perceiving information from the digital mockup (e.g., visual information of buttons), and the cognitive model generating motor actions upon the digital mockup (e.g., clicking a button). In this proposed work a prototyping tool was developed using MATLAB GUIDE (Graphical User Interface Design Environment, see Figure 45 (a)). MATLAB GUIDE allows creating UI designs with the support of common GUI objects including push buttons, toggle buttons, text, etc. These objects could be created graphically with drag-and-drop and edited by editing the property inspector of the selected object (see Figure 45 (b)). The behaviors of objects upon user actions (e.g., switching to a different screen once a button is clicked) could be defined by editing the callback function of the object. MATLAB GUIDE also supports WYSIWYG (What You See Is What You Get). The software user could check the current design's look and behavior at any time during the prototyping process. Figure 46 shows two digital mockups developed using the MATLAB GUIDE. The created design is saved in the format of MATLAB FIG-file and its associated m-file.



(a) MATLAB GUIDE Main Screen (b) Object Property Inspector

Figure 45. MATLAB GUIDE for UI design prototyping

Figure 46. Demonstration of two digital mockups created using MATLAB GUIDE

7.2.2 Front End: Software GUI

Once the digital mockups of the to-be-tested designs are created, the software user could setup the simulation using the software GUI. Figure 47 shows the software structure of the proposed work. At the front end is a graphic user interface (GUI) that allows the software users to setup and run the simulation in four steps. At the back end is the simulation environment with four components. The details of the front and back end of the software and the information flow between the components are described below.

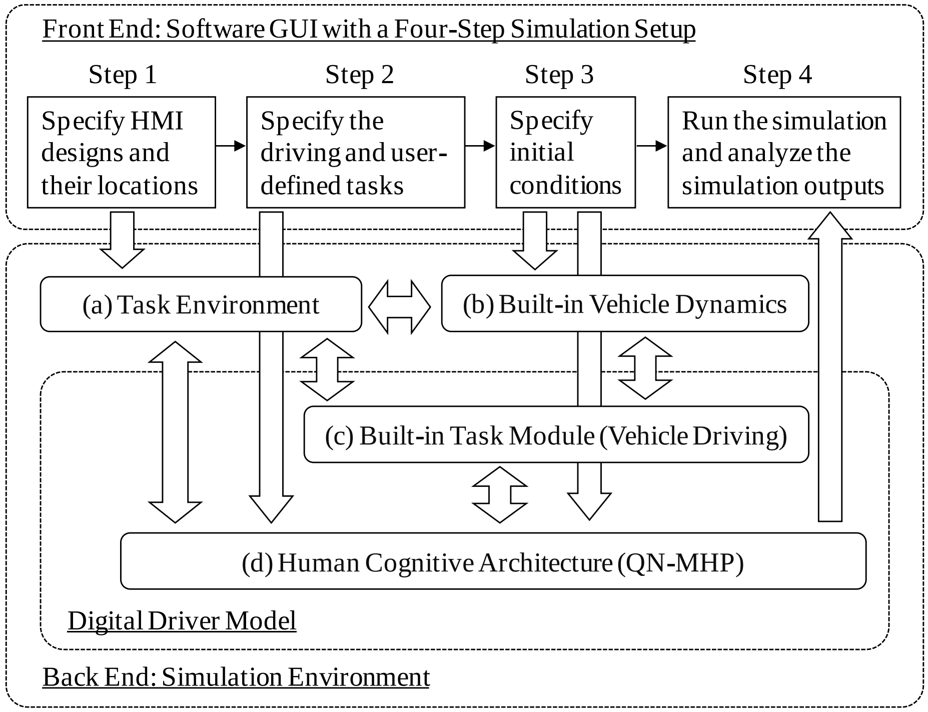


Figure 47. Software architecture

Step 1: The setup starts with adding virtual devices to the task environment. A virtual device could be added by filling out one row in the virtual device table (see Figure 48). A unique name is required for each virtual device, and the user can select one of the pre-defined device types (currently support physical panels, touch screens, and displays). The viewing distance refers to the distance from the digital human's eyes to the device. The profile of the device could be imported to the task environment by first clicking the "Import Device Profile" button, and then select the design profile saved earlier in the design prototyping. The software also allows the user to temporarily disable a device without deleting the device.

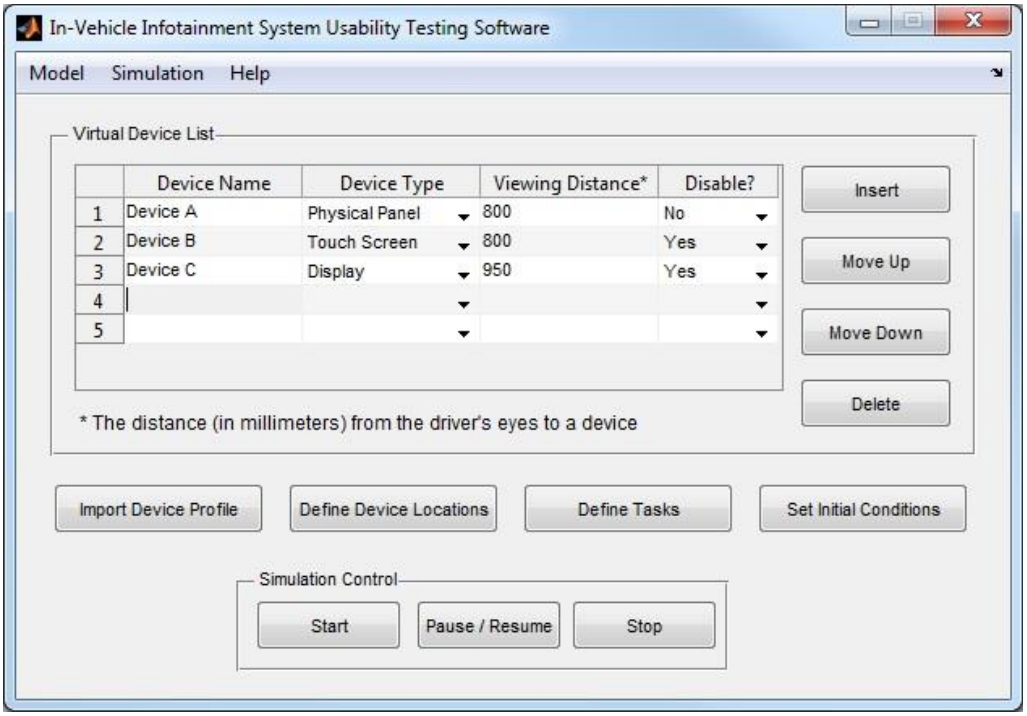


Figure 48. Main window of the software GUI

Once the devices are defined, the user needs to specify the locations of the devices in the task environment. This is done in the "Device Locations" window (see Figure 49) after clicking the "Define Device Location" button on the main screen. The locations are specified in terms of their physical distance and/or visual angles between the device to the steering wheel or the front road scene.

