

# **PREDICTIVE MODELS OF PROCEDURAL HUMAN SUPERVISORY CONTROL BEHAVIOR**

by

**YVES BOUSSEMART**

Bachelor of Computer Engineering, McGill University, 2002  
Master of Engineering, McGill University, 2005

Submitted to the Engineering Systems Division  
in partial fulfillment of the requirements for the degree of

**DOCTOR OF PHILOSOPHY**

at the

**MASSACHUSETTS INSTITUTE OF TECHNOLOGY**

January 2011

©2011 Yves Boussemart. All rights reserved.

The author hereby grants to MIT permission to reproduce and to distribute publicly paper and electronic copies of this thesis document in whole or part in any medium known or hereafter created.

Signature of Author: \_\_\_\_\_

Yves Boussemart  
Engineering Systems Division  
January 31<sup>st</sup>, 2011

Certified by: \_\_\_\_\_

M. L. Cummings  
Associate Professor of Aeronautics and Astronautics and Engineering Systems  
Thesis Supervisor

Certified by: \_\_\_\_\_

Nicholas Roy  
Associate Professor of Aeronautics and Astronautics  
Thesis Supervisor

Certified by: \_\_\_\_\_

Daniel Frey  
Associate Professor of Mechanical Engineering and Engineering Systems  
Thesis Supervisor

Accepted by: \_\_\_\_\_

Nancy Leveson  
Professor of Aeronautics and Astronautics and Engineering Systems  
Chair, Engineering Systems Division Education Committee

*[Page intentionally left blank]*

# **PREDICTIVE MODELS OF PROCEDURAL HUMAN SUPERVISORY CONTROL BEHAVIOR**

by

Yves Boussemart

Submitted to the Engineering Systems Division on 31/01/2011 in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Engineering Systems

## **ABSTRACT**

Human supervisory control systems are characterized by the computer-mediated nature of the interactions between one or more operators and a given task. Nuclear power plants, air traffic management and unmanned vehicles operations are examples of such systems. In this context, the role of the operators is typically highly proceduralized due to the time and mission-critical nature of the tasks. Therefore, the ability to continuously monitor operator behavior so as to detect and predict anomalous situations is a critical safeguard for proper system operation. In particular, such models can help support the decision making process of a supervisor of a team of operators by providing alerts when likely anomalous behaviors are detected.

By exploiting the operator behavioral patterns which are typically reinforced through standard operating procedures, this thesis proposes a methodology that uses statistical learning techniques in order to detect and predict anomalous operator conditions. More specifically, the proposed methodology relies on hidden Markov models (HMMs) and hidden semi-Markov models (HSMMs) to generate predictive models of unmanned vehicle systems operators. Through the exploration of the resulting HMMs in two distinct single operator scenarios, the methodology presented in this thesis is validated and shown to provide models capable of reliably predicting operator behavior. In addition, the use of HSMMs on the same data scenarios provides the temporal component of the predictions missing from the HMMs. The final step of this work is to examine how the proposed methodology scales to more complex scenarios involving teams of operators. Adopting a holistic team modeling approach, both HMMs and HSMMs are learned based on two team-based data sets. The results show that the HSMMs can provide valuable timing information in the single operator case, whereas HMMs tend to be more robust to increased team complexity. In addition, this thesis discusses the methodological and practical limitations of the proposed approach notably in terms of input data requirements and model complexity.

This thesis thus provides theoretical and practical contributions by exploring the validity of using statistical models of operators as the basis for detecting and predicting anomalous conditions.

Thesis Supervisor: M. L. Cummings

Title: Associate Professor of Aeronautics and Astronautics and Engineering Systems Division

*[Page intentionally left blank]*

# ACKNOWLEDGEMENTS

It would be impossible for me not to start this section by thanking my advisor, Prof. Missy Cummings. Missy, as hackneyed as it sounds, my Ph.D. would have been a sisyphian task without your guidance, advice, and, occasionally, much needed “motivational speeches<sup>1</sup>”. I am truly grateful that you taught me the ropes of academia and research, but the most valuable things I’ve received from you were life-skills: how manage people and how to have impact when delivering information, among so many others, will stay with me forever.

In addition, I also want to recognize my committee, Prof. Nicholas Roy and Prof. Dan Frey. For your help, support and advice throughout my (sometimes meandering) Ph.D. process, thank you. Ryan Castonia and Hank Huang, I would never have finished this work without leaning heavily on your Master thesis accomplishments; I feel deeply privileged to have collaborated closely with you both. To my UROPs who did the majority of the grunt-work in this thesis: Ned Twigg, Jonathan Las Fargeas and Scott Bezek, thank you for the long hours, the innumerable lines of code and the never-ending number crunching runs; thank you Lindley Graham for dealing with the eye-tracker (sorry). I feel incredibly fortunate I was able to lean on such brilliant and promising minds. Finally, to the international members of my academic family: Prof. Kristina Lundqvist in Sweden, Prof. Gilles Coppin in France, Prof. Duncan Campbell in Australia and Prof. Axel Schulte in Germany: they say a mind stretched by an idea never goes back to its original size, thank you for giving me lots of them.

On a more personal note, I would never have made it this far without the incredible love and support of a number of people. First, to my parents, no words can express how thankful I am for all you’ve given me. More than anyone, Mom, Dad, you defined the core of who I am. Of course, the rest of the family, grandparents, uncles, aunts and cousins spread all over the world between in France, Canada and the US, thank you for steadily believing in me during this long endeavor. To the people I consider an extension of my family, Seb, Jill & Tristan Gorelov, Giacomo & Alexa Corbo, Vikas Sharma, Fabrice Turcq, Darius Camp, David Twose and Dani Nascimento, you guys are the people I look up to and inspire me to make the very best of every day.

The people I met at MIT all share one thing: uncommon in common. To my fellow ESD’ers, current and soon-to-be Doctors, Mike Cardin, Sid Rupani, Katherine Dykes, Mat Silver, Dan Livengood, Erica Gralla, Lara Pierpoint, Dave Keith, Arzum Akkas, Philippe Bonnefoy, Rob Nicol, Mary-Beth Mills-Curran and so many others, it has been wonderful to share ideas, drinks, good times and hard times. My CrossFit buddies, Geoff Carrigan (it had to start with you), Dan McEachran, Andrew Benedick, Niek Beckers, Seb Dango, Cody Fleming, Florian Naegele, you guys made me dread 3-2-1 countdowns and girl names. Doing something that sucks everyday changed my life in a profound way, and I’m grateful for the large part of the suck that came via Coach Betty-Lou McClanahan and swim buddy Marie-Eve Rancourt. Merci aussi to the Francophone crew, Stephane Chong, Laure-Anne Ventouras, Mikael Bikard for bringing a little bit of France right here in Cambridge. To my fellow SCUBA instructors Roeland & Madeline Jaspers and Conan Campbell, I’m looking forward to meet and dive with you in your respective corners of the world. To my PADI buddies, Laurent Guerin, Noemie Chocat, Jeremie Bertaud, David Bergeron, Mark Avnet, call me whenever you feel like breathing some compressed air.

The Human and Automation Lab has been my home during all my Ph.D., and as my closest colleagues, I want to express my gratitude to HALiens past and present. To the Post-Docs, Stacey Scott, Mark Ashdown, Jake Crandall, Birsen Donmez, Luca Bertucelli and Kris Thornburg: I now fully appreciate the

---

<sup>1</sup> Yes, Missy, you were right that we did grow “old” together, but no, Missy, Emma did not finish her Ph.D. before me ☺

path you took and your continued success is an inspiration. To my fellow Ph.D.'s, Carl Nehme, Brian & Tuco Mekdeci, Farzan Sasangohar, Yale Song, Fei Gao and Jason Ryan: we are all going or went through the same gauntlet, keep up the good work and you too will soon see the light at the end of the tunnel. To the Masters students, Jason Rathje, Christin Hart, Jackie Tappan, Dave Pitman, Kim Jackson, Mariela Buchin, Anna Massie, Hudson Graham, Amy Brzesinski and Patricia Pina: working with you all on a day-to-day basis has been an absolute pleasure. A regular stream of Delft students also haunted the lab, Mark "Chewie" Duppen, Mark Visser, Pierre Maere and Luisa dos Santos Buinhas, all you definitely helped keep things interesting in HAL.

To the admins, Beth Milnes, Sally Chapman and Katherine Fischer, you somehow managed to abstract away most of MIT's notorious administrivia and things just wouldn't work without you guys. Thank you.

Finally, I want to express my gratitude to Boeing Research and Technology and the Office of Naval Research who sponsored this current research. Also, thanks to Air Force Research Lab – Human Effectiveness Directorate for providing the basis for a valuable collaboration.

*January 31, 2011, Cambridge MA, USA*  
*- Yves Boussemart*

# TABLE OF CONTENTS

Abstract .....	3
Acknowledgements .....	5
Table of Contents .....	7
List of Figures .....	11
List of Tables .....	13
List of Main Acronyms .....	14
Chapter 1 Introduction .....	15
1.1 Research Approach .....	18
1.2 Research Questions .....	19
1.3 Thesis Organization .....	19
Chapter 2 Literature Review .....	21
2.1 Procedural Human Supervisory Control .....	21
2.1.1 Procedures and Human Supervisor Control .....	22
2.1.2 Team Supervisory Control .....	23
2.1.3 Monitoring Human Supervisory Control Behaviors .....	24
2.2 Computational Models of Human Behavior .....	26
2.2.1 Important Characteristics of Modeling Techniques in PHSC settings .....	26
2.2.2 Deductive Models .....	27
2.2.3 Statistical Models .....	28
Artificial Neural Networks .....	29
Support Vector Machines .....	31
Auto-Regressive Moving Averages .....	34
Hidden Markov Models and Hidden Semi-Markov Models .....	36
Summary of the Methodologies .....	38
2.3 Hidden Markov Models .....	39
2.3.1 Formal definition .....	39
Computational Issues .....	40
2.3.2 Hidden Semi-Markov Models .....	44
Computational Issues .....	45
Learning the Model Parameters .....	46
2.4 Chapter Summary .....	47
Chapter 3 HMMs of Single PHSC Operators .....	49

3.1	Operator Models in a Static Environments .....	49
3.1.1	StrikeView Interface Description.....	49
3.1.2	Learning HMMs from PSCH Data .....	51
3.1.3	Grammatical Phase .....	51
3.1.4	Statistical Phase.....	52
	Model Learning and Selection .....	53
3.1.5	StrikeView Models .....	55
3.1.6	Model Validation .....	57
3.1.7	Performance Evaluation.....	58
3.2	Operator Models in Dynamic Environments .....	60
3.2.1	RESCHU Interface Description .....	60
3.2.2	RESCHU Grammar.....	61
3.2.3	RESCHU Models.....	62
3.2.4	RESCHU Models Validation .....	64
3.2.5	RESCHU Performance Evaluation .....	64
3.3	Chapter Summary .....	65
Chapter 4	Modeling a Single Operator Through HSMMs .....	67
4.1	Learning HSMMs for PHSC Data .....	67
4.1.1	HSMM Complexity Analysis.....	67
4.1.2	Sojourn Distributions as Gaussian Mixture Models .....	68
4.1.3	HSMM Learning Process.....	70
4.2	HSMM of RESCHU .....	71
4.2.1	Model Selection .....	71
4.2.2	Selected Model.....	72
4.2.3	Model Validation .....	74
4.2.4	Model Evaluation and MAS .....	75
	MAS Metric .....	75
	MAS Sensitivity.....	77
4.3	Chapter Summary .....	80

Chapter 5	Team Models of PHSC operators .....	83
5.1	Modeling Approach .....	83
5.2	Team-RESCHU .....	84
5.2.1	Team-RESCHU Grammar .....	85
5.2.2	Experimental Subjects.....	85
5.2.3	Team-RESCHU Models .....	85
5.3	AFRL Data Set.....	87
5.3.1	Team Structure and Roles .....	87
5.3.2	Communications .....	89
5.3.3	AFRL Grammar .....	89
5.3.4	Experimental Subjects.....	91
5.3.5	AFRL Models .....	91
5.4	Comparing Single Operator and Team Models .....	92
5.5	Chapter Summary .....	97
Chapter 6	Conclusions.....	99
6.1	Contributions.....	100
6.1.1	Applications .....	102
	Real-time supervisor decision support tool.....	104
6.2	Limitations .....	106
6.2.1	Training Data .....	106
6.2.2	User Interface Input Requirement.....	106
6.2.3	Grammar Construction.....	107
6.2.4	Visualization Complexity .....	107
6.2.5	Model Complexity .....	107
6.3	Future Work.....	109
6.4	Thesis Summary.....	110
Appendix A	Assumption Validations.....	111
A.1	Data Sufficiency.....	111
A.1.1	Eye Tracking and Behavioral Models.....	112
A.1.2	Experimental Procedure .....	113

A.1.3 Eye-tracking data processing .....	113
A.1.4 Modeling Results .....	115
A.2 First-Order Model Assumption .....	120
A.2.1 Markov Assumption and HMMs .....	120
A.2.2 Learning higher-order HMMs .....	122
A.2.3 Complexity analysis .....	127
A.2.4 Results .....	127
A.3 Learning Methodology .....	128
A.3.1 Classic Supervised Learning .....	129
A.3.2 Smooth Supervised Learning .....	130
A.3.3 Results .....	131
A.4 Summary .....	137
References .....	138

# LIST OF FIGURES

Figure 1.1 Human Supervisory Control Loop (adapted from (Sheridan, 1992))	15
Figure 1.2 Team of multi-UV operators and team supervisor	18
Figure 2.1 Graphical representation of an artificial neural network	30
Figure 2.2 SVM are maximum margin hyperplanes (Cyc, 2008)	32
Figure 2.3 Kernel trick for non-linearly separable data in 2D (Niissalo, 2010)	33
Figure 2.4: A Three-state Hidden Markov Model.	40
Figure 2.5 Progression through the lattice of hidden states	41
Figure 2.6: A 3-state hidden semi-Markov model	44
Figure 3.1. The StrikeView interface	50
Figure 3.2 Two-stage learning for HMMs of PSCH behavior from experimental data	51
Figure 3.3 HMM Learning Process	53
Figure 3.4 Model fit vs. model complexity	54
Figure 3.5 BIC scores for StrikeView models	55
Figure 3.6 5-state HMM for StrikeView	56
Figure 3.7 Model validation for StrikeView	57
Figure 3.8 Predictive performance for the StrikeView HMMs	59
Figure 3.9 Cumulative prediction rate for StrikeView	59
Figure 3.10: The RESCHU interface	60
Figure 3.11 BIC score for the RESCHU model	62
Figure 3.12 8-state HMM for RESCHU	63
Figure 3.13 Model validation for RESCHU	64
Figure 3.14 Predictive performance for RESCHU HMMs	65
Figure 4.1 Example of a bimodal Gaussian mixture model	69
Figure 4.2 HSMM Learning Process	70
Figure 4.3 BIC scores (lower is better) for the HSMMs and GMM-HSMM of different sizes	71
Figure 4.4 Transition probabilities in the 5-state 1-mode GMM-HSMM	72
Figure 4.5 Hidden state sojourn probabilities	73
Figure 4.6 Validation for HSMMs and GMM-HSMM of different sizes	75
Figure 4.7 Timing score scaling with a resolution of 1 standard deviation (Huang, 2009)	77
Figure 4.8: MAS for the 5-state 1-mode HSMM given different time resolutions and $\alpha$ values	78
Figure 4.9: MAS for the 4- and 6-state 1-mode HSMM given different time resolutions and $\alpha$ values	79
Figure 4.10: MAS with $\alpha = 1.0$ for 1-mode GMM HSMMs, HSMMs and HMM of different sizes	80

Figure 5.1 Team-RESCHU main display .....	84
Figure 5.2 BIC for HMMs of Team-RESCHU .....	86
Figure 5.3 BIC for HSMMs of Team-RESCHU .....	86
Figure 5.4 Mission map .....	87
Figure 5.5 Areas of Responsibility .....	88
Figure 5.6 Strike Operator GUI .....	90
Figure 5.7 BIC for HMM of AFRL .....	91
Figure 5.8 BIC for HSMMs of AFRL .....	92
Figure 5.9 HMM and HSMM performance of team and individual models .....	93
Figure 5.10 HSMMs performance of single and team behaviors .....	95
Figure 5.11 Event durations and rate, task arrival rate for single operator and teams .....	96
Figure 5.12 Models for single and teams of operators .....	97
Figure 6.1 DST interface (Castonia, 2010) .....	104
Figure 6.2 Experimental setup (Castonia, 2010) .....	105
Figure 6.3 Team as a set of individual models or as a single holistic model .....	108
Figure A.1 Example of fixation patterns during a 1 minute use of the StrikeView interface .....	114
Figure A.2 BIC curves for the models trained with and without eye tracking data .....	115
Figure A.3 $\chi^2$ values for model fit across the cross-validation sequences (lower is better) .....	116
Figure A.4 Information gained with respect to the maximum entropy model .....	118
Figure A.5 One step-ahead prediction rates for two models .....	119
Figure A.6 Graphical model representation of a first order HMM .....	121
Figure A.7 Graphical model representation of a second order HMM .....	121
Figure A.8 Higher order HMMs BIC comparison .....	128
Figure A.9 Supervised learning model of a single operator of multiple unmanned systems .....	133
Figure A.10 Smooth supervised learning model of a human operator of multiple unmanned systems .....	134
Figure A.11 Model fit in terms of test set likelihood for the three different training techniques .....	136

# LIST OF TABLES

Table 2.1 ANNs applied to human behaviors .....	31
Table 2.2 SVMs applied human behavior.....	34
Table 2.3 ARMA applied human behavior .....	36
Table 2.4 HMMs or HSMMs applied human behavior .....	37
Table 2.5 Summary of different pattern recognition methods applied to human behavior.....	38
Table 3.1 StrikeView grammar .....	52
Table 3.2 Emission function for state 3 for StrikeView.....	56
Table 3.3 RESCHU grammar. ....	61
Table 5.1 Team-RESCHU grammar .....	85
Table 5.2 AFRL grammar .....	90
Table 5.3 Selected models summary.....	93
Table A.6.1 Average normalized entropies of all possible hidden state sequences given the observations (unitless).....	117
Table A.6.2 2nd order HMM structure .....	122
Table A.6.3 Third order HMM structure .....	125
Table A.6.4 Higher-order model complexity analysis .....	127

# LIST OF MAIN ACRONYMS

AFRL	Air Force Research Lab
ANN	Artificial Neural Network
ARMA	Autoregressive Moving Average
ATC	Air Traffic Control
BIC	Bayesian Information Criterion
CTA	Cognitive Task Analysis
DST	Decision Support Tool
GMM	Gaussian Mixture Model
HALE	High Altitude Long Endurance Unmanned Vehicle
HMM	Hidden Markov Model
HSC	Human Supervisory Control
HSMM	Hidden Semi-Markov Model
ISR	Intelligence, Surveillance, and Reconnaissance
MALE	Medium Altitude Long Endurance Unmanned Vehicle
MAS	Model Accuracy Score
MLE	Maximum Likelihood Estimate
PHSC	Procedural Human Supervisory Control
RESCHU	Research Environment for Supervisory Control of Heterogeneous Unmanned Vehicles
RKHS	Reproducing Kernel Hilbert Space
RNN	Recurrent Neural Network
SME	Subject Matter Expert

# CHAPTER 1 INTRODUCTION

---

*“Machines know a good deal about human-machine processes, and this knowledge can permit machines to monitor human performance for errors, just as humans must be able to monitor machine performance for errors or failures.” -Charles E. Billings, 1997<sup>2</sup>*

*“Quis custodiet ipsos custodes?” -Juvenal, circa 100AD<sup>3</sup>*

Formally, Human Supervisory Control, or HSC, is the process by which one or more human operators intermittently interact with a computer, receiving feedback from and providing commands to a controlled process or task environment, which is connected to that computer (Sheridan, 1992). This control loop is represented in Figure 1.1.



**Figure 1.1 Human Supervisory Control Loop (adapted from (Sheridan, 1992))**

Because the computer allows operators and tasks to be decoupled both in time and space, operators in HSC settings often work under time-pressure and in high risk environments. Furthermore, this work is primarily cognitive and procedural, i.e., other than the occasional button press or lever engagement, most work happens via internal information processing that follows a set of pre-defined steps. Typical procedural HSC (PHSC) domains include military command and control, air traffic control, railway systems and process control including the operation of nuclear power plants. The systems under the supervision of the operators tend to be complex and usually consist of many tightly coupled components which may, at times, exhibit emergent properties that are beyond the operators’ ability to comprehend (Leveson, 2003). Such systems tend to generate large amounts of data and need continuous monitoring. This represents a supervisory challenge for the operator especially when compound failures take place. In

---

<sup>2</sup> NSF-HCS Workshop on human-centered systems: information, interactivity and intelligence, Arlington, VA; 1997.

<sup>3</sup> “Who will guard the guards themselves?”, in *Satires of Juvenal* (6.346-348)

such events, emergent behaviors of the system may rapidly exceed an operator's capacity to react to the situation. Furthermore, because PHSC systems are often mission and/or life-critical, operator failure could lead to disastrous outcomes.

The procedures inherent to PHSC applications influence the cognitive patterns of operators (Bruni, Boussemart et al., 2007) and provide a useful basis against which the correctness of an operator behavior can be tested. In fact, deviations from the procedures were shown to be a major contributing factor in a number of aeronautical incidents. In a 1994 review of aircrew-involved accidents, procedures were the single largest cause cited, contributing to 24% of the major accidents examined (National Transportation Safety Board, 1994). Similarly, an analysis of accidents by Boeing concluded that more than 50% of the major hull loss accidents from 1982 to 1991 could have been prevented by better procedure following (Moodi and Graeber, 1998). Both failure to comply to good procedures and compliance to poor procedures are large contributors to accidents in a number of other domains as well (Byrne and Davis, 2006), including medicine (Xiao and Mackenzie, 1995), the nuclear industry (Trager, 1988; Marsden, 1996; Park, Jung et al., 2002), manufacturing systems (Marsden and Green, 1996), construction (McDonald and Hrymak, 2002), and maritime industries (Perrow, 1984).

Unmanned Vehicles Systems (UVSs) form a representative application of time-pressured, mission-critical procedural human supervisory control. In addition to the challenges outlined previously, this domain is of interest because UVS operations are becoming increasingly ubiquitous both in civilian and military applications (DoD, 2007; Nehme, 2009). This is especially true in the military context where the use of Unmanned Air Systems (UASs) has increased tremendously in tasks ranging from Intelligence, Surveillance and Reconnaissance (ISR) to those that use lethal force. The number of hours flown by Predator-series UAS rose from 80,000 hours in 2006 to 295,000 hours in 2009, and cumulatively surpassed the 1 million flight hours mark in April 2010 (Jennings, 2010). As remarkable as this increase in flight-hours is, the demand for UVS operations grows even faster and has created a shortage of qualified operators (DoD, 2007). For this reason, a significant amount of resources and research has been devoted to leveraging automation in order to shift the current operating paradigm in which multiple operators control a single unmanned vehicle to one in which a single operator could control multiple vehicles (Dixon and Wickens, 2003; Mitchell and Cummings, 2005; Cummings, Bruni et al., 2007; Boussemart and Cummings, 2008). This radical change in paradigm represents a challenge both for operators and systems designers (Ollero and Maza, 2007).

An additional complicating factor lies in the fact that as the transition to controlling multiple unmanned vehicles occurs, multiple operators will likely work together, under the leadership of a team supervisor, to carry out coordinated tasks similar to present-day air traffic control (ATC) settings. This scenario is commonly proposed as the future of UAS operations with military and civilian platforms requiring deconfliction in the national airspace (McCarley and Wickens, 2005). While military applications of UASs concentrate on ISR, homeland security and law enforcement, civilian applications include traffic surveillance, weather monitoring, maritime patrol and disaster relief. In addition, UASs can undertake commercial applications such as freight, pipeline monitoring or agricultural management. The increased presence of UASs therefore poses legal and technical challenges in a national airspace already busy with civilian and military manned flights (Ravich, 2009). The technical aspects of this incorporation, notably in terms of the supervision of flight-path deconfliction and crew duties, remain a primary concern for safe operations (Weibel and Hansman, 2005).

In summary, the control of multiple UVSs, especially in a team-environment, compounds the potential for operator cognitive overload while simultaneously increasing the potential consequences of operator failure. Thus, constant monitoring of the operator behavior is critical for proper system behavior. That task is usually assigned to supervisors who typically rely on expert knowledge and experience in order to detect anomalous conditions. Because continuously monitoring the behavior of multiple operators while providing high-level guidance is a demanding task for the supervisor, the ability to automatically recognize the likely onset of an operator's off-nominal behavior, as defined by a deviation from an expected behavioral pattern determined by procedures, has immense practical value as potential serious accidents could be avoided.

The goal of this thesis is to address this supervisory problem by designing and validating a framework capable of providing automatic, continuous and predictive operator monitoring. By leveraging recent advances in processing power and machine learning algorithms, the framework and associated models can monitor one or more operators, each of whom may control one or more supervisory control tasks, which in the representative task include heterogeneous UVSs (Figure 1.2). In operational settings, teams of UV operators are likely to operate under a supervisor whose job it is to provide high-level coordination and monitor the correct behavior of the overall system. While it is unrealistic to expect the supervisor not to become overloaded if presented with the entirety of the operators' behavioral information, automation may be useful to assist the monitoring of operator performance.

In particular, automation can support the performance monitoring task by generating an alert to a team supervisor if the deviation from the expected behavior of the operators under his or her supervision exceeds a given threshold. The supervisor could then take the required corrective action (e.g. by providing assistance in a complex task or by reducing the number of UVs under the operator's control) in order to bring back operator behaviors closer to where they should be, and potentially avert catastrophic outcomes. The role of the supervisor is critical because while operator behaviors may be labeled as anomalous by the predictive models, they cannot be qualitatively assessed as "good" or "bad" because the algorithms can only detect the difference of the current behavior compared to the expected. Operators may react to novel external situations in a perfectly appropriate manner and yet raise an alert because the predictive models were never exposed to such behavioral patterns. Therefore, it is the responsibility of the team supervisor to assess the generated alerts and, if needed, take the appropriate action. The use of such a monitoring system is not limited to military settings and could be applied to a large portion of human supervisory control applications.

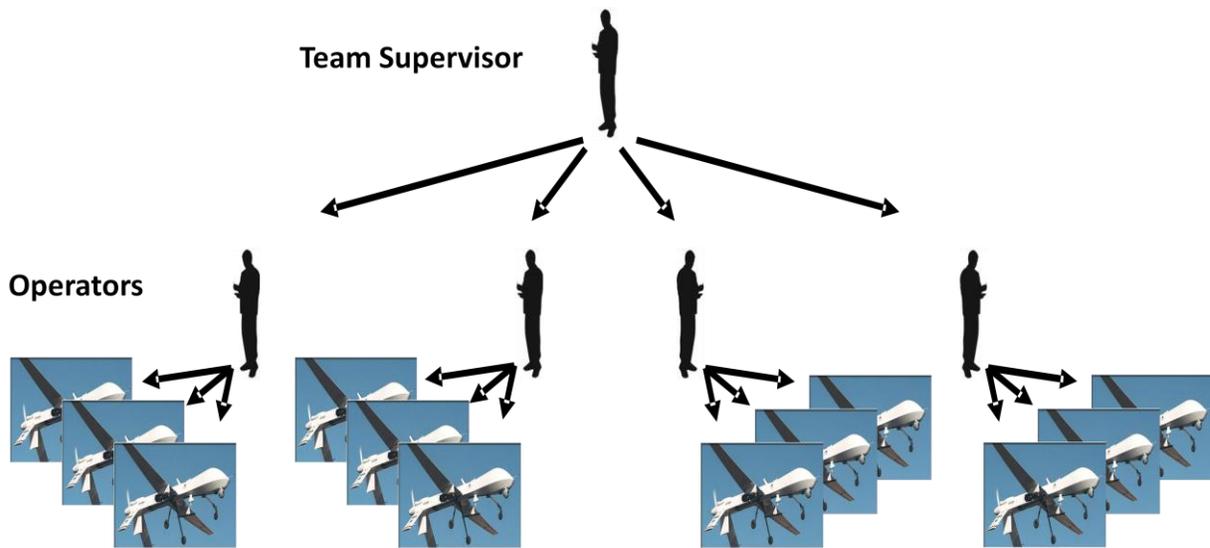


Figure 1.2 Team of multi-UV operators and team supervisor

## 1.1 Research Approach

In the vast majority of PHSC applications, UV operators are trained to follow a set of procedures in order to achieve a specific goal. Without the notion of a goal, the behavior of an operator can vary enormously. However, because the operator is trying to accomplish a specific task in collaboration with a machine in PHSC applications, the range of typical behavior is restrained considerably. As a result, it becomes possible to identify a set of interactions that will result in the task being accomplished. In PHSC settings,

such set of tasks are typically synthesized in Standard Operating Procedures (SOPs), and while they may not be consistently followed in practice, SOPs establish the basis for recurring behavioral patterns common to multiple operators (Boussemart and Cummings, 2008). Similarly, groups of operators can exhibit “habitual routines”, a situation where a group repeatedly exhibits a functionally similar pattern of behavior in a given stimulus situation without explicitly selecting it over alternative ways of behaving (Gersick and Hackman, 1990).

The overarching idea of this thesis is to frame the issue of detecting an operator’s off-nominal behavior as a pattern recognition and prediction problem. A number of machine learning methodologies can then be used to exploit the recurring patterns in operator behavior. In particular, the proposed methodology makes use of Hidden Markov Models (HMMs) and Hidden semi-Markov Models (HSMMs) in order to learn models of operator behavioral patterns from previously-seen data. These patterns correspond to statistically linked clusters of observable events, which we call operator states. The first step of the research consists of modeling the behavior of a single operator in different tasks, while verifying that the assumptions made by the mathematical models are valid. Then, the second step is to scale the approach and see how the proposed methodology can be extended to teams of operators.

## **1.2 Research Questions**

Given the potential benefits of being able to detect off-nominal PHSC operator behavior, this thesis proposes a methodology for learning HMMs and HSMMs of such behavior and answers the following research questions:

- How well can HMMs and HSMMs model the behavior of a single operator engaged in a PHSC task? Additionally, do methodological and model learning assumptions hold true for PHSC data?
- How well can HMMs and HSMMs model the behavior of teams of operators engaged in a PHSC task, and more generally how well does the approach scale to multiple operators?
- What are the limitations of the proposed approach?

## **1.3 Thesis Organization**

This chapter motivated this thesis by highlighting how off-nominal operator behavior detection could provide tremendous benefits in procedural human supervisory control contexts. Such methods could be especially useful in for UVSS operations, a representative application of real-time mission critical PHSC. A high-level overview of the research approach was provided along with the research questions that will be answered in this thesis. The remainder of this thesis is organized as follows:

- Chapter 2, *Literature Review*, consists of an overview of previous work related to this thesis. In particular, an overview of the related literature on human supervisory control and procedures are provided. Keeping an emphasis on PHSC applications, a comparison of the different methodologies for computational models of human behavior is then discussed. Since this thesis relies on HMMs and HSMMs in order to build the computational models of PHSC operator behavior, the mathematical foundations of both methods are examined in depth.
- Chapter 3, *HMMs of Single PHSC Operators*, describes the methodology needed to learn HMMs from single operator experimental data. Results of applying this methodology to two distinct data sets and the resulting models are then presented.
- Chapter 4, *Modeling a Single Operator through HSMMs*, takes the same data sets as Chapter 4 but shows how HSMMs, a more complex version of HMMs, can be used to provide more informative models. In addition, an evaluation metric for HSMMs is presented.
- Chapter 5, *Team Models of PHSC Operators*, examines how the HMM and HSMM methodologies used for single operators scales for teams of multiple operators. Two data sets are analyzed and the resulting models are evaluated.
- Chapter 6, *Conclusions*, closes the argument of this thesis. First, a discussion of the results is provided along with the possible applications such results could have. Potential avenues for future work are provided, and a conclusion summarizes this thesis.

# CHAPTER 2 LITERATURE REVIEW

---

*“Essentially, all models are wrong, but some are useful”*

*-George Box, 1987<sup>4</sup>*

This chapter first presents background information human supervisory control of unmanned systems. Because one of the main challenges of this thesis is to model the behavior of the UVS operators, previous methods for creating computational models of human behavior are presented. Looking at this problem from a pattern recognition and prediction perspective is particularly useful for real-time monitoring, and how different statistical learning techniques have been applied to model human behavior is discussed. An in-depth mathematical discussion of HMMs and HSMMs is then provided.

## 2.1 Procedural Human Supervisory Control

Procedural human supervisory control is a critical application domain for Human Systems Engineering (HSE) because it subsumes a majority of situations in which complex processes need to be monitored and intermittently adjusted for proper performance given a set of procedures. For example, operators of power plants, trains, autopilots, robots or unmanned vehicles all have to supervise complex systems by interfacing with automated tools. The role of automation is critical because failures in more complex processes often manifest themselves as unforeseeable emergent properties which may exceed the system’s operating range and leave human operators as the ultimate safeguard. Should the operator be unable to correct the situation, the system may fail and the consequences can be catastrophic (Perrow, 1984; Leveson, 2003). Similarly, the role of the procedures that guide the interaction between the operators and the system also plays a critical role (Marsden, 1996). Thus, PHSC applications are prime examples of complex and critical socio-technical systems where the procedural collaboration between humans and automated systems is critical for ensuring proper system behavior. The domain of UVS is a particularly salient example of such systems, in that UVS operators often operate highly automated platforms in life- and time-critical environments.

---

<sup>4</sup> Box, George E. P.; Norman R. Draper (1987). Empirical Model-Building and Response Surfaces. Wiley. pp. 688, p. 424. ISBN 0471810339.

### ***2.1.1 Procedures and Human Supervisor Control***

A procedure is “a set of steps, acts, or even sub-procedures that one intends to accomplish in order to achieve a goal” (Degani and Wiener, 1997; Moodi and Graeber, 1998). In general, procedures are used to manage the interaction of an operator with another system or systems (e.g., automation, roadways, a plant) in situations in which the interacting systems are not deterministically connected. In such situations, the range of interactions between the systems is delimited by procedures. Typical reasons for controlling the interaction between systems are to ensure safety, efficiency, standardization, or predictability.

Procedures are, in general, under-researched and poorly understood in relation to their importance in human machine systems, as they have been directly cited as a primary factor in a number of serious accidents (Rogovin, 1979; Degani and Wiener, 1997; Furuta, Sasou et al., 2000). Implicit in most complex activities, procedures are particularly crucial in systems where interactions between operators and systems must be controlled, such as most human supervisory control systems. A great deal of research mentions, but does not focus on, procedures. For example, procedures underlie operator response to alerting systems (de Winter, Wieringa et al., 2007; Lees and Lee, 2007; Stanton and Baber, 2008) and operator response to errors and accidents (Kanse, Van Der Schaaf et al., 2006; Patrick, James et al., 2006). In each of these cases, procedures are required for good performance, but are not considered directly in the research. One fundamental aspect of procedures that is not well understood is the relationship of procedure compliance and non-compliance to overall system performance (Roth, Mumaw et al., 1994).

Two opposing views of the relationship between compliance and non-compliance to procedures have been proposed in the literature. The classic position takes the “normalized” point of view in which operators should explicitly follow procedures and that deviation from procedures represents human error. In fact, de Brito et al. (2002, p. 233) states that “system reliability requires pilots to strictly follow procedures.” This view leads to the conclusion that one should design procedures better, document procedures better, and not accept deviations from procedure as a matter of system safety. The normalized view has been criticized for not recognizing that explicit rule following does not guarantee good system performance (Dekker, 2003), and that some deviations from procedures in response to external events may in fact lead to better system performance (Ockerman and Pritchett, 2004). In fact, noncompliance that results in good outcomes does not usually get reported, leading to the misperception that noncompliance almost inexorably leads to bad outcomes. Such criticisms gave rise to an alternative “immersed” view, which proposes that operators use procedure information as an input to guide behavior

and that deviation from procedure represents an attempt by operators to improve overall performance given an operational context.

The debate between normalized and immersed views is relevant for this thesis because the proposed methodology relies on the deviation from patterns in order to detect abnormal situations. While the set of normative behaviors learned from previously seen data is a central component of the methodology, the deviation from expected behaviors cannot be qualitatively labeled as “good” or “bad” in terms of operator performance. The detected abnormal situations are simply “different” from what was previously seen and may, in fact, be perfectly appropriate behaviors given a specific operational context. In line with the immersed point of view, the proposed methodology can provide alerts to a supervisor, but the nature of the actual response to the alert is left to human interpretation.

### ***2.1.2 Team Supervisory Control***

PHSC operators frequently work in teams, and sometimes in distributed teams (Bowers, Oser et al., 1996). This is especially true for most military UASs operations which currently rely on multiple operators controlling a single platform. For example, a crew of at least two operators is typically needed for a single Predator UAS (DoD, 2007). Teams are more than just a collection of individuals pursuing their own goals. A commonly accepted definition of “team” is a “collection of (two or more) individuals working together inter-dependently to achieve a common goal” (Salas, Dickinson et al., 1992). The main concepts in this definition are both the common goal and the interdependence needed to achieve this goal.

The difficulty of automation-mediated interactions with complex systems can be exacerbated in teamwork settings mainly for three reasons (Swezey and Salas, 1992; Bowers, Salas et al., 2006). First, team tasks are complex in that they require one operator to process several subtasks concurrently, such as performing their individual responsibilities while communicating to other team members if the automation fails in a group task. As a result, considerable communication and time might be required before the team can accurately diagnose the new state and figure out how to react to it, which could have devastating consequences in time critical situations (Gorman, Cooke et al., 2005). Secondly, operators in team environments need to be cognizant of the state of the rest of the team (Cooke and Gorman, 2006). Studies have shown that it is more difficult to maintain situation awareness in a team context than when individuals perform alone, especially when team members are not collocated (Jentsch, Barnett et al., 1999).

Finally, team dynamics may introduce biases in the decision making process and result in improper use of the available information (Dunbar and McDonnell, 2001; Mosier, Skitka et al., 2001). While it could be expected that the presence of additional team members could alleviate the risk of the information misinterpretation, studies have shown that this is not the case. Indeed, the “presence of a second team crewmember as well as a highly reliable automated system might actually discourage rather than enhance vigilance” (Mosier, Skitka et al., 2001, p. 3). Another instantiation of information misuse in teams is “groupthink”, which is “a mode of thinking that people engage in when they are deeply involved in a cohesive in-group, when the members' strivings for unanimity override their motivation to realistically appraise alternative courses of action” (Janis, 1972, p. 9).

Within the PHSC domain, a commonly-cited example of team failure is the *USS Vincennes* mistakenly shooting down an Iranian Airbus because the crew erroneously believed the airliner to be a hostile F-14 on an attack run (Klein, 1999). Another well known example is the Eastern Airlines 401 incident in which the aircraft performed a controlled flight into terrain after the flight crew became fixated on a malfunctioning landing gear position indicator and failed to realize that the autopilot was in the wrong mode (NTSB, 1973). While both accidents were initially attributed to “operator error”, subsequent studies pointed out that the accidents could not solely be attributed to the actions of a single individual. The widely-used term “operator error” can be misleading and should be understood as “operators’ error” or “team error” (Gardenier, 1981; Weick, 1990).

Additionally, teams do not operate in a vacuum. They are, most of the time, structured according to hierarchical chains of command in which subordinates are put under the responsibility of a supervisor, team leader or commander. A main role of the supervisor is to monitor the behavior of the team and take appropriate remedial measure should the performance of the team drop below acceptable levels (Brewer, Wilson et al., 1994).

### ***2.1.3 Monitoring Human Supervisory Control Behaviors***

Both group members monitoring each other and team supervisors monitoring a group have been shown to improve team performance by helping a group integrate related task activities, identify appropriate interruption opportunities, and notice when a team member requires assistance (Pinelle, Gutwin et al., 2003; Gutwin and Greenberg, 2004). One main role of team leaders is to take direct action during the team task and guide the team towards positive outcomes. Determining when and how often to intervene in team behaviors are key factors to optimizing team performance (Irving, Higgins et al., 1986; Brewer and Ridgway, 1998). Deciding when to intervene is non-trivial because the supervisors in most PHSC settings

cannot easily infer accurate team performance from simply observing the physical actions of the subordinates, since the majority of behaviors are cognitive and not directly observable. In the context of operators performing PHSC tasks, an individual's physical actions only involve activities such as interfacing with a computer. This disconnect between operators' behavior and possible ensuing consequences can be widened by the remote location of the physical outcomes. For example, the behavior of a UAS pilot operating in mode confusion (i.e. interacting with the system under the assumption it is in a mode different from what it actually is (Joshi, Miller et al., 2003)) is unlikely to differ from the normal behavior and therefore would be difficult for a supervisor to detect solely based on the observation of the operator's interactions with the ground control station. Furthermore, pilots remotely controlling a UAV platform in an operational field cannot benefit from the external cues (such as peripheral vision or environmental auditory information) which could indicate an abnormal condition of the aircraft.

In order to facilitate good team performance, all personnel involved have to form an adequate mental model of the situation. While this is a recognized and well-studied issue for single operators (Lee and Moray, 1992; Muir, 1994; Riley, 1996), the problem is more salient for team supervisors as they must synthesize information from multiple operators often while being under strict time constraints due to operational tempo. The main problem is then how to support supervisors of such tasks so that they are better able to understand what their team is doing (Scott, Rico et al., 2007; Cummings, Bruni et al., 2010). One way automation can assist a supervisor is through real-time monitoring that provides a better understanding of operator and team performance through the use of a decision support tool, seen as critical in most PHSC settings (Castonia, 2010; Cummings, Bruni et al., 2010). Since researchers recognize that the role of the team supervisor can be critical in improving team performance (Burns, 1978; Hackman, 2002), the advancement of supervisor decision support tools that exploit predictive models of operator behavior may also play a critical role in improving team performance.