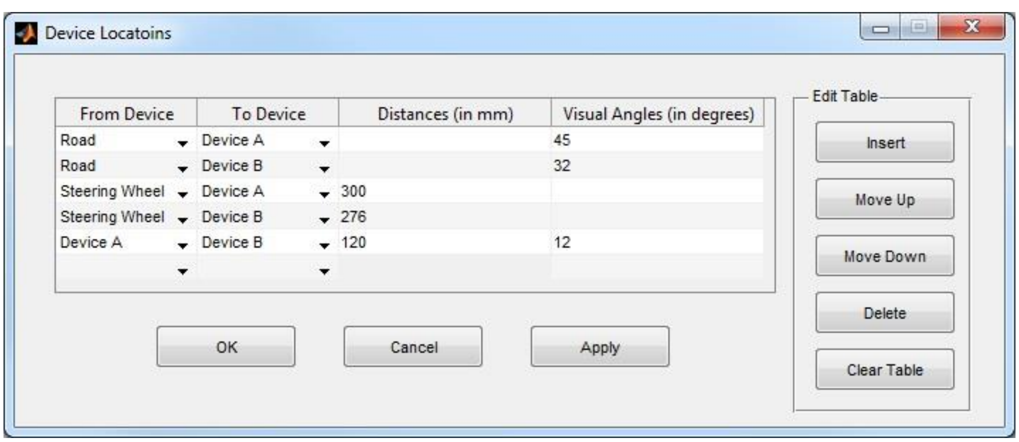


Figure 49. Device location window

Step 2: After the virtual devices and their locations are specified in Step 1, the related task information needs to be specified as the instructions to the digital driver on how to perform the tasks with the devices. A new task could be added by filling out one row in the task table in the "Manage Tasks" window (see Figure 50) after clicking the "Define Tasks" button on the main screen. A unique name is required for each task, and the user can select one of the two task types (either user-defined or driving). The details of a user-defined task can be specified in the "Task Component Table" in the "Define Tasks" window (see Figure 51) after clicking the "Define Task Details" on the "Manage Tasks" screen. The details of the task are defined by conducting an NGOMSL-style (Natural Goals, Operators, Methods, and Selection rules Language) task analysis which breaks down the task into QN-MHP-level operators. More details on the NGOMSL-style task analysis could be found on Liu et al. (2006). The software also allows the user to save a task and load it later. The built-in driving task could be included by simply creating a new task and selecting driving as its task type.

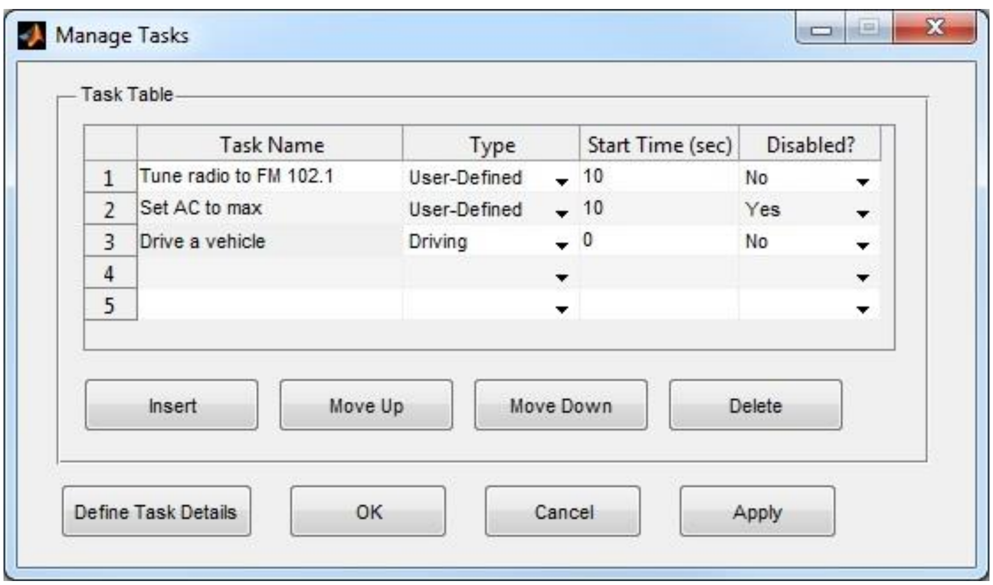


Figure 50. Task management window

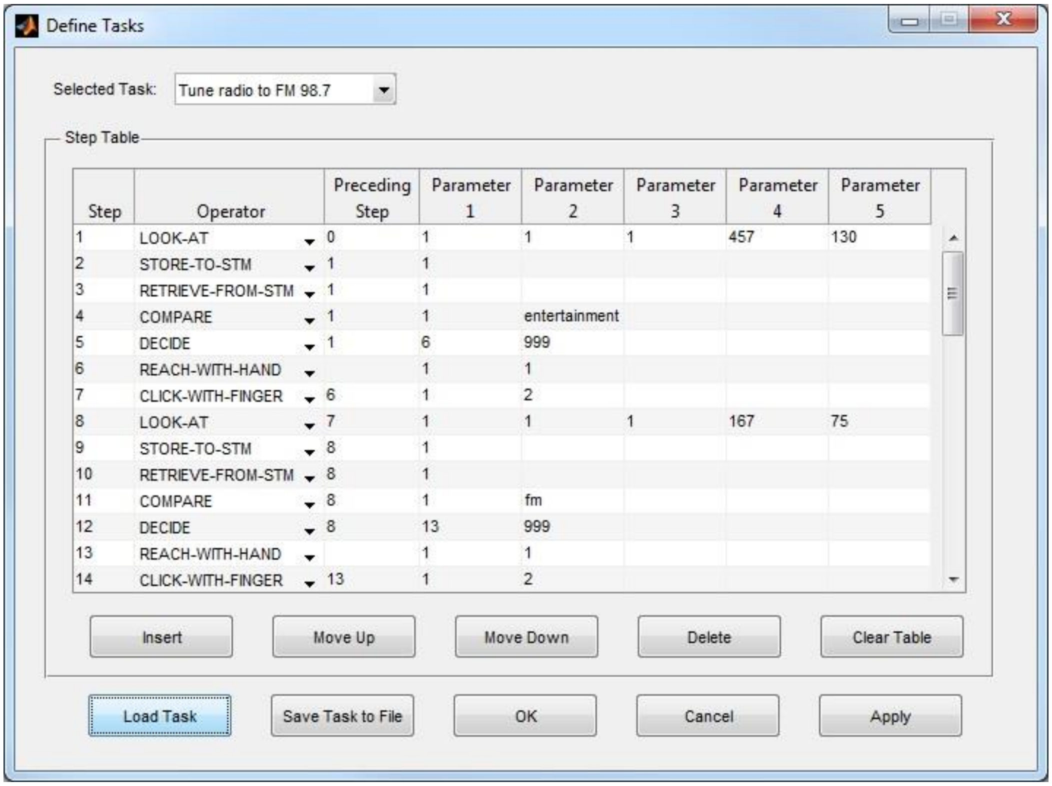
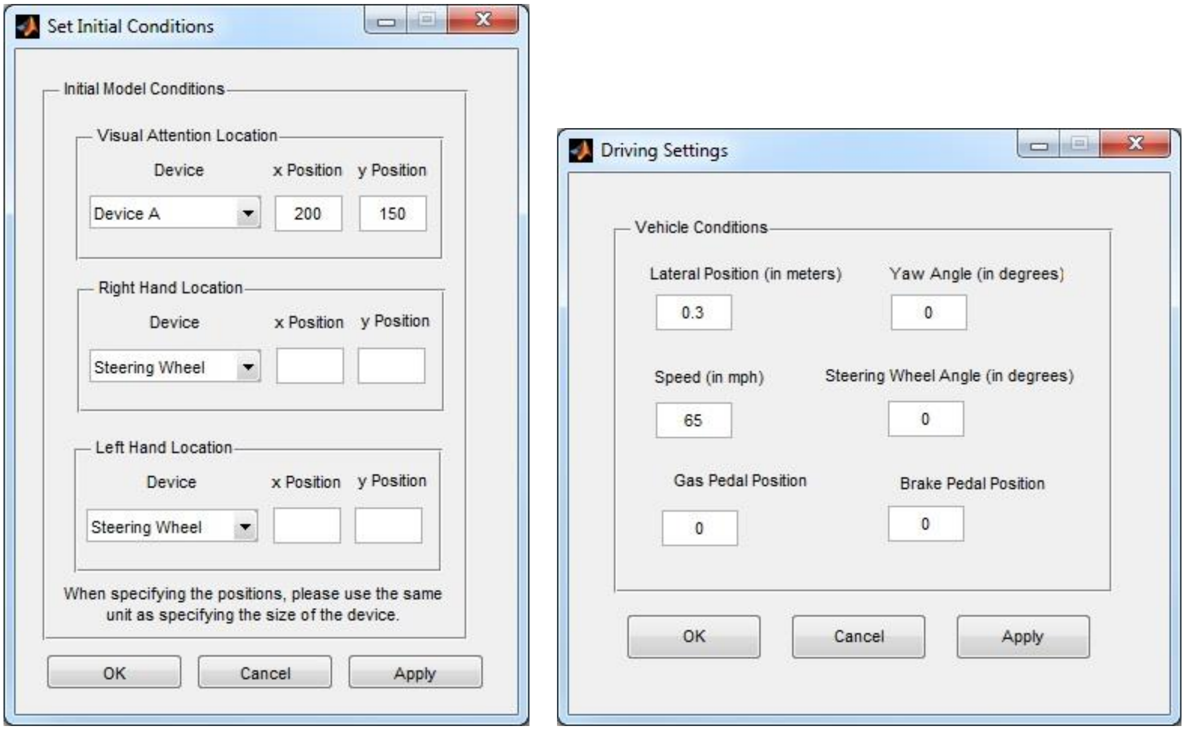


Figure 51. User-defined task definition window

Step 3: Once the task information is specified in Step 2, the software user needs to specify the initial conditions of the digital driver (e.g., where the digital driver's visual attention is located), the devices (e.g., a device is at a particular screen), and the vehicle (e.g., position on the road) at simulation time zero in Step 3. Step 4 is to run the simulation and generate the outputs. Throughout the course of the simulation the software is able to generate digital driver's instantaneous task performance and workload.



(a) Initial conditions of the digital human (b) Initial conditions of the vehicle

Figure 52. Setting the initial condition of the simulation

Once Steps 1-3 are completed, the software user can run the simulation by clicking the "Start" button on the main screen.

7.2.3 Backend: Software Architecture

At the back end of the software the task environment (box a in Figure 47) represents the environment with which the digital driver interacts. It stores the information of the mockup designs and the driving environments once they are specified during the simulation setup by following the 3 steps described above. During the simulation run it receives outputs from the QN-MHP's body part servers (e.g., the hand server to click a button), and supplies updated input stimulus to the QN-MHP (box d)'s perceptual servers (e.g., the visual input server).

The vehicle dynamics (box b) is a built-in module that receives input from the QN-MHP's driving related actions (e.g., steering), and generates the vehicle responses which would be used to update the driving environment in box a. Currently a three-Degree-Of-Freedom (longitudinal, lateral and yaw) bicycle model of vehicle dynamics was implemented for its simplicity and wide use. More details about the vehicle dynamics module can be found in Chapter 2.

The driving task module (box c) is a built-in extension of the QN-MHP that represents the skills of driving a vehicle. Its current implementation is adopted from Salvucci (2006)'s driving model, which assumes that a driver utilizes two distinct visual cues on the road (termed near point and far point, respectively) to determine how much to steer the vehicle. More details on the driving task module can be found in Chapter 2.

The QN-MHP (box d) represents the generic digital human. Its procedural long-term memory server stores the task information once it is specified in the simulation setup. During the simulation the task information is used as the instructions to the digital driver on how to perform the tasks. During the simulation, the QN-MHP is able to generate the task performance based on the information available at the queuing network, and estimate the workload based on the queuing network server utilizations (Wu & Liu, 2007).

7.3 **Software Outputs**

During the simulation the software is able to generate instantaneous outputs of the task performance of the digital human. Figure 53 shows the instantaneous simulation outputs of the digital driver performs a secondary task while driving in one simulation run. The model is able to generate outputs of vehicle states such as steering wheel angle, vehicle position in the lane (both lateral and longitudinal), driving performance measures such as Time to Lane Crossing (TLC), and driver behavior and states such as eye gaze location (on the road or off the road) and estimated mental workload.

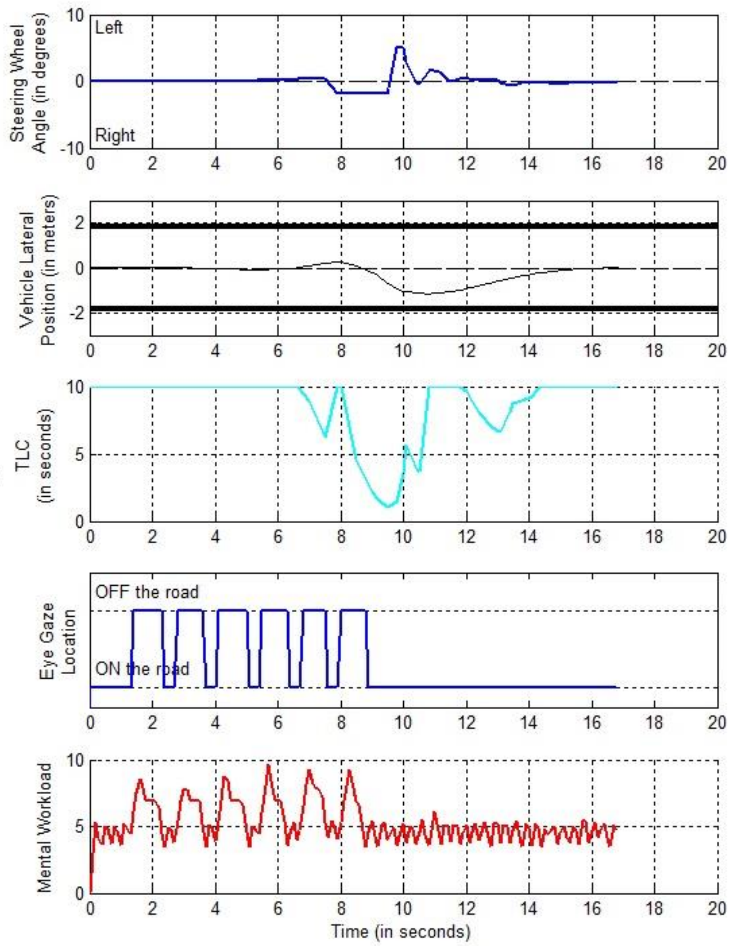


Figure 53. Instantaneous simulation outputs in one simulation run

During the simulation a trace window (see Figure 54) also shows the discrete events that have occurred in chronological order with the latest event on the top. This trace information could be useful for the software user to examine the progress of the task execution and get some insights of how the task is performed at the QN-MHP server level. This information could also be used by the model developers to debug and verify the cognitive model or extension modules at the model and software development stage.

At the end of the simulation, the software generates a report of the simulation results in terms of the task performance (see Figure 55). The detailed information of the digital human during the simulation is automatically saved in a spreadsheet that can be viewed later after the simulation (see Figure 56).

Time (in ms)	Events
3196	Task information for "Click the USB button" (Step # 5) is entering Server D for processing.
3196	The DECIDE operator is activated due to step 5 of task "Click the USB button" at Server D.
3196	The COMPARE operator is activated due to step 4 of task "Click the USB button" at Server D.
3196	The RETRIEVE-FROM-STM operator is activated due to step 3 of task "Click the USB button" at Server D.
3196	The STORE-TO-STM operator is activated due to step 2 of task "Click the USB button" at Server D.
3196	Task information for "Click the USB button" (Step # 1) is entering Server D for processing.
3196	The saccade is completed at the Eyes server with the visual attention now on device "Device 1" (310, 330).
3186	A saccade is starting to be executed at the Eyes Server (Saccade distance: 2.5 degrees, execution time: 10 ms).
3130	A saccade motor action starts to be prepared at Server W (Target location on device "Device 1" at (310, 330)).
3129	The LOOK-AT operator is activated due to step 1 of task "Click the USB button" at Server D (Target visual attention on device "Device 1" at
3129	Task information for "Click the USB button" (Step # 6) is entering Server D for processing.
3129	The VISUAL-SEARCH operator is activated due to step 6 of task "Click the USB button" at Server D.
3129	Task information for "Click the USB button" (Step # 5) is entering Server D for processing.
3129	A mis-match is found by the COMPARE operator at Server F.
3101	The "text" feature (value: "Tape") is RETRIEVE(-ed)-FROM-STM at Server B (Phonological Loop).
3039	The "text" feature (value: "Tape") is starting to be STORE(-ed)-TO-STM at Server B (Phonological Loop).