In summary, this section highlighted the critical nature of the work performed by PHSC operators, especially in the context of UV operations. The task of monitoring the performance of the UV operators typically falls to a team supervisor whose actions are critical to overall system performance. There is therefore value in drawing team supervisors' attention to operators whose behaviors deviate from the expected. In order to automatically detect such anomalous operator condition, computational models of the expected operator behaviors are needed. Different methodologies for obtaining such models are discussed in the following section.

## **2.2 Computational Models of Human Behavior**

In general, given a training set of known behavioral patterns, there are two alternatives to detect anomalous behaviors: 1) show that the observed pattern is similar to a known adversary pattern and 2) show that the observed pattern is dissimilar to a known normal pattern (Singh, Tu et al., 1996). The first option is impractical because it is, in general, difficult to generate an exhaustive list of adversary patterns in applications characterized by a large number of degrees of freedom such as PHSC settings. In contrast, predictive models of human behavior embody the known normal patterns and can therefore be used to detect and predict anomalous behaviors. In addition to providing anomaly detection capability, most predictive models also comprise a descriptive component. Therefore, a number of insights can be derived from a qualitative analysis of the models. Thus, within the context of PHSC behaviors, the real-time use of predictive models can support the performance monitoring task of a team supervisor by generating alerts when anomalous situations are predicted. The same models can also be analyzed off-line and provide a better understanding in the typical behavior of operators. The remainder of this section discusses different types of modeling techniques in light of a number of important characteristics for modeling human behaviors in PHSC contexts.

### ***2.2.1 Important Characteristics of Modeling Techniques in PHSC settings***

PHSC applications typically require operators to react to situations that are dynamic, time-sensitive and uncertain. The appropriate framework for the detection and prediction of anomalous behaviors in such settings should fit these characteristics. Therefore, possible modeling techniques should be examined in light of a number of criteria. The first two criteria pertain to the structure of the model while the following two relate to the learning of the model parameters.

1. Use of categorical data. The behavior of an operator is often recorded as a sequence of categorical actions such as mouse clicks or keyboard input. Therefore, models that rely on interval data (for example the range of real numbers) are not applicable in such a context because an ordering may not exist for user events.
2. Interpretability. While the aim of statistical learning methodologies is to provide models that provide good recognition and predictions rates, models should also provide explanatory factors pertaining to the underlying modeled process. Interpretable models thus afford descriptive capability which can be analyzed. Without interpretability in the context of human behaviors, statistical learning methods are effectively useless.
3. Use of temporal information. While the sequence of operator actions contains valuable information, the operational tempo (i.e. the inter-event arrival time) provides another dimension

of information that can be exploited by the models. This is especially important in time-critical domains in general and PHSC settings in particular.

4. Unsupervised learning. Because the information, or priors, regarding the underlying structures and processes that drive human behavior are often not known, methods that require a priori labeled data may suffer from biases compared to methods that rely solely on the statistical patterns contained in the data (Boussemart, Fargeas et al., 2010)

There exists a wide range of computational modeling techniques, and they can be divided in three main categories: symbolic models, architecture-based models and statistical models. The first two classes of model tend to be deductive (i.e. use of a top-down methodology relying on predefined theories) whereas statistical models tend to be inductive (i.e. a bottom-up approach and data driven).

### ***2.2.2 Deductive Models***

Symbolic modeling techniques represent different mental objects using variables and rules. The most commonly-used decision support tools relying on such methods are expert systems (Endsley, 1987). Expert systems use descriptive models designed to encapsulate a set of rules abstracted from human expert knowledge, and such systems have been used successfully to replicate complex decisions flows such as a physician's deductive process during a diagnosis (Weber and Coskunoglu, 1990; Miller, 1994). This methodology, however, suffers from its strict reliance on rules that must be correctly elicited from a subject matter expert (SME) a priori. This problem of knowledge elicitation is both time consuming and may introduce the bias of a given SME in the system.

Architecture-based models make use of theoretical frameworks aimed at replicating cognitive processes, and therefore serve as blueprints for intelligent agents. Goals, Operators, Methods, and Selection rules, or GOMS (Wayne, Bonnie et al., 1992), and Adaptive Control of Thought-Rational, or ACT-R (Anderson, 1993), are two such cognitive frameworks. ACT-R is an open-ended architecture with modules simulating different processes such as visuospatial working memory (Lyon, Gunzelmann et al., 2004). While ACT-R has been used to model low-level cognitive processes such as serial memory (Anderson and Matessa, 1997), and has mimicked patterns of brain activation during imaging experiments (Anderson, Qin et al., 2008), the practical use of ACT-R is limited because of sophisticated cognitive task modeling required to fit the framework.

In contrast, GOMS is an architecture that focuses on a user's interaction with a computer by breaking it down into elementary actions which can be physical, cognitive or perceptual. GOMS has been

successfully used in modeling the work of a telephone operator, and has predicted the impact on productivity of the introduction of a new type of workstation (Gray, John et al., 1992). However, GOMS is limited because it assumes that all users are deterministic and follow the same human processor model, which narrowly limits use to expert behaviors.

The main shortcoming of both symbolic and architecture-based methods lies in their use of a priori definition of rules or cognitive processes. Eliciting such rules or cognitive processes is problematic in PHSC settings because of the complexity of the decisions expected from an expert operator. Moreover, such approaches are inherently brittle in they cannot be used to describe or predict anomalous, never-before-seen events. In contrast, statistical models make use of an inductive, data-driven approach in the sense that they rely on the exploitation of the statistical patterns exhibited in the human behavior data stream in order to describe and predict possible future actions. The next section provides an overview of the different statistical learning methods that can be used to model human behavior.

### ***2.2.3 Statistical Models***

A significant body of work has focused on using statistical modeling techniques for human behaviors, relying on the idea that human actions can be appropriately modeled by serial processes because humans can solve only one complex problem at a time (Welford, 1952; Broadbent, 1958). Therefore, pattern recognition techniques have been used to model human behaviors ranging from large-scale populations patterns (Pentland, 2008) to detailed small-scale cognitive processes (Griffiths, Kemp et al., 2008). This range encompasses a large number of tasks; for example, computer system intrusion detection (Terran, 1999), ship navigation (Gardenier, 1981) or car driving (Pentland and Liu, 1999). Yet, even though the correctness of operator behavior in PSCH settings is often mission and life-critical, little work has been done using pattern recognition techniques in such contexts. Statistical techniques can be beneficial in the PSCH domain because, in contrast with qualitative models, they provide a formal, quantitative basis for describing human behavior patterns and for predicting future actions. This is especially true for PHSC application because the procedures provide structure to the behavior of an operator thereby facilitating the emergence of behavioral patterns that can be exploited by the statistical models.

Statistical models can be either generative or discriminative. Discriminative models are a class of models used in machine learning for modeling the dependence of an unobserved variable  $y$  on an observed variable  $x$ . Within a statistical framework, this is done by modeling the conditional probability distribution  $p(y|x)$ , which can be used for predicting  $y$  from  $x$ . Discriminative models differ from

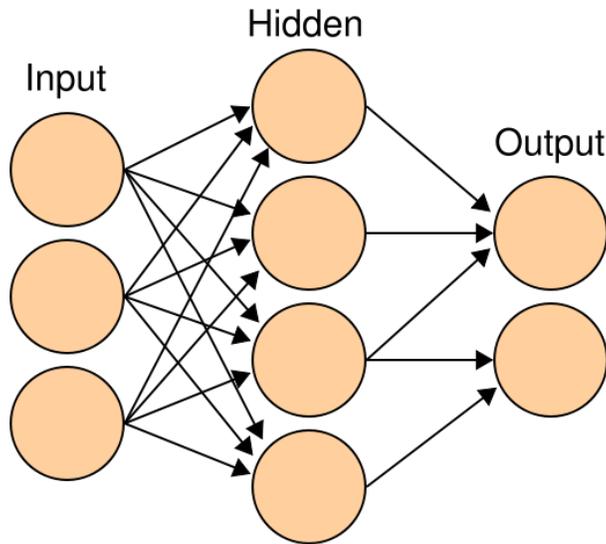
generative models in that they do not allow one to generate samples from the joint distribution  $p(y, x)$  (Ng and Jordan, 2001).

The distinction between generative and discriminative models is important in this thesis for multiple reasons. First, generative models tend to be more flexible than discriminative models in expressing dependencies in complex learning tasks at the expense of a greater complexity arising from the need to model the full joint distribution  $p(x, y)$  as opposed to the conditional distribution  $p(y|x)$  (Shannon, 1948). Within the scope of this thesis, while both discriminative and generative models could be used for anomalous operator behavior detection and prediction, generative models provide a more comprehensive model of the operator behavior because they model full joint distributions. Secondly, because of their reliance on conditional distributions, the predictive power of discriminative models is generally more limited than that of generative models. Finally, discriminative models usually rely on supervised learning and extending them to unsupervised contexts tend to be difficult. In PSCH setting, this can be problematic because there is no definitive way to perform unbiased a priori state labeling.

This section reviews four widely used pattern recognition and prediction techniques and evaluates how each could be used for PHSC operator modeling. The first two of the techniques are discriminative: Artificial Neural Networks (ANN) and Support Vector Machines (SVM), while the other two are generative: Auto-Regressive Moving Averages (ARMA) and Hidden Markov Models (HMM), along with a variation of HMMs called the Hidden Semi-Markov Models (HSMM).

### ***Artificial Neural Networks***

Artificial neural networks use a connectionist approach: they assume that the modeled processes can be described by a network of nodes. The nodes and connections are modeled after simplified biological neurons and synapses, and therefore each node outputs to the next layer a function (usually a sigmoid) of the weighted sum of the previous layer (McCulloch and Pitts, 1943; Minsky, 1954). With the use of one or more layers of hidden neurons, ANNs are in theory capable of representing any non-linear function (Bishop, 2006). Figure 2.1 shows a graphical representation of a typical ANN, with 3 input variables, a hidden neuron layer comprising 4 nodes and finally 2 output nodes.



**Figure 2.1 Graphical representation of an artificial neural network**

The learning process for an ANN consists of optimizing the weights between the different nodes in order to minimize a cost function. A wide range of paradigms exist for the training of ANNs, the most commonly used are back-propagation and reinforcement learning. Both back-propagation and reinforcement learning techniques are instances of supervised learning (i.e. some information about the desired output of the network for a given input must be defined a priori). ANNs can also be used in unsupervised settings, for instance through the use of self-organizing Kohonen maps for clustering tasks (Kohonen, 1982). Furthermore, with the addition of one or more delayed feedback loop from the input layer to the output layer, neural networks can exploit temporal data and model dynamic processes. A neural network with such a structure is a recurrent neural network (RNN).

Neural nets have been successfully used in diverse applications such as handwriting recognition (Gader, Mohamed et al., 1997) and detecting credit card fraud (Ghosh and Reilly, 1994). ANNs have also been used to model human behavior but with limited success. Yeung et al. (2006) trained ANNs capable of modeling operators taking single decisions in static environments. However, the ANNs in this work did not take the sequence or the timing of multiple actions into account. While there has been no prior use of ANNs for operator modeling in PSHC settings, one possible solution could involve an RNN in which each input neuron represents a specific operator event. The ANN could then determine whether the input is anomalous. For realistic PSHC systems however, such a structure would imply an extremely complex model that would likely require large amounts of training data, a commonly cited issue of neural networks (Pomerleau, 1993).

A significant drawback of neural nets lies in the way the network stores its knowledge as weights between nodes. These weights are usually not interpretable, which makes neural nets akin to a black-box that cannot provide explanatory power regarding the captured underlying process. This is problematic in the context of this thesis as ANNs lack this descriptive ability.

The discussion of ANNs for modeling human behavior is summarized in the Table 2.1. Neural networks can use categorical data and recurrent neural networks allow the explicit modeling of temporal information. However, ANNs are not interpretable and therefore behave as black box models. Finally, the use of unsupervised learning for ANNs is usually restricted to clustering approaches such as Kohonen self-organizing maps.

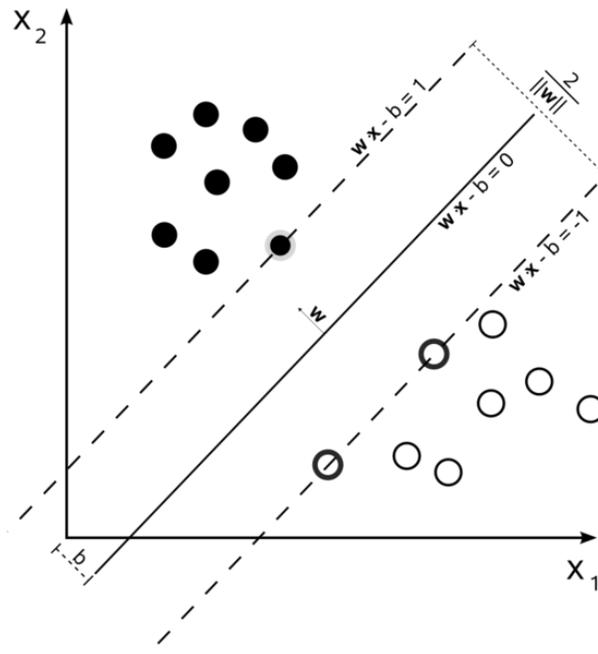
**Table 2.1 ANNs applied to human behaviors**

Use of categorical data	 (Recurrent ANNs)
Use of temporal information	 (Recurrent ANNs)
Interpretability	
Unsupervised Learning	 (self-organizing maps)
Other Limitations	Black box model

### ***Support Vector Machines***

One of the most recent techniques for discriminative modeling exploits reproducing kernel Hilbert spaces (RKHS) and the application of the so-called “kernel trick” in order to find the maximum margin hyperplane for different classes of objects in high dimensional (possibly infinitely dimensional) spaces (Vapnik, 2000). Such hyperplane-based algorithms are known as support vector machines. Figure 2.2

shows the maximum margin hyperplane separating the two classes of objects (the black and white dots in this case).



**Figure 2.2 SVM are maximum margin hyperplanes (Cyc, 2008)**

The decision vector  $\mathbf{w}$  is normal to the decision boundary. The distance of any given point to the boundary is expressed as:

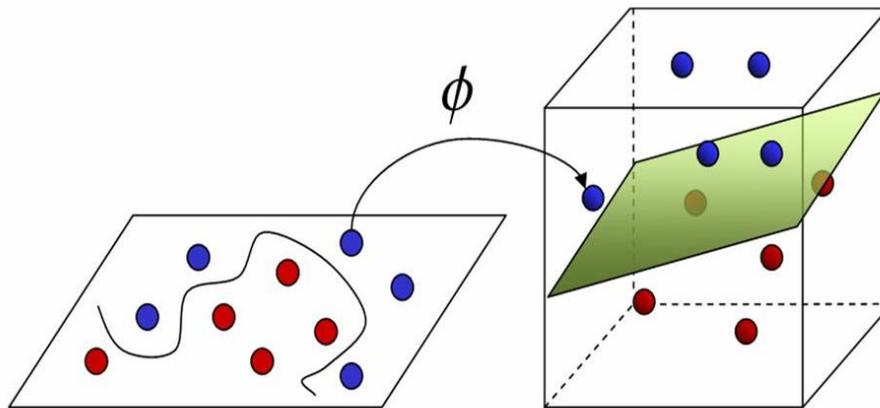
$$d = \mathbf{w} \cdot \mathbf{x} - b \tag{1}$$

where  $\|d\| > 1$  if the point is strictly outside of the margin. Furthermore, the norm of the margin is  $2/\|\mathbf{w}\|$  and  $b$  is the offset parameter of the hyperplane. Finally, the highlighted objects on the margin are the support vectors of the maximum margin hyperplane. In order to increase robustness to noise, SVMs can also have soft margins in which case the decision boundary is allowed to misclassify a number of points.

In their basic version<sup>5</sup>, SVMs can only model linearly separable data and therefore may not be usable to model non-linear human behaviors (especially for a categorical representation of operator behavior). However, in conjunction with the use of RKHS, the data can be projected in higher dimensions which allows the use of an hyperplane with data that is not linearly separable (Burges, 1998). Figure 2.3 shows

<sup>5</sup> Multiple variants of SVMs exist, such as Support Vector Regression (Drucker, Chris et al., 1997) to Structured SVMs (Tsochantaridis, Joachims et al., 2005). These methods share characteristics similar to regular SVMs.

this process, going from a 2D to a 3D space via a kernel  $\phi$ . The essence of the kernel trick is that the data becomes linearly separable when projected in a higher dimension space, a 3D space in this example. Thus, categorical representation of human behavior may be used in conjunction with the kernels. While finding the separating hyperplane, even in high dimensional spaces, remains computationally tractable, the issue arises with the design of the kernel itself: finding an appropriate kernel that allows a separating plane to be found while remaining simple enough remains non-trivial (Ayat, Cheriet et al., 2005).



**Figure 2.3 Kernel trick for non-linearly separable data in 2D (Niissalo, 2010)**

From an application perspective, SVMs have been used successfully in many fields such as a crowd monitoring system (Yogameena, Komagal et al., 2010) or for intrusion detection (Mukkamala, Janoski et al., 2002). With respect to human behaviors, SVMs have been used to determine whether a driver was distracted based on vehicle behavior and eye tracking data (Liang, Reyes et al., 2007), and more recently to discriminate whether a current driver is a legitimate owner of the car based on driving patterns (Qian, Ou et al., 2010). The SVMs presented in this work uses the car dynamics (i.e. Fourier transforms of acceleration, braking and steering data) to identify the driver correctly approximately 80% of the time. A methodology similar to the one proposed by Qian et al. could be used to model and detect anomalous operator behavior, but the use of SVM presents specific drawbacks for the PHSC context.

One of the major drawbacks of SVMs lies in the use of supervised learning. SVM require data to be labeled a priori which is typically an expensive endeavor. This is especially true in the context of PHSC behavior where no established methodology exists to establish proper labels. Recent efforts, however, have shown that it is possible to use unsupervised learning methods with SVMs, but the resulting training process scales poorly and can only deal with small data sets (Xiangwei, Kun et al., 2008). Furthermore,

SVMs are mostly used in static discriminative tasks and therefore exploit neither the temporal information contained in the data, nor models the dynamics of the underlying process. This represents a significant drawback for PHSC applications due to their time-sensitive nature.

Summarizing (see Table 2.2), SVMs can exploit categorical data but cannot model temporal data. While SVM may be interpretable through the definition of the support vectors of the separating hyperplane, their use with unsupervised learning technique only had limited success. Finally, the design of an appropriate kernel that projects the data in higher dimensions can be problematic.

**Table 2.2 SVMs applied human behavior**

Use of categorical data	
Use of temporal information	
Interpretability	
Unsupervised Learning	 (for small data set only)
Other Limitations	Need carefully designed kernel and parameters

### ***Auto-Regressive Moving Averages***

Auto-Regressive Moving Average (ARMA) models exploit the autocorrelations present in a time series (Hamilton, 1994). Formally, autoregressive models can be expressed as follows:

$$X_t = \sum_{i=1}^q \phi_i X_{t-i} + \sum_{j=1}^p \theta_j \epsilon_{t-j} + \epsilon_t \quad (2)$$

where  $X_t$  is a stochastic process expressed as a linear combination of its past  $q$  values and the current and past  $p$  values of the model error  $\epsilon_t$ . The noise model is assumed to be an independent identically distributed (IID) Gaussian with zero mean and variance  $\sigma_\epsilon^2$ . The parameters of the models are the auto-

regressive (AR) coefficients  $\phi_i$ , the moving average (MA) coefficients  $\theta_j$ , the models orders  $p$  and  $q$ , and finally the model variance  $\sigma_e^2$ . Three steps are accomplished in the process of fitting the ARMA model to a time series: (1) identification of the model, that is, the determination of the ARMA model orders  $p$  and  $q$ ; (2) estimation of the parameters (AR and MA coefficients and model variance); (3) application of a forecasting methodology to obtain new values of the time series. The critical stage of the process is model identification. This is usually accomplished by fitting several models ARMA of different orders  $p$  and  $q$  to the time series data and selecting one of them by applying some statistical criteria such as the Bayesian Information Criterion (Tsay, 2005).

In the context of human behaviors, ARMA has been used as for modeling skilled-based tasks such as target pursuit activities (Shinners, 1974; Abdel-Malek and Marmarelis, 1990), simple hand-tapping (Pressing and Jolley-Rogers, 1997) or word reading behaviors (Wagenmakers, Farrell et al., 2005). These represent low-level skill-based tasks and no prior work has been done for modeling and predicting higher-level cognitive reasoning tasks such as those in PHSC settings. Furthermore, ARMA models only work on interval data which could be problematic within the context of this thesis as PHSC behavioral events are represented as a discrete series of UI interactions, an inherently categorical scale. Autoregressive models could however be used to model reaction times, which can be categorized as an interval scale, in response to specific operational conditions.

While quite powerful, these methods need to be carefully tailored in order to fit the modeled data, and may not always be capable of distinguishing between two vastly different signals if they give rise to nearly identical power spectra, a commonly used property used to determine the values of  $p$  and  $q$  in the identification phase (Sulis and Combs, 1996, p. 47). Furthermore, because ARMA-like methods are based on regression, their use on categorical human behavioral data may be problematic.

In summary (see Table 2.3), ARMA-based methods typically rely on interval or ordinal scales and may not be usable on categorical representations of human behaviors. However, these methods can be used on time series and are interpretable through the analysis the different regression and correlation parameters.

**Table 2.3 ARMA applied human behavior**

Use of categorical data	✗
Use of temporal information	✓
Interpretability	✓
Unsupervised Learning	✓
Other Limitations	IID noise assumption

### ***Hidden Markov Models and Hidden Semi-Markov Models***

Hidden Markov models and hidden semi-Markov models are a sub-family of dynamic Bayesian networks based around an unobservable Markov chain. Each state of the Markov chain gives rise to an emission function of observable events (Rabiner and Juang, 1986). The emissions functions are probabilistic and can be discrete or continuous, which allows the categorical representation of the operator behavior. The space of observable events can therefore be categorical (e.g. representing a set of possible user actions) or interval (e.g. body position or reaction time). Classical HMMs, however, have a structural shortcoming in that they cannot explicitly take the timing of state transitions into account. In contrast, hidden semi-Markov models are a version of HMMs capable of explicitly modeling the timing of state transitions (Guedon, 2003). Both HMMs and HSMMs have been shown to be capable of capturing time-varying signal characteristics by statistically modeling the underlying dynamic of the signal (Rabiner, 1989). Importantly, HMMs and HSMMs are interpretable because (1) the emission function of each hidden state is expressed explicitly over the space of observable events and (2) the transitions between hidden states are also explicitly modeled.

One of the main assumptions for using HMMs is that the data should be independent identically distributed (IID). Although the IID assumption rarely holds in practice, HMMs and HMM-based methods have been successfully used in a number of applications (Chien and Furui, 2003; Allanach, Tu et al., 2004; Bilmes, 2006). In the context of human behaviors in particular, HMMs have been shown to

accurately classify and predict hand motions in driving tasks, which is a strong application of monitoring and prediction of sequential data (Pentland and Liu, 1995). In this work, however, the authors had access to the unambiguous ground truth linking the state of the model to the known hand positions. In another example, Hayashi et al. (2005) have used HMMs to model the gaze patterns of shuttle pilots. This work also used a priori labeled data sequences to guide model learning, and the resulting HMM was shown capable of detecting anomalous pilot behavior based on gaze pattern only. However, methods that rely on supervised training may not be appropriate in dynamic environments typical of PHSC settings, where the definitions of the states of the model are not known a priori, particularly for anomalous events.

In summary (see Table 2.4), HMMs and HSMMs can use categorical representations of operator behaviors and the HSMMs are capable of exploiting the temporal dimension of a time series. In addition, HMMs and HSMMs are interpretable through the analysis of the hidden state definition which can be learned via unsupervised methods. However, HMMs and HSMMs rely on the Markov assumption, and Appendix A discusses the practicality of this assumption applied to human behaviors.

**Table 2.4 HMMs or HSMMs applied human behavior**

Use of categorical data	✓
Use of temporal information	✓
Interpretability	✓
Unsupervised Learning	✓
Other Limitations	Markov assumption

Thus, the structure of the HMM is particularly suitable for inferring underlying, hidden cognitive processes from visible events extracted from human behavior, especially in unsupervised training contexts (Boussemart, Fargeas et al., 2010). For example, a UAS pilot may perform a sequence of observable actions such as selecting a UV, adding a waypoints and modifying the altitude, which could possibly be collapsed into a “threat avoidance” operator state. Yet, even with such a structural fit between a framework and a modeled process, little work has been done using HMMs and HSMMs in PHSC

settings. This presents a clear research opportunity because HMMs and HSMMs provide formal quantitative bases for providing both recognition and prediction of operator behavior in real-time (Boussemart and Cummings, 2008; Boussemart and Cummings, 2010; Boussemart, Fargeas et al., 2010).

**Summary of the Methodologies**

Table 2.5 provides a summary of for the pattern recognition human behavior modeling methodologies described in this section. The table shows that the HMMs and HSMMs techniques are the best fit for modeling human behavior in terms of the four important characteristics outlined at the beginning of this section for PHSC domains. In contrast, all the other techniques possess significant limitation in PHSC contexts that impair their use for the purposes of this thesis. The most important flaw of ANNs is that they are not interpretable and behave as black-boxes. In contrast, SVMs are interpretable but do not make use of temporal information. ARMA models are interpretable and use temporal information but cannot handle categorical data (such as user interface events). While HMMs and HSMMs address these shortcomings, they remain under-used in PHSC settings. The following section provides an in-depth review of the mathematical bases for HMMs and HSMMs.

**Table 2.5 Summary of different pattern recognition methods applied to human behavior**

	Discriminative Models		Generative Models	
	ANN	SVM	ARMA	HMM/HSMM
Use of categorical data	✓	✓	✗	✓
Use of temporal information	~ (Recurrent ANNs)	✗	✓	✓
Interpretability	✗	~	✓	✓
Unsupervised Learning	~ (self-organizing maps)	~ (for small data set only)	✓	✓
Other Limitations	Black box model	Need carefully designed kernel and parameters	Categorical data is problematic, IID noise assumption	Markov assumption

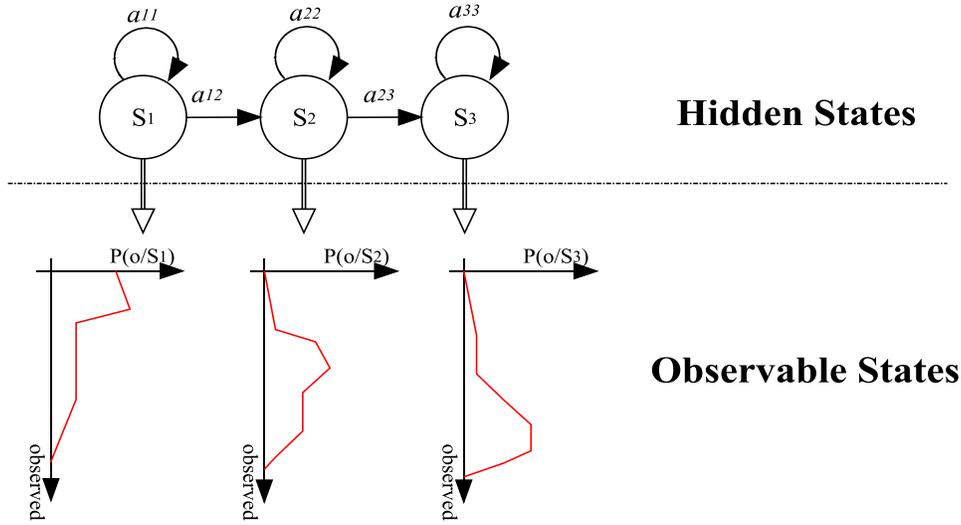
## 2.3 Hidden Markov Models

### 2.3.1 Formal definition

Hidden Markov models were popularized in a seminal paper by Rabiner et al. (1986). They consist of stochastic Markov chains based around a set of hidden states whose value cannot be directly observed. Each hidden state generates an observable symbol according to a specific emission function. Although the sequence of hidden states cannot be observed directly, the probability of being in a specific state can be inferred from the sequence of observed symbols. Transition functions describe the dynamics of the hidden state space. There are two types of probability parameters in HMMs: state transition probabilities and observable symbol output probabilities. Given a finite sequence of hidden states, all the possible transition probabilities and symbol output probabilities can be multiplied at each transition to calculate the overall likelihood of all the output symbols produced in the transition path up to that point. Summing all such transition paths, one can then compute the likelihood that the sequence was generated by a particular HMM. Adopting the classic notation from Rabiner et al. (1986), let  $N$  be the number of states  $S = \{S_1, S_2, \dots, S_N\}$  in the HMM and  $M$  be the number of observation symbols  $V = \{V_1, V_2, \dots, V_M\}$  (i.e. the dictionary size). Let  $S_i^t$  denote the property of being in state  $i$  at time  $t$ . The state transition probability from state  $i$  to state  $j$  is  $A = \{a_{ij}\}$  where  $a_{ij} = P(S_j^{t+1}, S_i^t)$ ;  $i, j = 1, \dots, N$ . The symbol output probability function in state  $i$  is  $B = \{b_i(c)\}$ , where  $b_i(c) = P(V_c | S_i)$ . The distribution  $b_i(c)$  may be continuous or discrete, but in the remainder of the thesis the emission functions will be assumed to be discrete. The model parameters must be valid probabilities and thus satisfy the constraints:

$$\sum_j^N a_{ij} = 1, \sum_c^M b_j(c) = 1 \quad (3)$$
$$a_{ij} \geq 0, b_j(c) \geq 0.$$

The initial probability of being in state  $i$  at time  $t = 0$  is  $\pi = \{\pi_i\}$ , where  $\pi_i = P(S_i^0)$  and  $\sum_i \pi_i = 1$ . Thus, an HMM is formally defined as the tuple:  $H = \{S, V, A, B, \pi\}$ . Figure 2.4 illustrates the HMM concept by showing a graphical representation of a 3-state model, where the set of hidden states'  $\{S_1, S_2, S_3\}$  transition probabilities are defined as a set of  $a_{ij}$ 's. Each state has a probability density function of emitting a specific observable.



**Figure 2.4: A Three-state Hidden Markov Model.**

An HMM is said to respect the first order Markov assumption if the transition from the current state to the next state only depends on the current state, i.e.  $P(S_j^{t+1} | S_i^t, S_m^{t-1} \dots S_n^0) = P(S_j^{t+1} | S_i^t)$ .

### **Computational Issues**

Three main computational issues need to be addressed with HMMs: model evaluation, most likely state path, and model learning. The first issue is the evaluation problem, i.e. the probability that a given sequence is produced by the model. This probability of a given sequence of data given the model is useful because, according to Bayes' rule, it is a proxy for the probability of the model given the data presented. We can thus compare different models and choose the most likely one by solving the evaluation problem. The evaluation problem is solved with the forward/backward dynamic programming algorithm. Let  $O^s$  be the  $s^{\text{th}}$  training sequence of length  $l_s$ , and the  $t^{\text{th}}$  symbol of  $O^s$  be  $O_t^s$ , so that  $O^s = \{O_1^s \dots O_{l_s}^s\}$ . We can define the forward probability  $\alpha_t(j)$  as the probability that the partial observable sequence  $O_1^s \dots O_t^s$  is generated and that the state at time  $t$  is  $j$ .

$$\alpha_t(j) = P(S_j^t, O_1^s \dots O_t^s ; H) \tag{4}$$

The forward probability can be recursively computed by the following method:

$$\alpha_t(j) = \sum_i^N a_{ij} b_j(O_{t+1}^s) \alpha_{t-1}(i), (t = 1, \dots, l_s) \tag{5}$$

$$\alpha_{l_s+1}(j) = \sum_i^N a_{ij} \alpha_{l_s}(i)$$

where  $\alpha_0(j) = 1$  if  $j$  can be the first state and  $\alpha_0(j) = 0$  otherwise.

Similarly, we can define the backward probability  $\beta_t(i)$  as the probability of the partial observable sequence  $O_{t+1}^s \dots O_{l_s}^s$  and that the state at time  $t$  is  $i$ .

$$\beta_t(i) = P(O_{t+1}^s \dots O_{l_s}^s | S_i^t ; H) \quad (6)$$

The backward probability can also be recursively computed as follows:

$$\beta_t(j) = \sum_j^N a_{ij} b_j(O_{t+1}^s) \beta_{t+1}(j), (t = l_s - 1, \dots, 0) \quad (7)$$

$$\beta_{l_s+1}(i) = \sum_i^N a_{ij} \beta_{l_s+1}(j)$$

where  $\beta_{l_s+1}(i) = 1$  if  $i$  can be the last state and  $\beta_{l_s+1}(i) = 0$ , otherwise.

We can now compute the likelihood that the given training sequence  $O^s$  is generated by HMM  $H$  and solve the state evaluation problem:

$$P(O^s ; H) = \sum_i \alpha_{l_s+1}(i) \beta_{l_s+1}(i) \quad (8)$$

This computation of the likelihood of a given sequence can also be seen as the computation of the probabilities along a lattice of hidden states. Figure 2.5 shows this lattice for 5 hidden states, the solid arrows represent the most likely path so far to state  $t$  and the dashed arrows represent the different predictions as to what the next most likely hidden state could be.

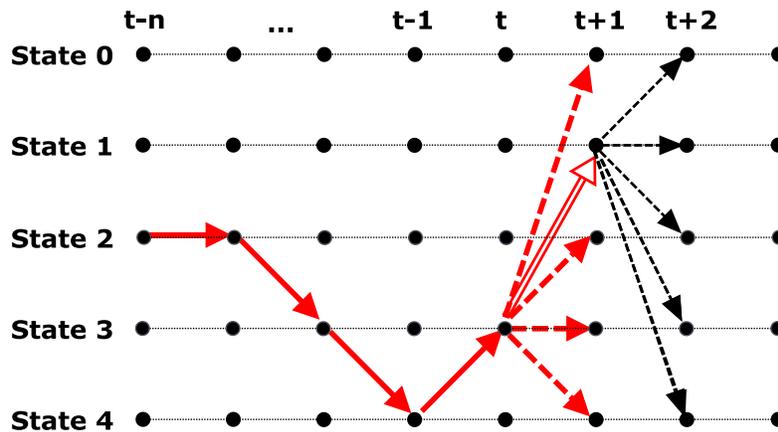


Figure 2.5 Progression through the lattice of hidden states

The second HMM computational issue consists of determining the most probable (“correct”) path of hidden states, given a sequence of observables. The most common way to solve this problem is by the Viterbi algorithm (Forney, 1973). The Viterbi algorithm is a dynamic programming algorithm that finds the most probable sequence of states  $S = \{S_1^1 S_2^2 \dots S_k^T\}$  given  $O = \{O_1^1 O_2^2 \dots O_k^T\}$  by using a forward-backward algorithm across the trellis of hidden states. More specifically, let  $\delta_t(i)$  be the highest probability path across all states which ends at state  $i$  at time  $t$ :

$$\delta_t(i) = \max_{S_1^1 S_2^2 \dots S_{t-1}^{t-1}} [P(S_t^t = i, O_1^1 O_2^2 \dots O_k^T ; H)] \quad (9)$$

The Viterbi algorithm uses a mechanism similar to the Forward/Backward algorithm and finds the maximum value of  $\delta_t(i)$  iteratively, and then uses a backtracking process across the hidden state lattice (Figure 2.5) to decode the sequence of hidden states taken along the path of maximum likelihood.

Finally the last computational problem is the learning of the model, such that given a sequence of observables, what is the maximum likelihood HMM that could produce this string? The parameters of an hidden Markov model  $H$ , i.e. the characteristics of the sequences of data being modeled, are trained to maximize  $\sum_S \log(P(O^S|H))$ , the sum of the posterior log-likelihoods of each training sequence  $O^S$ . For ease of notation, we introduce  $\gamma_t(i)$  as the probability that the sequence  $O^S$  is generated by the HMM and that the state at time  $t$  is  $i$ . We also define  $\xi_t(i, j)$  as the probability that the sequence  $O^S$  is generated by the HMM and that the state at time  $t$  and  $t + 1$  are  $i$  and  $j$  respectively:

$$\gamma_t(i) = P(S_i^t | O^S ; H) = \frac{\alpha_t(i)\beta_t(i)}{P(O^S ; H)} \quad (10)$$

$$\begin{aligned} \xi_t(i, j) &= P(S_i^t, S_j^{t+1} | O^S ; H) \\ &= \frac{\alpha_t(i)a_{ij}b_j(O_{t+1}^S)\beta_{t+1}(j)}{P(O^S ; H)}, (t = 0, \dots, l_S - 1) \end{aligned} \quad (11)$$

Define  $count(s \rightarrow s')$  as the number of times state  $s'$  follows state  $s$  and  $count(s \rightsquigarrow c)$  as the number of times state  $j$  is paired with emission  $c$ :

$$count(s \rightarrow s') = \sum_{j=1 \dots l_S-1} \llbracket S_j = s \wedge S_{j+1} = s' \rrbracket \quad (12)$$

$$count(s \rightsquigarrow c) = \sum_{j=1 \dots l_S} \llbracket S_j = s \wedge O_j = c \rrbracket \quad (13)$$

where  $\llbracket \cdot \rrbracket$  is the indicator function such that:

$$\begin{cases} \llbracket i = j \rrbracket = 1 \text{ iff } i = j \\ \llbracket i = j \rrbracket = 0 \text{ otherwise} \end{cases} \quad (14)$$

The Maximum Likelihood Estimates (MLEs) of  $\hat{a}_{ij}$  of  $a_{ij}$  then are:

$$\hat{a}_{ij} = \frac{\sum_{i=1 \dots N} \text{count}(s \rightarrow s')}{\sum_{i=1 \dots N} \sum_{s'} \text{count}(s \rightarrow s')} \quad (15)$$

Similarly, the MLE estimates  $\hat{b}_j(c)$  of  $b_j(c)$  are:

$$\hat{b}_j(c) = \frac{\sum_{i=1 \dots N} \text{count}(i, s \rightsquigarrow x)}{\sum_{i=1 \dots N} \sum_x \text{count}(i, s \rightsquigarrow x)} \quad (16)$$

The most commonly used algorithm for HMMs is a form of Expectation-Maximization (EM) called the Baum-Welch algorithm. The goal of the Baum-Welch algorithm is to maximize the posterior likelihood of the observed sequence  $O^s$  for a given HMM. More formally, Baum-Welch computes the optimal model  $H^*$  such that:

$$H^* = \underset{H}{\operatorname{argmax}} \left( \prod_s P(O^s ; H) \right) \quad (17)$$

Expectation maximization operates by hypothesizing an initial, arbitrary set of model parameters. These model parameters are then used to estimate a possible state sequence  $S_{O^s} = \{s^1 \dots s^{l_s}\}$  via the Viterbi algorithm. This is the expectation or E-Step of the EM algorithm. The model parameters are then re-estimated using Eq. (13) and (14), given the state labels  $S_{O^s}$ .

We could make the assumption that the state sequence  $S_{O^s}$  is correct. However, the state sequence can be uncertain if one or more of the most likely paths are close to being equiprobable. Thus, assuming that the state sequence is correct may lead to failures in determining the model parameters. The EM algorithm takes the uncertainty of the state sequence estimate into account by using the probability of being in state  $S_i$  at time  $t$  to estimate transition and emission probabilities. The probability  $\hat{a}_{ij}$  is re-estimated using  $\gamma_t(i)$  and  $\xi_t(i, j)$  based not on the frequency of state transitions from  $i$  to  $j$  in the data, but on the likelihood of being in state  $i$  at time  $t$  and the likelihood of being in state  $j$  at time  $t+1$ . Note that the frequencies or counts in Eq. (13) and (14) are not integer counts but likelihoods, and are therefore fractional. Similarly,  $\hat{b}_i(c)$  is re-estimated with  $\gamma_t(i)$  as the likelihood of being in state  $i$  when the observation was  $c$ . Through this iterative procedure, it can be proven that the Baum-Welch algorithm converges to a local optimum (Baum and Petrie, 1966).

One major shortcoming of HMMs is that they do not provide a way to explicitly deal with state durations. In fact, the probability of staying in a given state is structurally set to be geometrically distributed according to the state self-transition probability: the probability of staying in state  $i$  for  $j$  iterations is  $(a_{ii})^j$ . Assuming such a state sojourn distribution may not be valid in all contexts, which could be problematic in PHSC domains which often dictate that operators perform actions in time-pressured scenarios. Hidden semi-Markov models (HSMMs, also known as explicit duration hidden Markov models) address this specific issue (Rabiner, 1989; Guedon, 2003) and are discussed in the next section.

### 2.3.2 Hidden Semi-Markov Models

Structurally, a HSMM is similar to an HMM in that it is composed of an embedded Markov chain (usually first order) that represents the transitions between the hidden states  $\{S_t\}$ . In addition, an HSMM incorporates a discrete state occupancy distribution representing the sojourn time in non-absorbing states. The set of such distributions is noted  $D = \{d_j(u)\}$  and represents the probability of staying  $u$  units of time in state  $j$  which may be discrete or continuous. This thesis will focus on the discrete case. Figure 2.6 shows a 3-state hidden semi-Markov model including the sojourn distributions  $d_j(u)$  for all states.

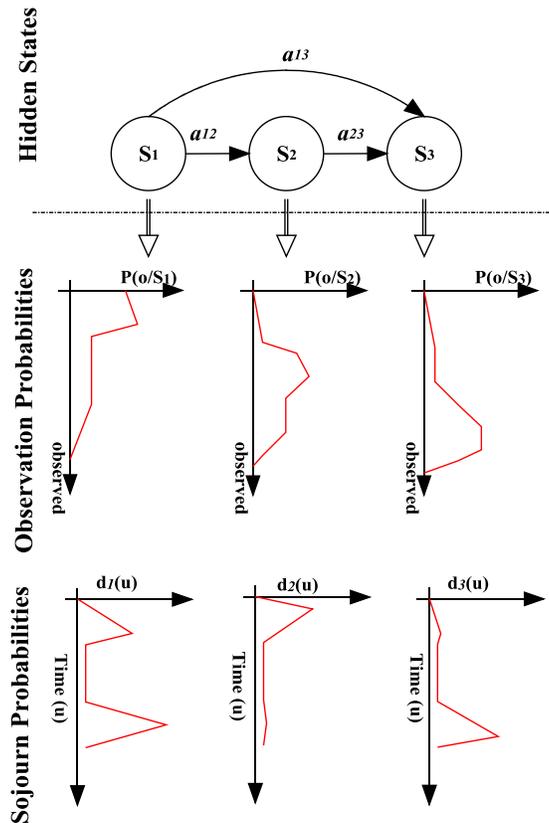


Figure 2.6: A 3-state hidden semi-Markov model

Formally, the sojourn distribution probability is defined as follows:

$$d_j(u) = P(S_{k \neq j}^{t+u+1}, S_j^{t+u-\nu}, \nu = 0, \dots, u-2 | S_j^{t+1}, S_{h \neq j}^t), \quad u = 1, \dots, M_j \quad (18)$$

where  $M_j$  is an upper bound to the time spent in state  $j$ . Then, assuming the process starts at  $t = 0$  in a given state  $j$ , the following relation is verified:

$$P(S_{k \neq j}^t, S_j^{t-\nu}, \nu = 1, \dots, t) = d_j(t) \pi_j \quad (19)$$

Equation (17) represents that the process enters a new state at time 0. The explicit expression of a state duration enforces that the underlying Markov chain contains no state self-transition. There can be no transition of the form  $a_{ii}$  in an HSMM as demonstrated in Figure 2.6. Furthermore, the conditional independence between the past and the future in HSMMs only holds when the process evolves from one state to another, while this property holds at each time step for HMMs. This distinction denotes the relaxation of the Markov assumption to a semi-Markov regime. However, due to their structural similarities, HSMMs can be unfolded and expressed as larger first order models with additional constraints due to the structure of the HSMMs (e.g., same distribution for parent and child states, can only to the next child state or a parent state different from the current one).

### ***Computational Issues***

Similarly to HMMs, the forward/backward algorithm is a central estimation mechanism for HSMMs. However, the addition of the duration probability makes the algorithm more complex than for HMMs. Guedon (2003) proposed a possible derivation of the quantities needed for the forward/backward algorithm. Recall that for HMMs:

$$\gamma_t(i) = P(S_i^t | O^s; H) = \frac{\alpha_t(i) \beta_t(i)}{P(O^s; H)} \quad (20)$$

For HSMMs, the formulation becomes:

$$\begin{aligned} \gamma'_j(t) &= P(S_{k \neq j}^{t+1}, S_j^t | O^s; H) \\ &= \frac{P(O^{t+1 \dots \tau-1} | S_{k \neq j}^{t+1}, S_j^t)}{P(O^{t+1 \dots \tau-1} | O^{0 \dots t})} P(S_{k \neq j}^{t+1}, S_j^t | O^{0 \dots t}) \\ &= \frac{\alpha'_t(i) \beta'_t(i)}{P(O^{t+1 \dots \tau-1} | O^{0 \dots t})} \end{aligned} \quad (21)$$

Eq. (21) has a form similar to Eq. (10) and can be separated into a forward and backward terms. The forward recursion can be written as:

$$\begin{aligned}
\alpha'_t(j) &= P(S_{k \neq j}^{t+1}, S_j^t | O^{0 \dots t}) \\
&= \frac{b_j(O^t)}{N_t} \left[ \sum_{u=1}^{\tau-1} \left\{ \prod_{v=1}^{u-1} \frac{b_j(O^{\tau-1-v})}{N_{t-v}} \right\} d_j(u) \sum_{i \neq j} a_{ij} \alpha'_{t-u}(i) \right. \\
&\quad \left. + \left\{ \prod_{v=1}^t \frac{b_j(O^{t-v})}{N_{t-v}} \right\} d_j(t+1) \pi_j \right]
\end{aligned} \tag{22}$$

The backward recursion is written as:

$$\begin{aligned}
\beta'_t(j) &= P(O^{t+1 \dots \tau-1} | S_{k \neq j}^{t+1}, S_j^t) \\
&= \sum_{k \neq j} \left[ \sum_{u=1}^{\tau-2-t} \beta'_{t+u}(k) \left\{ \prod_{v=0}^{u-1} \frac{b_k(O^{t+u-v})}{N_{t+u-v}} \right\} d_k(u) \right. \\
&\quad \left. + \left\{ \prod_{v=0}^{\tau-2-t} \frac{b_k(O^{\tau-1-v})}{N_{\tau-1-v}} \right\} d_k(\tau-1-t) \right] a_{jk}
\end{aligned} \tag{23}$$

where  $N_t$  is a normalization factor:

$$N_t = P(O^t | O^{0 \dots t-1}) \tag{24}$$

### ***Learning the Model Parameters***

The method for learning the parameters common to both HMMs and HSMMs (i.e. the sets of initial probabilities  $\pi = \{\pi_i\}$ , state transitions  $A = \{a_{ij}\}$  and emission distributions  $B = \{b_i(c)\}$ ) is the same as outlined previously for HMMs. The re-estimation formulation of the sojourn distributions  $D = \{d_j(u)\}$  is summarized as follows (Guedon, 2003):

$$d_j^{(k+1)}(u) = \frac{\eta_{j,u}^{(k)}}{\sum_v \eta_{j,v}^{(k)}} \tag{25}$$

where  $k$  is the re-estimation step and:

$$\begin{aligned}
\eta_{j,u}^{(k)} &= \left[ \sum_{t=0}^{\tau-2} P(S_{k \neq j}^{t+u+1}, S_j^{t+u-v}, v = 0, \dots, u-1, S_{l \neq j}^t | O^{0 \dots \tau-1}) \right] \\
&\quad + P(S_{m \neq j}^u, S_j^{u-v}, v = 1, \dots, u | O^{0 \dots \tau-1})
\end{aligned} \tag{26}$$

It is more convenient to estimate each of the two terms of this expression separately. In both cases, different cases must be considered depending on the value of the sojourn duration  $u$ . As explained in

(Guedon, 2003), care must be taken to set the boundary conditions properly for  $u = \tau - 2 - t$  and  $u = \tau - 1$  which correspond to the beginning and the end of a specific time interval.

## 2.4 Chapter Summary

This chapter describes the field of procedural human supervisory control and in particular focused on unmanned vehicle operations. UVSs are representative PHSC systems and are frequently time and mission critical. In addition, the shift towards single operators controlling multiple UVs reinforces the need for automatic, continuous monitoring of the operators as the consequence of operator failure is high.. However, such monitoring systems require models of expected operator behaviors so that anomalous behaviors can be detected and predicted. Such settings are well suited to pattern matching algorithms due to their procedural nature.

A review of previous pattern detections and modeling methodologies shows that hidden Markov and hidden semi-Markov models in particular, provide both an original research approach and an appropriate structure to model PHSC behaviors. In contrast, artificial neural networks do not provide model interpretability, support vector machines do not support temporal data and ARMA-based models cannot be used with categorical data.

The last section of this chapter discussed in detail the algorithms used to learn an HMM or an HSMM. However, in order to obtain a practical model from a given training data set, the typical process first involves learning a large number of different models and then selecting the most appropriate model from the obtained set of models. The next chapter presents this process in the context of both static and dynamic unmanned vehicle planning for a single operator, and evaluates the predictive capabilities of the resulting models.

*[Page intentionally left blank]*

## CHAPTER 3      HMMS OF SINGLE PHSC OPERATORS

---

*“In an information-rich world, the wealth of information means a dearth of something else: a scarcity of whatever it is that information consumes. What information consumes is rather obvious: it consumes the attention of its recipients. Hence a wealth of information creates a poverty of attention”*

*-Herbert Simon, 1971<sup>6</sup>*

The previous chapter provided the motivation for using statistical models, HMMs and HSMs in particular, for representing PSCH behaviors. This chapter provides the details of the methodology needed to learn HMMs from raw experimental human behavioral data. Each step of the methodology is first illustrated through its application to a static PHSC scenario. This scenario, StrikeView, is representative of generic static, automation-aided PHSC resource allocation tasks. Then, the same methodology is applied to a more complex dynamic scenario. The representative data set in this case is collected through RESCHU, a single operator, multi-UVPHSC scenario in a dynamic environment. Learning HMMs of operator behavior first in a static and then in a more complex dynamic environment allows assessing the scalability of the methodology.

### **3.1      Operator Models in a Static Environments**

This section discusses in detail the methodology proposed to obtain models of operator behaviors. Each step of the methodology is applied to a static resource allocation task called StrikeView, a mission planner for missile-target assignment.

#### ***3.1.1 StrikeView Interface Description***

A typical missile strike is planned by a coordinator whose main task consists in pairing a set of pre-planned missions with missiles of various capabilities available aboard different launchers such as submarines or cruising ships. This constitutes a complex, multivariate resource allocation problem, in which a human operator must not only satisfy a set of matching constraints, but also optimize the

---

<sup>6</sup> Simon, H. A. (1971), "Designing Organizations for an Information-Rich World", in Martin Greenberger, Computers, Communication, and the Public Interest, Baltimore, MD: The Johns Hopkins Press, ISBN 0-8018-1135

mission-missile assignments to minimize operational costs or enhance the quality of the overall plan. In an effort to decrease strike coordinators' workload and improve the quality of the resulting plan, a decision-support system called StrikeView was developed (Bruni and Cummings, 2005; Bruni and Cummings, 2006). StrikeView (Figure 3.1) allows an operator to create the solutions with the help of an automated planner. Although the interface is specific to the mission-missile assignment task, StrikeView is representative of most generic static resource allocation tasks such as human resource staffing or material warehousing.

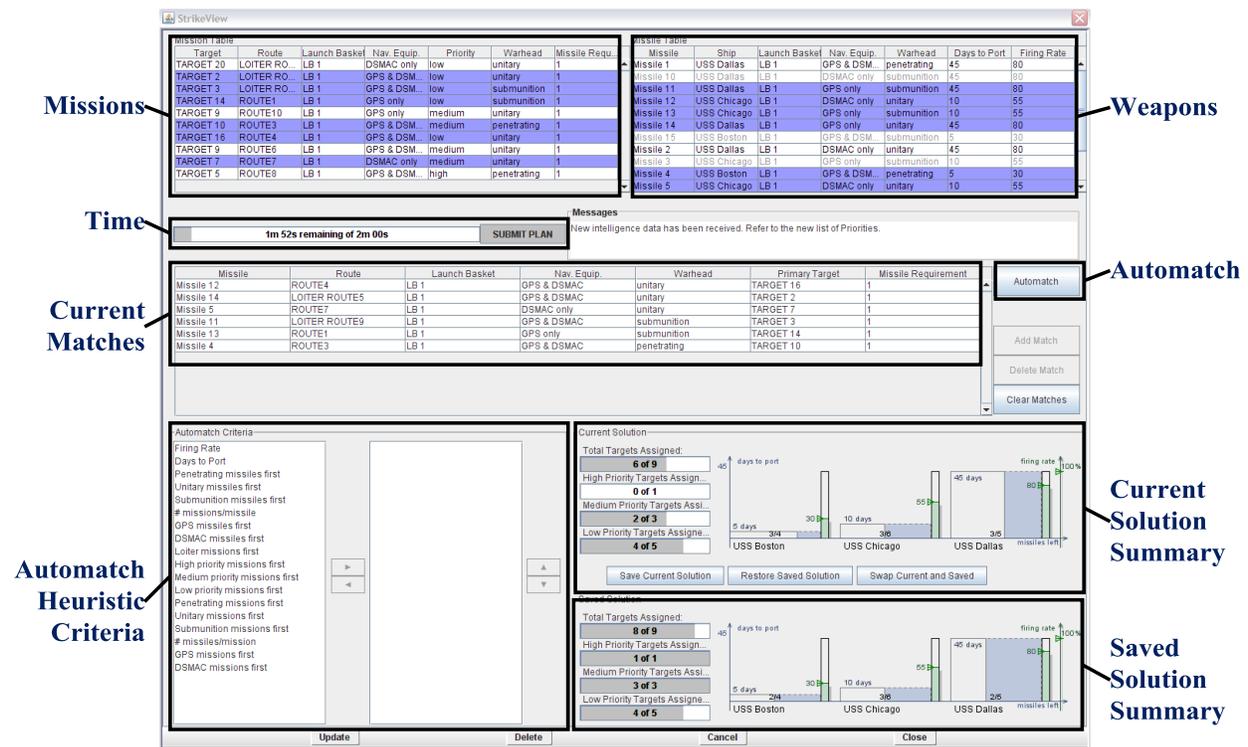


Figure 3.1. The StrikeView interface

The automated decision support function (called Automatch) provides the user with a heuristic-based computer-generated solution that only takes into account hard constraints along with a limited set of additional criteria which can be selected through the lower left portion of the interface. The solution provided is not guaranteed to be optimal, but it always exhibits correctness with respect to hard constraints. The lower right portion of the interface contains the summary information of both the current solution and of a previously saved solution. This allows two sets of solutions to be evaluated against each other. Finally, a time bar gives subjects a visual indication of how much time they have left to generate their solution.

Because the HMMs need a training data set, a user experiment was conducted in order to gather real behavioral data. The experimental population consisted of 10 MIT undergraduate students. Overall, 2050 user interface events such as mouse clicks on specific missions or missiles were collected from the 10 subjects.

### 3.1.2 Learning HMMs from PSCH Data

Learning the parameters of HMMs requires training the model on the observed behavioral data. For the purposes of this thesis which focuses on supervisory control in proceduralized environments, the raw behavioral data consists of logged user interface events and possibly communication data. This information cannot be used directly by the learning algorithms and must be pre-processed. Figure 3.2 shows how an HMM is built from raw data, which includes both a grammatical and a statistical phase. The grammatical phase translates the low level observed user interactions into abstract events, which then form the basis of the observable state space for the statistical phase. In this phase, the hidden Markov model learning algorithms are applied in order to obtain a model. These two phases are explained in more detail below and illustrated through their application to StrikeView.

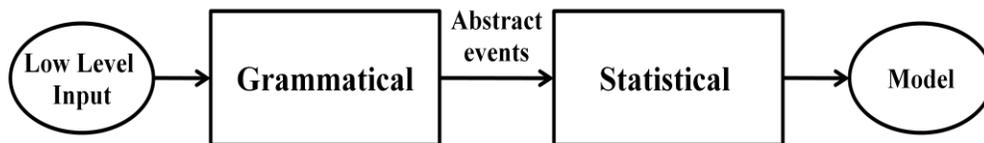


Figure 3.2 Two-stage learning for HMMs of PSCH behavior from experimental data

### 3.1.3 Grammatical Phase

The first step of the process consists of parsing low-level input information (such as mouse clicks on a screen) into abstract events according to a set of grammatical rules. The role of the grammar is thus to abstract low level user interface interactions into a set of meaningful tasks that can both be learned by the algorithm, as well as interpreted by a human modeler. Thus, the grammar represents feature extraction which reduces the size of the state space. It also defines the scope of the observable space usable by the machine learning process (Eads, Glocer et al., 2005). For application to PHSC settings, we propose that the grammar should take the form of a 2D space where the rows defines a set of operands (i.e. entities that are acted on) while the columns delineates a set of operations (i.e. what is being performed). A set of operations can be established through a general Task Analysis (Kirwan and Ainsworth, 1992) or a more specialized cognitive task analysis (CTA) (Schraagen, Chipman et al., 2000). This orthogonal

representation of the state space is essentially a generic ontology that represents type of objects and their relations.

Within the context of StrikeView, operator interactions were functionally grouped into seven operations (evaluate, backtrack, browse, select, filter, create and automatch), which represent the operations in this task. Since these operations could be carried out on different object abstractions (e.g., a user could elect to create a single match or a group of matches), these were crossed with what is termed “operands”, which included data item, data cluster, individual match, group of matches, individual criterion or group of criteria. The resulting 2D table represents the set of observables states for the algorithm (Table 3.1). For example, a click on a missile would be translated as a selection of a data item and deleting a previously created match would correspond to a backtrack action on an individual match. During the trials, the incoming raw events were parsed by a grammar, thereby encoding the raw events into intermediate level descriptors.

**Table 3.1 StrikeView grammar**

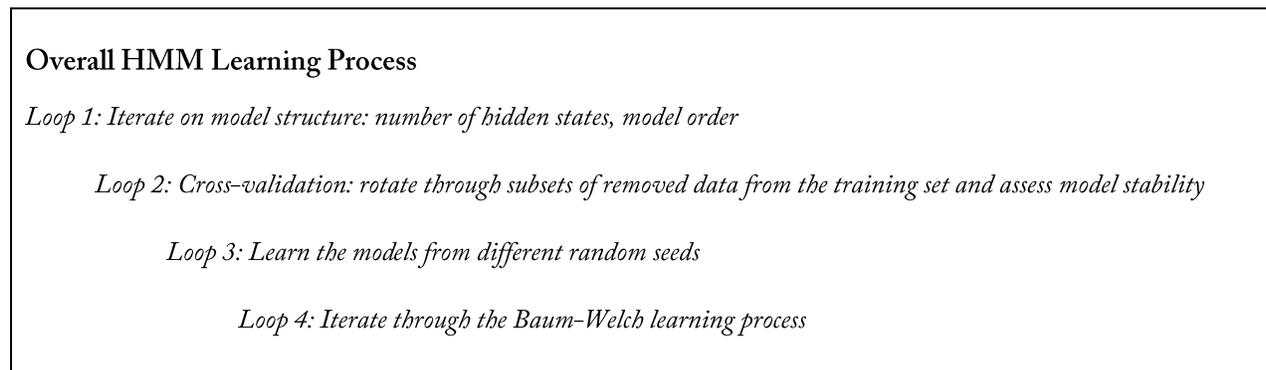
<i>Group of Criteria</i>							
<i>Individual Criterion</i>							
<i>Groups of Matches</i>							
<i>Individual Matches</i>							
<i>Data Cluster</i>							
<i>Data Item</i>							
<b><i>Operands / Operations</i></b>	<b>Evaluate</b>	<b>Backtrack</b>	<b>Browse</b>	<b>Select</b>	<b>Filter</b>	<b>Create</b>	<b>Automatch</b>

### ***3.1.4 Statistical Phase***

The learning algorithms for HMMs described in Chapter 2 assume that the model structure (e.g. the number of hidden states or model order) is known in advance. In most practical settings, this assumption is unrealistic and the structure of the model must be determined through a process called model selection (Burnham and Anderson, 2004). This involves first learning a number of models with varying structural properties and secondly, selecting the most likely model from the learned set.

### ***Model Learning and Selection***

Given a set of training data, the most likely HMM can be learned through the process illustrated in Figure 3.3. The outer loop iterates across a number of model structures. For HMMs, the structural differences consist of the number of hidden states embedded in the model and the order of the model. Then, the training data set is split and a number of sequences are reserved for cross-validation. Cross-validation is a technique used for assessing how a model obtained from a training data set will generalize to other unseen data sets. Generalizable models should be stable in the sense that they should not change significantly if a relatively small subset<sup>7</sup> of the training data is removed (Kohavi, 1995). Typically, multiple rounds of cross-validation are performed by rotating the sequences not used for training. Furthermore, because the Baum-Welch algorithm is akin to a gradient descent search (Baum and Petrie, 1966), models have to be learned from a large number of random seeds so as to avoid local minima. The number of random seeds used is usually balanced against computational requirements<sup>8</sup>. The Baum-Welch process is also iterative, and a set number of training iterations can be determined by measuring when the Kullback-Leibler distance (Bishop, 2006) between the models obtained across two successive iterations goes below a specific threshold.



**Figure 3.3 HMM Learning Process**

Once all the models have been obtained through the process described in Figure 3.3, precisely which model should be used remains to be determined. While there are many different criteria used to determine the quality of a model, an information-theoretic metric is adopted called the Bayesian Information Criterion (BIC) (Burnham and Anderson, 2002).

$$BIC = -2 \log(\mathcal{L}(H)) + P \log(K) \quad (27)$$

---

<sup>7</sup> The ratio of sequences used for training and cross-validation vary, but the number of reserved sequences would typically range from a single sequence (leave-one-out cross-validation) up to a quarter of the training set (k-fold cross validation).

<sup>8</sup> A typical number of random seeds used in this work is 10,000.

This metric allows for the comparison of different models, in particular with different number of hidden states, that are trained on the same underlying data. As shown in Eq. 27, the BIC penalizes the likelihood  $\mathcal{L}(H)$  of the model  $H$  by a complexity factor proportional to the number of parameters  $P$  in the model and the number of training observations  $K$ . Model selection through the BIC is a form of regularization<sup>9</sup> and corresponds closely to the notion of Occam's razor, or *lex parsimoniae*, which states that when competing hypotheses are equal in other respects, the principle recommends selection of the hypothesis that introduces the fewest assumptions and postulates the fewest entities while still sufficiently answering the question. In the context of statistical models, the use of the BIC supports the intuition that a model with fewer parameters is less prone to overfitting the training data and thus more likely to generalize to unseen data points. Figure 3.4 provides an illustrative example of the trade-off between model fit and complexity. In this example, a set of data points are generated from a linear function with added noise. Then, these points are fitted both by a linear regression ( $R^2 = 0.8245$ ) and by a 6<sup>th</sup> order polynomial ( $R^2 = 0.9325$ ). For example, in Figure 3.4 although the more complex 6<sup>th</sup> order polynomial appears to be a better fit to the training data in terms of  $R^2$ , the graphical representation of the polynomial seems to indicate that the additional parameters of this model fit the noise in the data such that the 6<sup>th</sup> order polynomial seems unlikely to generalize well to other unseen data points. This intuition is supported by computing the BIC of both models; the BIC of the linear regression is 1.24 whereas that of the 6<sup>th</sup> order polynomial is 5.45. These BIC results<sup>10</sup> show that, as expected, the simpler linear model is indeed a better fit for this data.

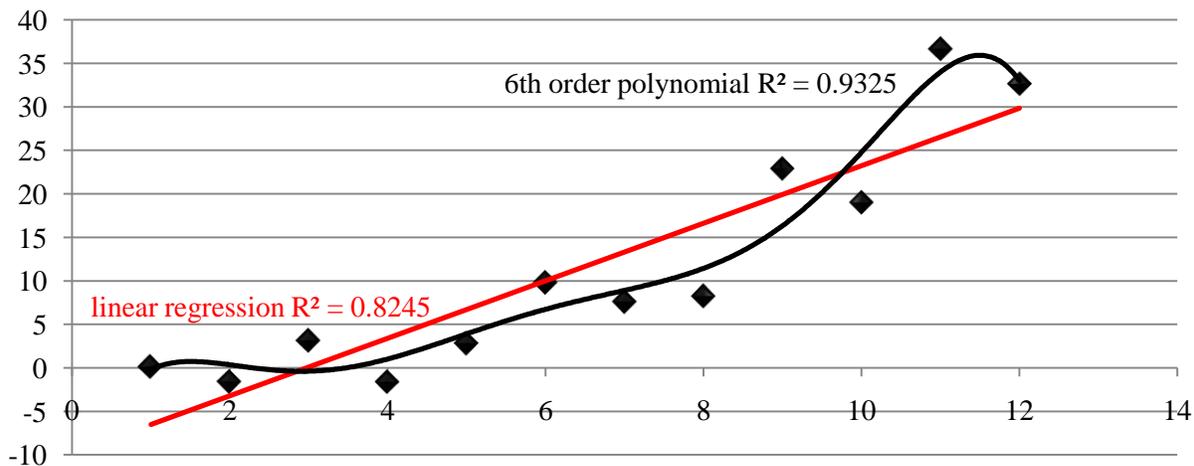


Figure 3.4 Model fit vs. model complexity

<sup>9</sup> Regularization involves introducing additional information in order to prevent overfitting. This information is usually of the form of a penalty for complexity, such as restrictions for smoothness or bounds on the vector space norm (Bousquet, Boucheron et al., 2004).

<sup>10</sup> Recall that lower values of the BIC metric imply better models.

### 3.1.5 StrikeView Models

Within the context of StrikeView, the methodology outlined in Figure 3.3 was used to learn a set of first-order HMMs from the training data gathered in the experimental sessions. Figure 3.5 shows the BIC score (see Eq. 27) for models of different sizes. The minimal value of the BIC occurs for the 5-state model ( $BIC = 6851.56$ ).

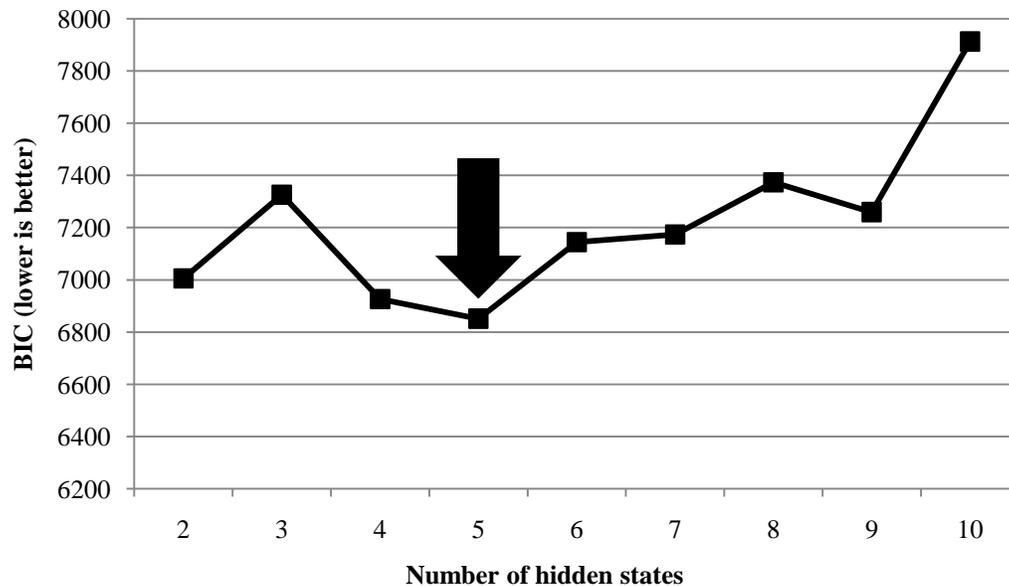


Figure 3.5 BIC scores for StrikeView models

The 5-state model extracted from the behavioral data is graphically shown in Figure 3.6. All the state transitions with probabilities less than 0.05 have been removed for legibility purposes. An analysis of each hidden state emission function provided the labels for the states. This process is illustrated in Table 3.2 shows the emission function of the third hidden state of the HMM for StrikeView. The emission function shows that the third state has a ~91% chance of producing the observable “Browse Data Cluster”, thereby providing the label for that state.

The model shows a number of interesting features. First, the action of “browsing a data cluster” is split into two separate states. This is interesting because there is a deterministic transition between these two states. In doing so, the HMM incorporates memory relating to performing the action of browsing a data cluster. The model thus suggests that there are always at least two consecutive “browse data cluster” actions unless the previous action was “evaluate data item & filter data cluster”. This corresponds to an information seeking procedure in which an operator filters a cluster and then searches, possibly repeatedly

for a desired match. Similarly, the high probabilities of the self-transitions of the states “select data item & create individual match” (0.78) and “backtrack data item” (0.702) suggest that these actions tend to occur in clusters.

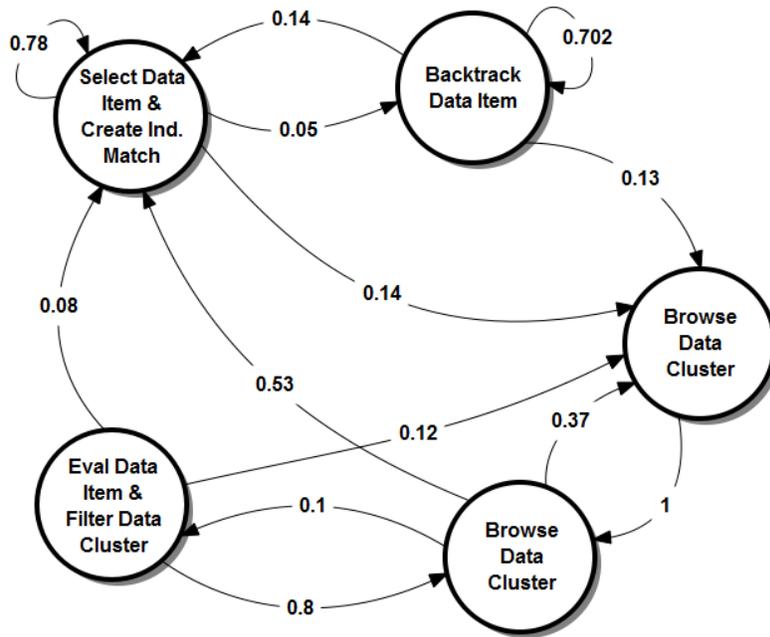


Figure 3.6 5-state HMM for StrikeView

Table 3.2 Emission function for state 3 for StrikeView

<i>Group of Criteria</i>	0	0	0	0	0	0	0
<i>Individual Criterion</i>	0	0	0	0	0	0	0
<i>Groups of Matches</i>	0	0	0	0	0	0	0
<i>Individual Matches</i>	0	0.003	0	0	0	0	0
<i>Data Cluster</i>	0	0	0.912	0	0	0	0
<i>Data Item</i>	0.085	0	0	0	0	0	0
<i>Operands / Operations</i>	<b>Evaluate</b>	<b>Backtrack</b>	<b>Browse</b>	<b>Select</b>	<b>Filter</b>	<b>Create</b>	<b>Automatch</b>

### 3.1.6 Model Validation

While the BIC score is a useful metric for comparing the goodness of different models, it is also important to validate that the model with the best BIC score captures the underlying event distribution present in the training data. In fact, using the BIC to select a model from a set of poor candidates will still yield a poor model. A practical measure to validate that the selected model is reasonable given a data set is the steady state distribution of observable events (McCane and Caelli, 2004). The steady-state distributions can be generated from the model through Monte-Carlo simulations. These distributions can then be compared via a  $\chi^2$ -test with that of the training data (Reiser and Lin, 1999). The better the model represents the training data, the more similar the simulated and training data will be.

Figure 3.7 shows the  $\chi^2$  values for the different 5-states models obtained during the cross-validation sequences on the StrikeView data set. None of the  $\chi^2$  values were significant ( $\chi^2 = 30.06, p = 0.85$ , was the largest  $\chi^2$  value,  $\chi^2_{crit} = 53.38, dof=38$ ). This means that the methodology provided models that represent their respective training data sets correctly while avoiding overfitting. It is therefore appropriate to assume that the models are properly trained.

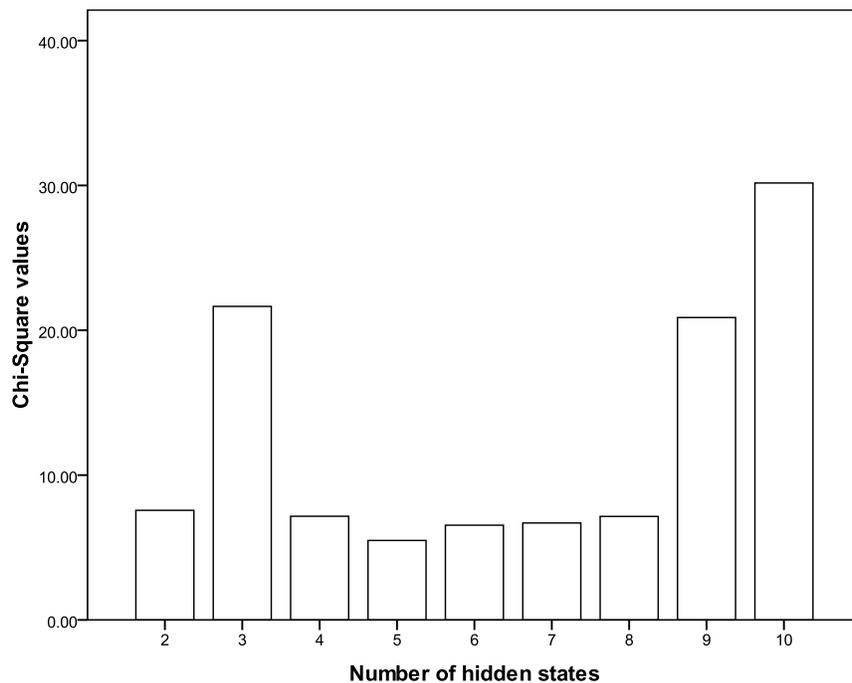


Figure 3.7 Model validation for StrikeView

In summary, this section described the details of a methodology capable of learning models of operator behaviors in static environments, in order to predict likely future behaviors. The most representative model of this behavior comprised 5 hidden states and was capable of accurate predictions ( $MAS = 88.11$ ). However, in the StrikeView scenario, the operators were faced with a single resource allocation task with a set of static constraints and low temporal stress. In contrast, most PHSC situations are characterized by their time-sensitive nature, and operators must perform replanning tasks in response to a dynamic environment, which is substantially more complex than the static mission planning scenario. The application of the proposed methodology to a dynamic mission planning data set, which incorporates these elements in an experimental scenario, is discussed in this section.

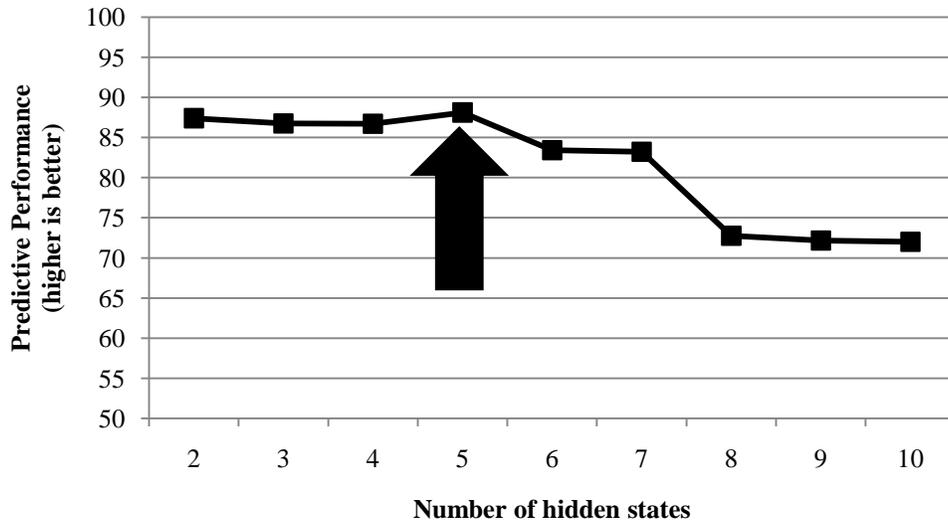
### ***3.1.7 Performance Evaluation***

The previous section presented a description of the 5-state model and valuable qualitative information was gathered from the structure of the model. However, the predictive capability of the models is the critical metric for their use in PHSC scenarios. The predictive performance of the model can be formulated as the accuracy of one-step-ahead observable predictions made by the model. In accordance to Huang's work (2009), the range of the prediction performance was chosen to be [50, 100] in order to promote a human operator's understanding by mimicking a prediction accuracy percentage where a score of 50 would mean no better than chance while a score of 100 would represent perfect predictions.

Specifically, the prediction performance is computed by determining if the current event is within the top five<sup>11</sup> predicted events at the previous iteration and scaled according to the ranking of the prediction. For example, if the current event was the top ranked in the predictions, the maximum score of 100 is assigned. Similarly, if the actual event corresponds to the 2<sup>nd</sup> most-likely event, a score of 90 is given. Thus, a predictive performance of 100 would mean that the actual event was always the top prediction. A predictive performance of 90 or greater would mean that the actual event was, on average, within the first 2 most-likely events. Figure 3.8 shows the predictive performance scores obtained by models of different sizes.

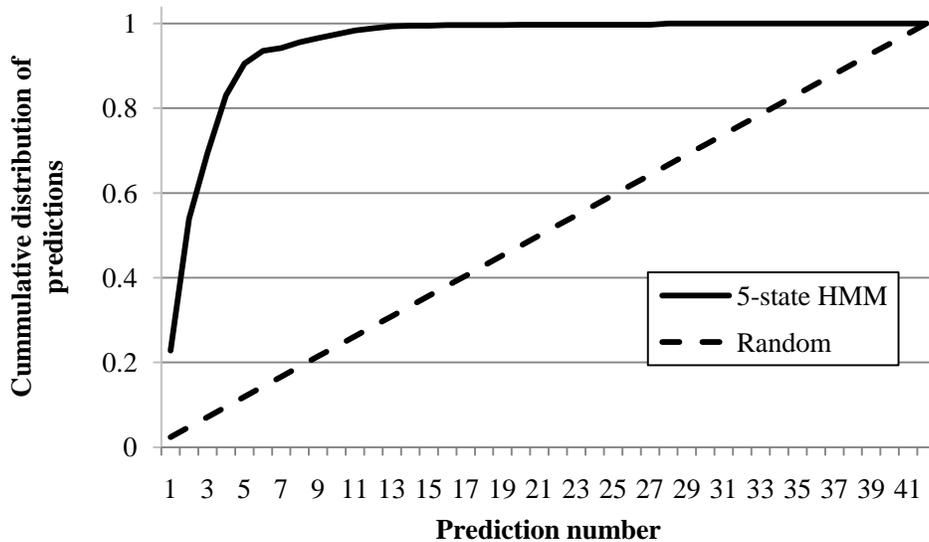
---

<sup>11</sup> Following Huang's work (2009), the top five events are considered in the metric in order to balance the penalty incurred for inaccurate predictions.



**Figure 3.8 Predictive performance for the StrikeView HMMs**

The results show that the selected 5-state model obtains the highest score of 88.11. The interpretation of this predictive performance is further illustrated in Figure 3.9 which shows the cumulative prediction rate for different prediction ranks for the 5-state HMM (solid line) and for a random model (dashed line). In particular, the figure shows that the first 5 predictions for the 5-state HMM covers 90% of all predictions. Therefore the increased prediction rate of the model compared to random is illustrated by the area between the solid and the dashed line.



**Figure 3.9 Cumulative prediction rate for StrikeView**

## 3.2 Operator Models in Dynamic Environments

This section discusses how the methodology introduced in Section 3.1 applies in dynamic PHSC environments. The Research Environment for Supervisory Control of Heterogeneous Unmanned-Vehicles (RESCHU) interface is used as a representative example of such scenario and can be generalized to other PHSC tasks in which iterative replanning must be performed in a dynamic setting.

The RESCHU data set was obtained from a previous experiment (Nehme, Crandall et al., 2008). While the goal of the original experiment was to validate a discrete event simulation model of an operator controlling multiple heterogeneous unmanned vehicles, the recorded user interface interactions represent a rich corpus of supervisory control behaviors.

### 3.2.1 RESCHU Interface Description

In the experiment, a single human operator controlled a team of UVs composed of unmanned air and underwater vehicles (UAVs and UUVs). The user interface is shown in Figure 3.10.

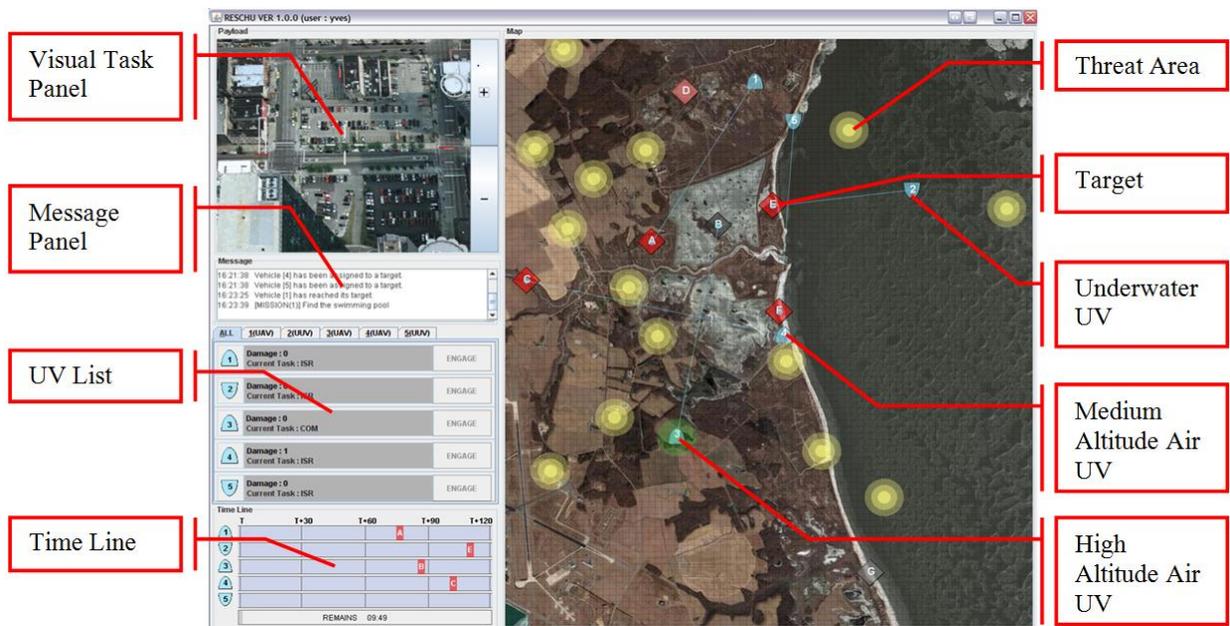


Figure 3.10: The RESCHU interface

In this interface, the UVs perform surveillance tasks with the ultimate goal of locating specific objects of interest in urban, coastal, and inland settings. UAVs can be of two types: one that provides high level sensor coverage (High Altitude Long Endurance or HALE), while the other provides more low-level

target surveillance and video gathering (Medium Altitude Long Endurance or MALE). In contrast, UUVs are all of the same type, with a similar goal of searching for targets of interest. Thus, the single operator controls a heterogeneous team of UVs which may consist of up to three different types of platforms.

In this simulation, the HALE performs a target designation task (simulating some off-board identification process). Once designated, operators use either the MALEs or UUVs to perform a visual target acquisition task, which consists of looking for a particular item in an image by panning and zooming the camera view. Once a target is visually identified, an automated planner chooses the next target assignment, creating possibly non-optimal target assignments that the human operator can correct. Furthermore, threat areas appear dynamically on the map, and entering such an area could damage the UV, so the operator can optimize the path of the UVs by assigning a different goal to a UV or by adding waypoints to a UV path in order to avoid threat areas.