Figure 54. A fraction of the simulation trace during a secondary task

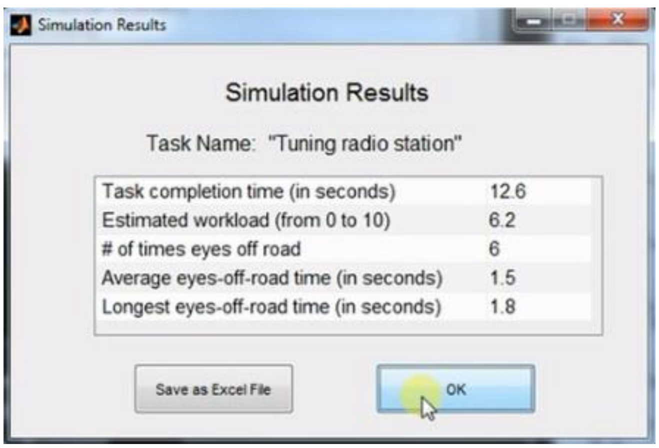


Figure 55. Simulation results for the user-defined non-driving task

Time (s)	Visual Atten	Visual Att	Visual Att	Right Han	Right Han	Right Han	Vehicle P	Vehicle P	Yaw Angle	Eyes Off R	Workload
5.7	1	263	725	1	168.4105	75.14615	0.078527	159.8818	0.151734	1	0.082892
5.75	1	263	725	1	168.4105	75.14615	0.078527	159.8818	0.151734	1	0.082892
5.8	1	263	725	1	168.4105	75.14615	0.078527	159.8818	0.151734	1	0.082892
5.85	1	263	725	1	168.4105	75.14615	0.078527	159.8818	0.151734	1	0.082892
5.9	1	263	725	1	168.4105	75.14615	0.078527	159.8818	0.151734	1	0.082892
5.95	1	263	725	1	168.4105	75.14615	0.078527	159.8818	0.151734	1	0.082892
6	1	263	725	1	168.4105	75.14615	0.120943	174.4165	0.182674	1	0.082892

Figure 56. A spreadsheet shows a fraction of the system states during a simulation run

7.4 Discussions

This chapter describes the development of a CAE software for human machine interface (HMI) designers to predict and benchmark the usability of in-vehicle infotainment systems. At the front end of the software a GUI was developed that allows HMI designers to create digital mockups of designs and setup the tasks for simulation. At the back end a digital driver was created for simulating the driver cognition and performance based on the cognitive architecture of QN-MHP. The software is able to simulate a driver performing in-vehicle secondary tasks (e.g., tuning radios) while steering a vehicle, make quantitative predictions of the driver's task performance and workload.

The software does not require the user to know any programming language to simulate tasks with specified designs, as all the simulation setup is conducted in the software's easy-to-use GUIs in four steps. The software also adopts the MATLAB GUIDE as the design prototyping tool, which allows the user to design UIs graphically with drag-and-drop and WYSIWYG (What You See Is What You Get). These features enable more UI designers to use the software as a usability testing tool with less training and reduced modeling time.

This work demonstrates the potential of the software as a useful usability testing tool for UI designers. We are exploring several further developments of the software in several directions. First, currently the software requires the user to be able to conduct a NGOMSL task analysis (described in detail in Liu, et al, 2006). This may require some initial training time to the software user, and the accuracy of the model outputs may be affected by the accuracy of the task analysis by the user. In the future we are planning to add "templates" for specifying typical in-vehicle operations (e.g., clicking a button with visual guidance) at a higher level than the GOMS's perceptual, cognitive, and motor level. So the users could use these built-in templates with higher level task components to create the tasks to be modeled. This will help reduce the training time of conducting task analysis, and reduce the potential errors and discrepancy during the task analysis.

Second, we are currently using the MATLAB GUIDE for prototyping the UI designs to be tested. In some applications, the designs are already implemented on some other UI languages (e.g., HTML, XML, QML). To test these UIs, the software user first needs to recreate the UIs using MATLAB GUIDE. In the future we may explore the techniques to support direct import of popular UI formats into the software, so that the UI designs created in other formats could be directly tested in their original formats. This would further reduce the time and workload for using this usability testing software.

Third, the software structure treats the driving task (box c in Figure 47) as a built-in extension module to the task-independent cognitive architecture of QN-MHP (box d in Figure 3). This structure allows the software developer to add other extension modules in the future if the needs arise. For example, a potential module for flying an airplane could be added by the software developer as another built-in task that users could specify and simulate. This flexibility makes it possible to extend this software from the usability testing of in-vehicle systems to systems in other domains with human-in-the-loop systems.

Fourth, a simple vehicle dynamics module was used (box b in Figure 47), and its outputs (the updated near/far angle values) are fed directly into the cognitive model, as if they are being "seen". In the future we may be able to connect the cognitive model with a driving simulator (the driver simulator would replace box b and part of box a in Figure 47). In this way we could ask the human subjects to performance the tasks in the same setting as the simulation, and compare the task performance of the human subjects and the cognitive model side-by-side. This would be useful to further calibrate and validate the model, and illustrate the usefulness of the software.

Chapter 8.
Conclusion

8.1 **Dissertation Summary**

In the automotive and some other domains (e.g., aviation, railway), it is a common task scenario when a human needs to interact with some device (e.g., finding an item on an electronic map or dialing a number on a phone) while simultaneously performing another continuous task (e.g., steering a vehicle). It is of great importance to understand the mechanisms of human multitasking behavior in order to design the task environments and user interfaces (UIs) that facilitate human performance and minimize potential safety hazards such as driver distractions.

Extensive research has been conducted to study the behavior and mechanism of human visual search task. But most of these studies considered visual search as the only task performed. Less research has been done to study visual search as one of several tasks that are performed simultaneously and the interference among the tasks. In this dissertation I investigated and modeled human multitask performance with a vehicle steering task and several typical in-vehicle secondary tasks.

Two experiments were conducted to investigate how various display designs and control modules affect the driver's eye glance behavior and performance. In the first experiment I employed experimental work that investigates how a driver’s eye glance behavior and task performance is affected during a visual search task by two basic design parameters of a touch screen device (number of buttons and their sizes). The findings from this study make contributions to the existing knowledge of how the human’s eye glance behaviors are affected by increased visual search difficulty (induced by a larger amount of buttons displayed on the screen) and increased reaching difficulty (induced by a smaller size of buttons) when performed simultaneously with a continuous vehicle steering task.

A computational model based on the cognitive architecture of Queuing Network-Model Human Processor (QN-MHP) was built to account for the findings from the experiments. A modeling mechanism for flexible task activation (rather than strict serial activations) was developed to allow task component activation to be based on the status of other task components. A task switching scheme was built to allow segmentations of tasks to model time-sharing nature of multitasking as observed from the experiment. These extensions offer new theoretical insights into visual search in multitask situations and enable the model to simulate parallel processing both within one task and among multiple tasks. The QN-MHP cognitive architecture was implemented in MATLAB/Simulink, which is a general-purpose numerical computing platform that is widely used in both academic and research institutions and industry.

In the second experiment I investigated the effect of three common control modules on the eye glance behavior, task performance for a typical real-world radio-tuning task during simulated driving. The findings from this study make contributions to the existing knowledge of how the human's eye glance behaviors, task performance, and workload are affected by different features of control modules (physical buttons vs. virtual buttons) and different input methods (pressing buttons vs. turning knobs). This experiment was modeled using the QN-MHP model with the multitasking features described above. The validation results show that the model could account for the observed behavior and performance differences from the empirical data.

Based on this model, a computer-aided engineering toolkit was developed to enable the UI designers of in-vehicle infotainment systems to evaluate, predict, and benchmark the usability of design concepts and prototypes. This toolkit supports the usability testing of the UI designs created using MATLAB's GUI design environment (GUIDE). At the front end of the software a GUI was developed that allows HMI designers to create digital mockups of designs and setup the tasks for simulation. At the back end a digital driver was created for simulating the driver cognition and performance based on the cognitive architecture of QN-MHP. From the engineering application and practical value perspective, the new toolkit has great advantages over the traditional usability testing methods with human subjects. It enables the UI designers to explore a larger design space and address usability issues at the early stages with lower cost both in time and manpower. This work was based on a generic cognitive architecture modeling approach that has the potential to be applicable to other multitasking domains.