Participants maximized their score by 1) avoiding dynamic threat areas, 2) completing as many of the visual tasks correctly, 3) taking advantage of re-planning when possible to minimize vehicle travel times between targets, and 4) ensuring a vehicle was always assigned to a target whenever possible. The data of interest for this work consisted of user interactions with the interface for a 10 minute experiment in the manner of clicks, such as operator UV selections on the map or on the left sidebar, waypoint operations (add, move, delete), goal changes and the start and end of visual tasks, as seen in Figure 3.10. Overall, the 48 subjects participating in the 10 minute long experiment yielded a data set containing 3420 data points.

### 3.2.2 RESCHU Grammar

Clusters of cognitive events were analyzed and resulted in the grammatical space shown in Table 3.3.

Table 3.3 RESCHU grammar.

<i>All UVs</i>						
<i>Underwater UV</i>						
<i>MALE</i>						
<i>HALE</i>						
<i>Operands / Operations</i>	<b>Select Sidebar</b>	<b>Select Map</b>	<b>Waypoint Edit</b>	<b>Waypoint Add/Del</b>	<b>Goal</b>	<b>Visual Task/Engage</b>

User interactions were first categorized by operands, i.e. the type of UV under control (All UVs, UUVs, MALEs or HALEs) and define the rows of Table 3.3. Then, the interactions with each of the UV types were separated into different operations in Table 3.3: selection on either the sidebar or on the map, waypoint manipulation (addition, deletion and modification), goal changes, and finally, the visual task engagement. These different operations define the columns in Table 3.3. The table of operands and operations represent all possible user interactions with the system.

### 3.2.3 RESCHU Models

Using the methodology outlined in Figure 3.3, a set of HMMs was learned for sizes ranging from 2 to 15 hidden states. Figure 3.11 shows the BIC scores of these models. The results show that the model with the best BIC of the learned set is the 8-state HMM ( $BIC = 12183.93$ ).

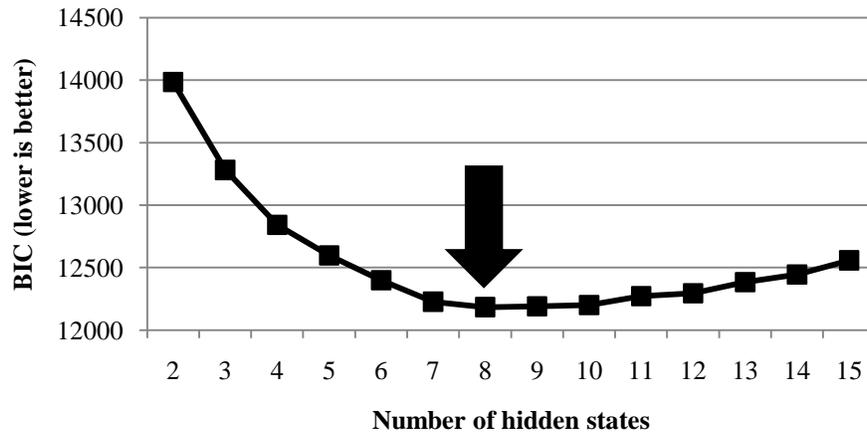
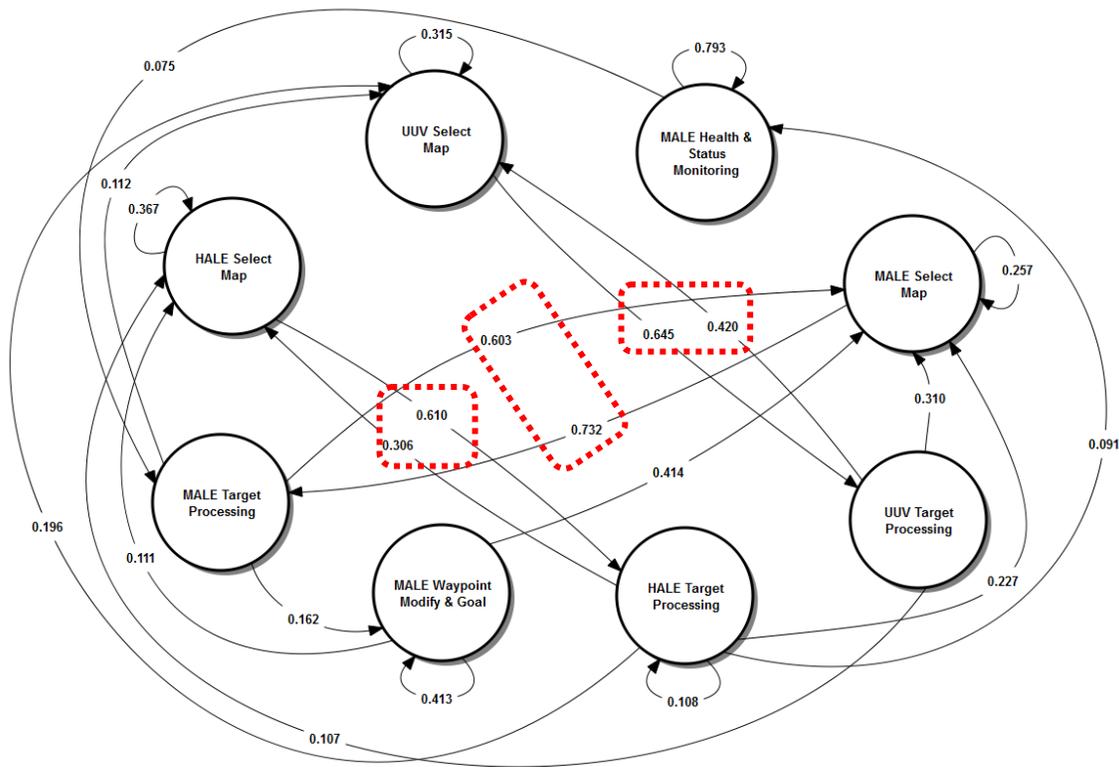


Figure 3.11 BIC score for the RESCHU model

Figure 3.12 shows a graphical representation of the 8-state HMM for RESCHU. All transitions with probabilities lower than 0.05 are removed for legibility purposes. Furthermore, the most likely transitions between states are purposefully graphed within the state graph and lower transitions are graphically routed on the outside of the state graph. This representation highlights the important loops between the states. The first observation of interest concerns the partitioning of states across the different types of UVs: UUVs are represented with 2 states, HALEs with 2 states, and MALEs with 4 states. This makes sense given the nature of the RESCHU scenario in which the interactions with the faster-moving MALEs were more frequent than with the other two slower types of UVs. This functional differentiation is represented by the number of states devoted to each type of vehicle.

Furthermore, it is interesting to look at the transition within each group, as defined by UV type, of states. For both UUVs and HALEs, the state structure and transitions are the same in that there is a strong cyclical loop between map selection and target processing<sup>12</sup> behaviors. This supports the idea that operators tend to pay attention to one specific type of UV before moving on to another type. The state structure for the MALEs is similar to that of UUVs and HALEs in that two of the MALE states also exhibit a similar cyclical loop between map selection and target processing. These cyclical loops are highlighted in Figure 3.12 by the dashed outlines for all three types of UVs. In addition, the MALEs are also represented by an additional two states. The first one corresponds to a specific state for waypoint modification and goal, which then mostly leads back to the main MALE cycle between map selection and target processing. Finally the last MALE state represents MALE health and status monitoring, which corresponds to the sidebar UV list selection as shown in Figure 3.10. These events were comparatively less frequent than the other events and this is reflected by the low probability of accessing this state.



**Figure 3.12 8-state HMM for RESCHU**

<sup>12</sup> Target processing corresponds to a cluster of Waypoint Add/Delete, Goal and Engage for a single type of UV.

### 3.2.4 RESCHU Models Validation

The results in Figure 3.13 show that none of the expected steady-state distributions from the models exhibit significant differences from the observed. At worst, the  $\chi^2 = 0.89, p > 0.999$  ( $\chi^2_{crit}=28.87, \text{dof}=18$ ). Similarly to the models obtained from the StrikeView data set, these results suggest that the proposed methodology is capable of generating appropriate models of single operator behaviors engaged in procedural human supervisory control in dynamic environments such as the one presented in the RESCHU task.

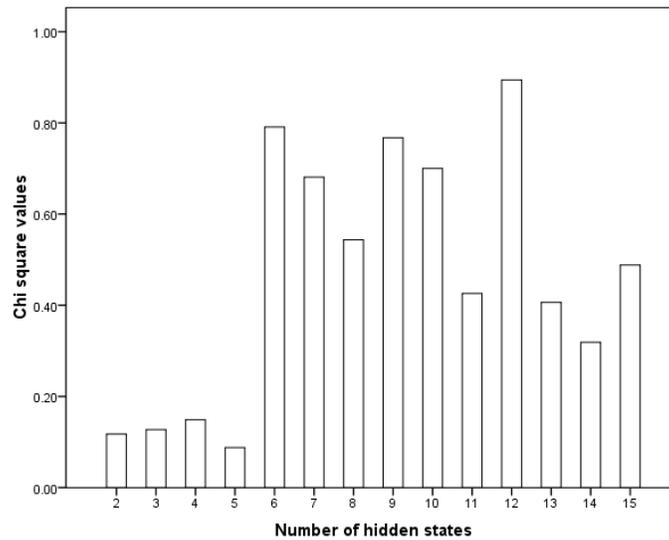
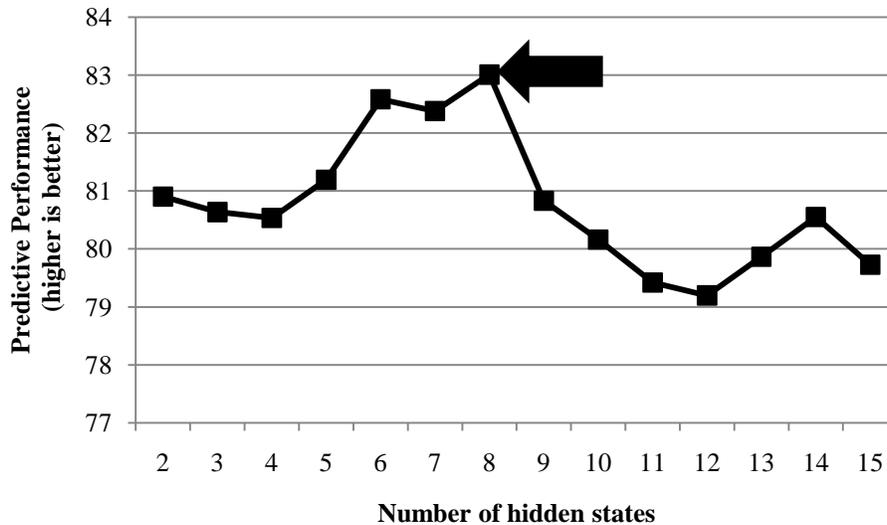


Figure 3.13 Model validation for RESCHU

### 3.2.5 RESCHU Performance Evaluation

The one-step-ahead prediction performance can again be used to measure the predictive performance of the HMMs learned for RESCHU. The prediction performance results in Figure 3.14 show that the 8-state HMM provides the highest score of 83.01. This result is slightly lower than that obtained in that static scenario (prediction performance was 88.11 for the 5-state HMM of StrikeView). This difference is likely due to the more complex dynamic nature of the RESCHU task.



**Figure 3.14 Predictive performance for RESCHU HMMs**

### 3.3 Chapter Summary

This chapter presented the methodology used to learn models of single operator behavior in PHSC settings. Each step of the methodology was first illustrated through its application to a static PHSC mission planning/resource allocation scenario called StrikeView. Then, the same methodology was applied in a more complex dynamic mission planning/resource allocation environment called RESCHU. For both data sets, the structure of the most representative HMM provided valuable insights regarding the behavior of the operators. In addition, the evaluation of the predictive performance showed that the obtained HMMs were capable of accurately predicting future operator behaviors. In fact, the increased complexity of the dynamic RESCHU scenario had minimal impact on the predictive performance of the models when compared to the simpler static case.

One of the structural weaknesses of using HMMs is that they do not explicitly exploit the state duration information, a critical factor in time-sensitive PHSC domains. The next chapter discusses how the HMM methodology can be extended to generate more complex models capable of using temporal data, and what the impact of the increased model complexity is on model predictive ability.

*[Page intentionally left blank]*

# CHAPTER 4 MODELING A SINGLE OPERATOR THROUGH HSMMS

---

*“The only reason for time is so that everything doesn't happen at once.”*

*–Albert Einstein, 1930<sup>13</sup>*

Hidden Markov models representations of single operator behaviors were presented in Chapter 3. HMMs, however, are structurally limited by an implied exponential distribution for the duration of the states. This assumption may be problematic in PHSC settings characterized by a time-sensitive nature. As outlined in Section 2.3.2, hidden semi-Markov models address this issue by explicitly modeling the state duration as a distinct probability function learned from the data, and as such, HSMMS may be particularly suited to time-critical supervisory control domains. This chapter examines the semi-Markov model learning process and presents the HSMMS obtained on the RESCHU data set. This chapter also introduces the Model Accuracy Score, a metric that can be used to measure the predictive capability of HSMMS.

## 4.1 Learning HSMMS for PHSC Data

The learning process for HSMMS is similar to that of HMMs described in Section 3.1.2. The process also consists of a grammatical and a statistical phase. While the grammatical phase remains unchanged, the statistical learning process must be adapted to fit the additional complexity required from the explicit expression of the state sojourn distribution.

### 4.1.1 HSMMS Complexity Analysis

The ability of HSMMS to extract information from timed-events<sup>14</sup> comes at the expense of model complexity since HSMMS typically need a significantly higher number of parameters than regular HMMs in order to explicitly represent the state durations as histograms. As can be expected, learning HSMMS is significantly more difficult than learning HMMs. The first issue is generalizability in that HSMMS contain significantly more parameters than HMMs with identical numbers of hidden states, and are

---

<sup>13</sup> New York Times Magazine (9 November 1930)

<sup>14</sup> In this thesis, the timing of the events is considered discrete and the event step size is determined in accordance to the maximum event rate in the data set in order to minimize errors due to the discretization process.

therefore more prone to overfit the training data (Guedon, 2003). An  $n$ -state HMM with a  $d$ -sized dictionary has  $n + n^2 + dn$  number of parameters ( $d \gg n$  usually). A similar HSMM with a maximum state duration  $M_t$  has  $n + n^2 + dn + nM_t$  parameters ( $M_t \gg d \gg n$  usually). For practical models (i.e. with relatively small  $n$ ), the dominant factors become the size of the dictionary  $d$  and the maximum state duration  $M_t$ .

In contrast with the size of the dictionary, the maximum state duration can be traded against time resolution granularity because it is computed in terms of time-steps. Obtaining fine-grained time resolution (i.e. multiple time-steps per second) can become expensive if some states have long durations. The higher number of parameters means that achieving a parsimonious and generalizable model is difficult and requires more training data, often a problem in small sample settings. Additionally, from a purely computational perspective, learning the model can be impractical. Looking at the cost of a forward/backward pass, a  $n$ -state HMM with a  $d$ -sized dictionary will typically have a run-time of  $O(n^2d)$ . In contrast, the same run-time for an HSMM with a maximum state duration  $M_t$  will be  $O(n^2dM_t^2)$  (Mitchell, Harper et al., 1999). As mentioned earlier,  $M_t \gg d \gg n$  is a typical scenario, so computation time can be a significant problem for HSMMs.

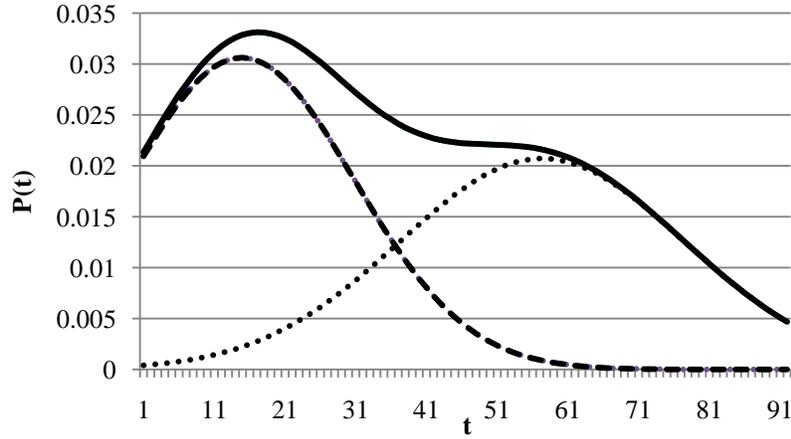
The solution to both of these problems, i.e. model generalizability and computational complexity, lies in reducing the number of parameters that need to be learned. As shown earlier, a significant proportion of the number of parameters in an HSMM is devoted to defining a set of sojourn distributions explicitly represented as histograms. One way to reduce the number of parameters in the model is to use parameterized distributions (i.e. having a closed form), such as Gaussian mixture models, in order to describe the sojourn probabilities (Marin, Mengerson et al., 2005). This reduction promotes model generalizability and reduces the computation load at the cost of imposing additional constraints on the expression of the sojourn probability distribution function.

#### ***4.1.2 Sojourn Distributions as Gaussian Mixture Models***

Gaussian mixture models (GMMs) are defined as a weighted sum of independent normal distributions. The GMM definition of the sojourn duration is as follows:

$$d_j(u) = \sum_{k=1}^{M_m} \phi_{jk} * \frac{e^{-\frac{(u-\mu_{jk})^2}{2\sigma^2}}}{\sqrt{2\pi}\sigma_{jk}^2} \quad (28)$$

where  $M_m$  is the number of modes in the GMM and  $\phi_{jk}$  represents the weighting parameter of the  $k^{th}$  Gaussian in state  $j$ , which has a mean  $\mu_{jk}$  and a standard deviation  $\sigma_{jk}$ . A graphical representation of a 2-mode Gaussian mixture model (solid line) is shown in Figure 4.1 along with its 2 Gaussian sub-components (dashed lines). The GMMs parameters, i.e.  $\phi_{jk}$ ,  $\mu_{jk}$  and  $\sigma_{jk}$ , can be learned by a process of expectation maximization identical to the one used for the other parameters of the HSMM. For a GMM with a single mode, the solution can be computed by taking the partial derivative of the  $Q$  function and setting it to 0 (Marin, Mengersen et al., 2005).



**Figure 4.1 Example of a bimodal Gaussian mixture model**

For GMMs with more than one mode, the derivation of the re-estimation equations remains similar to the single mode case, with the additional requirement of computing the appropriate weight parameter  $\phi_{jk}$ . The first step is to derive the  $Q$  function, which is the expectation of the log of the sojourn probability (Eq. 29):

$$\begin{aligned}
 Q_j &= \sum_{k=1}^{M_m} \sum_u \eta_{ju} * \log d_j(u) \\
 &= \sum_{k=1}^{M_m} \sum_{u \in T_k} \eta_{ju} \left( \log \phi_{jk} + \frac{(u - \mu_{jk})^2}{2\sigma_{jk}^2} - \log(\sqrt{2\pi}\sigma_{jk}^2) \right)
 \end{aligned} \tag{29}$$

The mean re-estimation is:

$$\frac{dQ_j}{d\mu_j} = 0 \rightarrow \mu_j = \mu_{jk} = \frac{\sum_{u \in T_k} u \eta_{ju}}{\sum_{u \in T_k} \eta_{ju}} \tag{30}$$

The standard deviation re-estimation is:

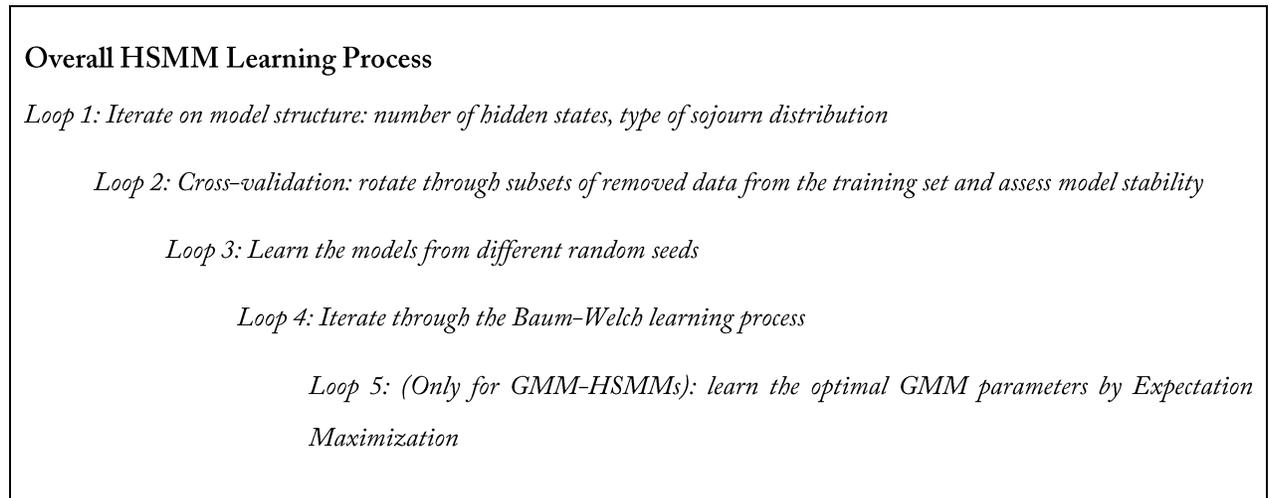
$$\frac{dQ_j}{d\sigma_j} = 0 \rightarrow \sigma_j = \frac{\sum_{u \in T} \eta_{ju} (u - \mu_j^*)^2}{\sum_{u \in T} \eta_{ju}} \quad (31)$$

The re-estimation formulae for the scaling function  $\phi_{jk}$  must be evaluated separately for each number of modes by setting the derivative of the  $Q$  function to 0 for the different modes:

$$\frac{dQ_j}{d\phi_{jk}} = 0 \quad (32)$$

### 4.1.3 HSMM Learning Process

The detailed HSMM learning algorithms provided in Section 2.3.2 assume a known model structure. This is not the case in practical settings and a model selection process similar to the one used for HMMs must be established. Figure 4.2 provides the HSMM version of the HMM process established in Figure 3.3. There are two main changes in the process. First, the model structure can be iterated along different number of states as well as along the type of sojourn distribution (histogram-based vs. parametric, and if parametric, the distribution to be used). Within the scope of this thesis, the parameterized distributions will be limited to GMM because most human processing times have been shown to follow normal distributions (Carroll, 1993). Secondly, should a parametric distribution be used for expressing the state durations, an inner loop needs to be added to the learning algorithm in order to find the most likely parameters of the sojourn distributions.



**Figure 4.2 HSMM Learning Process**

Finally, the optimal model can be chosen from the set of learned models via the BIC method (Eq. 27), which balances out the model fit and model complexity.

## 4.2 HSMM of RESCHU

The HSMM model learning methodology shown in Figure 4.2 is applied to the RESCHU data set, and both histogram-based and parametric models are learned. The parametric models use Gaussian mixture models with up to 3 modes<sup>15</sup> in order to express the state sojourn probability function. Similar to HMMs, a number of HSMMs needs to be learned and the best one can be chosen through the process of model selection.

### 4.2.1 Model Selection

The results in Figure 4.3 show that the BIC scores of the histogram-based HSMMs (from 2 to 10 hidden states) are higher than that of any GMM-HSMM, regardless of the model size. The poorer scores of the histogram-based HSMM are due to the large number of parameters needed to specify every point in the distribution of the sojourn time.

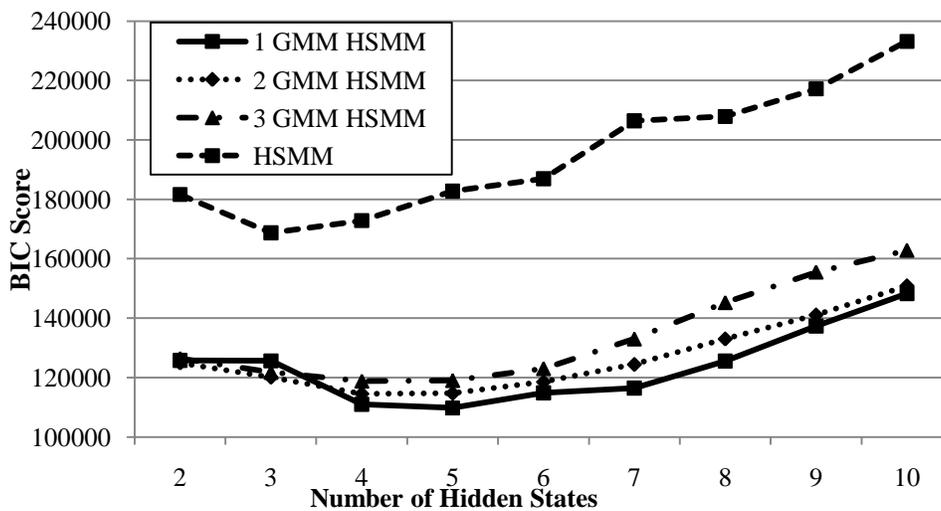


Figure 4.3 BIC scores (lower is better) for the HSMMs and GMM-HSMM of different sizes

In contrast, the GMM-HSMMs, regardless of the number of modes used to specify their sojourn distribution, have fewer parameters and their BIC scores indicate that they are likely to generalize better than their histogram-based counterparts. Within the group of GMM-HSMMs, we see a similar trend where the simpler models tend to have a better BIC score. Thus for the RESCHU data, the BIC metric indicates that a 5-state 1-mode GMM-HSMM used to define the sojourn distribution provides the best

<sup>15</sup> A maximum of 3 modes for the GMM-HSMM was chosen because hidden states in HMMs and HSMMs typically represent less than 3 different modes of operation.

HSMM model for this particular UV application, and that requiring full specification of all the parameters of the sojourn time distribution can be detrimental to model generalizability. However, using a parametric function to specify this distribution also imposes an additional assumption with regards to the form of the sojourn time expression. This may not be appropriate in applications that require highly specific time distributions, i.e., very tight tolerances for user interactions.

While the BIC of the 5-state 1-mode GMM-HSMM is the lowest of all HSMMs with a score of BIC=109775 (Figure 4.3), the best HMM model trained on the same data set, built around 8 hidden states (see Figure 3.12), is an order of magnitude lower (BIC=13420). Although the BICs cannot be compared directly due to the rescaling of the training data with the HSMM time resolution, the results suggest that the less complex HMMs are likely to generalize better to unseen data than HSMMs. HMMs, however, are not capable of using and providing timing information data, which are often critical in PSCH settings. Thus, whether to use HMMs or HSMMs presents a trade space between external validity and model fit.

#### 4.2.2 Selected Model

Figure 4.4 presents an overview of the selected model, highlighting the different hidden states while graphically showing the state transition probability matrix.

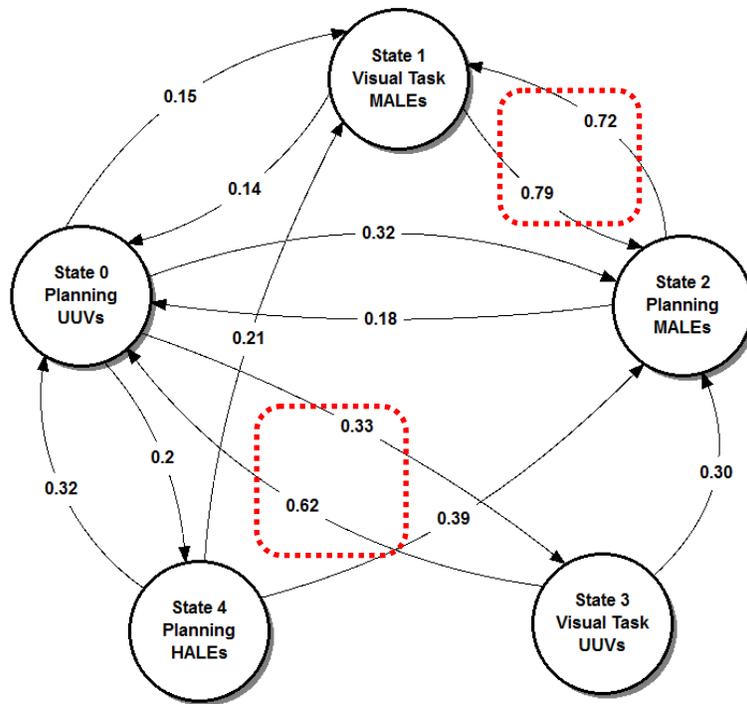
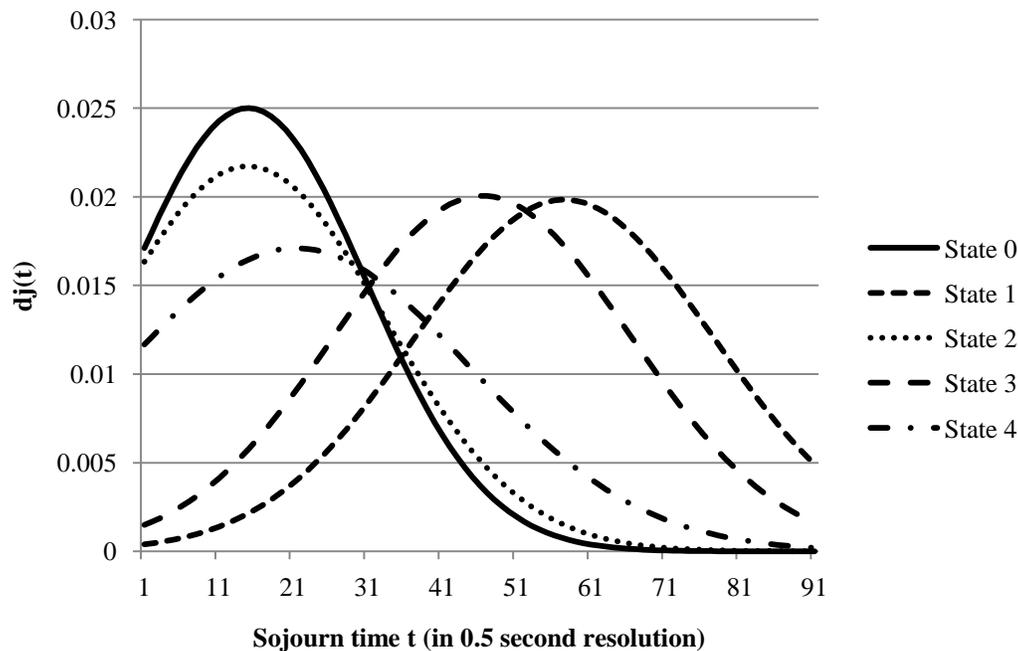


Figure 4.4 Transition probabilities in the 5-state 1-mode GMM-HSMM

The 5 hidden states of the selected GMM-HSMM along with the state transition probabilities  $A = \{a_{ij}\}$  are presented. All the transitions with less than 5% probability have been removed for legibility purposes; all states are otherwise fully connected. Note that because HSMMs explicitly model state durations, there are no self-transitions for the hidden states. The hidden states are also labeled according to their emission functions. The transition probabilities between the hidden states provide valuable insight into operator behavior. As highlighted in Figure 4.4, the model suggests that the planning and visual task states are heavily linked both for UUV and MALE types, and therefore expresses the idea that operators alternate regularly between these two activities. For both types of UVs, there is a high likelihood of engaging in planning behavior with a vehicle of the similar type after a given visual task (0.79 for the MALEs and 0.62 for the UUVs). This demonstrates that the first action an operator does after finishing a visual task is to send the vehicle towards another target, a typical replanning strategy for RESCHU. While the transition between the planning and visual tasks for the MALEs is strong (0.72), the transition between UUV planning and the visual task is comparatively weaker (0.33). This result is not surprising as UUVs are slower vehicles in RESCHU. Thus, an operator is less likely to perform a visual task right after retasking such a vehicle because the UUV will take longer to reach the assigned goal.



**Figure 4.5 Hidden state sojourn probabilities**

Figure 4.5 shows the duration distribution functions  $D = \{d_j(u)\}$  for the different states of the 5-state model in Figure 4.4. The x-axis is labeled in 0.5s intervals because this is the time resolution needed to parse out all the events in distinct discrete time steps. In other words, at most 2 events happened in the same second in the training data set, and therefore a time resolution of 0.5s is needed to put them in different time intervals. The y-axis shows the probability of staying in a given state for a duration  $x$ . Figure 4.5 demonstrates that the planning tasks (states 0, 2, and 4) require, on average, much less time to accomplish than the visual tasks of states 1 and 3, which agrees with observed data (as well as real world UAV operations). The mean duration of a planning state is 8.03s whereas the mean duration of the visual task states is 25.43s. These sojourn times thus present distinctly separate modes of operator behavior.

In addition, one of the most interesting features of the model is that three of five the hidden states in Figure 4.4 represent operator planning and replanning operator behavior with each of the three types of UVs (HALEs, MALEs and UUVs). The last 2 hidden states represent the visual tasks for both MALEs and UUVs (recall HALEs do not perform a visual task). The expected durations of the visual task states for MALEs and UUVs are comparatively longer (28.5s and 22.5s respectively) than that of the interaction states (around 8s). The fact that the learning algorithm was able to segregate the visual task states as different from the planning states highlights the insights that can be obtained from patterns contained in such a data set.

In comparison to the simpler 8-state HMM, the additional complexity of the HSMM given the same amount of data resulted in a 5-state 1-mode GMM-HSMM. Thus, while HSMMs may provide less detailed synthesis of an operator's sequence of action, the explicit modeling of the state durations provides timing information which may be critical in time-sensitive PHSC domains.

Overall, the qualitative interpretation of the model selected as the most likely is consistent with the task and suggests that the learning algorithm was capable of extracting coherent and valuable information from the sequences of behavioral data used in model training.

### ***4.2.3 Model Validation***

Similarly to regular HMMs, the HSMMs model can be validated by verifying their steady state distributions. Figure 4.6 shows that the  $\chi^2$  values were all non-significant ( $p > 0.75$ ) for all models with respect to the experimental data. This suggests that the trained models captured the underlying distributions properly. In addition, these results show that the distributions generated by the models are not statistically different from the data, and therefore support the conclusion that the 5-state 1-model

GMM HSMM is a valid representation of the operator behavior in controlling the multiple heterogeneous UVs. However, the histogram-based HSMM tends to produce smaller count deviations than the GMM-HSMMs. By increasing the number of modes used by the GMM HSMM, the parametric models tend to approximate their histogram-based counterparts. However, modeling the state durations with more than one mode does not seem to provide worthwhile added value as measured by the BIC. In practical terms, these results suggest that the states tend to exhibit homogenous timing characteristics, reflecting the single-operation nature of the hidden states (e.g. planning or visual task). Other environments may lead to multi-mode distributions if the hidden states aggregate multiple operations.

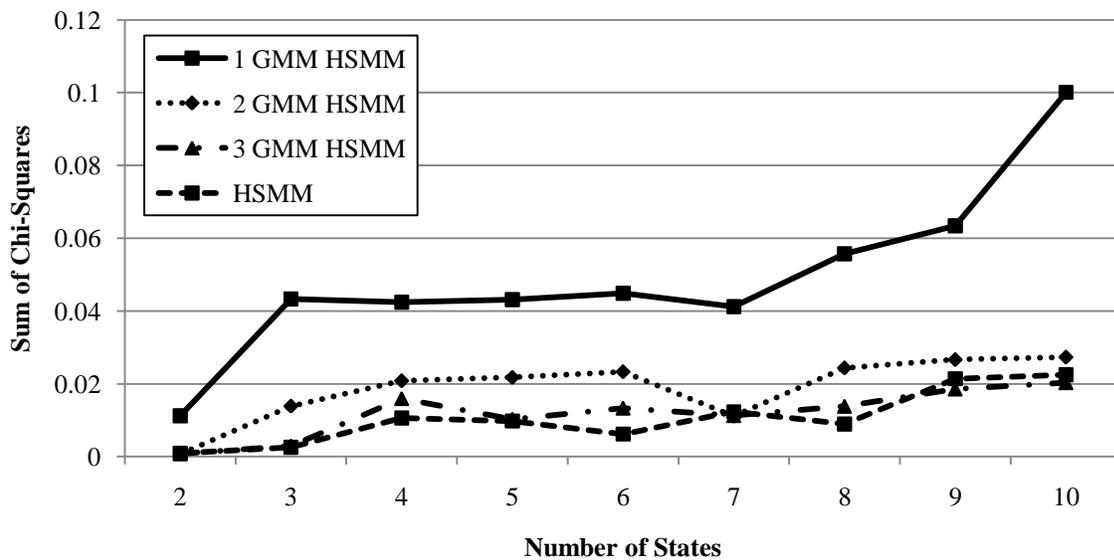


Figure 4.6 Validation for HSMMs and GMM-HSMM of different sizes

#### 4.2.4 Model Evaluation and MAS

While the qualitative analysis of the model description is interesting, the real value of using such models lies in their predictive abilities. Models capable of accurately predicting future operator behavior could be of great value in PHSC settings which often can be life or mission critical. In order to measure model predictive capability, we introduce the Model Accuracy Score, a metric that weighs the quality and timing of the predictions according to a weighting parameter  $\alpha$ .

##### MAS Metric

Measuring the predictive capabilities of a regular HMM is a straight-forward process. The next  $n$  actions can be predicted from the model parameters and verifying if the predictions are correct is straightforward.

However, measuring the predictive capabilities of HSMMs is more complex than for HMMs because the predictions are made on two independent dimensions: the first measures if the predicted event is correct (or at least of high probability), and the second dimension measures the timing of the prediction, i.e., did the prediction timing coincide with the occurrence of next event? The Model Accuracy Score (Huang, 2009), or MAS, is an aggregate metric that considers both dimensions, i.e. quality and timing of the predictions. The MAS assesses the predictions capability of a model according to the following equation:

$$MAS(t) = \frac{\sum_{i=t, \dots, t-n} \alpha \times Quality(i) + (1 - \alpha) \times Timing(i)}{n} \quad (33)$$

The MAS is a running average of  $n$  subscores, where the  $\alpha$  parameter is the weighting factor used to balance the respective importance of quality and timing of the predictions. In the current application, we determined that  $n = 10$  provided a good balance between smoothing and sensitivity, and the effects of the  $\alpha$  parameter on the results will be discussed in the following section. The range of the MAS is [50, 100], and each MAS sub-score is computed every time a user event is logged. The range of values of the MAS was chosen to promote a human operator's understanding by mimicking a prediction accuracy percentage where a score of 50 would mean no better than chance while a score of 100 would represent perfect predictions. For example, a MAS of 90 indicates that the model's prediction of the human's next action in controlling the unmanned vehicles is well within the set of expected actions (both in actual state transition and in timing of action). Conversely, a MAS of 50 predicts that the next action is outside the expected set of states or required time window for action. However, it is unlikely that a single MAS prediction is useful, as it is a running average and decision-makers will likely require further context and a temporal representation of the MAS to make an informed decision.

The MAS comprises two sub-scores that represent quality and timing. The quality of the prediction is computed by determining if the current event is within the top five<sup>16</sup> predicted events at the previous iteration and scaled according to the ranking of the prediction. For example, if the current event was the top ranked in the predictions, the maximum score of 50 is assigned. In contrast, if the event is the 5<sup>th</sup> in the ranking, a score of 10 is assigned, and any event out of the top five is given a score of 0. Thus, the quality sub-score of the prediction is exactly equivalent to the prediction performances metric used to evaluate models in Chapter 3. The timing of the prediction is evaluated by measuring the difference between the predicted and the actual state duration. Specifically, that difference is measured in terms of number of standard deviations away from the predicted mean state duration, both of which can be

---

<sup>16</sup> Following Huang's work (2009), the top five events are considered in the metric in order to balance the penalty incurred for inaccurate predictions.

computed from the  $D = \{d_j(u)\}$  distributions. The timing score is not penalized if the event happens within one standard deviation before or after the prediction. The 1 standard deviation standard was chosen because given the means and standard deviations of the durations of the planning and visual task operator states (states 0, 2, 4 and states 1 and 3 respectively), the chances of type I and II errors were 8% and 9% respectively for the most distant states (states 0 and 1), low by human modeling standards. Any timing deviation further than one standard deviation is penalized according to the Gaussian cumulative tail probability. For example, if an event arrives within one standard deviation of the predicted, the assigned score for the timing of the prediction is 50. In contrast, should a state duration be between 1 and 2 standard deviations away from the predicted, the timing score received will be 27.2, which is computed based on the area under the Gaussian curve between 1 and 2 standard deviations. Deviations larger than 3 standard deviations from the predicted mean receive a 0 score for the timing metric. This process is summarized in Figure 4.7.

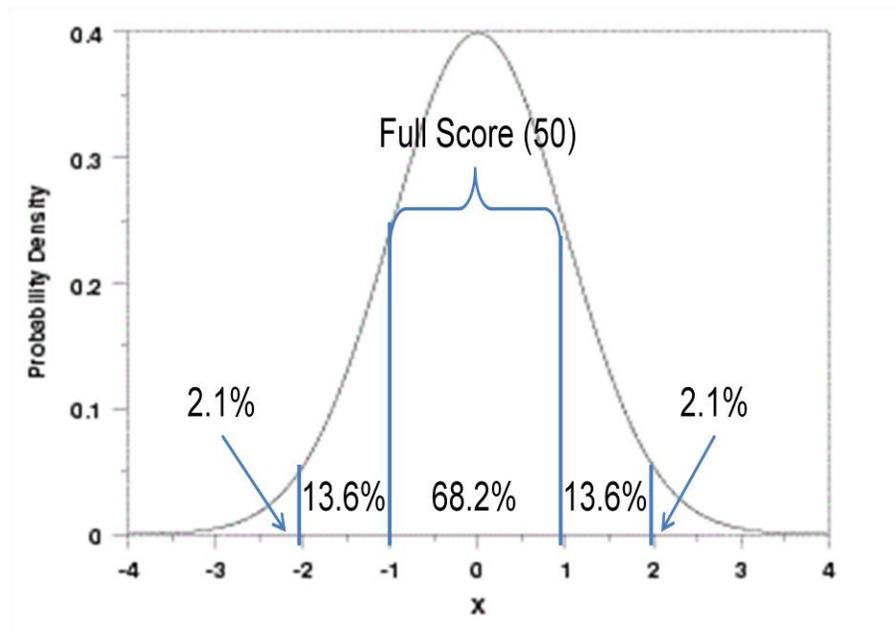


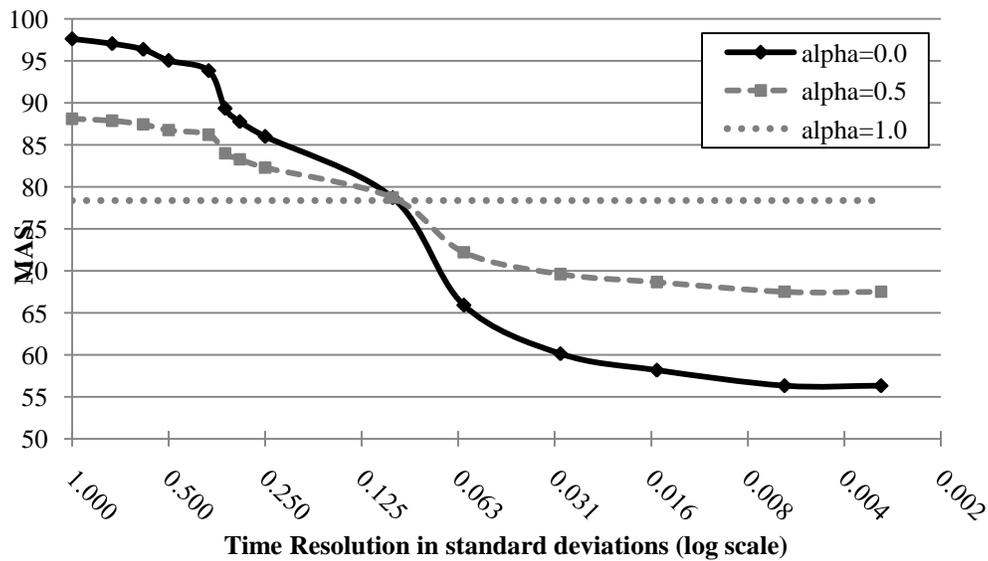
Figure 4.7 Timing score scaling with a resolution of 1 standard deviation (Huang, 2009)

### ***MAS Sensitivity***

Figure 4.8 explores the sensitivity of the MAS to the weighting of the quality and timing of the predictions sub-scores (in particular the 1 standard deviation rule for the timing sub-score), both of which are essentially subjective components. Specifically, Figure 4.8 shows the MAS obtained for the 5-state 1-mode GMM HSMM given different values of  $\alpha$  (0.0, 0.5 and 1.0) and different time resolutions ranging from 0.003 to 1 standard deviation. With an  $\alpha$  value of 1.0, the MAS only considers how well the model is able to predict the next events with no consideration of timing. In this case, the value of the MAS is

unaffected by the change in time resolution as shown by the constant MAS of 78.31. In contrast, with an  $\alpha$  value of 0.0, the MAS only measures how well the states durations are predicted, and is more sensitive to the changes in time resolution. Finally, an  $\alpha$  value of 0.5 weighs both quality and timing of the prediction equally.

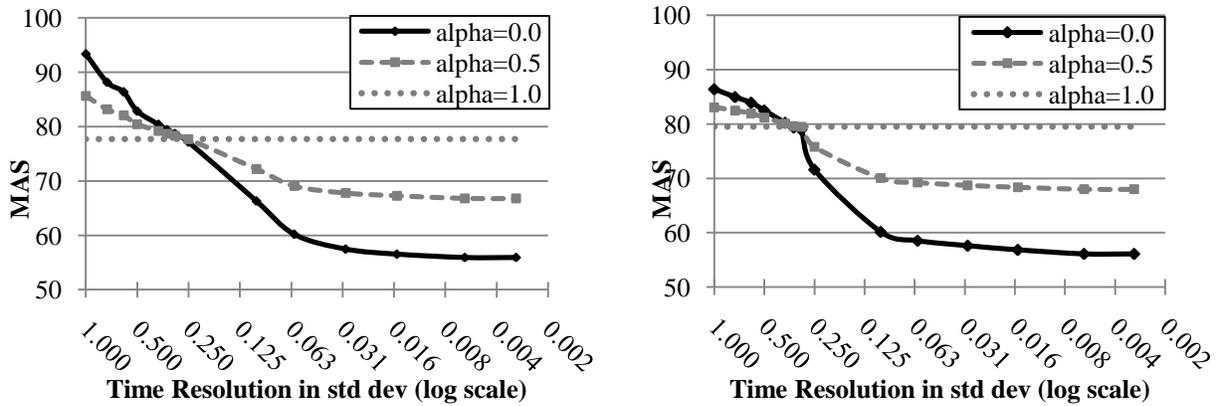
As expected, the MAS scores decrease monotonically with finer-grained time resolution due to the increased penalty for falling outside of the full-score interval. For all values of  $\alpha < 1.0$ , the maximum MAS is obtained when the time resolution is 1 standard deviations (97.31 for  $\alpha = 0.0$  and 88.10 for  $\alpha = 0.5$ ). The MAS values then decrease and plateau for time resolutions finer than 0.03 standard deviations with MAS ranging from 55 to 60 for  $\alpha = 0.0$  and from 69 to 67 for  $\alpha = 0.5$ . The MAS curves obtained for different values of  $\alpha$  intersect the ordinate at the constant MAS score obtained for  $\alpha = 1.0$  (78.31) and the abscissa at a time resolution of 0.1 standard deviations. This intersection marks the time resolution setting at which the timing part of the metric stops contributing to the MAS. Finer-grained resolutions lead to a decreased MAS due to more stringent penalties for inaccurate predictions. In other words, at a time resolution smaller than 0.1 standard deviations, the timing of the prediction becomes inaccurate compared to the quality of the prediction and the overall MAS score is decreased.



**Figure 4.8: MAS for the 5-state 1-mode HSMM given different time resolutions and  $\alpha$  values**

A similar analysis can be carried out for models of different sizes. Figure 4.9 (a) and (b) show the same results as Figure 4.8 but were obtained given a 4-state 1-mode GMM HSMM and a 6-state 1-mode GMM HSMM, the 2 closest models to their 5-state counterpart that exhibited the highest BIC score. Figure 4.9 (a) shows that the 4-state 1-mode GMM HSMM can provide accurate timing predictions for time

resolutions ranging from 1 to 0.25 standard deviations. Similarly, the 6-state 1-mode GMM HSMM provides accurate timing predictions for resolution up to 0.33 standard deviations (Figure 4.9 (b)).

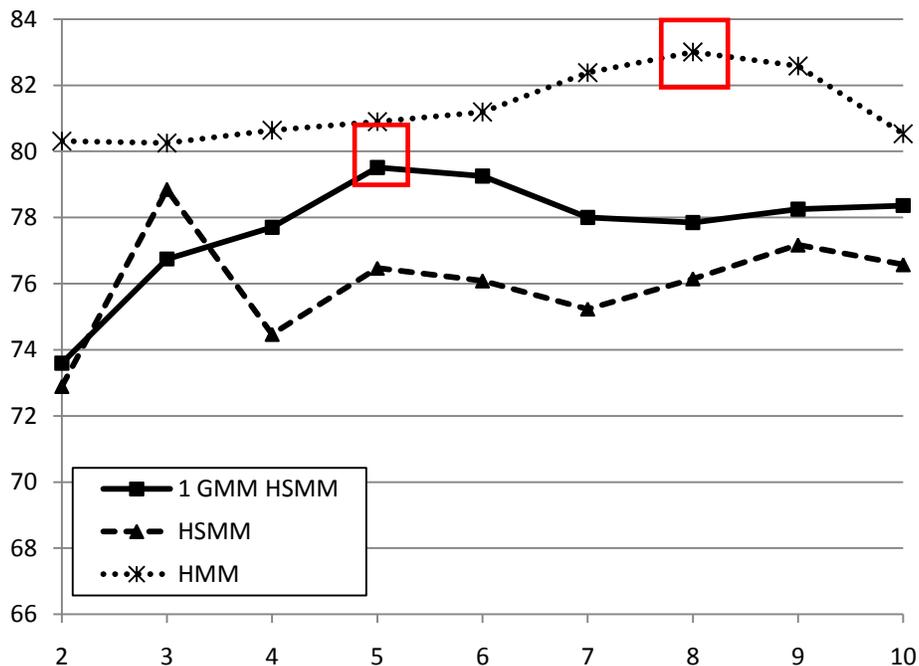


(a) 4-state 1 mode GMM HSMM

(b) 6-state 1-mode GMM HSMM

**Figure 4.9: MAS for the 4- and 6-state 1-mode HSMM given different time resolutions and  $\alpha$  values**

For reference, Figure 4.10 compares the MAS scores of regular HMMs, 1-mode GMM HSMMs and non-parametric HSMMs trained on the same data set when  $\alpha = 1.0$ . The comparison is only valid for this specific value of  $\alpha$  because HMMs cannot provide timing information and therefore their MAS can only consider the quality of the prediction.



**Figure 4.10: MAS with  $\alpha = 1.0$  for 1-mode GMM HSMMs, HSMMs and HMM of different sizes**

Figure 4.10 shows that the MASs of the HSMMs are generally lower than that of the HMMs, which means that the additional number of parameters that need to be learned to specify  $D = \{d_j(u)\}$  hinders the HSMM ability to accurately predict state durations. More specifically, the simpler HMM models provide marginally higher MAS (MAS=83.0 for an 8-state HMM vs. MAS=79.5 for the 5-state 1-mode GMM-HSMM) at the expense of not providing timing information. Thus, comparing the predictive capability of the HMM and the selected GMM-HSMM provides insight in the practical consequences of using a larger number of parameters to define the model. Setting the value of  $\alpha$  to 1.0 provides a valid basis for comparison against an HMM because the metric then does not require measuring the timing of the prediction, information that the HMM is not capable of providing.

The MAS scores of the HMM were higher than any of the GMM-HSMMs. This performance delta highlights the trade-off between simple models which focus solely on quality of the prediction and more complex HSMM models which incorporate timing information. In our specific case, the difference in performance was marginal between the HMMs and the HSMMs (i.e., 3.5 points on the MAS scale). Still, our results suggest that simpler HMMs are likely preferable in non-time critical applications. HMMs are simpler and perform marginally better than HSMMs at predicting next states, while being more computationally manageable and better capable of generalizing from a smaller training data set. However, HSMMs are capable of providing valuable information for time-sensitive applications that HMMs cannot, and for our UV application, the impact of the increased complexity on state predictions was relatively minor.

### **4.3 Chapter Summary**

This chapter presented the methodology for learning a set of hidden semi-Markov models in procedural human supervisory control settings, which led to the selection of the optimal model using an information theoretic measure. The selected model not only captured the underlying distribution of events in the training data, but also segregated qualitatively different behaviors (such as a differentiating a visual task from a planning action). This chapter also showed how the predictive capability of such HSMM models in PHSC settings can be evaluated via a Model Accuracy Score, which is a flexible aggregate metric that weighs both quality and timing of the predictions. An analysis of the MAS scores in an applied setting demonstrated that the HSMM model is capable of reliably predicting the next observable state both in terms of quality and timing of the prediction. While the results show that HSMMs can be used to model

and predict operator behaviors in PHSC environments where temporal information is critical, HSMMs tend to be more complex than HMMs. Thus, if timing information is not specifically required for such a model, HMMs may be preferred because they are simpler and therefore likely to generalize better to unseen data.

So far, these results have concentrated on HMMs and HSMMs of single PHSC operators. However, operators rarely work in isolation: they typically work in teams. The next chapter applies the methodologies shown in the current and previous chapters for single operators and scales the task complexity to data sets that represent that represent the behaviors of teams of operators.

*[Page intentionally left blank]*

## CHAPTER 5 TEAM MODELS OF PHSC OPERATORS

---

*“In the long history of humankind (and animal kind, too) those who learned to collaborate and improvise most effectively have prevailed” –Charles Darwin 1871<sup>17</sup>*

The previous two chapters presented HMMs and HSMMs of single PHSC operator behavior. However, most current UV operations are performed by multiple operators in team structures. From a qualitative perspective, group behaviors are typically more complex than single operator behaviors. In addition, team behavioral data differ from that of single operator in two significant ways. First, the operators not only interact with the computer interface but also with other operators. Communication data therefore needs to be taken into account in addition to the UI events. Secondly, because teams comprise multiple members, the overall emergent team behavior may exhibit more complex patterns. In light of these differences, the goal of this chapter is to discuss how the proposed methodology scales to team environments.

In order to guide this discussion, this chapter first presents how the single operator modeling approach is modified so as to extend to a team context. Then, a set of HMMs and HSMMs are developed from two separate team data sets. The first data set, Team-RESCHU, is a 3-person team version of the RESCHU game. The second data set was obtained from an Air Force Research Lab/Human Effectiveness Directorate (AFRL/HE) experiment in which teams of five operators participate in an air battle defense simulation. Finally, the team models are compared with the single operator models, and the overall scalability of the proposed methodology is discussed.

### 5.1 Modeling Approach

Because the goal in this chapter is to see how the proposed methodology scales to teams of operators, the modeling procedure follows the same grammatical and statistical steps outlined previously in this thesis. In addition, the overall team is modeled holistically as a single entity in order to provide team models that are comparable to single operator models. Therefore, the states are not individual “operator states” but “team states”. This is an important distinction because it implies that the multiple, simultaneous operator behaviors are serialized in a single time sequence for analysis. The implications and limitations of this modeling structure are discussed further in the next chapter.

---

<sup>17</sup> The Descent of Man, and Selection in Relation to Sex (1st ed.), London: John Murray, ISBN 0801420857

## 5.2 Team-RESCHU

Team-RESCHU, a team version of the RESCHU simulation was created by Mekdeci et al. (2009). In this simulation, teams of 3 operators have access to 3 different types of UV. Each operator uniquely prosecutes a specific type of target: friendly, enemy or unknown. With these 3 types of UVs, the team of operators has to process contacts that appear intermittently over a map. Should an unidentified target appear on the map, operators have to dispatch a scouting UV capable of labeling the target as either friendly or enemy. Then, depending on the assigned label, operators have to determine whether to engage the target either by delivering aid packages or dropping weapons. Figure 5.1 shows the main display through which the operators direct vehicles and coordinate with other team members.



Figure 5.1 Team-RESCHU main display

Determining which unmanned vehicle to assign to a particular task requires some level of coordination amongst the operators. The medium for such coordination is a text “chat” messaging channel in which operators can broadcast written messages to the rest of the team (no voice communication was allowed). In addition, each operator can monitor the entire set of UVs by actively requesting positional updates of the other team members’ UVs. The overall objective for each team is to process the maximum number of contacts appearing on the map in a set amount of time.

### 5.2.1 Team-RESCHU Grammar

The grammar used to translate game events into a set of observable events was devised according to the principled methodology presented in Section 3.1.3. However, due to the nature of the team collaboration, the rows that define the operands in the single operator case is modified to distinguish between the different operators. The columns are unchanged and define the set of possible operations for the operators. For Team-RESCHU, the set of possible operator actions were 1) move own UV, 2) monitor team members' UVs, 3) engage a target, or 4) chat (Table 5.1). Within the scope of this data set, the content of each chat message was not analyzed, only the discrete event occurrences were considered. This approach was chosen because the nature of the communications was very homogeneous, i.e., operators generally just discussed coordinating which UV to send to a target.

Table 5.1 Team-RESCHU grammar

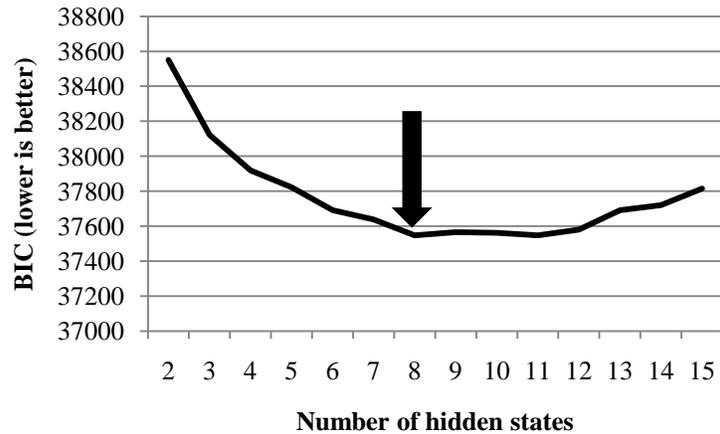
<i>Operator 2</i>				
<i>Operator 1</i>				
<i>Operator 0</i>				
<b>Operators/ Operations</b>	<b>Move</b>	<b>Monitor</b>	<b>Engage</b>	<b>Chat</b>

### 5.2.2 Experimental Subjects

Ten 3-member teams were recruited for the experiment and were between the ages of 18 and 35 (mean 21.7). The teams were then given two 10 minute-long practice scenarios before proceeding with the real set of 4 10 minute-long experiments. This data set consists of 40 10-minute long experimental sessions, which yielded a data set containing 8116 events.

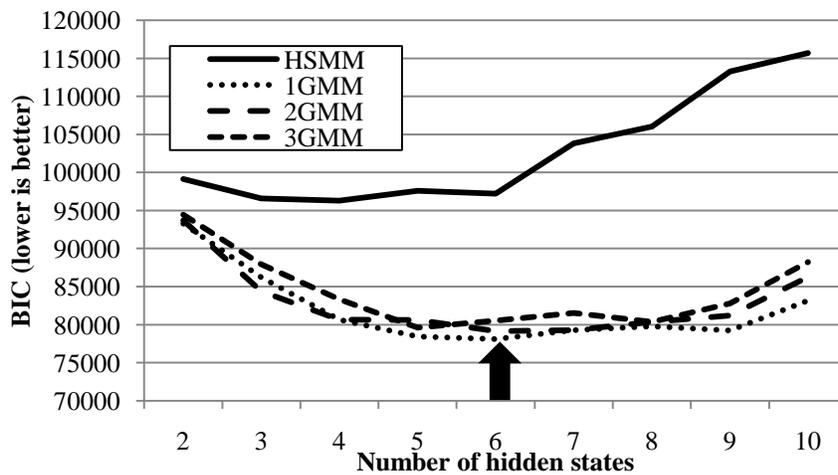
### 5.2.3 Team-RESCHU Models

Models of team behaviors in the Team-RESCHU environments can be learned using the algorithms described in Chapter 2. Figure 5.2 shows that the most representative HMM comprises 8 hidden states for a BIC of 37546.38. In contrast, Figure 5.3 shows that the 6-state 1-mode GMM HSMM (BIC=78099.43) provides the best balance between training data fit and model complexity for HSMMs. In comparison, the BICs of the histogram-based HSMMs are significantly higher than those of the GMM-HSMMs. In contrast, the BICs of the GMM-HSMMs tend to behave similarly and therefore seem to have a comparatively smaller impact. These relationships between HMMs, HSMMs and GMM-HSMMs are thus similar to the single operator case.



**Figure 5.2 BIC for HMMs of Team-RESCHU**

The difference in BIC of the most representative 8-state HMMs and the 6-state 1-mode GMM HSMMs suggests that the HMMs provide models that are likely to generalize better to unseen data. In fact, the simpler structure of the HMMs balances the higher number of hidden states compared to HSMMs. Thus, the HMMs may provide a more detailed representation of the team behaviors. HMMs may therefore provide more accurate state predictions. However, the HSMMs incorporate temporal information and therefore may be capable of providing timing predictions unavailable with regular HMMs.



**Figure 5.3 BIC for HSMMs of Team-RESCHU**

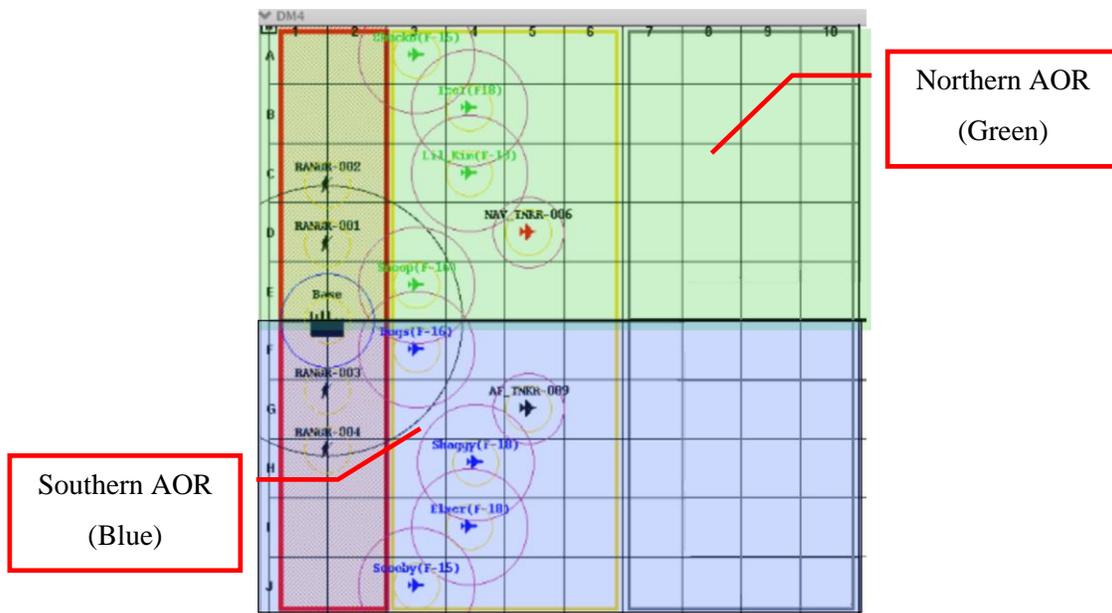
The evaluation of the models through the MAS methodology will be presented in the Section 5.4.



- Tanker Operator (TO): Carries out orders from the WDs by maneuvering two tanker aircraft that contain replacement fuel and weapons for the fighter aircraft.

Each fighter has limited amounts of fuel and ammunition. A fighter can refuel and restock weapons either from one of the two tankers or by returning to base. Thus, a typical scenario would involve a WD requesting an SO to move assets to a given grid coordinate and engage a target, while the ordering the TO and another SO to coordinate the refueling of an asset.

The tasks are further divided by two Areas of Responsibility (AOR), the Northern and Southern halves of the map. Each AOR is under the exclusive control of a WD. The WD in charge of the Northern AOR controls the Green Team whereas the WD in charge of the Southern half of the map controls the Blue Team. This division is summarized in Figure 5.5. Should Strike Operators or Tanker Operators move an aircraft from one AOR to another, the operators must notify the corresponding WD.



**Figure 5.5 Areas of Responsibility**

In addition to having role specific tasks, WDs, SOs and TOs are presented with different types of information on their respective interfaces. WDs get a complete view of the enemies. SOs can only see enemies within sensor range. TOs cannot see enemies at all. The limited information available to SOs and TOs enforces coordination with WDs in order to properly intercept longer range targets. Each team member interacts with a GUI via a point-and-click interface. For illustrative purposes, Figure 5.6 shows the SO interface

### **5.3.2 Communications**

Due to the collaborative nature of the task, the players have to exchange information with one another. In this experiment, players were capable of 1) broadcasting messages on an open audio channel and 2) using a public text-based chat. Each utterance, vocal or typed, was recorded by the simulation. Furthermore, the players were trained to start their communications with the ID of the intended recipient of the message. For example, a Blue WD would warn the Green WD that a given aircraft is being handed to his or her AOR as follows: “Green WD, Fighter15 is headed north to take care of Mig 335”.

Thus, the AFRL procedure differs from the Team-RESCHU scenario in multiple critical aspects. The first main aspect is team size: Team-RESCHU involves 3-member teams whereas AFRL uses 5-member teams. Secondly, the mode of communication is different. While Team-RESCHU operators were limited to text chats, AFRL operators were also allowed to vocally communicate with each other, thereby influencing the bandwidth of possible communications between the players. Third, the amount of coordination needed between the players differs significantly. With the exception of coordinating the dispatch of a specific UV to a given target, the operators in the Team-RESCHU task could operate mostly independently. In contrast, the diversity in the different roles of the AFRL operators enforced a higher level of communication and collaboration between the players.

### **5.3.3 AFRL Grammar**

Following the procedure shown in Figure 3.2, a grammar was developed in collaboration with domain experts from the AFRL in order to translate the operators’ action into an observable state space for the statistical learning phase. In addition to UI interactions, inter-operator communication was taken into account. Table 5.2 shows the resulting grammar. Following the same principled design outlined in Section 3.1.3, the observable space is represented by a 2D matrix where the y-axis represents the different operators, WDs Blue or Green, SO Blue or Green or TO. The x-axis of the table defines the set of possible operations for the players. The first 3 categories on that axis correspond to UI interactions: Move, Refuel/Restock (RF/RS) and Attack.

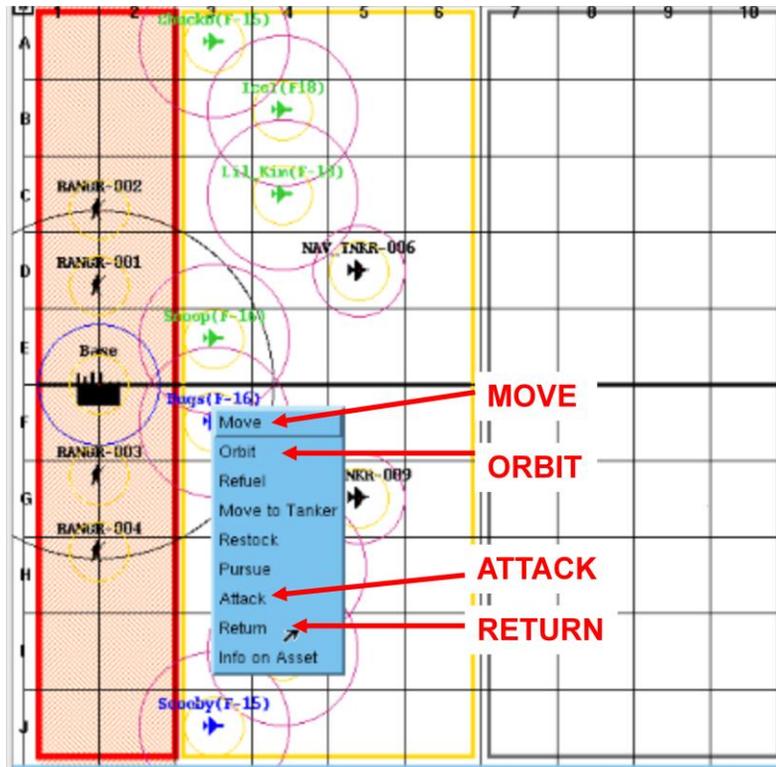


Figure 5.6 Strike Operator GUI

The next group of actions corresponds to inter-operator communications. In contrast with the grammar defined for Team-RESCHU, the communications for this data set were parsed according to their content. More specifically, the communication were labeled as either Move, Restock/Refuel, Attack and finally other types of Information request or exchange. All the voice and chat communications were manually encoded into this grammar. This step was necessary in the AFRL data set because the team-members are heavily dependent on each other for mission completion.

Table 5.2 AFRL grammar

<i>WD Blue</i>							
<i>WD Green</i>							
<i>SO Blue</i>	<i>UI Interactions</i>			<i>Communications</i>			
<i>SO Green</i>							
<i>TO</i>							
<b>Operators/ Operations</b>	<b>Move</b>	<b>RF/RS</b>	<b>Attack</b>	<b>Move</b>	<b>RF/RS</b>	<b>Attack</b>	<b>Info</b>

### 5.3.4 Experimental Subjects

The experimental data set consists of 4 distinct 5-player teams doing 6 day-long sessions. In each session the teams were presented with 15 scenarios that lasted 10 minute. The data used in this thesis corresponds to the last of those day-long sessions for each team. Thus, the data in this last session corresponds to the behavior of trained teams who have 5 days of previous experience in the task. In total, these 5 days of prior experience correspond to ~12.5 hours of training with the simulator, which is significantly higher than the typical amount of training provided in the single operator scenarios. Therefore, the teams were considered trained and experienced with the interface and the task at hand. This is important because prior studies have shown that team coordination and routine emerges with practice (Gersick and Hackman, 1990). Overall, this data set contains 13435 data points and corresponds to ~50 hours of single operator data.

### 5.3.5 AFRL Models

Models of team behaviors can be learned via the same methodology and algorithms illustrated in the previous chapters.

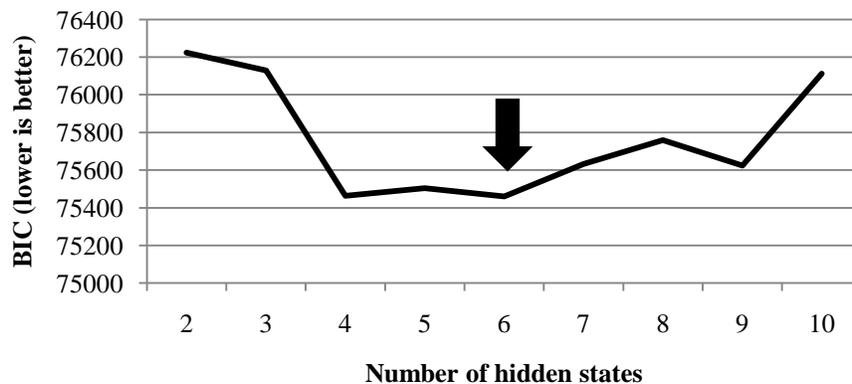


Figure 5.7 BIC for HMM of AFRL

Figure 5.7 shows the BIC for the HMM of the AFRL data set. The results show that a 6-state HMM provides the most representative model with a score of 75460.07.

Figure 5.8 shows the BIC scores for HSMM models, both histogram-based and using between 1 and 3 Gaussian mixture models. As in the single operator scenario and for Team-RESCHU, the BIC of the histogram-based HSMMs is significantly higher than those of the GMM-HSMMs. The number of modes

for the GMM-HSMMs has a comparatively smaller impact, and the single mode GMM-HSMMs tend to perform better than other HSMMs, parametric or not, for most numbers of hidden states.

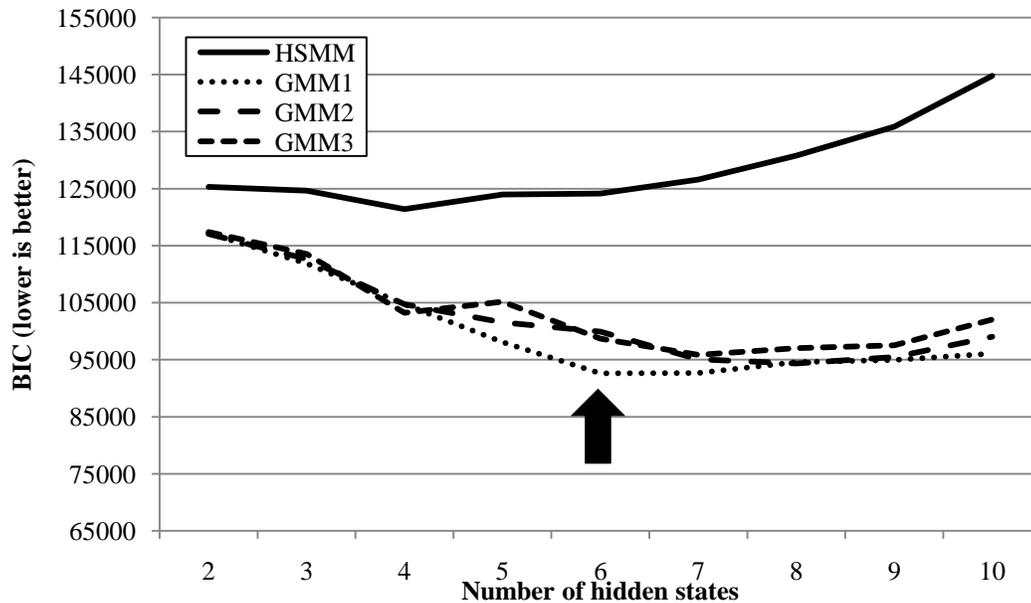


Figure 5.8 BIC for HSMMs of AFRL

For comparison, the BIC of the 6-state 1-mode GMM-HSMM is 92594.46, whereas the BIC of the 6-state HMM is significantly lower (75463.54). As in the Team-RESCHU data set, the BIC of the HMMs are significantly lower than that of the HSMMs, which seems to indicate that the simpler HMMs are likely to generalize better than the more complex HSMMs. However, in contrast with the models obtained with the other data sets, both the selected HMM and 1-mode GMM-HSMM have the same number of hidden states. Therefore, the HMM may not, in this case, provide a more detailed description of the behavior of the team. In practical terms, however, the HSMMs have the advantage of providing temporal predictions, a significant factor in PHSC scenarios. The evaluation of the models through the MAS methodology will be presented in the following section.

## 5.4 Comparing Single Operator and Team Models

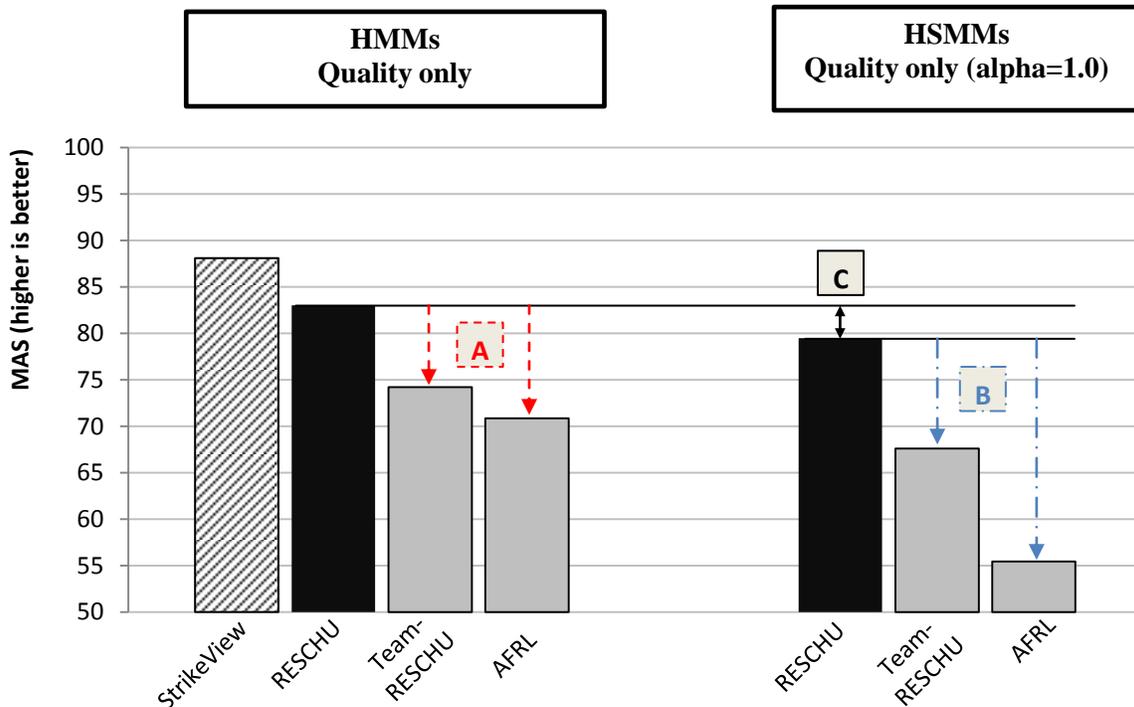
The main goal of this chapter is to examine how the proposed methodology extends to teams of PHSC operators. The MAS metric (with different values of  $\alpha$ ) described in Section 4.2.4 is a useful measure of how well HMMs and HSMMs can predict operator behaviors. Recall the MAS measures both the quality and the timing of the predictions for HSMMs, but is limited to measuring the quality of the prediction in HMMs. This section compares the most representative HMMs and HSMMs for the 4 data sets in this

thesis. The structures of the considered models are presented in Table 5.3. The data sets represent a range of scenario complexity, from single operator in a static mission planning environment (StrikeView) and dynamic resource allocation environments (RESCHU) to similar team scenarios with 3-member teams (Team-RESCHU) and 5-member teams (AFRL) in dynamic settings. This increasing scenario complexity provides representative sample points for a wide range of PHSC activities.

**Table 5.3 Selected models summary**

	Single Operator Static Context	Single Operator Dynamic Context	Team of 3 Operators	Team of 5 Operators
	StrikeView	RESCHU	Team-RESCHU	AFRL
Selected HMM	5-state	8-state	8-state	6-state
Selected HSMM	n/a	5-state 1-mode	5-state 1-mode	6-state 1-mode

Due to the static environment, timing data was not available for the StrikeView data set and only HMMs were developed. All the HSMM results were obtained with a default resolution of 0.25 standard deviations. HMMs do not provide timing information, thus the MAS scores for the HMMs and the HSMMs can be compared when the MAS solely measures the quality of the predictions ( $\alpha = 1.0$ ).

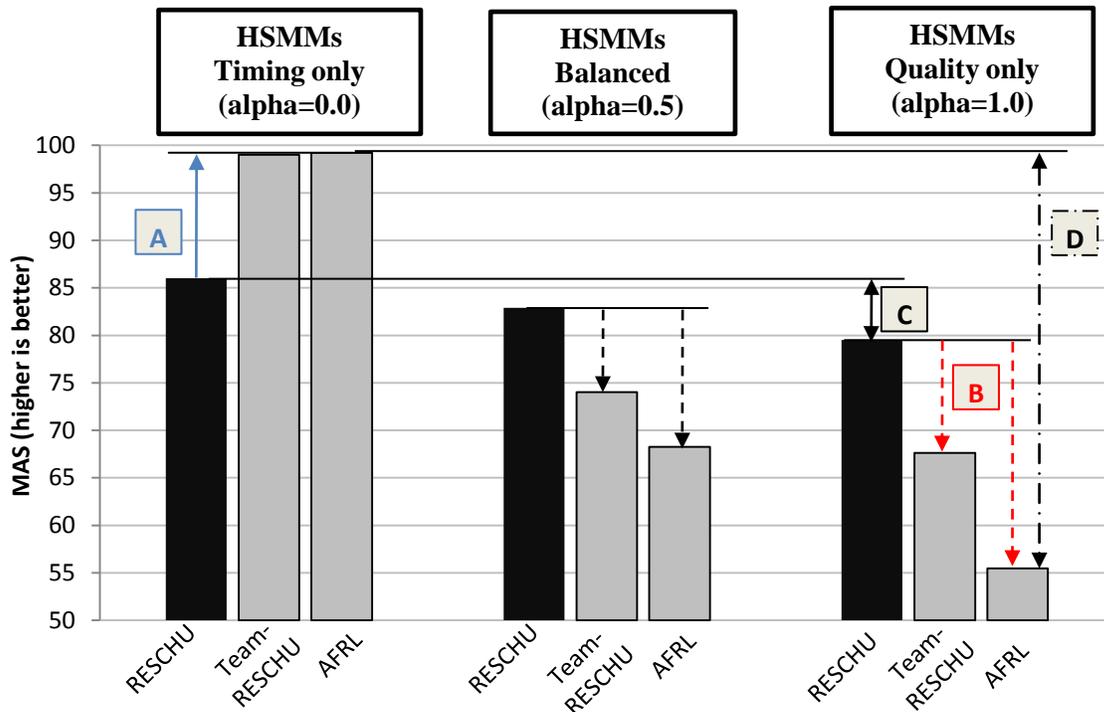


**Figure 5.9 HMM and HSMM performance of team and individual models**

Figure 5.9 illustrates the differential in prediction quality between single operator and team models for both HMMs and HSMMs (in the latter case,  $\alpha = 1.0$ ). First, the quality of the predictions of the most representative HMMs across the data sets shows that the MASs for StrikeView and RESCHU are 88.1 and 83.01 respectively, whereas that of Team-RESCHU and AFRL are comparatively lower at 74.21 and 70.85 respectively (labeled A in Figure 5.9). Thus, the predictive quality of the HMMs decreases as the complexity of the scenario increases. A similar trend is observed for the HSMMs, but the decrease in quality of prediction is more pronounced (labeled B in Figure 5.9). The MAS score for RESCHU is 79.51, and those of Team-RESCHU and AFRL are 66.27 and 55.46 respectively. In addition, for each data set, the predictive quality of the HMMs is higher than that of the HSMMs (illustrated for RESCHU in the single operator case by the label C in Figure 5.9).

These results therefore suggest that HSMMs are more sensitive to scenario complexity than HMMs. While HSMMs of single operators perform only slightly worse than their HMM counterparts, they do much worse when modeling teams of operators. This may be due to the fact that the learning algorithms have to estimate a higher number of parameters for the HSMM given a fixed amount of data. However, this increased complexity also allows HSMMs to provide timing predictions, a capability that HMMs do not have. The question is whether the value of the timing information balances the decrease in predictive quality.