8.2 **Conclusions**

- A new driver steering module was developed within the QN-MHP framework based on the latest findings in the driver behavior and modeling work. The results shows the model is able to capture the main steering behaviors in terms of the temporal characteristics of vehicle lateral position and steering wheel angle that were captured by other studies as well as our own driving simulation study.
- A simulated driving experiment was designed and conducted to examine the driver's eye glance behavior and task performance during a visual search task on a touch screen device. The results show that the two basic design parameters being tested, namely the number of buttons and their sizes, may affect the driver's eye glance behavior during the multitasking scenario. The results provide insights into the driver's eye glance strategy and the mechanism of task switching between a visual-based continuous task (e.g., steering a car) and a simple visual-motor self-terminating task.
- The QN-MHP model with the new driving module and has successfully modeled the visual search task during driving. The model is able to generate similar eye glance behaviors and task performance compared with the human data.
- To further validate the model's capability in modeling realist in-vehicle tasks, a simulated driving experiment was designed and conducted to examine the driver's eye glance behavior, task performance, and mental workload for a typical radio-tuning task. Compared with the first visual search task, the radio-tuning task requires multiple button clicking or knob turning, and the task completion time as a single task is much longer than the previous visual search task. The results show that the three typical control modules used for the task, namely physical buttons, virtual buttons, and physical knob, may induce different eye glance behavior, task performance, and mental workload. The results also provide insights into the driver's eye glance strategy and the mechanism of task switching between a visual-based continuous task (e.g., steering a car) and a realistic multi-step visual-motor self-terminating task.
- The QN-MHP model has successfully modeled the radio-tuning task during driving. The model is able to generate very similar eye glance behavior, task-switching behavior, task performance, and estimated mental workload compared with the collected human data.

- Based on the modeling work, a usability testing software package was developed for the UI designers of in-vehicle infotainment systems. The software features an easy-to-use graphical user interface for the software users to create digital mockups of the designs to be tested and setup the simulation. The software is able to make quantitative predictions of the driver's task performance and workload.

8.3 **Future Research**

8.3.1 Model the effect of learning

Currently the QN-MHP model only simulates the "expert" human who possesses perfect knowledge and skills needed to perform the specified task. In the future it would be beneficial to expand the cognitive servers of the QN-MHP to include a mechanism of learning. This would allow the model to predict the level of intuitiveness of an UI design from a prospective of a novice or experienced user. With this capability it would also make it possible to predict UI performances over the course of the ownership of a to-be-tested, which is very important but rarely done even with human subjects.

8.3.2 Multimodal interaction

Currently the QN-MHP model utilizes vision as the primary perceptual channel and generates basic motor actions including reaching an object, clicking a button, turning a knob, etc. In the future it would be beneficial to expand the model's perceptual and motor servers to support a wider range of multimodal interactions, following the trend of emerging in-vehicle infotainment systems. Examples include, but are not limited to, the use of sound/voice, 2-dimension or 3-demension gesture controls, haptic feedback, and proprioception (i.e., the sense of the position and movement of one's body parts, essential for modeling body actions without looking, e.g., touch-typing).

8.3.3 More realistic driver models

Currently the QN-MHP driver module includes only the steering component of the driving task. In the real world, however, driving is a lot more complicated, including not only the lateral control of steering, but also longitudinal control of pressing the gas or brake pedals (either with or without a leading vehicle), negotiating a curve, changing lanes, passing another vehicle, visual checking the road environment (e.g., traffic lights, potential obstacles), etc. Many individual models have been developed in the past decades for specific subtask or set of subtasks of driving. But it is still a challenge to incorporate these separate models into a unified driver model. Currently the QN-MHP driver module only represents a “typical” driver. There’s yet parameter to account for the behavior difference between different driver populations, for example, the novice and skilled driver, the conservative and aggressive driver, or the younger and older driver, etc. More work could be done in this regard to account for the variability of the driver population. Also the current model assumes that only the visual information is used as model inputs, while studies have found other sensory inputs may also play a role in the steering task. As mentioned earlier in Chapter 2, studies have found that the vestibular and kinesthetic channels add useful information to improve the driving performance. More work can be done to incorporate these sensory channels and their functions to the driver model. All these work requires a combination of effort from the field of both empirical study of driver behaviors and unified-theory based driving modeling. But it is strongly desired in order to further improve the accuracy and applicability of the driver model.

8.3.4 Model validation with a wider range of realistic secondary tasks

Currently the model is only validated using a few typical in-vehicle tasks under simplified driving conditions (e.g., steering on a straight road with no other vehicles). More validations would be desired in the future to further calibrate the model for a wider range of in-vehicle tasks and driving scenarios. For example, a potential testing case could be using a touch screen infotainment system to answer an incoming call in a car-following scenario. This work is essential for further improving the model applicability and credibility.

8.3.5 Further improvements on the CAE usability testing tool

The CAE usability testing software described in Chapter 7 could be improved in several ways. First, the currently software requires the user to conduct a NGOMSL-style task analysis which decomposes the task into QN-MHP level operators. This requires the software using to have some fairly good understanding of the NGOMSL task analysis method and familiarity to the QN-MHP operators. The accuracy of the model outputs may decrease if a less accurate task analysis is used. A potential way to solve this problem is to add “templates” for specifying typical in-vehicle operations (e.g., clicking a button with visual guidance) at a higher level than the QN-MHP operator level. The users could use these built-in templates to create the tasks to be modeled. And the software would compile the user task analysis by automatically decomposing the templates into the QN-MHP operators and inserting necessary operator so the cognitive model could use. This will help reduce the training time of conducting task analysis, and reduce the potential errors and discrepancy during the task analysis.

Second, currently the MATLAB GUIDE is used for prototyping the to-be-tested UI designs. In some applications, the designs are already implemented on some other UI languages (e.g., HTML, XML, QML). To test these UIs, the software user first needs to recreate the UIs using MATLAB GUIDE. In the future we may explore the techniques to support direct import of popular UI formats into the software, so that the UI designs created in other formats could be directly tested in their original formats. This would further reduce the time and workload for using this usability testing software.

8.3.6 Model human performance in other domains

Although the main application of the modeling work described in this dissertation is on the infotainment systems for automobiles. The structure of the model treats the driving task (box c in Figure 4) as a built-in extension module to the task-independent cognitive architecture of QN-MHP (box d in Figure 4). This structure allows the modeler to add other extension modules in the future if the needs arise. For example, a potential module for flying an airplane could be added by the software developer as another built-in task that users could specify and simulate. This flexibility makes it possible to extend this software from the usability testing of in-vehicle systems to systems in other domains with human-in-the-loop systems.

APPENDIX

Vehicle dynamics module (Rajamani, 2012)

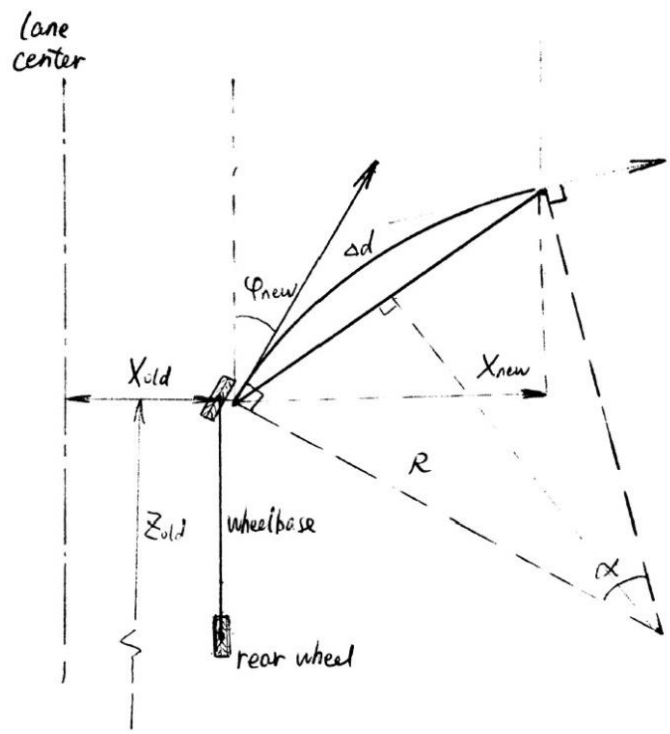


Figure 57. Vehicle dynamics module

$$\Delta d = speed * \Delta t$$

In which:

Δd is the distance traveled since the last cycle.