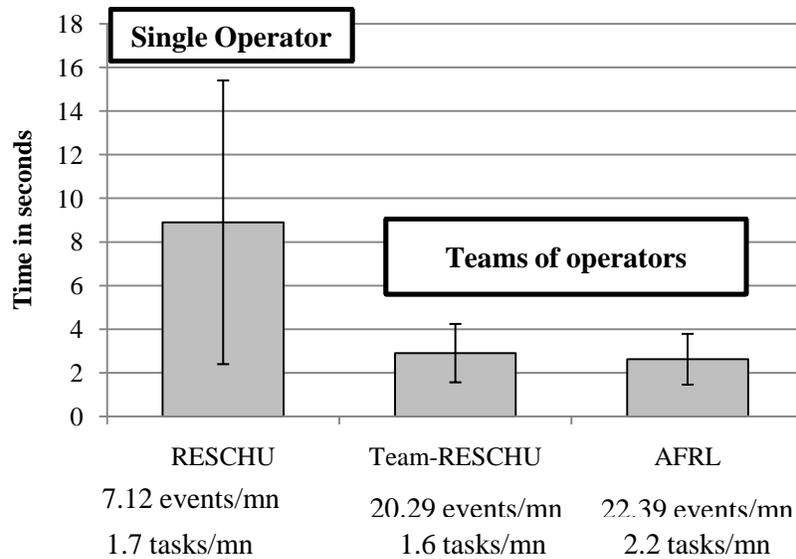
In order to investigate this question, Figure 5.10 shows the HSMMs' MASs that incorporate the timing of the predictions across the different scenarios. As a reminder, the  $\alpha$  parameter balances the quality and the timing subscores. When  $\alpha = 1.0$ , the metric measure only the quality of the prediction. In contrast, when  $\alpha = 0.0$ , the metric only considers the timing of the prediction. With  $\alpha = 0.5$ , the quality and timing are balanced equally. The "Timing only" results in Figure 5.10 show that the team models are capable of extremely accurate timing predictions (the MAS of Team-RESCHU and AFRL are 99.0 and 97.55 out of 100 respectively). In comparison, the MAS of the single operator case is lower at 86.02 than that of team models (label A in Figure 5.10). This contrasts with the previously shown MAS results that consider only the quality of the prediction (i.e.  $\alpha = 1.0$ ) which showed that the MAS for team models tend to be lower than that of single operators (labeled B in Figure 5.9).



**Figure 5.10 HSMMS performance of single and team behaviors**

In the latter case, HSMMS provide higher scores for single operator than for teams of operators (label B in Figure 5.10). When the MAS is balanced ( $\alpha = 0.5$ ), the results are an average between results that consider either timing or quality uniquely. Thus, Figure 5.10 suggests that while HSMMS can provide accurate timing in team settings, the quality of the prediction is higher in single operator scenarios. In addition, the difference in timing and quality-only MAS between the single and team scenarios increases with the complexity of the situation (labels C and D in Figure 5.10). These results pose the question of why the timing predictions in the team situations are so high.

Figure 5.11 shows an analysis of the mean and standard deviations of the state durations in RESCHU, Team-RESCHU and AFRL. In addition, both the average task arrival rates (the rate of system-generated tasks presented to the team of operators in tasks per minute) and the average number of user events per minute (i.e. the rate of events generated by the team in response to the system-generated tasks in events per minute) are provided for each scenario. Figure 5.11 shows that the average time between subsequent events is 2.76s (standard deviation 2.5s) for teams compared to 8.9s (standard deviation 13.01s) in single operator cases. Thus, because the state of an HSMMS is updated with every event arrival, the state durations tends to be markedly longer and more variable for single operators compared to team situations.



**Figure 5.11 Event durations and rate, task arrival rate for single operator and teams**

Correspondingly, the teams of operators in the data sets produce on average 3 times as many events as a whole compared to the single operator case (7.21 for single operators vs. 21.34 on average for both team scenarios). It is also important to note that the shorter state durations are not simply due to the number of tasks generated by the system (1.7 tasks/minute for RESCHU, 1.6 tasks/minute for Team-RESCHU and 2.2 tasks/minute for AFRL). Given that the single operator and the teams had a comparable number of tasks to perform in a given amount of time, the shorter state durations can be attributed to the nature of the collaborative task, and in particular to 1) having multiple operators interacting with the system simultaneously and 2) the additional coordination task between operators.

In terms of modeling, the more consistent state durations in the team scenarios make it easier of the models to predict when the next actions are likely to take place. This explains the highly accurate timing predictions of the HSMMs in the team conditions observed in both the Team-RESCHU and AFRL data sets. However, the high timing accuracy questions the value of the information contained in the state sojourn distribution. The high consistency of the state durations is akin to a uniform distribution and implies that a low amount of information is conveyed by the temporal component of the signal in team situations. In fact, these results also suggest the complexity of HSMMs is not warranted in situations where the modeled events are uniformly distributed and have similar durations.

Thus, in a team context with uniform state durations, the use of HMMs could be advantageous compared to HSMMs because (1) they can provide more accurate state predictions (as shown by label C in Figure

5.9) and (2) the mean state duration can provide an appropriate estimate of the actual state durations (Figure 5.11) thereby negating the usefulness of the timing predictions of the HSMMs. Conversely, because the state durations for single operators are both longer and more variable, accurately predicting the occurrence of future actions is critical, especially in time-sensitive PHSC contexts. Thus, in the individual case, the HSMM ability to accurately predict the timing of future states has great practical value. Summarizing, while HSMMs seem to be more appropriate for single operator scenarios, HMMs seem more appropriate in team situations. This conclusion is notionally illustrated in Figure 5.12.

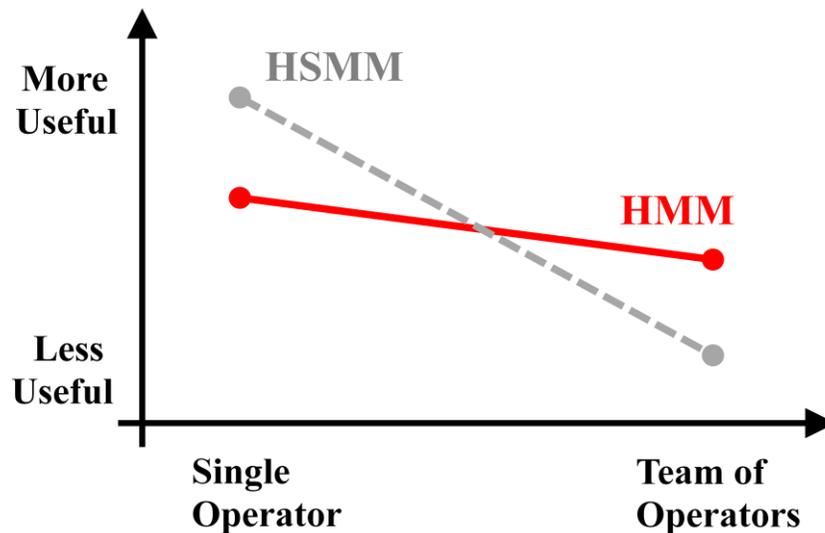


Figure 5.12 Models for single and teams of operators

It must be noted however that these results likely stem from the holistic modeling approach adopted in this thesis. Another approach would have been to model each operator in a team independently, which likely would have resulted in non-uniform state duration similar to those observed in RESCHU. However, this independent modeling approach would not capture the inherent degree of dependency in the team-tasks. The implications of the holistic approach are discussed in more details in the following chapter.

## 5.5 Chapter Summary

This chapter presented the results of using the proposed methodology on two data sets representing the behavioral patterns of teams of operators involved in PHSC tasks. The first data set was Team-RESCHU, modified version of the RESCHU game presented in Chapter 3 in which teams of 3 operators collaborated in order to process the maximum number of target on a map. The second data set was obtained from an Air Force Research Lab team experiment. In this experiment, teams of 5 operators had to protect friendly

airspace from an enemy intrusion. The team modeling results showed that the HMM approach tended to provide more robust prediction quality than HSMMs. However, HMMs cannot provide timing predictions, a task that HSMMs, in contrast, performed with high accuracy in the team scenarios.

Comparing the HMMs and HSMMs for team and single operators highlights a number of factors in the scalability of the methodology. The first factor is the complexity of the scenario. The 4 data sets used present a range of complexity from the more simple single operator in a static and dynamic environments (StrikeView and RESCHU) to more complex teams comprising 3 and 5 members in a dynamic situation (Team-RESCHU and AFRL). The results in this chapter show that the predictive power of the models seems inversely proportional to the complexity of the underlying process. In other words, the quality of the prediction for models of team behaviors tended to be lower than those of single operators. This is especially true for the more complex HSMM team models.

The second scalability factor is the timing characteristic of the modeled process. A comparative analysis of the single operator and teams data sets showed that the average and standard deviations of the team behaviors were markedly lower than those of single operators. A further analysis of the incoming system-generated task rate and operator event rate in response to those tasks suggests that the uniformity of state durations in team scenario is due to the simultaneous nature of the operators' work and to the added coordination required by the team task. The low variability of the team states is critical because it allows the HSMMs to provide highly accurate predictions. However, the low variability in state duration also implies that the mean state duration provides a reasonably accurate estimate. In such a case, it becomes conceivable to replace the explicit modeling of the state duration by the mean state durations. Using this structure, the simpler HMMs (which do not provide explicit timing information) could be used to forecast what the next states would be and the mean state duration could be used to estimate when these states would occur. In contrast, the inter-event timing for single operators is more variable. Replacing the explicit modeling of the state duration by the mean state duration would not be appropriate. Therefore, the use of HSMMs capable of providing accurate timing information provides more value in the single operator case than in the team scenarios.

The next chapter concludes this thesis by discussing the implication of the presented results along with possible lines of future work.

## CHAPTER 6 CONCLUSIONS

---

*“We shall not cease from exploration  
And the end of all our exploring  
Will be to arrive where we started  
And know the place for the first time.” – T.S. Eliot, 1942<sup>18</sup>*

*The most exciting phrase to hear in science, the one that heralds new discoveries, is not  
'Eureka!' but 'That's funny...!' – Anonymous*

This thesis presented a methodology capable of learning hidden Markov models and hidden semi-Markov models of human supervisory control behaviors in proceduralized contexts. The main idea behind the proposed method is to exploit pattern recognition and prediction techniques in order to learn statistical models of operator behaviors. Then, by leveraging behavioral patterns, such statistical models can detect and predict possibly anomalous operator conditions. HMMs are useful because they provide computationally efficient algorithms to infer the path through a lattice of hidden states from a sequence of observable behaviors. In the context of operator modeling, HMMs can infer operator states from observable behaviors such as user interface interactions or communications. In other words, operator states represent clusters of statically-linked observables. HMMs, however, suffer from a strong structural limitation in that they do not take temporal information into account. This can be especially problematic in typically time-sensitive PHSC contexts. Because of this structural limitation, HSMMs, a more complex version of HMMs capable of explicitly modeling the state durations, can be used. Although HMMs and HSMMs are established methodologies in such applications as voice recognition and protein analysis, the use of such techniques to model and predict human behaviors in a PHSC context is novel. As such, the central part of this thesis was to formally establish and validate the proposed methodology in the PHSC context.

The purpose of this final chapter is to summarize and synthesize the results presented in this thesis. Both academic and practical contributions of this work are first discussed. Then, important limitations of the

---

<sup>18</sup> Four Quartets - Little Gidding

proposed methodology are examined before closing this thesis by highlighting a number of possible future research areas.

## 6.1 Contributions

As described in Chapter 1, this thesis set out to answer a number of research questions regarding the use of HMMs and HSMMs for modeling PHSC operator behaviors. The first research question was:

*“How well can HMMs and HSMMs model the behavior of a single operator engaged in a PHSC task?”*

As described in Chapters 3 and 4 respectively, the proposed methodology was capable of learning suitable HMMs and HSMMs. In particular, HMMs were learned for two distinct single operator data sets. The first data set was a static resource allocation problem. The second data set included a dynamic environment in which an operator performed resource allocation and scheduling replanning tasks. The learned HMMs of both were shown capable of accurately predicting operator behaviors. In addition, qualitative analyses of the models provided valuable insights into the patterns expressed in the operators’ behaviors. Because of the HMM’s inability to represent temporal state information, more complex HSMMs were applied to the dynamic data set. The temporal information embedded in HSMMs can be critical in typically time-sensitive PHSC environments. In addition to properly synthesizing the sequences of operator behavior, the HSMMs were also shown capable of learning the explicit expressions of the state durations.

A subset of this first question, critical to address in any modeling effort, was:

*“Do methodological and model learning assumptions hold true for PHSC data?”*

Appendix A validates three main methodological assumptions in the PHSC context. First, the proposed methodology relies uniquely on easily accessible user interaction and communication data. The question was whether finer-grained data (e.g. psycho-physiological data) would provide benefits to the models. Section A.1 compares models built uniquely with UI data to those built with UI data and eye tracking data. The results show that the diminished signal-to-noise ratio of the combined UI and eye tracking data set produced models that were less useful than those built solely on UI data.

The second assumption concerns the Markov property of independence. This assumption of memorylessness is central to the computational tractability of HMMs but does not hold for human behaviors in PHSC settings. The question is whether the assumption is valid in practice. Higher order models can be used to mitigate this assumption of memorylessness but they also lead to more complex models. Section A.2 compares first, second and third order models of operator behaviors for a representative scenario, and concludes that the increased model complexity of the higher order models is not balanced by the increased fit to the data. The results therefore suggest that the use of models exploiting the first order Markov assumption is preferable from a generality standpoint.

Finally, the last assumption addressed the use of unsupervised learning and unlabeled data in order to obtain the models. This is an important issue because while supervised learning methods tend to be computationally easier, they also rely on a priori labeled data. This labeling process is problematic in the PHSC context because the ground-truth of operator states is not accessible. In addition, the labeling process typically relies on expert knowledge and is an expensive process. Section A.3 compares unsupervised models to models learned with two supervised learning methods using data hand-labeled by a subject matter expert. The results show that the unsupervised models outperformed the supervised models possibly due to the bias introduced in the labeling process.

The second research question was:

*“How well can HMMs and HSMMs model the behavior of teams of operators engaged in a PHSC task, and more generally, how well does the approach scale to multiple operators?”*

The scalability of the methodology was tested in Chapter 5 by learning behavioral models of teams of PHSC operators based on 2 distinct data sets. The first one, Team-RESCHU had 3-person team perform a task similar to that of RESCHU. The second data set was obtained from an Air Force Research Lab experiment and represents 5-person teams defending friendly airspace against intruders. The team modeling results showed that the HMM approach tended to provide more robust prediction quality than HSMMs. However, HMMs cannot provide timing predictions, a task that HSMMs, in contrast, performed with high accuracy in the team scenarios.

Chapter 5 also compared the HMMs and HSMMs for team and single operators across different scenarios of varying complexity. More specifically, the scenarios in question ranged from a single operator in static or dynamic environments to 3-person or 5-person teams. These comparisons highlighted a number of

factors in the scalability of the methodology. The results showed that while HSMMs provided valuable timing predictions in the single operator case, their usefulness was mitigated in team situations because of the markedly smaller variance in state durations. In contrast, while they cannot provide timing data, HMMs appear to be more robust models in team scenarios. These results were summarized in Figure 5.12 (reproduced below) and present a critical take-away message of this thesis.

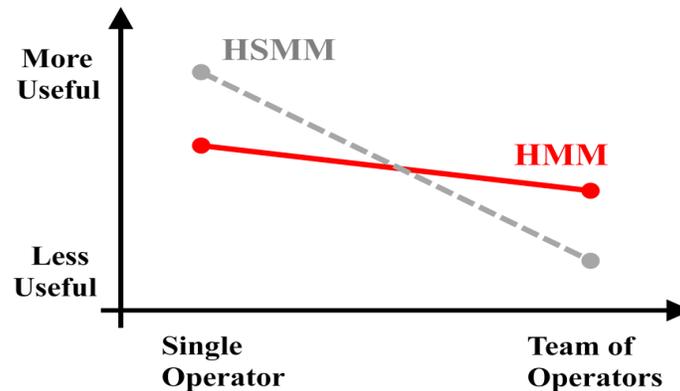


Figure 5.12 Models for single and teams of operators

In summary, this thesis showed that HMMs and HSMMs could be used to learn models of operator behaviors in proceduralized supervisory control settings. The next section presents a number of practical applications that could exploit such models.

### 6.1.1 Applications

The results presented in this thesis suggest that accurate models of PHSC operator behaviors can be obtained via the methodology described in Chapter 3. The developed HMMs and HSMMs synthesize the behavioral patterns seen in the training data. Then, computationally efficient algorithms (such as the Forward/Backward algorithm described in Section 2.3) can be used to compute the likelihood of a sequence of observables given the model. This likelihood is useful in PHSC settings because it provides a direct quantitative measure of how close the current operator behavior is to those synthesized in the model. The likelihood of a sequence of observables can be used for post-hoc analysis or in real-time.

As a post-hoc analysis tool, the likelihood of sequence of operator behaviors measures how close they are to those on which the model was trained. Therefore, assuming the models represent a set of desired behavioral patterns, the likelihood the sequence provides a quantitative assessment of how close the operator is to the desired behavior. This can be useful for monitoring student performance in PHSC training environments where the desired behaviors are typically expressed as standard operating

procedures and therefore known a priori. Because the assessment relies on behavioral patterns, the focus is shifted from outcome-based evaluations to process-based evaluations. In other words, the models can evaluate a trainee not only on what the final solution to the problem is but also on how the problem was solved. Thus, the proposed methodology might be particularly suited to training scenarios because Chapter 5 showed that models tend to perform better in such static environments where time pressure is not critical.

Furthermore, the methodology proposed in this thesis provides the ability to autonomously learn what the desired patterns are from expert behavior. These expert models are useful for two reasons. In situations where no SOPs are available, the expert models can provide a set of desirable behavioral patterns for trainees. In doing so, the progression of the behavioral patterns similarity between trainees and experts can be objectively quantified across the training sessions. Conversely, if the SOPs are known a priori, the expert models can establish whether the SOPs are actually used in practice. This diagnostic use of the models can therefore be used as a SOP quality assurance check. For example, should the model suggest that the steps in a given procedures are consistently performed out of order by experts, it may be appropriate to review the adequacy of that procedure.

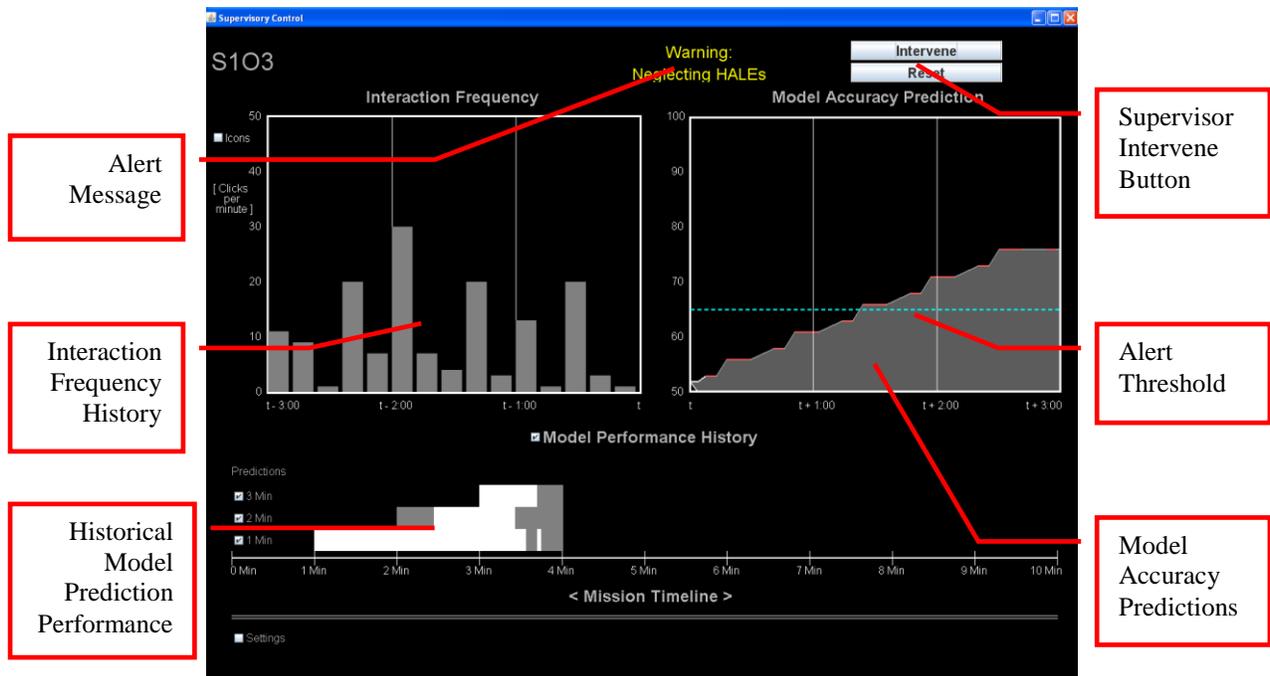
The real-time use of the models corresponds to a scenario in which the likelihood of an incoming stream of events is computed, such as in an air traffic control setting where controller behaviors are monitored in real-time. Should the computed likelihood of an expected sequence of behaviors fall below a given threshold, the current behavior could be flagged as possibly anomalous. However, the notion of “anomalous behavior” in this methodology does not necessarily imply improper behavior. A sequence of events with a low likelihood given a model only means that this sequence is different from those on which the model was trained. Therefore, the appropriate response to an alert generated by the model should be left to a human operator capable of qualitatively judging whether the current operator behavior could have detrimental consequences. In addition, because HMMs and HSMMs are generative models, it is possible to compute the likelihood of future actions and therefore forecast the most likely sequence of future operator behaviors for different time horizons. These predictions can then be verified against the actual behavior of the operator thereby providing an historical estimate of the model prediction performance. The real-time use of the HSMM models was tested in a user experiment by Castonia (2010) and the experimental protocol and results are summarized in the following section.

### *Real-time supervisor decision support tool*

The original application of the proposed methodology was the development of decision support tools for supervisor of teams of unmanned vehicles. Castonia (2010) designed the interface of the Decision Support Tool (DST) which was then implemented by Huang (2009). This DST interface relied on the models developed in this thesis in order to generate alerts to the supervisor when anomalous situations were detected. The models and the DST were tested in a user study in which an experimental subject was charged with monitoring a team of 3 UV operators interacting with the RESCHU simulation.

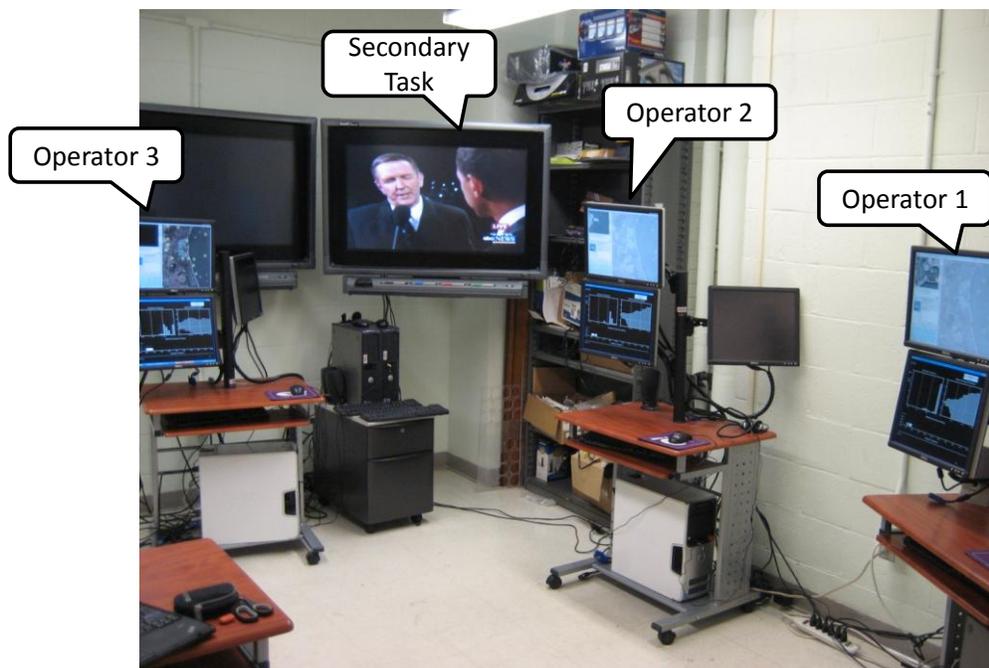
Figure 6.1 shows the DST tool. The top left bar graph represents an historical user interaction frequency. The top right line graph represents the current and projected model accuracy values. The uncertainty of the projection is shown by the gray area in that graph. Finally, the bottom timeline shows a history of the predictive performance of the model over diverse time horizons.

In this experiment, the UV operators were assumed to be remotely located and the supervisor had access to either a feed from the operators' RESCHU interface along with the associated DST or only the feed of the operator's RESCHU interface.



**Figure 6.1 DST interface (Castonia, 2010)**

Figure 6.2 shows the experimental setup in which the test subject had to monitor and detect anomalous operator behaviors. The results of this experiment showed that the overall alert system, i.e. the models developed through the proposed methodology along with the DST, improved team supervisor performance in terms of increased decision accuracy, decreased incorrect interventions, and decreased response times in single alert scenarios. In addition, the overall system was also shown to decrease the number of incorrect interventions, while having no affect on decision accuracy and total response time scenarios when the supervisor faced multiple simultaneous alerts. However, the experimental results did not show the same positive results for scenarios in which multiple alerts were generated. This may have been due to a cognitive bias in not intervening without an alert. In addition, an analysis of the post-hoc debriefs showed that the design of the display (especially the MAS and confidence history plots) was difficult to understand by some subjects. However, these results demonstrate the practical benefits of the proposed methodology.



**Figure 6.2 Experimental setup (Castonia, 2010)**

The proposed methodology could also be used in a number of situations in which the operator's performance needs to be monitored either for its own sake or as the input to another system (such as in an adaptive automation scenario). There are, however, a number of important limitations which restrict the generalizability of the proposed method, discussed in the following section.

## **6.2 Limitations**

The structure of the methodology and the assumptions of the associated models impose a number of constraints to practical use and generalizability to other contexts. This section discussed five main limitations of this method.

### ***6.2.1 Training Data***

The models presented in this thesis were developed from data obtained in research settings. There are three main implications of using such data. First, the data may not be ecologically valid in that most of these experiments were run in laboratory settings so experimental conditions were designed to influence the behaviors of the test subjects for example by varying the amount of provided automated support or induced time pressure (Boussemart, Donmez et al., 2009). Therefore, the recorded behaviors correspond to narrow slices of human performance on controlled settings. Secondly, while the proposed methodology relies on the patterns in operator behaviors, such patterns typically develop over time through the slow acquisition of expertise. Although all users involved in the data sets received some level of training, the amount of exposure to the task falls far short of what would normally qualify as “expert behaviors”. Finally, the most significant implication of using experimental data is the limited size of the data sets. Performing human-in-the-loop experiments is a notoriously complex and expensive endeavor, and obtaining large data sets is often impossible. Thus, all the models presented in this thesis may be improved if larger training data sets were available.

### ***6.2.2 User Interface Input Requirement***

One critical limitation of the proposed methodology is its reliance on user interface events. While this thesis showed that UI-based models did not benefit from additional eye tracking data, there is an implicit assumption that the operational setup provides a number of UI events for models processing. This may not be the case in some HSC situations that mostly involve monitoring such as operators who spend most of their time watching displays, and never actually touching a control, e.g. nuclear power plant operators under full plant load operation. In such cases, using additional sources of information (e.g. body and eye tracking, skin conductance, EEGs) would be required in order to gather sufficient amounts of data. Therefore, the proposed method is only applicable to situations in which the operator interacts intermittently with an interface.

### ***6.2.3 Grammar Construction***

The grammar is used to reduce the size of the problem space to a set of observable events that can be used by the statistical learning algorithms, and it is also critical for interpretation of results. The definition of the grammar is the first step in the methodology and represents the foundation on which the rest of the algorithms operate. Therefore, the grammar is critical to the rest of the modeling process. In this thesis, the grammar generically takes the form of a 2D matrix where the rows represent a set of either operands in single operator scenario or operators in team conditions. In contrast, the columns represent a set of operations feasible in the space. Thus, the grammar represents the possible observable events as a combination of an action (how) either on a specific item (what) or performed by a specific operator (who). While a principled Task Analysis or Cognitive Task Analysis provides the basis of defining the set of possible operations (i.e. the columns of the matrix), the definition of the operands or operators remains subjective. In fact, the subjectivity introduced in the definition of the grammar is, to some extent, similar to that introduced by a data labeling process involved in supervised learning and should be investigated further

### ***6.2.4 Visualization Complexity***

Castonia (2010) designed an interface capable of leveraging the models of operator behaviors in order to provide a real-time decision support tool to supervisors of teams of UV operators. While the decision support was shown to provide value to the team supervisor, one of the common remarks during the post-experimental debriefing was that the interface was hard to understand. This is a critical problem for the practical use of the proposed methodology. The outputs of the HMMs and HSMMs are dynamic probability densities over a set of observables. These probabilistic representations are notoriously difficult for humans to understand (Tversky and Kahneman, 1974), and even the best models are useless if their recommendations are not followed or trusted by the human operator (Lee and See, 2003). Therefore, one of the limitations of the methodology is how to communicate such information effectively to the human decision maker.

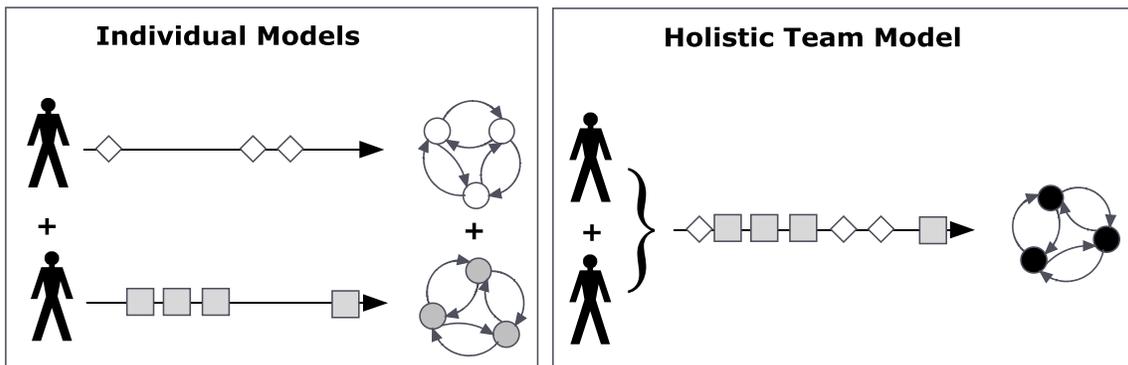
### ***6.2.5 Model Complexity***

The results in Chapter 5 show that the predictive power of the models decreases as the complexity of the underlying process increases. From a practical standpoint, this raises the issue of the nature, both in terms of dimensionality and metrics, of the complexity of the underlying process.

For single operators, the models performed worse in dynamic environments than in static environments. The nature of the environments therefore represents one dimension in complexity. This leads to the following question: what are the other dimensions of complexity that influence the performance of the model? A number of possible dimensions are likely to have a significant impact, such as the rate of arrival and the cognitive complexity of the tasks, the uncertainty of the environment, or the level of adherence to procedures. From a practical perspective, the question becomes, given a set amount of data, how complex a behavior can HMMs or HSMMs reliably model?

The question of complexity dimensions is even more pronounced for team situations. While our results show that team models tend to perform worse than single operator models, this performance differential is likely to be influenced by the nature of the teamwork. Therefore, the amounts of task sharing and operator collaboration are dimensions that may influence the predictive ability of the models. In theory, the behavior of an operator in a team could be indistinguishable from that of single, team-less operators if the group tasks are fully independent and disjointed. In a situation with independent tasks, the use of a set of independent individual models may be an appropriate representation of a team as illustrated in the left-hand side of Figure 6.3. Then, if the operators need some low level of cooperation, the independent models might influence each other slightly<sup>19</sup>. Should the collaboration between operators increase, so should the inter-model dependency.

This thesis took a diametrically opposed approach (1) by assuming that the teams are holistic entities and (2) by analyzing the patterns of team events in a single univariate data stream. This approach is shown in the right-hand side of Figure 6.3. Thus, the manner in which the team behaviors are aggregated is a critical characteristic in the application of the methodology to the team data



**Figure 6.3 Team as a set of individual models or as a single holistic model**

<sup>19</sup> From a methodological perspective, coupled HMMs have been developed precisely to deal with such situations (Brand, Oliver et al., 1996).

Furthermore, another limitation to the holistic approach adopted in thesis is that it ignores the notion of concurrent work. The inter-event arrival time defines the duration of a state, regardless of the source of the event. This is especially critical for teams in which the roles of each operator are loosely defined, i.e. each operator can perform the same task as any others. Taking a UV-centric example, one operator could take an anomalously long time to perform a visual task, but this timing discrepancy may not be detected by the current model if the other operators keep interacting normally.

The application of the methodology to teams is therefore an extremely complex endeavor. This holistic approach to team in this thesis was chosen because it was the closest to single operators and therefore provided results that could be compared more readily. Yet, these results only scratch the surface of this enormously complex issue, and a large number of teamwork dimensions could be taken into account in order to learn more detailed models.

## **6.3 Future Work**

While the results shown in this thesis are promising, they also opened the door to a number of exciting research questions. In particular, the previous section highlighted a number of limitations to the proposed methodology which directly define a number of future areas of possible research questions such as:

- How different would the models be if a larger amount of data was available? What would be the impact on the predictive capability of the models if they were built using expert or novice behaviors only?
- What is the minimum rate of UI interaction is needed in order to obtain models that are useful in practice?
- The process by which the grammar is defined is somewhat subjective. What would be the impact of a slightly different grammar? And if this impact can be measured, could a grammar be defined autonomously?
- Given the trade-offs between (1) the complexity of the underlying data and the predictive ability of a model and (2) the amount of training data and the complexity of the model, would it be possible to get measure of the underlying data complexity and estimate first the type of model needed and secondly, the amount of data needed to train such a model?
- What is the optimal way to display the results of the methodology to an operator, and how would the display need to be adapted for real-time or post-hoc use of the models?

- In applying the methodology to teams of operators, a number of team factors could be taken into account. What would those factors be and what would be the impact of including them in the modeling approach?

In addition to these research questions stemming from the limitations of the model, another global question in the application of this methodology remains its generalizability. Future work should use the proposed methodology both in different domains and for different purposes. Such an effort has already started with the use of statistical models in training scenario, but more work is needed in order to validate the usefulness of the methodology across varied procedural human supervisory control domains.

## **6.4 Thesis Summary**

This thesis presented a methodology for learning HMMs and HSMMs of operator behaviors in procedural human supervisory control contexts. Such models provide significant benefits in the context of procedural human supervisory control because they can automatically monitor operator behavior in real-time, thereby detecting and predicting anomalous operator conditions. Because PHSC settings typically are mission and life critical, this automatic monitoring capability is paramount for more efficient and reliable supervisory control systems. In addition to real-time use, the models may also be used as post-hoc analysis tools in applications such as operator training. In this case, the models can monitor the progress of a trainee compared to the expected or expert behavior. The proposed methodology is thus generic and may be applied in other procedural human supervisory control environment in which operators interact intermittently with the system.

From an academic perspective, the two main contributions of this thesis are 1) to develop HMMs and HSMMs methodologies so that they can be successfully used to model PHSC behaviors both for single operators and teams, and 2) to validate that the methodological assumptions needed by the HMMs and HSMMs hold in the context of procedural human supervisory control. The core of this thesis consisted of learning HMMs and HSMMs both for single and teams of operators. A comparison of these models showed the existence of a trade-off between the complexity of the model and that of the operators' behavior given a certain amount of data. The more complex HSMMs were shown to be a better fit for the simpler time-critical single operator data, whereas the simpler HMMs were shown more appropriate for complex team situations. Through the exploration of the theoretical and the practical aspects of the methodology, this thesis paves the way for a wider use of machine learning techniques in the field of procedural human supervisory control.

# APPENDIX A      ASSUMPTION VALIDATIONS

---

*“I don't believe it. Prove it to me and I still won't believe it.”*  
- Douglas Adams, 1982<sup>20</sup>

The methodology by which the HMM models were obtained (see Chapter 3) operates on a number of assumptions. Three are of particular importance for this work. The first assumption is that of data sufficiency. Because the models rely solely on UI interaction, whether such data is sufficient for building useful behavioral models remains a valid question. Secondly, first order HMMs exploit the Markov independence assumption in order to maintain computational tractability. While mathematically convenient, the first-order Markov assumption is theoretically not valid for human behavior. The question then becomes whether the benefits of higher order models outweigh the increase in complexity. Finally, the proposed methodology uses unsupervised learning because the hand-labeling of the training sequences may introduce biases and therefore yield less useful models. This chapter explores those three assumptions in turn and provides a justification for the validity of the proposed approach.

## A.1 Data Sufficiency

The models presented in Chapter 3 relied solely on user interface events. From a modeling standpoint, such events represent the observable manifestation of a number of low-level cognitive processes. The question is whether the information contained in the high-level UI events is sufficient to create useful behavioral models of PHSC operators. In other words, could additional, finer-grained data such as psycho-physiological measures provide valuable information? The StrikeView experiment presented in Section 3.1.1 provided user interaction data from which behavioral models could be built. In addition to UI interactions, the experiment also recorded a user's gaze patterns using an eye-tracking system. With such data, it becomes possible to create models on a combination of UI events and eye tracking data. The usefulness of these models can be compared to those built only from UI data in order to determine the practical value of the additional information contained in the eye tracking data (Boussemart and Cummings, 2010).

---

<sup>20</sup> Life, the Universe, and Everything, Chapter 12, ISBN 0-345-39182-9

### ***A.1.1 Eye Tracking and Behavioral Models***

Eye tracking, a popular psychophysiologic measure (Andreassi, 1989), refers to recording the eye (and sometimes head) position of a participant in order to extract fixation and gaze patterns. It is compelling because it is seen as a window into an individual's cognition (van Gompel, Fischer et al., 2007). In the context of operator modeling, such information is valuable because fixation patterns can provide detailed insight to the source and sequence of information processed by the operator. However, it is commonly noted that using eye tracking data for modeling purposes can be problematic, notably in terms of the effort needed to gather, process, and analyze the fixation patterns (Schnipke and Todd, 2000; Sibert and Jacob, 2000; Poole and Linden, 2005; Bartels and Marshall, 2006). Most eye trackers function by detecting the pupil of a user and, after initial calibration, indicate a user's point of visual focus. Previous research has demonstrated that eye trackers can provide valuable behavioral insight in diverse fields such as network management tool analysis (Pretorius, Calitz et al., 2005), usability testing (Nakamichi, Shima et al., 2006) or marketing (Duchowsky, 2002). In the context of cognitive modeling (i.e., the formalization of human cognitive processes for a given activity) eye trackers have been used to generate descriptive models of varied tasks, from simple visual search (Hornof and Halverson, 2003) to more complex activities such as a driving while tuning a radio or dialing a phone number (Salvucci, 2005).

From a data analysis standpoint, extracting the required information from raw eye tracking signals is challenging due to the saccadic nature of the human visual system. High-frequency components (saccades) must be removed in order to extract fixation points (Salvucci and Goldberg, 2000), which in turn must be clustered into gazes and regions of interest (Santella and DeCarlo, 2004). Then, the bulk of the modeling effort remains in the analysis of such gaze clusters and a wide range of techniques have been used in the past. Researchers have published practical guidelines aimed at helping choosing the appropriate methodology (Goldberg and Kotval, 1999; Poole and Linden, 2005). Most of the techniques devised so far have ranged from simple scan pattern averaging (Hembrooke, Feusner et al., 2006) and analysis of percent coverage of the user's field of view (Wooding, 2002), to more complex methods such as principal component analysis (Rajashekar, Cormack et al., 2002) and hidden Markov models (Cooke, Russell et al., 2004; Hayashi, Beutter et al., 2005; Simola, Salojärvi et al., 2008). In particular, Hayashi et al. used HMMs to model space shuttle crewmember scanning behavior with an eye tracker and was able to detect deviation from the expected patterns (Hayashi, Beutter et al., 2005). In the latter work, the hidden states in the HMM were defined a priori and the models were trained via supervised learning. This approach was only possible because the researchers had access to a large amount of domain information used to create the models. This is, however, typically not the case in other contexts and, in addition, poses the risk of introducing human labeling bias in the state definition (Boussemart, Fargeas et al., 2010).

Simola et al. also used HMMs, with a priori defined hidden states for information searching tasks (Simola, Salojärvi et al., 2008). They showed that eye tracking data contained enough information to distinguish between word, sentence and title search. That study focused solely on discriminating between different kinds of information search tasks and thus did not consider user actions through some kind of input device. In contrast, this thesis focuses both on (1) PHSC applications where an operator intermittently physically interacts with the system thus creating unambiguous observable states, and (2) a different metric for success, namely how well models can predict future operator behavior.

### ***A.1.2 Experimental Procedure***

The StrikeView experimental procedure was the same as the one described in Section 3.1.1 with the addition of the use of the eye-tracker for collecting gaze data. In particular, participants were fitted with an ISCAN eye-tracking device and a user-specific calibration was performed. The ISCAN system is a dark-pupil eye tracker that uses low-level IR to illuminate the participants' eye. It is based on the RK-829PCI board capable of capturing images of the pupil at 60Hz. The refresh period is 17ms. The retina is captured with a 1500x1200 overlay and the tracker is precise to +/-1 degree of visual angle (VisionTRAK, *Polhemus by ISCAN*). The calibration comprised two steps: first the eye-tracker camera gain was adjusted to ensure proper image captures of the pupil. Secondly, the eye tracker, the Polhemus magnetic head tracker and the surface of interest was calibrated using a laser-based system in order to verify where they were with respect to each other. The remainder of the training process remained the same. The participants then proceeded to the 5 minute experimental session in which both UI interaction events and eye-tracking data were gathered.

### ***A.1.3 Eye-tracking data processing***

While the user-interface interactions were logged transparently by the interface, the participants' eye movement data were simultaneously recorded with a head-mounted eye tracker. In total, the user experiments yielded 7550 eye tracking data points in addition to the 2050 UI interaction events used to build the models shown in Chapter 3. The raw eye tracking data were processed with the software provided by ISCAN, the eye tracker manufacturer. Saccades were removed and in accordance to established methodical standards (Poole and Linden, 2005), only fixations longer than 200ms were considered. Fixations sequences over tables in the interface that were mostly horizontal were translated into evaluation modes. We made this assumption since by our definition, the evaluation of an object entailed reading lines in a table in order to understand if match criteria were met, and reading has been shown to be associated mostly horizontal fixation patterns (Simola, Salojärvi et al., 2008). In contrast, we

made the assumption that browsing corresponded to less goal-directed, more stochastic fixation patterns. A classification was needed and we validated these assumptions during pilot testing.