Δt is the time elapsed since the last cycle.

speed is the speed of the vehicle. It's currently set as a constant.

$$\varphi_{new} = \varphi_{old} + \Delta\varphi$$

In which:

φ_{old} and φ_{new} are the steering wheel position before and after the update.

$$R = \begin{cases} \dfrac{wheelbase}{\sin(\varphi_{new})} & if\ \varphi_{new} <> 0 \\ 1000{,}000 & if\ \varphi_{new} = 0 \quad (avoid\ being\ devided\ by\ zero) \end{cases}$$

In which:

R is the vehicle turning radius (negative value means turning to the left).

wheelbase is the distance between the centers of the front and rear wheels. It’s a constant for a given vehicle.

$$\alpha = \frac{\Delta d}{R}$$

In which:

α is the change of vehicle heading direction since the last cycle (in radian)

$$X_{new} = X_{old} + 2 * |R| * \sin\left(\frac{\alpha}{2}\right) * \sin(Yaw_{old} + \frac{\alpha}{2})$$

$$Z_{new} = Z_{old} + 2 * |R| * \sin\left(\frac{\alpha}{2}\right) * \cos(Yaw_{old} + \frac{\alpha}{2})$$

$$Yaw_{new} = Yaw_{old} + \alpha$$

In which:

X is the distance from the center of the vehicle to the lane center.

Z is the distance traveled from the origin point along the lane center.

Yaw: the angle between the direction of the vehicle heading and the direction of the lane center.

$$\theta_{near} = \operatorname{atan}\left(\frac{X_{new}}{nearDistance}\right) + Yaw_{new}$$

$$\theta_{far} = \operatorname{atan}\left(\frac{X_{new}}{farTHW * speed}\right) + Yaw_{new}$$

In which:

θ_{near} is the updated near angle.

θ_{far} is the updated far angle.

nearDistance is near point location (in distance) on the road, and set as a constant (10 m).

farTHW is the far point location (in time headway) on the road, and set as a constant (4 s).

BIBLIOGRAPHY

AAM (Alliance of Automobile Manufacturers). (2006). Statement of Principles, Criteria, and Verification Procedures on Driver Interactions with Advanced In-Vehicle Information and Communications Systems, Draft Version 2.1 (with updates), Driver Focus-Telematics Working Group.

Aasman, J. (1995). Modelling driver behaviour in Soar. Leidschendam, The Netherlands: KPN Research.

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., & Qin, Y. (2004). An integrated theory of the mind. *Psychological Review*, 111(4), 1036-1060.

Annunziato, M., & Fusi, S. (1998). Queuing theory for spike-driven synaptic dynamics. In L. Niklasson, M. Boden, & T. Ziemke (Eds.), *Proceedings of the 8th International Conference on Artificial Neural Networks*, Skovde, Sweden, Berlin: Springer-Verlag. 1, 117-122.

Bi, X., Li, Y., & Zhai, S. (2013), FFitts law: modeling finger touch with Fitts' law, *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 1363-1372.

Boyle, L.N., Lee, J.D., Peng, Y, Ghazizadeh, M., Wu, Y., Miller, E. and Jenness, J. (2013). Text reading and text input assessment in support of the NHTSA visual-manual driver distraction guidelines. Report No. DOT HS 811 820. Washington, D.C. NHTSA.

Byrne, M. D., & Anderson, J. R. (1998). Perception and action. In J. R. Anderson & C. Lebiere (Eds.), *The Atomic Components of Thought*. Mahwah, NJ: Lawrence Erlbaum. 167-200.

Cao, S., & Liu, Y. (2011a). Integrating Queueing Network and ACT-R cognitive architectures. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 55(1), 836-840.

Cao, S., & Liu, Y. (2011b). Mental workload modeling in an integrated cognitive architecture. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 55(1), 2083-2087.

Cao, S., & Liu, Y. (2012a). ACTR-QN: Integrating Queueing Network and ACT-R cognitive architectures. In W. D. Gray (Organizer), *Symposium on human performance modeling*. The 11th International Conference on Cognitive Modeling, Berlin, Germany.

Cao, S., & Liu, Y. (2012b). An integrated cognitive architecture for cognitive engineering applications. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 56, 323-327.

Cao, S., & Liu, Y. (2012c). QN-ACTR modeling of multitask performance of dynamic and complex cognitive tasks. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 56, 1015-1019.

Cao, S., & Liu, Y. (2013a). Queueing network-adaptive control of thought rational (QN-ACTR): An integrated cognitive architecture for modelling complex cognitive and multi-task performance. *International Journal of Human Factors Modelling and Simulation*, 4(1), 63-86.

Cao, S., & Liu, Y. (2013b). Concurrent processing of vehicle lane keeping and speech comprehension tasks. *Accident Analysis and Prevention*, 59, 46-54.

Cao, S., & Liu, Y. (2013c). Queueing Network-ACTR modeling of concurrent tasks involving multiple controlled processes. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 57, 768-772.

Card, S., Moran, T. P., & Newell, A. (1983). *The Psychology of Human-Computer Interaction*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Chaffin, D. B., (2007), Human Motion Simulation for Vehicle and Workplace Design, *Human Factors and Ergonomics in Manufacturing*, Vol. 17 (5) 475 C484

Donges, E. (1978). A two-level model of driver steering behavior. *Human Factors*, 20, 691-707.

Feyen, R. (2002). *Modeling human performance using the queuing network-model human processor (QN-MHP)*. Doctoral Dissertation. Department of Industrial and Operations Engineering. University of Michigan.

Feyen, R. & Liu, Y. (2001). Modeling task performance using the Queuing Network-Model Human Processor (QN-MHP). *Proceedings of the Fourth International Conference on Cognitive Modeling*. E. M. Altmann, A. Cleermans, C. D. Schunn, and W. D. Gray Eds. Lawrence Erlbaum Associates, Mahwah, NJ.

Feyen, R., Liu, Y., Chaffin, D., Jimmerson, G., & Joseph, B. (2000). Computer aided ergonomics: a case study of incorporating ergonomics analysis in workplace design, *Applied Ergonomics*, 31, 291-300.

Feng, F., & Liu, Y. (2013). Computational modeling of feature and conjunction visual search tasks using Queuing Network-Model Human Processor (QN-MHP), *2nd International Digital Human Modeling Symposium Proceedings*.

Feng, F., Liu, Y., Chen, Y., Filev, D., & To, C. (2014) Computer-aided usability evaluation of in-vehicle infotainment systems, *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 58, 2285-2289.

Freed, M. A., & Remington, R. (2000). Making human-machine system simulation a practical engineering tool: an APEX overview. *Proceedings of the 3rd International Conference on Cognitive Modeling*, Veenendaal (NL).

Fuller, H. J. A., Reed, M. P., & Liu, Y. (2010). Integrating physical and cognitive human models to represent driving behavior. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 54, 1042-1046.

Fuller, H. J. A., Reed, M. P., & Liu, Y. (2012). Integration of physical and cognitive human models to simulate driving with a secondary in-vehicle task. *IEEE Transactions on Intelligent Transportation Systems*, 13(2), 967-972.

Gil, G.-H. (2010). *An accessible cognitive modeling tool for evaluation of human-automation interaction in the systems design process*. Doctoral Dissertation. North Carolina State University.

Green, P. (1999). Estimating compliance with the 15-second rule for driver interface usability and safety. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*. 43(1), 987-991

Greenberg, J., Artz, B., & Cathey, L. (2003). The effect of lateral motion cues during simulated driving. *Proceeding of the Driving Simulator Conference North America*, Dearborn, MI

Hart, S. G. (2006). NASA-task load index (NASA-TLX): 20 years later. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 50, 904-908.

Hart, S. G. & Staveland, L. E. (1988) Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. In P. A. Hancock and N. Meshkati (Eds.) *Human Mental Workload*. Amsterdam: North Holland Press.

Horowitz, T. S., & Wolfe, J. M. (2005). Visual search: the role of memory for rejected distractors. In L. Itti, G. Rees, & J. Tsotsos (Eds.), *Neurobiology of attention*. San Diego, CA: Academic Press/Elsevier. 264-268.

Horrey, W., & Wickens, C. D. (2004). Driving and side task performance: the effects of display clutter, separation, and modality. *Human Factors*, 46(4), 611-24.

Horrey, W. J., & Wickens, C. D. (2007), In-vehicle glance duration distributions, tails, and model of crash risk. *Transportation Research Record*, 2018, 22-28.

Hudson, S. E., John, B. E., Knudsen, K., & Byrne, M. D. (1999). A tool for creating predictive performance models from user interface demonstrations. *Proceedings of the ACM Symposium on User Interface Software and Technology*, 93-102.

Jin, Z. X., Plocher, T., & Kiff, L. (2007). Touch screen user interfaces for older adults: button size and spacing, *Proceedings of the 4th International Conference on Universal Access in Human Computer Interaction: Coping with Diversity*, 933-941.

John, B. E., & Gray, W. D., (1995). CPM-GOMS: An analysis method for tasks with parallel activities, *Conference Companion on Human Factors in Computing Systems: CHI 1995*, 393-394.

John, B. E., & Kieras, D. E. (1996). The GOMS family of user interface analysis techniques: comparison and contrast. *ACM Transactions on Computer-Human Interaction*, 3(4), 320-351.

John, B. E., Prevas, K., Salvucci, D. D., & Koedinger, K. (2004). Predictive human performance modeling made easy. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 455-462

John, B., Vera, A., Matessa, M., Freed, M., & Remington, R. (2002). Automating CPM-GOMS. In Human Factors in Computing Systems: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 147-154.

Kieras, D. E. (1996). A Guide to GOMS model usability evaluation using NGOMSL. In M. Helander, T. Landauer, and P. Prabhu (Eds.), *Handbook of Human-Computer Interaction*. (Second Edition). Amsterdam: North-Holland. 733-766.

Kieras, D. & Meyer, D.E. (1997). An overview of the EPIC architecture for cognition and performance with application to human-computer interaction. *Human-Computer Interaction*, 12(4), 391-438.

Kieras, D.E., Meyer, D.E., Ballas, J.A., & Lauber, E.J. (2000). Modern computational perspectives on executive mental processes and cognitive control. Where to from here? In S. Monsell and J. Driver (Eds.), *Control of cognitive processes: Attention and Performance XVIII*. Cambridge, MA: MIT Press. 618-712

Kieras, D. E., Wood, S. D., Abotel, K., & Hornof, A. (1995). GLEAN: A computer-based tool for rapid GOMS model usability evaluation of user interface designs. *International Journal of Man-Machine Studies*, 22, 365-394.

Klauer, S.G., Dingus, T. A., Neale, V. L., Sudweeks, J.D., & Ramsey, D.J. (2006). The impact of driver inattention on near-crash/crash risk: an analysis using the 100-car naturalistic driving study data. (Report No. DOT HS 810 594). Washington, DC: National Highway Traffic Safety Administration.

Klauer, S.G. Guo, F., Sudweeks, J. & Dingus, T.A. (2010). An analysis of driver inattention using a case-crossover approach on 100-car data: final report, DOT HS 811 334.

Kristjansson, A. (2000). In search of rememberance: evidence for memory in visual search. Psychological Science, 11(4), 328–332.

Laird, J. E., Newell, A., & Rosenbloom, P. S. (1987). Soar: an architecture for general intelligence. *Artificial Intelligence*, 33(1), 1-64.

Land, M. F., & Horwood, J. (1995). Which parts of the road guide steering? *Nature*, 377, 339-340.

Lansdown, T. C., Brook-Carter, N., & Kersloot, T., (2004). Distraction from multiple in-vehicle secondary tasks: vehicle performance and mental workload implications. *Ergonomics*, 47(1), 91-104.

Lim, J., & Liu, Y. (2004a). A queueing network model of menu selection and visual search. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 47, 1846-1850.

Lim, J., & Liu, Y. (2004b). A queueing network model for eye movement. *Proceedings of the Sixth International Conference on Cognitive Modeling*, 154-159.

Lim J., & Liu, Y. (2009). Reinforcement learning in eye movement: Modeling the cyclic influences of top-down and bottom-up processes, *IEEE Transactions on Systems, Man, and Cybernetics*, 39(4), 706-714.

Lim, J., Tsimhoni, O., & Liu, Y. (2010). Investigation of driver performance with night vision and pedestrian detection systems, Part 2: Queueing network performance modeling, *IEEE Transactions on Intelligent Transportation Systems*, 11(4), 706-714.

Liu, Y. (1996a). Queueing network modeling of elementary mental processes. *Psychological Review*, 103, 116-136.

Liu, Y. (1996b). Quantitative assessment of effects of visual scanning on concurrent task performance, *Ergonomics*, 39, 382-399.

Liu, Y. (1996c). Interactions between memory scanning and visual scanning in process monitoring, *Ergonomics*, 39, 1038-1053.

Liu, Y. (1997). Queueing network modeling of human performance of concurrent spatial and verbal tasks. *IEEE Transactions on Systems, Man, Cybernetics*, 27, 195-207.

Liu, Y. (2009). QN-ACES: Integrating queueing network and ACT-R, CAPS, EPIC, and Soar architectures for multitask cognitive modeling. *International Journal of Human-Computer Interaction*, 25(6), 554-581.

Liu, Y., Feyen, R., & Tsimhoni, O. (2006). Queueing Network-Model Human Processor (QN-MHP): a computational architecture for multitask performance in human-machine systems. *ACM Transactions on Computer-Human Interaction*, 13(1), 37-70.

Liu, Y., & Wickens, C. (1992). Visual scanning with or without spatial uncertainty and selective and divided attention, *Acta Psychologica*, 79(2), 131-153.

MacAdam, C. C. (2003). Understanding and modeling the human driver. *Vehicle System Dynamics*, 40(1-3), 101-134.

MacKenzie, I.S., Buxton, W., (1992) Extending Fitts' law to two-dimensional tasks. *Proceedings of the ACM Conference on Human Factors in Computing Systems - CHI 1992 ACM*, 219-226

Mattia, M., & Del Giudice, P. (2000). Efficient event-driven simulation of large networks of spiking neurons and dynamical synapses. *Neural Computation*, 12, 2305-2329.

McLane, R. C., & Wierwille, W. W., (1975) The Influence of Motion and Audio Cues on Driver Performance in an Automobile Simulator, *Human Factors*, 17(5), 488-501.

Mehler, B., Reimer, B. & Dobres, J. (2012). An evaluation of the visual demands of portable telematics technologies among young adult drivers, *MIT Age Lab White Paper*.

Meyer, D. E., & Kieras, D. E. (1997). A computational theory of executive cognitive processes and multiple-task performance: part 1. basic mechanisms. *Psychological Review*, 104(1), 3-65.

Neisser, U., Novick, R., & Lazar, R. (1964). Searching for novel targets. *Perceptual and Motor Skills*, 19, 427-432.

Newell, A. (1973). *You can't play 20 questions with nature and win: Projective comments on the papers of this symposium*. New York: Academic Press.

Newell, A. (1990). *Unified Theories of Cognition*. Cambridge, MA: Harvard University Press.

NHTSA. (2010). Visual-manual NHTSA driver distraction guidelines for in-vehicle electronic devices, docket no. NHTSA-2010-0053. Washington, DC: National Highway Traffic Safety Administration.

NHTSA, (2013). 2012 motor vehicle crashes: overview, *Traffic Safety Facts - Research Note*, (Report No. DOT HS 811 856). Washington, DC: National Highway Traffic Safety Administration.

Nothdurft, H. (2000). Salience from feature contrast: variations with texture density, *Vision Research*, 40(23), 3181–3200.

Nyamsuren, E., & Taatgen, N. A. (2013). Pre-attentive and attentive vision module. *Cognitive Systems Research*, 24, 62-71.