Because the eye tracker has an accuracy of 1 degree of visual angle, fixations in the interface regions allowed us to determine the level of information detail, even though the precise data element could not be identified. For example, a fixation on the table of matches identifies the nature of the data being accessed (i.e., matches), without necessarily needing to know precisely which match or which line in the table is being evaluated. This simple set of interpretations rules was chosen so as to provide the basis for a constrained set of behaviors, which translates into a more compact state space for the machine learning algorithms. Figure A.1 shows a typical pattern of fixations over the StrikeView interface, where each fixation is represented by a circle whose radius is indicative of the fixation duration.

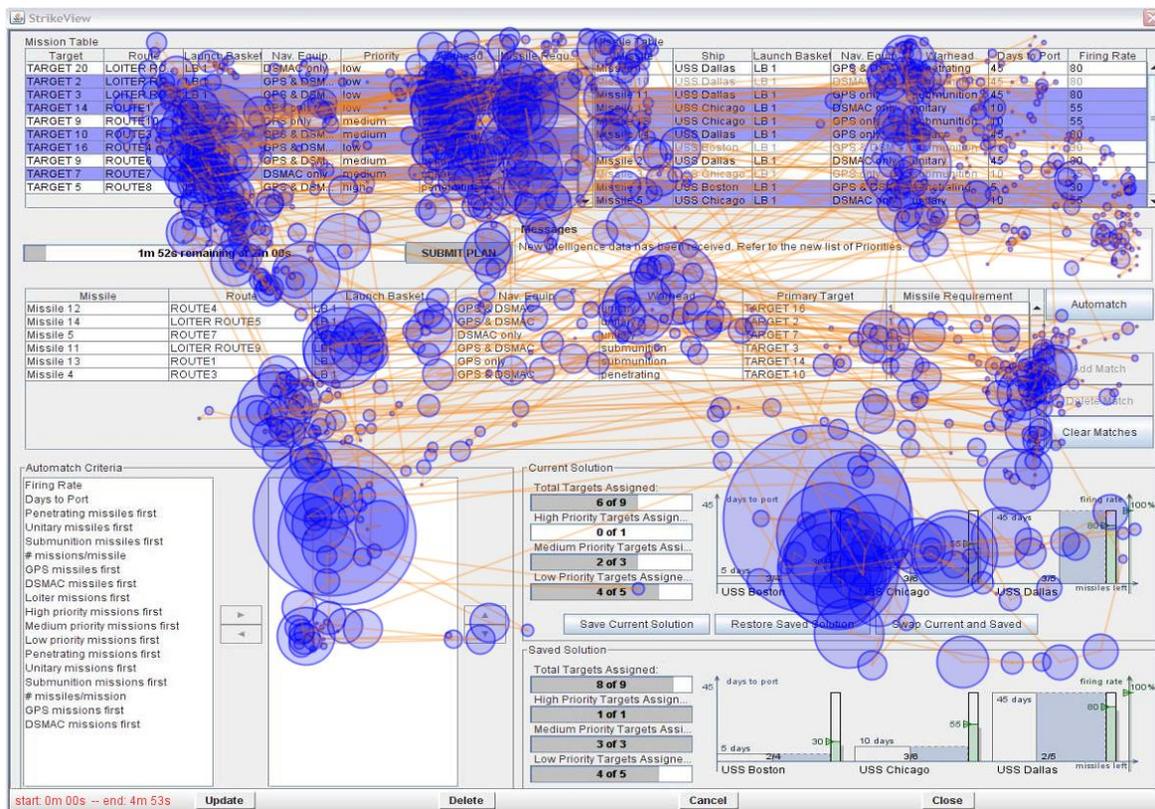


Figure A.1 Example of fixation patterns during a 1 minute use of the StrikeView interface

### A.1.4 Modeling Results

Because the main objective of this section is to determine the value of additional information contained in eye tracking data for HMM models of operator behavior, we built and compared two distinct HMMs: 1) an HMM based on UI events only (i.e., mouse clicks), and 2) an HMM with UI and eye tracking events.

#### Model Selection

We determined the optimal structure (i.e., the number of hidden states) for both models with and without eye tracking data by using the BIC metric. The BIC curves (Figure A.2) are created running models from size 2 to 24 and computing their respective BIC score. As previously established in Chapter 3, the optimal structure for the mouse click-only model is a 5-state model. In contrast, when the eye tracking information is incorporated in the data sets, the optimal model structure is best represented by a more complex 8-state model.

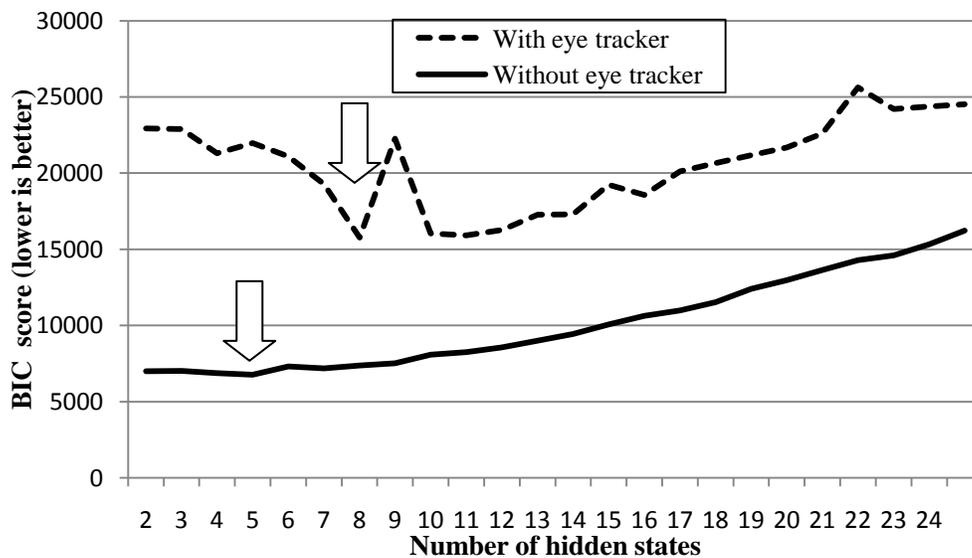
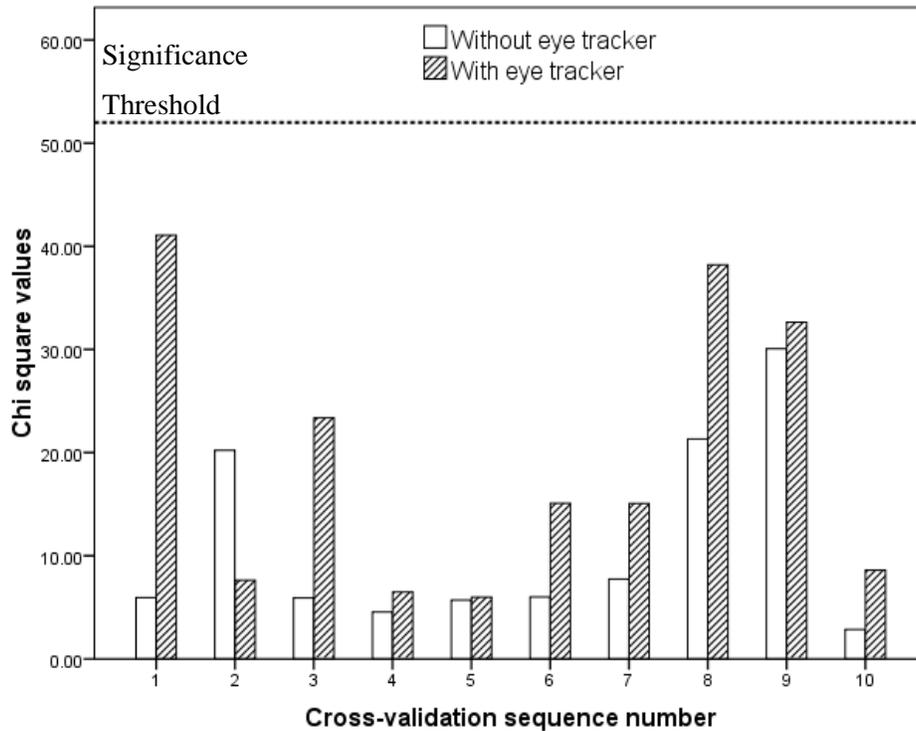


Figure A.2 BIC curves for the models trained with and without eye tracking data

#### Model Validation

The models were validated by running Monte-Carlo simulations in order to generate steady state observable distributions.



**Figure A.3**  $\chi^2$  values for model fit across the cross-validation sequences (lower is better).

These simulation-based distributions can be checked with the experimental distributions via a  $\chi^2$  test. The results (Figure A.3) show that none of the  $\chi^2$  values for the cross-validated models were significant ( $X^2=41.61$ ,  $p=0.35$ , was the largest  $\chi^2$  value for the eye tracking model and  $X^2=30.06$ ,  $p=0.85$  was the largest value for the model without eye tracking), which means that the two data sets are statistically likely to be no different. Thus, our model training process provides models that represent their respective training data sets correctly for both models while avoiding overfitting. It is then appropriate to assume that the models are appropriately trained and that comparisons can be made.

### ***Model Information Requirement***

The training process described above provides a diagnostic measure of the model learning through the posterior log-likelihood of the training data given the model. Although the log-likelihood of the training data is often used to assess the quality of a learned model across training iterations, this measure of model quality suffers from a practical weakness: the log-likelihoods obtained are data-set specific. Although the data sets used to train our models were generated from the same experimental data, the data used for the eye tracker model includes additional fixation information not available to the UI-only model. Therefore, the log-likelihood of the data given the model cannot be used as a valid comparison between the two

models. Similarly, most of the metrics derived from maximum likelihood measures such as the BIC, perplexity measures or Fisher information cannot be used to compare directly two models trained on different underlying data sets (Csiszár and Shields, 2000; Burnham and Anderson, 2002). Furthermore, although an information theoretic distance between two models can be computed by the Kullback-Leibler distance (Rabiner, 1989; Falkhausen, Reininger et al., 1995), also known as the KL divergence, this measure is computed for both models by using the same sequence of input data, which is again not appropriate in our case because the models were trained on different, albeit related, training data sets.

In order to objectively compare our models, we use (1) an entropy-based metric that can compare models across different data sets and (2) the predictive capabilities of the models, discussed next. We start by looking at the entropy  $H$  of the distribution of all possible sequences of length  $T$  of hidden states  $S^T = (S_1, S_2 \dots, S_T)$  that could have generated a set of  $T$  observations  $O^T = o^t$  given a model  $\lambda$ . This measure is written as  $H(S^T|O^T = o^t; \lambda)$  and can be computed as follows:

$$H(S^T|O^T = o^t; \lambda) = - \sum_{S^T} p(S^T = s^T|O^T = o^t; \lambda) \cdot \log_e p(S^T = s^T|O^T = o^t; \lambda) \quad (34)$$

The higher the entropy, the higher the uncertainty involved in tracking the hidden process with the model (Hernando, Crespi et al., 2005; Bishop, 2006). Alternatively, the measure  $H(S^T|O^T = o^t; \lambda)$  also describes the average information required to describe any hidden state sequence given the set of observations (in units of nats with the use of log base  $e$ ). To be meaningful,  $H$  should be normalized to  $\bar{H}$  with respect to the maximum entropy model  $H_{max}$ , i.e., the least informative model which is the one with equiprobable parameters (Eq. 35).

$$\bar{H}(X) = \frac{H(X)}{H_{max}(X)} \quad (35)$$

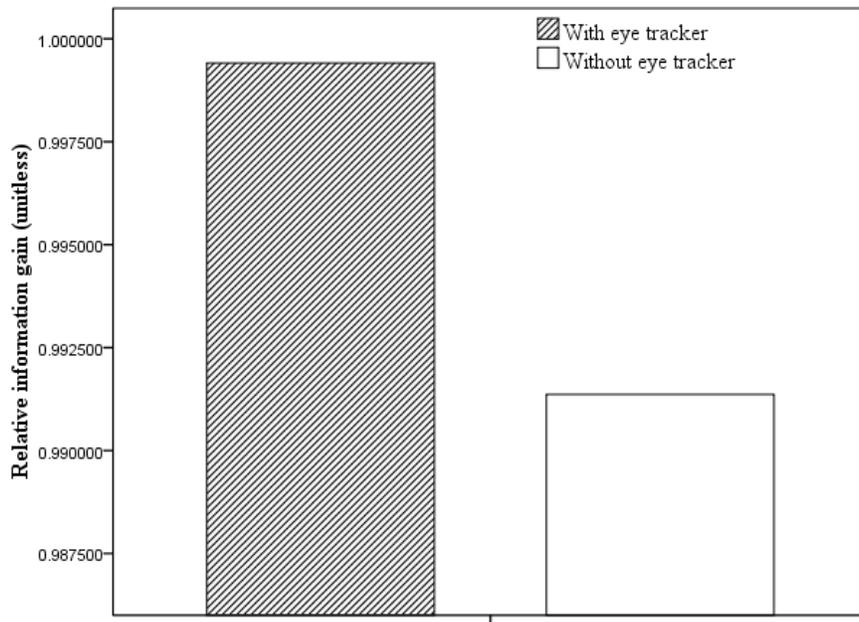
**Table A.6.1 Average normalized entropies of all possible hidden state sequences given the observations (unitless)**

	<b>With eye tracker</b>	<b>Without eye tracker</b>
$\bar{H}(S^T O^T = o^t; \lambda)$	0.588E-3	8.636E-3

The average normalized entropies of all the possible hidden state sequences (see Table A.6.1) show that the model trained with eye tracking data exhibits lower entropy than the model based only on UI events.

This means that the average number of nats (the unit of information entropy based on the natural logarithm) required to describe the state sequence of the model based on just the UI events is higher than for the model that takes eye tracking data into consideration. Conversely, the information content  $I$  gained by providing a model  $\lambda$  for modeling the training data can be estimated by comparing the entropy  $H$  of the trained models with that of the maximum entropy model. It is, however, more convenient to compute  $I$  with the normalized entropy  $\bar{H}$  (Eq. 36).

$$I = 1 - \bar{H}(S^T | O^T = o^t; \lambda) \quad (36)$$



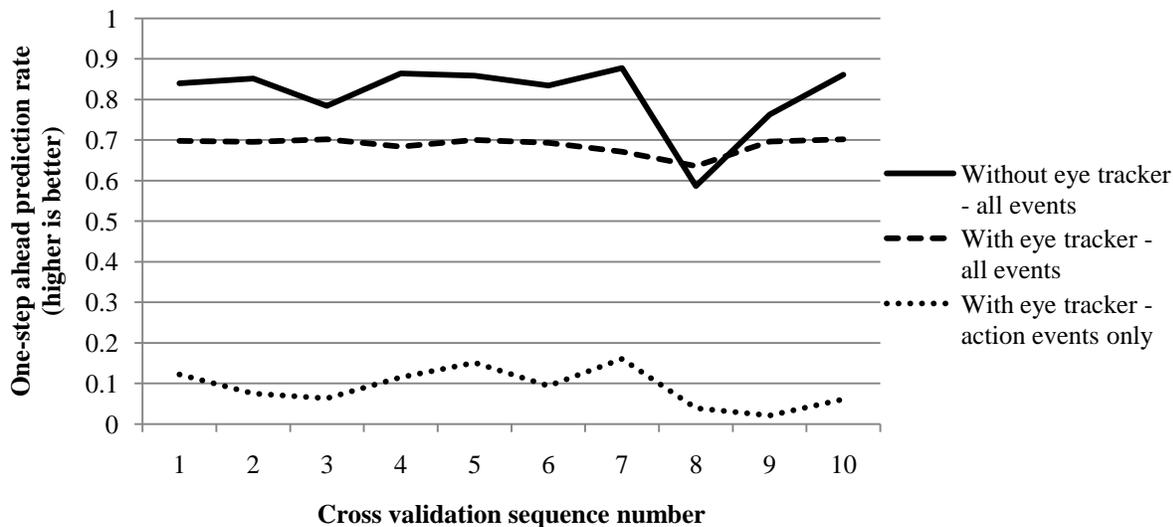
**Figure A.4 Information gained with respect to the maximum entropy model**

The results (Figure A.4) show that the model that makes use of the eye tracking data has a higher information gain (relative to the maximum entropy model) for modeling the training data than the model which relies on UI events only. This means that the eye tracking model provides more information than the UI-only model, which is not surprising given the larger amount of information contained in fixation patterns.

***Model one step-ahead predictive performance***

Whereas model entropy provides insights into information content, more important is a model’s ability to predict likely future deviations from the expected behavioral patterns. We can determine the one-step-

ahead prediction performance for both models by comparing the most likely observable given the model, and the observations that actually occurred at each time step in a test sequence. In the case of the model based on both UI and eye tracking events, we can look at either the overall prediction rate or at the prediction rate uniquely for user action events. In contrast, the model based only on UI data cannot be used to predict future eye movement because it has not been trained to do so. This distinction is important because in the context of PHSC, predicting user actions is more critical than predicting where the user will look next. Figure A.5 shows the results of both models' one-step-ahead predictive metric.



**Figure A.5 One step-ahead prediction rates for two models**

The results show that the action predictions are better with the UI event-only model (about 81% on average of correct one-step-ahead predictions) than with either the overall or action-only predictions of the eye tracker and mouse model (about 68% and 10% of correct predictions, respectively). This difference in prediction performance validates our claim that the model trained solely on UI events should be preferred in the PHSC context where user physical interactions are intermittent. Furthermore, these results show that the information content of the eye tracking data is, in fact, detrimental to the model's predictive power, likely due to the inclusion of the noisier eye tracking signal. The only exception is the 8<sup>th</sup> test sequence, which is an anomalous situation in which the user performed only one automatch action and submitted the resultant matches. This was the only occurrence of such behavior and, as evidenced by the consistently lower predictive score for the 8<sup>th</sup> sequence, both models were perplexed by this behavior. Interestingly, the drop in prediction rate was much higher for the mouse-only model, which could indicate that the mouse-only model not only provides better predictions, but also is more sensitive to anomalous behaviors.

### ***Summary***

The results shown in this section empirically validate the assumption that using UI events provides more useful models compared to those that comprise additional fine grained eye tracking data. Although it may seem counterintuitive that providing more data to a training set could result in a less useful model, feeding noisy data into a learning algorithm will decrease its ability to model the underlying process. The key point to consider is the quality, or relevance, of the additional data being supplied. Similar results were shown in speech recognition where models tended to be highly susceptible in the noise in the training data (Varga and Moore, 1990; Sanches, 2000). In the case of eye tracking, our results show that providing additional fixation data does add information to a model, while simultaneously decreasing its predictive ability. It is our contention that the issue lays in the low signal-to-noise ratio of the resulting data which results in degraded models. Thus, within the context of human supervisory control applications where user interactions are intermittent, we have shown that the inclusion of eye tracking data may add information to a model while degrading the model fit and ultimately limit the practical usefulness of model for predictive purposes.

## **A.2 First-Order Model Assumption**

The aim of this section is to investigate the appropriate model order for PHSC behaviors. HMMs rely on the first order Markov assumption which implies memoryless transitions from one state to another. This distinction is important for PHSC context because the assumption of memorylessness is unlikely to hold for PHSC operators. Yet, the question is whether the first order assumption provides a good enough approximation in exchange of simplified computations. Although HMMs have been widely used in the literature, the majority of the previous work used first order HMMs without specifically justifying the use of first order Markov models (Li and Biswas, 1999; Antonello, Manuele et al., 2002; Hayashi, 2003).

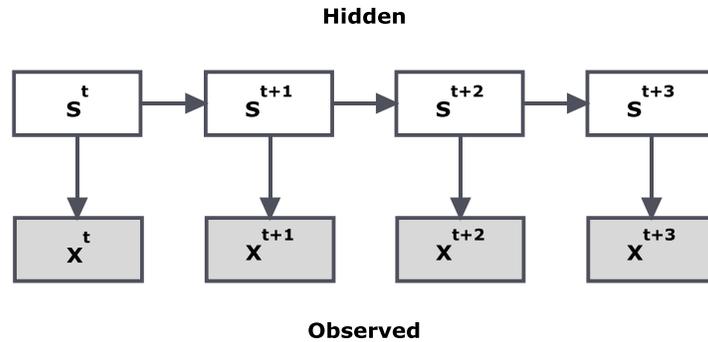
### ***A.2.1 Markov Assumption and HMMs***

The Markov property is central to the formulation of HMMs. This assumption can be formally stated as follows:

$$P(S^{t+1}|S^t, S^{t-1} \dots S^0) = P(S^{t+1}|S^t) \quad (37)$$

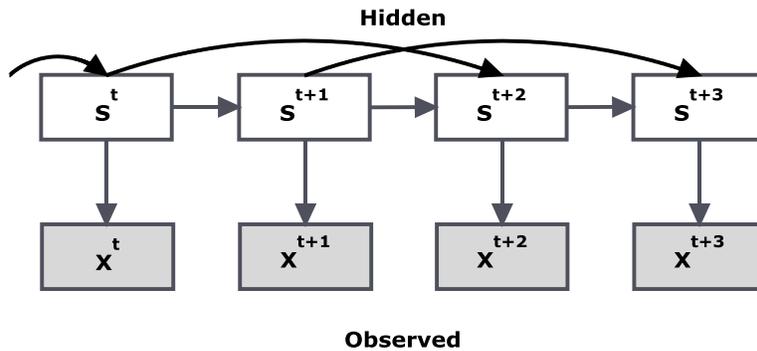
where  $S^t$  is the state at time  $t$ . In other words, the future states of the system are independent of the past states conditioned on the current state.

Figure A.6 shows the graphical model representation of a first order HMM highlighting this conditional independence (the arrows represent the dependencies). In particular, the graphical model clearly shows that  $S^{t+2} \perp S^t \mid S^{t+1}$ , i.e. that  $S^{t+2}$  is independent of  $S^t$  conditioned on  $S^{t+1}$ .



**Figure A.6** Graphical model representation of a first order HMM

From a computational perspective, the Markov assumption is exploited both in the forward/backward and the EM algorithms, which results in a computationally tractable dynamic programming implementation. The first order Markov assumption can be relaxed by using higher-order models. For instance, Figure A.7 shows a graphical model representation of a 2<sup>nd</sup> order HMM, which shows that  $S^{t+3} \perp S^t \mid (S^{t+1}, S^{t+2})$ . In other words, the 2<sup>nd</sup> order Markov assumption incorporates the notion of memory in the system.



**Figure A.7** Graphical model representation of a second order HMM

The Markov assumption has a significant impact on the structure of the models, and the order of the model should be chosen to match the properties of the underlying data process. However, while higher-order models may capture additional dependencies from the training data, they also involve a significant increase in model complexity which may mitigate their benefits in practice. In the context of PHSC operator models, a second order model would consider the current and the previous events in order to forecast future operator actions. In contrast, a first order model bases this forecast solely on the current

event. Therefore, higher order models may capture more in sophisticated behavioral patterns at the expense of model complexity. This section will investigate this issue by learning 2<sup>nd</sup> and 3<sup>rd</sup> order HMMs of the RESCHU data set. Then, the balance between model fit and model complexity can be established by using the BIC methodology. The next subsection provides the learning algorithms for 2<sup>nd</sup> and 3<sup>rd</sup> order HMMs.

### ***A.2.2 Learning higher-order HMMs***

Higher-order HMMs can be learned with algorithms similar to the ones used for first order HMMs previously shown in Section 2.3. The higher-order learning algorithms must be adapted to fit the state dependency structure imposed by the relaxation of the memory-less property. This section provides a description of the algorithms needed to learn second and third order models.

#### ***Second Order HMMs***

Second order HMMs are built through the following property:

$$P(S^{t+1}|S^t, S^{t-1} \dots S^0) = P(S^{t+1}|S^t, S^{t-1}) \quad (38)$$

In other words, each state transition depends not only on the current state but also on the previous state. From a structural perspective, an N-state 2<sup>nd</sup> order HMMs  $H$  with a dictionary size  $M$  and an observation sequence of length  $T$  is therefore defined by the parameters in Table A.6.2:

**Table A.6.2 2nd order HMM structure**

Initial Probability	$\pi_i = P(q_1 = s_i H)$ ( $i: 0 \dots N$ )
Initial State Transition	$\bar{a}_{ij} = P(q_2 = s_j q_1 = s_i, H)$ ( $i, j: 0 \dots N$ )
State Transition	$a_{ijk} = P(q_{t+2} = s_k q_{t+1} = s_i, q_t = s_j, H),$ ( $i, j, k: 0 \dots N, t: 0 \dots T$ )
Emission Probability	$b_k(l) = P(O_t = v_l q_t = s_k, H)$ ( $k: 0 \dots N, l: 0 \dots M, t: 0 \dots T$ )

The forward and backward equations for a 2<sup>nd</sup> order HMM can be extended from the first order HMM methodology (Kriouile, Mari et al., 1990; Watson and Chunk Tsoi, 1992; Thede and Harper, 1999).

### Second Order Forward Algorithm

Defining the forward parameter  $\alpha$  as the probability of the partial observation sequence from time  $1$  to  $t$  and transitions  $s_i s_j$  at times  $t - 1, t$  given the model  $H$ .

$$\alpha_t(i, j) = P(O_1^s \dots O_t^s, S_i^{t-1}, S_j^t; H) \quad (39)$$

The forward parameter can be computed via the following recursive process:

1- Initialization:

$$\begin{aligned} \alpha_1(i) &= \pi_i b_i(O_1), \text{ for } 1 \leq i, j \leq N \\ \alpha_2(i, j) &= \alpha_1(i) \bar{a}_{ij} b_j(O_2), \text{ for } 1 \leq i, j \leq N \end{aligned} \quad (40)$$

2- Recursion:

$$\alpha_{t+1}(j, k) = \left( \sum_{i=1}^N \alpha_t(i, j) a_{ijk} \right) b_k(O_{t+1}), \text{ for } 2 \leq t \leq T - 1 \quad (41)$$

3- Termination:

$$P(O|\lambda) = \sum_{i=1}^N \sum_{j=1}^N \alpha_T(i, j) \quad (42)$$

### Second Order Backward Parameters:

Similarly, defining the backward parameter  $\beta$  as the probability of the partial observation sequence from  $t + 1$  to  $T$ , given transitions  $s_i s_j$  at times  $t - 1, t$  and the model  $H$ .

$$\beta_t(i, j) = P(O_t^s \dots O_T^s | S_i^{t-1}, S_j^t; H) \quad (43)$$

The backward parameter can be computed via the following recursive process:

1- Initialization

$$\beta_T(i, j) = 1, \text{ for } 1 \leq i, j \leq N \quad (44)$$

2- Recursion

$$\beta_t(i, j) = \sum_{k=1}^N a_{ijk} b_k(O_{t+1}) \beta_{t+1}(j, k), \text{ for } T - 1 \geq t \geq 2 \quad (45)$$

The forward and backward parameters provide the basis for the Baum-Welch algorithm, and the parameter re-estimation for a 2<sup>nd</sup> order HMM are provided below.

### Second Order Re-estimation

In addition to the usual  $\xi$  and  $\gamma$  parameters, it is useful to define  $\eta_t(i, j, k)$  as the probability of being in states  $s_i$ ,  $s_j$  and  $s_k$  respectively at times  $t - 1, t, t + 1$  given the model and the observation sequence.

$$\begin{aligned}\eta_t(i, j, k) &= P(q_{t-1} = s_i, q_t = s_j, q_{t+1} = s_k | O, H) \\ \eta_t(i, j, k) &= \frac{\alpha_t(i, j) a_{ijk} b_k(O_{t+1}) \beta_{t+1}(j, k)}{P(O|H)}, \text{ for } 1 \leq t \leq T - 1\end{aligned}\quad (46)$$

The 2<sup>nd</sup> order definition of the parameters  $\xi$  and  $\gamma$  are similar to that of the first order HMMs. The parameter  $\xi_t(i, j)$  represents the probability of being in state  $s_i$  at time  $t$  and in state  $s_j$  at time  $t + 1$ , given the model and the observation sequence.

$$\begin{aligned}\xi_t(i, j) &= P(q_t = s_i, q_{t+1} = s_j | O, H) \\ \xi_t(i, j) &= \sum_{k=1}^N \eta_{t+1}(i, j, k)\end{aligned}\quad (47)$$

$$\begin{aligned}\gamma_t(i) &= P(q_t = s_i | O, H) \\ \gamma_t(i) &= \sum_{j=1}^N \xi_t(i, j)\end{aligned}\quad (48)$$

Finally, the parameters of a 2<sup>nd</sup> order HMM are re-estimated as follows:

$$\begin{aligned}\pi_i &= \gamma_1(i) \\ \bar{a}_{ij} &= \frac{\xi_1(i, j)}{\gamma_1(i)} \\ a_{ijk} &= \frac{\sum_{t=1}^{T-3} \eta_{t+1}(i, j, k)}{\sum_{t=1}^{T-3} \xi_t(i, j)} \\ b_k(l) &= \frac{\sum_{t=1, O_t=v_l}^T \gamma_t(k)}{\sum_{t=1}^T \gamma_t(k)}\end{aligned}\quad (49)$$

### Third Order HMMs

The third algorithms needed for 3<sup>rd</sup> order HMMs can be directly extended those used for 2<sup>nd</sup> order HMMs. The 3<sup>rd</sup> order Markov assumption that guides the structure of the HMM can be written as:

$$P(S^{t+1} | S^t, S^{t-1} \dots S^0) = P(S^{t+1} | S^t, S^{t-1}, S^{t-2})\quad (50)$$

The structure of an N-state 3<sup>rd</sup> order HMMs  $H$  with a dictionary size  $M$  and an observation sequence of length  $T$  is as shown in Table A.6.3:

**Table A.6.3 Third order HMM structure**

Initial Probability	$\pi_i = P(q_1 = s_i H)$ ( $i: 0 \dots N$ )
Initial State Transitions	$\bar{a}_{ij} = P(q_2 = s_j q_1 = s_i, H)$ ( $i, j: 0 \dots N$ ) $\bar{a}_{ijk} = P(q_3 = s_k q_1 = s_i, q_2 = s_j, H)$ ( $i, j, k: 0 \dots N$ )
State Transition	$a_{ijkl} = P(q_{t+3} = s_l q_{t+2} = s_k, q_{t+1} = s_j, q_t = s_i, H)$ ( $i, j, k, l: 0 \dots N, t: 0 \dots T$ )
Emission Probability	$b_k(l) = P(O_t = v_l q_t = s_k, H)$ ( $k: 0 \dots N, l: 0 \dots M, t: 0 \dots T$ )

*Third Order Forward/Backward Algorithm*

The 3<sup>rd</sup> order forward algorithm proceeds as follows:

Definition

$$\alpha_t(i, j, k) = P(O_1 O_2 \dots O_t, q_{t-2} = s_i, q_{t-1} = s_j, q_t = s_k | H)$$

Initialization

$$\begin{aligned} \alpha_1(i) &= \pi_i b_i(O_1), \text{ for } 1 \leq i, j \leq N \\ \alpha_2(i, j) &= \alpha_1(i) \bar{a}_{ij} b_j(O_2), \text{ for } 1 \leq i, j \leq N \\ \alpha_3(i, j, k) &= \alpha_2(i) \bar{a}_{ijk} b_k(O_3), \text{ for } 1 \leq i, j, k \leq N \end{aligned}$$

Recursion

$$\alpha_{t+1}(j, k, l) = \left( \sum_{i=1}^N \alpha_t(i, j, k) a_{ijkl} \right) b_l(O_{t+1}), \text{ for } 3 \leq t \leq T - 1$$

Termination

$$P(O|H) = \sum_{i=1}^N \sum_{j=1}^N \sum_{k=1}^N \alpha_T(i, j, k)$$

(51)

And similarly, the 3<sup>rd</sup> order backward algorithm is as follows:

Definition

$$\beta_t(i, j, k) = P(O_{t+1} O_{t+2} \dots O_T | q_{t-2} = s_i, q_{t-1} = s_j, q_t = s_k, H)$$

Initialization

$$\beta_T(i, j, k) = 1, \text{ for } 1 \leq i, j, k \leq N$$

(52)

Recursion

$$\beta_t(i, j, k) = \sum_{l=1}^N a_{ijkl} b_l(O_{t+1}) \beta_{t+1}(j, k, l), \text{ for } T-1 \geq t \geq 3$$

*Third Order Re-estimation*

Extending the 2<sup>nd</sup> order formulation, it is useful to define  $\zeta_t(i, j, k, l)$  as the probability of being in states  $s_i, s_j, s_k$  and  $s_l$  respectively at times  $t-2, t-1, t, t+1$  given the model and the observation sequence.

$$\begin{aligned} \zeta_t(i, j, k, l) &= P(q_{t-2} = s_i, q_{t-1} = s_j, q_t = s_k, q_{t+1} = s_l | O, H) \\ \zeta_t(i, j, k, l) &= \frac{\alpha_t(i, j, k) a_{ijkl} b_l(O_{t+1}) \beta_{t+1}(j, k, l)}{P(O|H)}, \text{ for } 1 \leq t \leq T-1 \end{aligned} \quad (53)$$

The extension of the parameters  $\eta, \xi$  and  $\gamma$  for 3<sup>rd</sup> order HMMs is straightforward:

$$\begin{aligned} \eta_t(i, j, k) &= P(q_{t-1} = s_i, q_t = s_j, q_{t+1} = s_k | O, H) \\ \eta_t(i, j, k) &= \sum_{l=1}^N \zeta_{t+1}(i, j, k, l) \end{aligned} \quad (54)$$

$$\begin{aligned} \xi_t(i, j) &= P(q_t = s_i, q_{t+1} = s_j | O, H) \\ \xi_t(i, j) &= \sum_{k=1}^N \eta_{t+1}(i, j, k) \end{aligned} \quad (55)$$

$$\begin{aligned} \gamma_t(i) &= P(q_t = s_i | O, H) \\ \gamma_t(i) &= \sum_{j=1}^N \xi_t(i, j) \end{aligned} \quad (56)$$

Finally the parameter re-estimation for 3<sup>rd</sup> order HMMs can be written as:

$$\begin{aligned} \pi_i &= \gamma_1(i) \\ \bar{a}_{ij} &= \frac{\xi_1(i, j)}{\gamma_1(i, j)} \\ \bar{a}_{ijk} &= \frac{\eta_2(i, j, k)}{\xi_1(i, j)} \\ a_{ijkl} &= \frac{\sum_{t=1}^{T-3} \zeta_{t+2}(i, j, k, l)}{\sum_{t=1}^{T-3} \eta_{t+1}(i, j, k, l)} \end{aligned} \quad (57)$$

$$b_l(h) = \frac{\sum_{t=1}^T \gamma_t(l)}{\sum_{t=1}^T \gamma_t(l)}$$

### A.2.3 Complexity analysis

Table A.6.4 provides a complexity comparison between first, second and third order HMMs with  $N$  hidden states and a dictionary of size  $D$ . This information is valuable because it provides an idea of the significant increase in complexity with each model order increment.

**Table A.6.4 Higher-order model complexity analysis**

	Number of parameters	Run Time
First order HMM	$N + N^2 + ND$	$O(N^2D)$
Second order HMM	$N + N^2 + N^3 + ND$	$O(N^3D)$
Third order HMM	$N + N^2 + N^3 + N^4 + ND$	$O(N^4D)$

Table A.6.4 shows that each model order increment leads to a geometric increase in the number of parameters and runtime. This is important for computational reasons as more complex models will take longer to train. More importantly, from a training data perspective, the increase in the number of model parameters means that a significantly larger data set is needed in order to elicit the higher-order relationship synthesized by the models.

### A.2.4 Results

The models of different orders can be compared via the BIC metric which balances the model fit to the training data and the model complexity.

Figure A.8 shows the BIC obtained for the first, second and third order HMMs trained on the RESCHU data set described in Chapter 3. In general, learning models that contain a larger number of parameters than training data points is not recommended due to the overfitting. For this reason, models that contain more than 3420 parameters were not considered. For 2<sup>nd</sup> order and 3<sup>rd</sup> models, this bounded the number of hidden states to 14 and 8 respectively.

The first order BIC curve is the same as the one shown in Figure 3.11, and shows that the minimal BIC is reached for an 8-state first order HMM. In contrast, the minimal BIC is reached for 5 hidden states

(BIC=32133.15) and 2 hidden states (BIC=42348.05) for the 2<sup>nd</sup> and 3<sup>rd</sup> order HMM respectively. Additionally, the increase in BIC scores as the number of states goes higher markedly different for the first, second and third order model. The increased penalty incurred by the more complex models is readily apparent from the graphs, even for 2-state models. Therefore, according to the BIC criteria, the additional relationships captured by higher order models do not balance out the significant increase in model complexity. These results suggest that, given the RESCHU data set, the use of first order HMMs for modeling PHSC behaviors provides a practical approximation of higher order models.

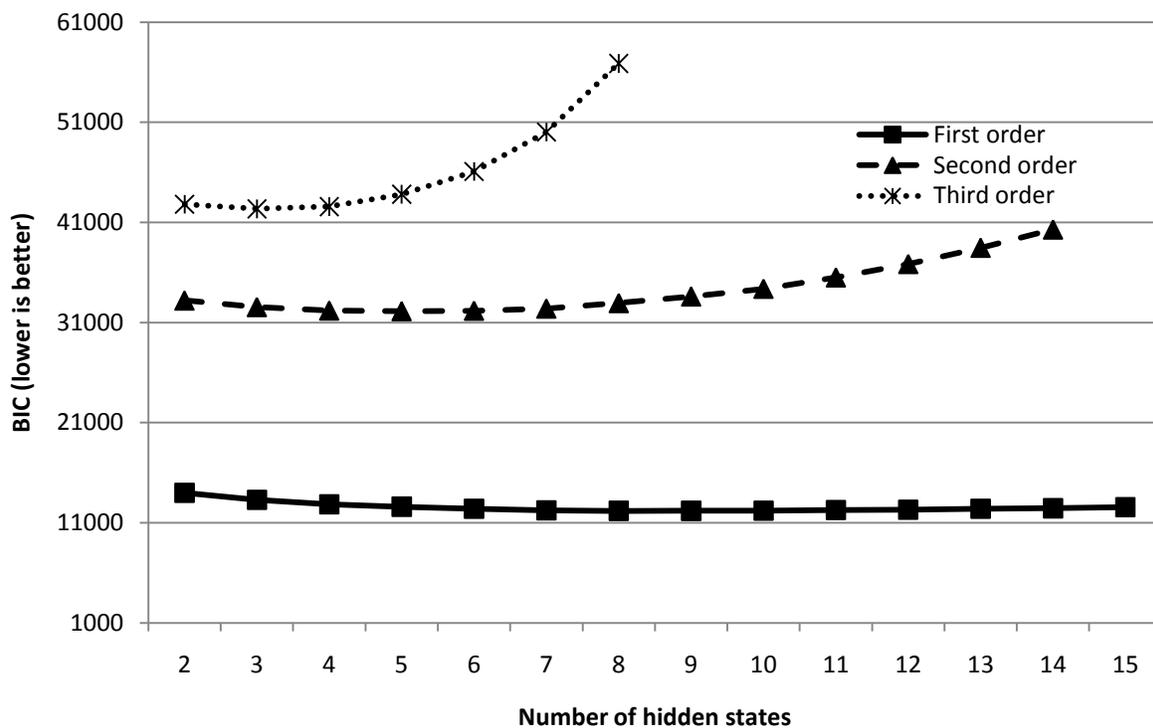


Figure A.8 Higher order HMMs BIC comparison

### A.3 Learning Methodology

The methodology described in Chapter 3 relies on unsupervised learning technique in which the algorithm only makes use of the information contained in the training data set to extract the optimal set of model parameters. The alternate learning algorithms are “supervised” in that they require the data to be augmented with *a priori* information, or labels, in order to guide the learning process. The labels usually consist of input data associated with the expected model output, defined by a subject matter expert. The supervised methodology has been favored by the machine learning community in the past for two

reasons: (1) the simplest supervised learning methods offer better computational efficiency compared to unsupervised learning methods, and (2) the labels in the training data are assumed to be derived from reliable ground-truth, thereby increasing the amount of information captured in the model.

The methodology proposed in Chapter 3 relies on unsupervised learning techniques because of the assumption that it is fundamentally impossible to correctly label the training data when operator cognitive states are not observable in the context of supervisory control behavior. Without reliable ground-truth, human bias (Tversky and Kahneman, 1974; Kahneman and Tversky, 1979) is unavoidably introduced into training data labeling, which greatly influences the learning process and may generate uninformative or incorrect models.

In order to support the use of unsupervised learning in the proposed methodology, this section compares the models obtained via unsupervised learning with those obtained via two supervised learning techniques applied to the RESCHU data set: purely supervised learning (Rabiner and Juang, 1986) and smooth supervised learning (Hiroshi, 1997).

### ***A.3.1 Classic Supervised Learning***

Classic supervised learning is the simplest way to extract model parameters from labeled data. Assuming the training data consists of sequence of observations  $O^s$ , it can be shown that the MLE of the emission probability distribution given the training data is distributed according to the frequency of emissions in the data (Aldrich, 1997). These frequencies can be obtained by counting how often an observation was generated by a given state. Similarly, the most likely transition probabilities are distributed according to the frequency of state transition observed in the training data. In the case of HMMs, the frequency of state transitions or observation emissions from a particular state cannot be counted because states are hidden. However, supervised learning makes the assumption that during training, we have access to the underlying state sequence and can therefore “label” each observation in  $O^s$  with the corresponding true, hidden state. The transition matrix of  $A = \{a_{ij}\}$  can therefore be computed directly by counting the relative frequency of the transition between all states  $i$  and  $j$ . Similarly, the emission functions  $B = \{b_j(c)\}$  can be computed by counting the number of times a specific observation  $c$  has been observed given a state  $j$ . More formally, recall from Section 2.3 the definition of  $count(s \rightarrow s')$  as the number of time state  $s'$  follows state  $s$  and  $count(s \rightsquigarrow c)$  as the number of time state  $j$  is paired with emission  $c$ :

$$\text{count}(s \rightarrow s') = \sum_{j=1 \dots l_s-1} \llbracket S_j = s \wedge S_{j+1} = s' \rrbracket \quad (58)$$

$$\text{count}(s \rightsquigarrow c) = \sum_{j=1 \dots l_s} \llbracket S_j = s \wedge O_j = c \rrbracket \quad (59)$$

The MLE estimates  $\hat{a}_{ij}$  of  $a_{ij}$  are:

$$\hat{a}_{ij} = \frac{\sum_{i=1 \dots N} \text{count}(s \rightarrow s')}{\sum_{i=1 \dots N} \sum_{s'} \text{count}(s \rightarrow s')} \quad (60)$$

Similarly, the MLE estimates  $\hat{b}_j(c)$  of  $b_j(c)$  are:

$$\hat{b}_j(c) = \frac{\sum_{i=1 \dots N} \text{count}(i, s \rightsquigarrow x)}{\sum_{i=1 \dots N} \sum_x \text{count}(i, s \rightsquigarrow x)} \quad (61)$$

This supervised learning technique to compute the HMM model parameters is relatively simple and runs in  $O(l_s)$ , where  $l_s$  is the length of a sequence.