Peng, Y., Boyle L. N., & Hallmark S. L. (2013). Driver's lane keeping ability with eyes off road: insights from a naturalistic study. *Accident Analysis Prevention*, 50, 628–634.

Perez, M., Owens, J., Viita, D., Angell, A., Ranney, T. A., Baldwin, G. H. S., Parmer, E., Martin, J., Garrott, W. R., & Mazzae, E. N. (2013). Radio tuning effects on visual and driving performance - Simulator and test track studies. (Report No. DOT HS 811 781). Washington, DC: National Highway Traffic Safety Administration.

Rajamani, R. (2012). Lateral Vehicle Dynamics, *Vehicle Dynamics and Control*, 15-46.

Ranney, T. A., Mazzae, E., Garrott, R., & Goodman, M. (2000). NHTSA driver distraction research: Past, present, and future, National Highway Traffic Safety Administration.

Reed, M., Faraway, J., Chaffin, D. B., and Martin, B. (2006). The HUMOSIM Ergonomics Framework: A New Approach to Digital Human Simulation for Ergonomic Analysis, *Proceedings of the 2006 SAE Digital Human Modeling Conference*.

Reed-Jones, J.G., Trick, L.M., & Matthews, M. (2008). Testing assumptions implicit in the use of the 15-second rule as an early predictor of whether an in-vehicle device produces unacceptable distraction, *Accident Analysis and Prevention*, 40 (2), 628–634.

Roland, P. E., Larsen, B., Lassen, N. A., & Skinhøj, E. (1980). Supplementary motor area and other cortical areas in organization of voluntary movements in man. *Journal of Neurophysiology*, 43(1), 118-136.

SAE J2364 Feb. (2004). Navigation and Route Guidance Function Accessibility While Driving.

SAE J2365 Apr. (2002). Calculation of the Time to Complete In-Vehicle Navigation and Route Guidance Tasks.

SAE J2396 Jul. (2000). Definitions and Experimental Measures Related to the Specification of Driver Visual Behavior Using Video Based Techniques.

Salvucci D. D. (2001). An integrated model of eye movements and visual encoding. *Cognitive Systems Research*, 1, 201-220.

Salvucci, D. D. (2005). A multitasking general executive for compound continuous tasks. *Cognitive Science*, 29, 457-492.

Salvucci, D.D. (2006). Modeling driver behavior in a cognitive architecture. *Human Factors*, 48(2), 362-80.

Salvucci, D. D. (2009). Rapid prototyping and evaluation of in-vehicle interfaces. *ACM Transactions on Computer-Human Interaction*, 16(2), 9: 1-33.

Salvucci, D. D., & Lee, F. J. (2003). Simple cognitive modeling in a complex cognitive architecture, *CHI 2003 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 265-272

Salvucci, D. D., & Gray, R. (2004). A two-point visual control model of steering. *Perception*, 33(10), 1233-1248.

Salvucci, D. D., & Taatgen, N. A. (2008). Threaded cognition: An integrated theory of concurrent multitasking. *Psychological Review*, 115(1), 101–130.

Salvucci, D. D., Zuber, M., Beregovaia, E., & Markley, D. (2005). Distract-R: Rapid prototyping and evaluation of in-vehicle interfaces. *In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: CHI 2005*, 581-589.

Sebok, A., Wickens, C., Sarter, N., Quesada, S., Socash, C., & Anthony, B. (2012). The Automation Design Advisor Tool (ADAT): Development and validation of a model-based tool to support flight deck automation design for NextGen operations. *Human Factors and Ergonomics in Manufacturing and Service Industries,* 22(5), 378-394.

Sivak, M. (1998). The information that drivers use: Is it indeed 90 percent visual? *The UMTRI Research Review*, 29, 1.

Steelman K, McCarley J, & Wickens C, (2009). N-SEEV A Computational Model of Attention and Noticing, *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 53, 774-778

Sun, R. (2003). A Tutorial on CLARION 5.0. Technical Report, Cognitive Science Department, Rensselaer Polytechnic Institute. http://www.cogsci.rpi.edu/~rsun/sun.tutorial.pdf

Tijerina, L., Barickman, F. S., & Mazzae, E. N. (2004). Driver eye glance behavior during car following, DOT HS 809 723. Washington, D.C. NHTSA.

Tivesten, E., & Dozza, M. (2014). Driving context and visual-manual phone tasks influence glance behavior in naturalistic driving, *Transportation Research Part F: Traffic Psychology and Behaviour*, 26, part A, 258-272.

Treisman, A., & Gelade, G. (1980). A feature-integration theory of attention, *Cognitive Psychology*, 12, 97-136.

Tsimhoni, O., & Green, P. (2001). Visual demand of driving and time execution of display intensive in-vehicle tasks. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 45, 1586-1590.

Tsimhoni, O., & Liu. Y. (2003), Modeling steering using the Queueing Network-Model Human Processor (QN-MHP), *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 47, 1875-1879.

Tsimhoni, O., Smith, D. & Green, P. (2004). Address entry while driving: Speech recognition versus a touch-screen keyboard. *Human Factors*, 46(4), 600-610.

Victor, T. W., Harbluk, J. L., & Engström, J. A. (2005). Sensitivity of eye-movement measures to in-vehicle task difficulty. *Transportation Research Part F: Traffic Psychology and Behaviour*, 8, 167-190.

Vlaskamp, B. N., & Hooge, I. (2006). Crowding degrades saccadic search performance. *Vision Research*. 46 (3), 417-25.

Weir, D. & McRuer, D. (1973). Measurement and Interpretation of Driver/Vehicle System Dynamic Response. *Human Factors*, 15, 367-378.

Wickens, C. D. (2002). Multiple resources and performance prediction. *Theoretical Issues in Ergonomics Science*, 3(2), 159-177.

Wickens, C. D. (2008). Multiple resources and mental workload. *Human Factors*, 50, 449–455.

Wierwille, W. W. (1993a). An initial model of visual sampling of in-car displays and controls. In A. G. Gale, I. D. Brown, C. M. Haslegrave, H. W. Kruysse, & S. P. Taylor (Eds.), *Vision in Vehicles* 4. Amsterdam: North-Holland. 271-282.

Wierwille, W. W. (1993b). Visual and manual demands of in-car controls and displays. In B. Peacock & W. Karwowsk (Eds.), *Automotive Ergonomics*, London: Taylor and Francis, 299-320.

Wierwille, W. W., & Tijerina, L. (1998). Modelling the relationship between driver in-vehicle visual demands and accident occurrence, *Vision in vehicles - VI*, 233-244.

Wolfe, J. (1994). Guided search 2.0 A revised model of visual search, *Psychonomic Bulletin & Review*, 1(2), 202-238.

Wolfe, J. (2007). Guided Search 4.0: Current progress with a model of visual search. In W. Gray (Ed.), *Integrated Models of Cognitive Systems*. New York: Oxford, 99-119.

Wu, C. (2007). *Queueing network modeling of human performance and mental workload in perceptual-motor tasks*. Doctoral Dissertation. Department of Industrial and Operations Engineering. University of Michigan.

Wu, C., & Liu, Y. (2007a). Queueing network modeling of driver workload and performance. *IEEE Transactions on Intelligent Transportation Systems*, 8(3), 528-537.

Wu, C., & Liu, Y. (2007b). Interface makeover of a software tool for modeling human performance and workload. *Ergonomics in Design*, 15(2), 8-14.

Wu, C., & Liu, Y. (2008a). Queuing network modeling of the psychological refractory period (PRP). *Psychological Review*, 115(4), 913-954.

Wu, C., & Liu, Y. (2008b). Queuing network modeling of transcription typing, *ACM Transactions on Computer Human Interaction*, 15(1), 1-45.

Wu, C., & Liu, Y. (2009). Development and evaluation of a software package for predicting human performance and workload in human-machine interface design and evaluation. *Computers and Industrial Engineering*, 56(1), 323-333.

Wu, C., Liu, Y., & Quinn-Walsh, C. M. (2008). Queuing network modeling of a real-time psychophysiological index of mental workload - P300 in event-related potential (ERP). *IEEE Transactions on systems, Man, and Cybernetics-Part A: Systems and Humans*, 38(5), 1068-1084.