### A.3.2 Smooth Supervised Learning

Smooth supervised learning was first introduced by Baldi et al. (1994) in order to avoid issues with sudden jumps or absorbing probabilities of 0 during the parameter update process. The absorption property of null probabilities is an issue because once a transition or emission function is set to 0, it cannot be used again. The idea for the supervised case is to minimize the distance between the *a priori* labels and the labels estimated as most likely by the HMM. This algorithm can be tailored for sequence discrimination (Hiroshi, 1997), and we can parameterize  $a_{ij}$  and  $b_j(c)$  with functions of  $\omega_{ij}$  and  $v_j(c)$  defined as (with  $\lambda$  being a constant):

$$a_{ij} = \frac{e^{\lambda \omega_{ij}}}{\sum_k e^{\lambda \omega_{ik}}} \quad (62)$$

$$b_j(c) = \frac{e^{\lambda v_j(c)}}{\sum_k e^{\lambda v_j(k)}}$$

Let  $p_s = P(O^s|H)$  be the target value of the likelihood of the pre-labeled observations and associated symbols given the HMM  $H$ . The probability  $p_s$  will depend on the length of the sequence, so we introduce  $\delta$  which scales the probability  $p_s$  with respect to the length of the sequence.  $C_a$  and  $C_b$  are constants that normalize  $\delta$  for different observation sequence sizes:

$$\begin{aligned} p_s &= \log(P(O, S|H)) \\ \delta_s &= 1 - C_a \frac{p_s}{(C_b - p_s^2)} \end{aligned} \quad (63)$$

The algorithm thus tries to maximize  $\delta$  in order to maximize the fit of the model to the data. Given  $\eta_\omega$  and  $\eta_\nu$  as learning rates, the update rules for  $\omega_{ij}$  and  $\nu_j(c)$  are as follows:

$$\begin{aligned} \Delta\omega_{ij} &= \eta_\omega \sum_s \delta_s \sum_{t=0}^{l_s} (\varepsilon_t(i, j) - a_{ij}\gamma_t(i)) \\ \Delta\nu_j(c) &= \eta_\nu \sum_s \delta_s \sum_{t=0}^{l_s} (\gamma_t(j)_{O_t^s=c} - b_j(c)\gamma_t(j)) \end{aligned} \quad (64)$$

In order to reach convergence, the constants and learning rates need to be adapted for each training set. Because the solution space is highly-non-linear, there is no analytical method to choose these parameters appropriately. As a result, the constants and rates have to be found by a time-consuming process of trial and error to maximize the model likelihood.

### ***A.3.3 Results***

#### ***State Labeling***

For the HMM leveraging supervised learning, labels are needed. However, due to the futuristic nature of the single operator-multiple unmanned vehicle system, no subject matter expert is available to label the data by hand. Through a cognitive task analysis, we derived an initial set of labels known to be recurrent based on sets of previously-identified common cognitive functions for UAV tasks (Nehme, Crandall et al., 2007), such that user-interface interactions could be grouped as clusters or states. For the supervised learning portion of this work, the *a priori* labels consisted of:

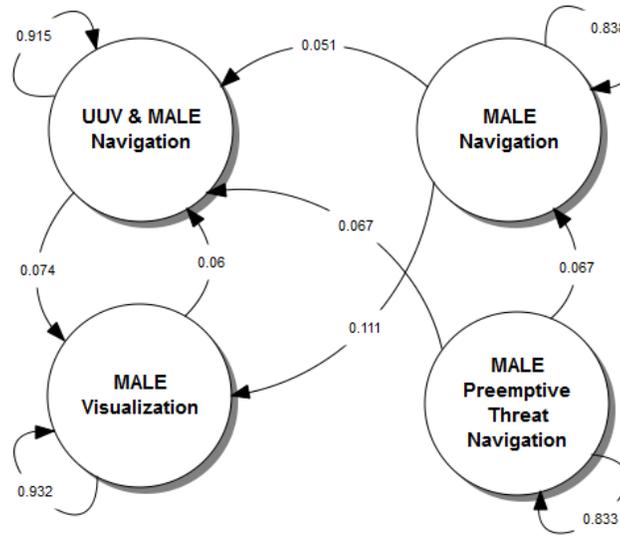
1. Navigation: map selections and interactions with goals

2. Monitoring: interaction with the UVs based on selection on the sidebar or the map,
3. Visual Task: set of action that results in the visual task engagement action , and
4. Preemptive Threat Navigation: adding a series of waypoints in order to change the course of the vehicle should the need arise.

This labeling scheme covered about 80% of the training sequences, and observables that did not get labeled were dropped from the training set. These labels align with those basic underlying operator functions that form the core of supervisory control of unmanned vehicles which include navigation, vehicle health and status monitoring, and payload management (Cummings, Bruni et al., 2007). In RESCHU, the payload management task is the visual task.

### *Classic Supervised Model*

Supervised learning algorithms find the most likely set of parameters for state transitions and the emission functions given a training data set. The model in Figure A.9 represents the HMM obtained with the classic supervised learning method. All transitions with a weight under 5% are not shown for legibility purposes. Models under this learning paradigm contain four states which correspond to interaction types as defined by the a priori patterns of most likely observable states. The annotated arrows between the states represent the probability of going from one state to another. The supervised learning process leverages the pre-defined state labels and learns the most likely set of parameters for the HMM. In the case of Figure A.9, the HMM comprises 4 hidden states. The first state contains both UUV and MALE navigation (repeated map selections and interaction with goals). The aggregation of the MALE and UUV operands is likely due to the comparatively lower interaction frequencies between the operator and the UUVs. The second state focuses uniquely on similar navigation interactions, but for MALEs only. In contrast, the third state embodies MALE threat navigation, which is adding a series of waypoints in order to change the course of the vehicle if needed. This differs from navigation tasks in that the operator only acts on waypoints and does not interact with goals. Finally, the last state is the MALE visualization, which corresponds to the visual target identification task in RESCHU. The obtained model shows that operator interactions with the HALEs do not appear as a distinct state, even though we know that they exist. While operators interacted with the HALEs less than they did with the MALE UAVs and the UUVs, because HALE use was required prior to use of a MALE for unknown targets, we anticipated that this would be a state with a clearly assigned meaning. However, this was not seen in this supervised learning model.



**Figure A.9 Supervised learning model of a single operator of multiple unmanned systems**

***Smooth-Supervised Models***

The model in Figure A.10 represents the model obtained with the smooth supervised learning method. Again, all transitions with a weight under 5% are not drawn for legibility purposes. The model obtained is somewhat different from the one obtained through classic supervised learning (Figure A.9), although three of the four states are the same. The first state aggregates MALE visualization and UUV navigation tasks. This aggregation denotes that UUV navigation and MALE visualization are statistically clustered together and indicates that a number of MALE visual tasks tended to either precede or follow a UUV navigation interaction. This is possibly due to the spatial distribution of the targets along the water body on the map. The second state expresses MALE visual tasks and the third state represents MALE normal navigation, and finally the last state corresponds to MALE threat navigation. While the transition probabilities between hidden states is less deterministic (as indicated by the higher number of likely transitions between hidden states) than in the classic supervised model, operator interactions with the HALEs again disappear and do not appear as a distinct state as defined by the learning algorithm.

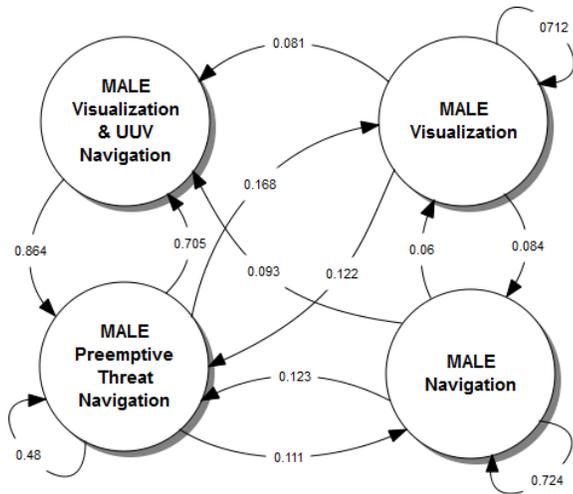


Figure A.10 Smooth supervised learning model of a human operator of multiple unmanned systems

*Discussion*

For reference, the 8-state HMM obtained through unsupervised learning is shown in Figure 3.12, reproduced below.

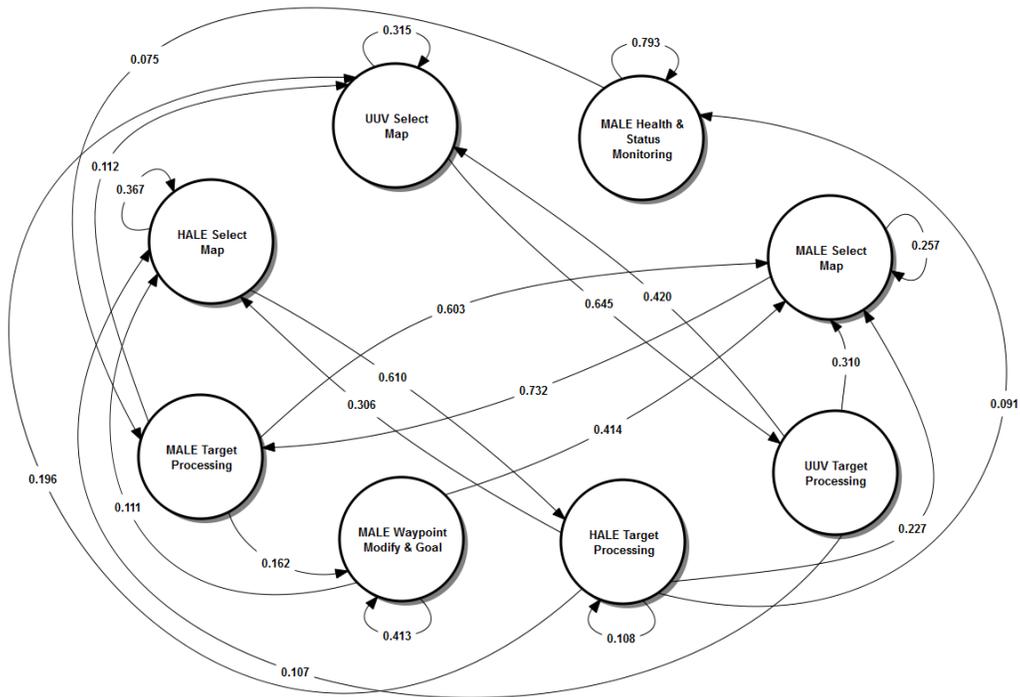


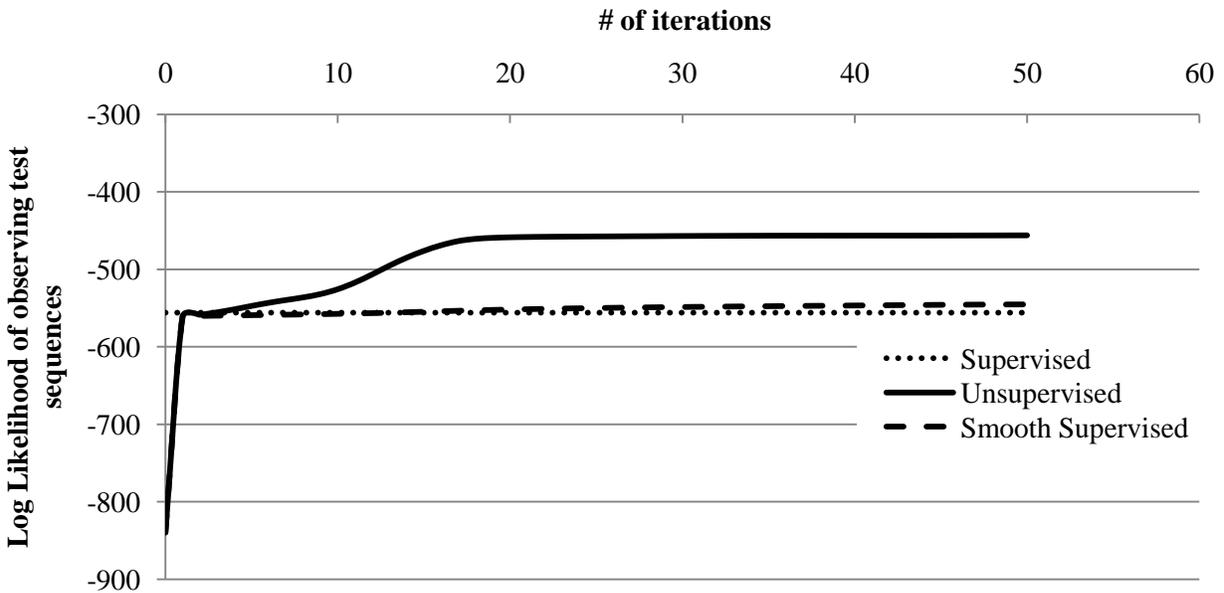
Figure 3.11 8-state HMM for RESCHU

When compared to the supervised models, the unsupervised model is markedly different in a number of respects. First, the model contains 8 states indicating that the increased model complexity is balanced by a better fit to the training data. Furthermore, due to the additional number of hidden states contained in this model, the interactions with UUVs and HALEs are explicitly modeled by two distinct states each whereas they were not apparent in the supervised and smooth-supervised models. This denotes that the unsupervised model was able to recognize the much less frequent interactions with UUVs and HALEs as a distinct state and qualitatively different from that with the MALEs.

Figure A.11 show the likelihood of the models obtained through each learning method on a test-set of sequences across a number of learning iterations. As expected, the supervised algorithm converges in the first iteration and provides a constant performance baseline. The first few iterations of the smooth supervised algorithm, conversely, are quite poor. However, at the 25th iteration, the smooth supervised model surpasses the classic supervised model and plateaus at around the 30th iteration. The first few learning iterations of the unsupervised model behave very closely to the smooth supervised. After the 3rd iteration, however, while the smooth supervised model plateaus for the first time, the unsupervised algorithm log likelihood continues to increase and converges at the 20th learning iteration. In terms of log likelihood, the performance differences are clear in that the unsupervised learning method gives rise to a model that is more likely than both supervised methods. The smooth supervised model provides slightly superior posterior log likelihoods than the classic supervised one.

Adopting a human-centric and cost-benefit point of view, it is interesting to compare how much human effort was required to generate the above models. For both supervised methods, the cost of labeling the data was quite high, as our initial undertaking was to execute a cognitive task analysis of the single operator - multiple unmanned systems in order to define a likely set of behaviors. Cognitive task analyses are labor intensive and are somewhat subjective, so there is no guarantee that the outcome behaviors are correctly identified. Moreover, these *a priori* defined patterns then had to be tagged in all the sequences in order to construct the corpus of training and testing data. In order to avoid the known risks of human judgment bias (Tversky and Kahneman, 1981) in the state definition process, an iterative approach was adopted in which multiple acceptable sets of state definitions were compared to the data. The set of definitions that provided the better explanation for the states was then chosen. It is important to note that expert knowledge of the task was required in all phases of this lengthy process. Thus, in addition to being extremely time-intensive, it is recognized that expert labeling is a costly and sometimes subjective process that can unnecessarily constrain the resulting models to the types of behaviors seen as important

by human experts (Hoey, 2007), which could ultimately be flawed especially for any attempt to label cognitive or operator states.



**Figure A.11 Model fit in terms of test set likelihood for the three different training techniques**

In addition to the quantitative metrics such as convergence speed and performance, it is interesting to analyze the models for the explanatory mechanism they can provide. For the supervised models, the results obtained are similar in that they emphasize the role of the MALEs and UUVs. Both supervised models, based on human-biased grammar, disregard a major part of the problem space: the existence of a 3<sup>rd</sup> vehicle category (the HALEs). The unsupervised learning technique, on the contrary, segregated the HALE and UUV interactions in separate states (2 hidden states for each of the vehicle types). The unsupervised technique also detected the regular patterns between map selection and target processing for each type of UVs. Furthermore, the unsupervised models also synthesized the comparatively higher number of interactions with MALEs by devoting 4 out of the 8 states to that type of UV. Such examples show the richness of the interpretation that can be obtained from analyzing a non-biased model that is based on statistical properties of operator interactions. Such unsupervised approaches could actually be used to augment cognitive task analyses in order to provide more objective results in what is known to be a very subjective process.

These results, both quantitative (i.e. model likelihood) and qualitative (i.e. model interpretation), demonstrate that for the purpose of modeling PHSC operator states, the use of supervised learning is

likely flawed. Not only did the supervised models yield poorer prediction rates, but also failed to capture important characteristics of operator behavior. The poor results could be blamed, quite rightly, to poor *a priori* labeling of the states, and that the results could have been very different with better labeling. However, this again highlights the subjective nature of expert state labeling in the presence of uncertainty. In the specific context of human supervisory control modeling, the results support that it is very difficult, if not impossible, to obtain a correct set of labels. Therefore, within the scope of PHSC applications, the use of unsupervised learning techniques should be favored of potentially biased supervised methods.

## **A.4 Summary**

This appendix validated three major assumptions needed to model PHSC behavior through HMMs. First, the assumption that UI events provide a rich source of data for modeling was validated by comparing models based solely on UI data and models based on UI data in conjunction with eye tracking data. The results showed that the models based on UI data only were as good, if not better, than the models that incorporate larger but noisier sources of information. The second section of this chapter validated the use of first-order HMMs, i.e. models that follow the first order Markov assumption. First, second and third order models were built and their fit vs. complexity was evaluated via the BIC. The results showed that the increased complexity incurred by 2<sup>nd</sup> and 3<sup>rd</sup> order models was not balanced by an increased fit to the training data. Finally, the last section of this chapter validated the use of unsupervised learning methods for HMMs via a comparison with two supervised learning techniques. The results showed that unsupervised learning that does not rely on possibly biased data provided better models.

## REFERENCES

- Abdel-Malek, A. and V. Marmarelis (1990). A model of human operator behavior during pursuit manual tracking-what does it reveal? IEEE International Conference on Systems, Man and Cybernetics.
- Aldrich, J. (1997). "R. A. Fisher and the making of maximum likelihood 1912--1922." Journal of Statistical Science **12**(3): 162--176.
- Allanach, J., H. Tu, S. Singh, P. Willett and K. Pattipati (2004). "Detecting, Tracking, and Counteracting Terrorist Networks via Hidden Markov Models." IEEE Aerospace Conference Proceedings: 3246-3257.
- Anderson, J. R. (1993). Rules of the Mind. Hillsdale, New Jersey, Lawrence Erlbaum Associates.
- Anderson, J. R. and M. P. Matessa (1997). "A production system theory of serial memory." Psychological Review **104**: 728-148.
- Anderson, J. R., Y. Qin, J. M. Fincham and A. Stocco (2008). "A central circuit of the mind." Trends in Cognitive Science **12**(4): 136-143.
- Andreassi, J. L. (1989). Psychophysiology : human behavior and physiological response. Hillsdale, N.J., L. Erlbaum Associates.
- Antonello, P., B. Manuele and M. Vittorio (2002). A Hidden Markov Model-Based Approach to Sequential Data Clustering. Proceedings of the Joint IAPR International Workshop on Structural, Syntactic, and Statistical Pattern Recognition, Springer-Verlag.
- Ayat, N. E., M. Cheriet and C. Y. Suen (2005). "Automatic model selection for the optimization of SVM kernels." Pattern Recognition **38**(10): 1733-1745.
- Baldi, P. and Y. Chauvin (1994). "Smooth On-Line Learning Algorithms for Hidden Markov Models." Neural Computation **6**: 307-318.
- Bartels, M. and S. P. Marshall (2006). Eye tracking insights into cognitive modeling. Proceedings of the 2006 symposium on Eye tracking research & applications. San Diego, California, ACM.
- Baum, L. W. and T. Petrie (1966). "Statistical Inference for Probabilistic Functions of Finite State Markov Chains " The Annals of Mathematical Statistics **37**(6): 1554-1563.
- Bilmes, J. (2006). "What HMMs Can Do." Institute of Electronics, Information and Communication Engineers - Transactions on Information Systems **E89-D**(3): 869-891.
- Bishop, C. M. (2006). Pattern recognition and machine learning. New York, Springer.
- Bousquet, O., S. p. Boucheron and G. b. Lugosi (2004). Introduction to Statistical Learning Theory. Advanced Lectures on Machine Learning: 169-207.
- Boussemart, Y. and M. L. Cummings (2008). Behavioral Recognition and Prediction of an Operator Supervising Multiple Heterogeneous Unmanned Vehicles. Humans Operating Unmanned Systems, HUMOUS'08, Brest, France.

- Boussemart, Y. and M. L. Cummings (2010). "Predicting Supervisory Control Behavior with Hidden Markov Models and Eye Tracking Data." User Modeling and User Adapter Interactions (**under review**).
- Boussemart, Y. and M. L. Cummings (2010). "Predictive models of human supervisory control behavior using hidden semi-Markov models." Engineering Applications of Artificial Intelligence (**under review**).
- Boussemart, Y., B. Donmez, M. L. Cummings and J. Las Fargeas (2009). Effects of Time Pressure on the Use of an Automated Decision Support System for Strike Planning. 15th International Symposium on Aviation Psychology (ISAP'09). Dayton, Ohio.
- Boussemart, Y., J. L. Fargeas, M. L. Cummings and N. Roy (2010). "Comparing Learning Techniques for Hidden Markov Models of Human Supervisory Control Behavior." AIAA Journal of Aerospace Computing, Information, and Communication (JACIC) (**Accepted**).
- Bowers, C. A., R. L. Oser, E. Salas and J. A. Cannon-Bowers (1996). Team Performance in Automated Systems. Automation and Human Performance, Theory and Applications. R. Parasuraman and M. Mouloua. Mahwah, New Jersey, Lawrence Erlbaum Associates: 243-263.
- Bowers, C. A., E. Salas and F. Jentsch (2006). Creating high-tech teams : practical guidance on work performance and technology. Washington, DC, American Psychological Association.
- Brand, M., N. Oliver and A. Pentland (1996). Coupled hidden markov models for complex action recognition. IEEE Computer Vision and Pattern Recognition.
- Brewer, N. and T. Ridgway (1998). "Effects of Supervisory Monitoring on Productivity and Quality of Performance " Journal of Experimental Psychology: Applied **4**(3): 211-227.
- Brewer, N., C. Wilson and K. Beck (1994). "Supervisory behavior and team performance amongst police patrol sergeants." Journal of Occupational and Organizational Psychology **67**: 69-78.
- Broadbent, D. E. (1958). Perception and Communication. Oxford, Pergamon.
- Bruni, S., Y. Boussemart, M. L. Cummings and S. Haro (2007). Visualizing Cognitive Strategies in Time-Critical Mission Replanning. HSIS 2007: ASNE Human Systems Integration Symposium, Annapolis, MD, USA.
- Bruni, S. and M. Cummings (2005). Human Interaction with Mission Planning Search Algorithms. Human Systems Integration Symposium, Arlington, VA.
- Bruni, S. and M. L. Cummings (2006). Tracking Resource Allocation Cognitive Strategies for Strike Planning. COGIS 2006 - Cognitive Systems with Interactive Sensors, Paris.
- Burges, C. (1998). "A Tutorial on Support Vector Machines for Pattern Recognition." Data Mining and Knowledge Discovery **2**: 121-167.
- Burnham, K. P. and D. R. Anderson (2002). Model Selection and Multimodel Inference, a Practical Information Theoretic Approach. New York, Springer.
- Burnham, K. P. and D. R. Anderson (2004). "Multimodel Inference: Understanding AIC and BIC in Model Selection " Sociological Methods & Research **33**(2): 161-304.
- Burns, J. M. (1978). Leadership. New York, Harper&Row.

- Byrne, M. D. and E. M. Davis (2006). "Task Structure and Postcompletion Error in the Execution of a Routine Procedure." Human Factors **48**: 627-638.
- Carroll, J. B. (1993). Human cognitive abilities : a survey of factor-analytic studies. Cambridge ; New York, Cambridge University Press.
- Castonia, R. (2010). The Design and Evaluation of a HSMM-based Operator State Monitoring Display. Department of Aeronautics and Astronautics. Cambridge Massachusetts Institute of Technology. **S.M. Thesis**.
- Chien, J.-T. and S. Furui (2003). Predictive Hidden Markov Model Selection for Decision Tree State Tying Eurospeech 2003, Geneva.
- Cooke, N., M. Russell and A. Meyer (2004). Evaluation of hidden Markov models robustness in uncovering focus of visual attention from noisy eye-tracker data. Proceedings of the 2004 symposium on Eye tracking research & applications. San Antonio, Texas, ACM.
- Cooke, N. J. and J. C. Gorman (2006). Assessment of Team Cognition. 2nd Edition - International Encyclopedia of Ergonomics and Human Factors. P. Karwowski, Taylor & Francis LTD: 270-275.
- Csiszár, I. and P. C. Shields (2000). "The consistency of the BIC Markov order estimator." The Annals of Statistics **28**(6): 1601-1619.
- Cummings, M. L., S. Bruni, S. Mercier and P. J. Mitchell (2007). "Automation architecture for single operator, multiple UAV command and control." The international Command and Control Journal **1**(2): 1-24.
- Cummings, M. L., S. Bruni and P. J. Mitchell (2010). Human Supervisory Control Challenges in Network Centric Operations. Reviews of Human Factors and Ergonomics. D. H. Harris. Santa Monica, CA, Human Factors and Ergonomics Society. **6**: 34-78.
- Cyc. (2008). "Graphic showing the maximum separating hyperplane and the margin." Retrieved 10/10/2010, from [http://en.wikipedia.org/wiki/File:Svm\\_max\\_sep\\_hyperplane\\_with\\_margin.png](http://en.wikipedia.org/wiki/File:Svm_max_sep_hyperplane_with_margin.png).
- de Brito, G. (2002). "Towards a model for the study of written procedure following in dynamic environments." Reliability Engineering & System Safety **75**(2): 233-244.
- de Winter, J. C. F., P. A. Wieringa, J. Kuipers, J. A. Mulder and M. Mulder (2007). "Violations and errors during simulation-based driver training." Ergonomics **50**(1): 138-158.
- Degani, A. and E. L. Wiener (1997). "Procedures in complex systems: the airline cockpit." Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on **27**(3): 302-312.
- Dekker, S. (2003). "Failure to adapt or adaptations that fail: contrasting models on procedures and safety." Applied Ergonomics **34**(3): 233-238.
- Dixon, S. R. and C. D. Wickens (2003). Control of Multiple-UAVs: A Workload Analysis. 12th International Symposium on Aviation Psychology, Dayton, OH.
- DoD (2007). Unmanned Systems Roadmap (2007-2032). Washington, D.C., Office of the Secretary of Defense.

DoD (2007). Unmanned Systems Roadmap (2007-2032), Office of the Secretary of Defense, Washington D.C.

Drucker, H., Chris, B. Kaufman, A. Smola and V. Vapnik (1997). Support vector regression machines. Advances in Neural Information Processing Systems 9 NIPS 1996, Vancouver, BC, MIT Press.

Duchowsky, A. T. (2002). "A breadth-first survey of eye tracking applications." Behavior Research Methods, Instruments, and Computers (BRMIC) **34**(4): 455-470.

Dunbar, M. and L. McDonnell (2001). "Aircrews and automation bias: the advantage of teamwork?" The International Journal of Aviation Psychology **11**(1): 1-14.

Eads, D., K. Glocer, S. Perkins and J. Theiler (2005). Grammar-guided feature extraction for time series classification. Neural Information Processing Systems (NIPS) '05. Vancouver, BC.

Endsley, M. (1987). The application of human factors to the development of expert systems for advanced cockpits. Human Factors Society 31st Annual Meeting, Santa Monica, CA.

Falkhausen, M., H. Reininger and D. Wolf (1995). Calculation Of Distance Measures Between Hidden Markov Models. Eurospeech '95, Madrid, Spain.

Forney, G. D. (1973). "The Viterbi Algorithm." Proceedings of the IEEE **61**(3): 268-278.

Furuta, K., K. Sasou, R. Kubota, H. Ujita, Y. Shuto and E. Yagi (2000). "Human Factor Analysis of JCO Criticality Accident." Cognition, Technology & Work **2**(4): 182-203.

Gader, P. D., M. Mohamed and C. Jung-Hsien (1997). "Handwritten word recognition with character and inter-character neural networks." Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on **27**(1): 158-164.

Gardenier, J. S. (1981). Ship navigational failure detection and diagnosis. Human Detection and Diagnosis of System Failure. J. Rasmussen and W. B. Rouse. Boston, Plenum. **49-74**.

Gersick, C. J. G. and J. R. Hackman (1990). "Habitual Routines in Task-Performing Groups." Organizational Behavior and Human Decision Processes **47**: 65-97.

Ghosh, S. and D. L. Reilly (1994). Credit Card Fraud Detection with a Neural-Network 27th Annual Hawaii International Conference on System Science, Hawaii, IEEE Computer Society Press, Los Alamitos, CA.

Goldberg, J. H. and X. P. Kotval (1999). "Computer interface evaluation using eye movements: methods and constructs - Its psychological foundation and relevance to display design." International Journal of Industrial Ergonomics **24**: 631-645.

Gorman, J. C., N. J. Cooke, H. K. Pedersen, O. O. Connor and J. A. DeJoode (2005). Coordinated Awareness of Situation by Teams (CAST): Measuring Situation Awareness of a Communication Glitch. Human Factors and Ergonomics Society.

Gray, W. D., B. E. John and M. E. Atwood (1992). The Precip of Project Ernestine or an overview of a validation of GOMS. SIGCHI Conference on Human Factors in Computing Systems '92. New York, NY, USA, ACM Press: 307-312.

- Griffiths, T. L., C. Kemp and J. B. Tenenbaum (2008). Bayesian models of cognition. Cambridge Handbook of Computational Cognitive Modeling. R. Sun, Cambridge University Press.
- Guedon, Y. (2003). "Estimating Hidden Semi-Markov Chains From Discrete Sequences " Journal of Computational & Graphical Statistics **12**(3): 604-639.
- Gutwin, C. and S. Greenberg (2004). The Importance of Awareness for Team Cognition in Distributed Collaboration. Team Cognition: Understanding the Factors That Drive Process and Performance. E. Salas and S. M. Fiore. Washington, D.C., American Psychological Association (APA): pp. 177-201.
- Hackman, J. R. (2002). Leading teams : setting the stage for great performances. Boston, Mass., Harvard Business School Press.
- Hamilton, J. D. (1994). Time series analysis. Princeton, N.J., Princeton University Press.
- Hayashi, M. (2003). Hidden Markov Models to Identify Pilot Instrument Scanning and Attention Patterns. IEEE International Conference on Man and Cybernetics.
- Hayashi, M., B. Beutter and R. S. McCann (2005). Hidden Markov Model analysis for space shuttle crewmember's scanning behavior. IEEE International Conference on Systems, Man and Cybernetics. Waikoloa, Hawaii: 1615-1622.
- Hembrooke, H., M. Feusner and G. Gay (2006). Averaging scan patterns and what they can tell us. Proceedings of the 2006 symposium on Eye tracking research & applications. San Diego, California, ACM.
- Hernando, D., V. Crespi and G. Cybenko (2005). "Efficient computation of the hidden Markov model entropy for a given observation sequence." Information Theory, IEEE Transactions on **51**(7): 2681-2685.
- Hiroshi, M. (1997). Supervised learning of hidden Markov models for sequence discrimination. Proceedings of the first annual international conference on Computational molecular biology. Santa Fe, New Mexico, United States, ACM.
- Hoey, J. (2007). "Value-Directed Human Behavior Analysis from Video Using Partially Observable Markov Decision Processes." IEEE Transactions on Pattern Analysis and Machine Intelligence **29**(7): 1118-1132.
- Hornof, A. J. and T. Halverson (2003). Cognitive strategies and eye movements for searching hierarchical computer displays. Proceedings of the SIGCHI conference on Human factors in computing systems. Ft. Lauderdale, Florida, USA, ACM.
- Huang, H. (2009). Developing an Abstraction Layer for the Visualization of HSMM-Based Predictive Decision Support Electrical Engineering and Computer Science. Cambridge Massachusetts Institute of Technology. **Masters: 118**.
- Irving, R. H., C. A. Higgins and F. R. Safayeni (1986). "Computerized performance monitoring systems: use and abuse." Commun. ACM **29**(8): 794-801.
- Janis, I. (1972). (1972) Victims of Groupthink. Houghton, Mifflin Company.

- Jennings, G. (2010). "Predator-series UAVs surpass one million flight hours." Retrieved Sept. 9th, 2010, from [http://www.janes.com/news/defence/jdw/jdw100409\\_1\\_n.shtml](http://www.janes.com/news/defence/jdw/jdw100409_1_n.shtml).
- Jentsch, F., J. Barnett, C. A. Bowers and E. Salas (1999). "Who is flying this plane anyway? what mishaps tell us about crew member rol assignment and air crew situation awareness." Human Factors **41**(1): 1-14.
- Joshi, A., S. P. Miller and M. P. E. Heimdahl (2003). Mode Confusion Analysis of a Flight Guidance System using Formal Methods. 22nd IEEE Digital Avionics Systems Conference (DASC'2003). Indianapolis, IN.
- Kahneman, D. and A. Tversky (1979). "Prospect Theory: An Analysis of Decision under Risk." Econometrica **47**(2): 263-292.
- Kanse, L., T. W. Van Der Schaaf, N. D. Vrijland and H. Van Mierlo (2006). "Error recovery in a hospital pharmacy." Ergonomics **49**(5): 503 - 516.
- Kirwan, B. and L. K. Ainsworth (1992). A Guide to Task Analysis: The Task Analysis Working Group. Bristol, PA, Taylor & Francis.
- Klein, G. (1999). Sources of Power: How People Make Decisions. Cambridge, MA, The MIT Press.
- Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. Proceedings of the 14th international joint conference on Artificial intelligence - Volume 2. Montreal, Quebec, Canada, Morgan Kaufmann Publishers Inc.
- Kohonen, T. (1982). "Self-organized formation of topologically correct feature maps." Biological Cybernetics **43**: 59-69.
- Kriouile, A., J. F. Mari and J. P. Haon (1990). Some improvements in speech recognition algorithms based on HMM. Acoustics, Speech, and Signal Processing, 1990. ICASSP-90., 1990 International Conference on.
- Lee, J. D. and N. Moray (1992). "Trust, control strategies and allocation of function in human-machine systems." Ergonomics **35**(10): 1243-1270.
- Lee, J. D. and K. A. See (2003). "Trust in Automation: Designing for Appropriate Reliance." Human Factors **46**(1): 50-80.
- Lees, M. N. and J. D. Lee (2007). "The influence of distraction and driving context on driver response to imperfect collision warning systems." Ergonomics **50**(8): 1264-1286.
- Leveson, N. G. (2003). A new accident model for engineering safer systems. Working Paper Series - Internal Symposium. Cambridge, MA, Engineering Systems Division, Massachusetts Institute of Technology.
- Li, C. and G. Biswas (1999). Finding Behavior Patterns from Temporal Data Using Hidden Markov Model based on Unsupervised Classification. International ICSC Symposium on Advances in Intelligent Data Analysis (AIDA'99), Rochester, N.Y., USA.

- Liang, Y., M. L. Reyes and J. D. Lee (2007). "Real-Time Detection of Driver Cognitive Distraction Using Support Vector Machines." Intelligent Transportation Systems, IEEE Transactions on **8**(2): 340-350.
- Lyon , D. R., G. Gunzelmann and K. A. Gluck (2004). Emulating a visuospatial memory field using ACT-R. 6th International Conference of Cognitive Modeling.
- Marin, J.-M., K. Mengersen, C. P. Robert, D. K. Dey and C. R. Rao (2005). Bayesian Modelling and Inference on Mixtures of Distributions. Handbook of Statistics. North-Holland, Amsterdam, Elsevier. **25**: 459-507.
- Marin, M., K. Mengerson and C. P. Robert (2005). Bayesian Modelling and Inference on Mixtures of Distributions. Handbook of Statistics. D. Dey and C. R. Rao. North-Holland, Amsterdam, Elsevier. **25**: 15840-15845.
- Marsden, P. (1996). Procedures in the Nuclear Industry. Human Factors in Nuclear Safety. N. Stanton. Bristol, PA, Taylor & Francis: 99-117.
- Marsden, P. and M. Green (1996). "Optimising procedures in manufacturing systems." International Journal of Industrial Ergonomics **17**(1): 43-51.
- McCane, B. and T. Caelli (2004). "Diagnostic tools for evaluating and updating hidden Markov models." Pattern Recognition **37**(7): 1325-1337.
- McCarley, J. S. and C. D. Wickens (2005). Human Factors Implications of UAVs in the National Airspace. Savoy, IL, University of Illinois.
- McCulloch, W. S. and W. Pitts (1943). " A logical calculus of the ideas immanent in nervous activity " Bulletin of Mathematical Biology **5**: 115-133.
- McDonald, N. and V. Hrymak (2002). Safety Behaviour in the Construction Sector. Dublin, Ireland, Health and Safety Authority.
- Mekdeci, B. and M. L. Cummings (2009). Modeling Multiple Human Operators in the Supervisory Control of Heterogeneous Unmanned Vehicles. 9th Conference on Performance Metrics for Intelligent Systems. Gaithersburg, MD: 7.
- Miller, R. A. (1994). "Medical diagnostic decision support systems--past, present, and future: a threaded bibliography and brief commentary." Journal of the American Medical Informatics Association **1**(1): 8-27.
- Minsky, M. (1954). Neural Nets and the Brain Model Problem, Princeton. **Ph.D. Thesis**.
- Mitchell, C., M. Harper and L. Jamieson (1999). "On the complexity of explicit duration HMMs." Speech and Audio Processing, IEEE Transactions on **3**(3): 213-217.
- Mitchell, P. J. and M. L. Cummings (2005). Management of Multiple Dynamic Human Supervisory Control Tasks. The 10th International Command and Control Research and Technology Symposium (ICCRTS), McLean, VA.
- Moodi, M. and R. C. Graeber (1998). Understanding flight crew adherence to procedures: the Procedural Event Analysis Tool (PEAT). Flight Safety Foundation, IFA/IASS. South Africa.

- Mosier, K. L., L. J. Skitka, M. Dunbar and L. McDonnell (2001). "Aircrews and Automation Bias: The Advantages of Teamwork." International Journal of Aviation Psychology **11**(1): 1-14.
- Muir, B. M. (1994). "Trust in automation. Part I. Theoretical issues in the study of trust and human intervention." Ergonomics **37**(11): 1905-1922.
- Mukkamala, S., G. Janoski and A. Sung (2002). Intrusion detection using neural networks and support vector machines. Neural Networks, 2002. IJCNN '02. Proceedings of the 2002 International Joint Conference on.
- Nakamichi, N., K. Shima, M. Sakai and K.-i. Matsumoto (2006). Detecting low usability web pages using quantitative data of users' behavior. Proceeding of the 28th international conference on Software engineering. Shanghai, China, ACM.
- National Transportation Safety Board (1994). A review of flightcrew-involved major accidents of U.S. air carriers 1978-1990. Washington, DC, NTSB.
- Nehme, C. (2009). Modeling Human Supervisory Control in Heterogeneous Unmanned Vehicle System. Dept of Aeronautics and Astronautics. Cambridge, MA, MIT. **Ph.D. Thesis**.
- Nehme, C. E., J. Crandall and M. L. Cummings (2008). Using Discrete-Event Simulation to Model Situational Awareness of Unmanned-Vehicle Operators. 2008 Capstone Conference. Norfolk, VA.
- Nehme, C. E., J. W. Crandall and M. L. Cummings (2007). An Operator Function Taxonomy for Unmanned Aerial Vehicle Missions. International Command and Control Research and Technology Symposium, Newport, Rhode Island.
- Ng, A. and M. Jordan. (2001). "On Discriminative vs. Generative classifiers: A comparison of logistic regression and naive Bayes." from citeulike-article-id:3329371  
<http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.19.9829>.
- Niissalo, A. (2010). "Principles of Support Vector Machines (SVM)." Retrieved 10/10, 2010, from <http://www.imtech.res.in/raghava/rbpred/svm.jpg>.
- NTSB (1973). Aircraft Accident Report, Eastern Airlines Inc, L-1011 N310EA, Miami FL, December 19, 1972. Washington, DC, NTSB.
- Ockerman, J. J. and A. R. Pritchett (2004). "Improving performance on procedural tasks through presentation of locational procedure context: an empirical evaluation." Behaviour and Information Technology **23**: 11-20.
- Ollero, A. and I. Maza (2007). Multiple Heterogeneous Unmanned Aerial Vehicles. Berlin Heidelberg, Springer.
- Park, J., W. Jung, J. Ha and C. Park (2002). "The step complexity measure for emergency operating procedures: measure verification." Reliability Engineering & System Safety **77**(1): 45-59.
- Patrick, J., N. James and A. Ahmed (2006). "Human processes of control: tracing the goals and strategies of control room teams." Ergonomics **49**(12/13): 1395-1414.
- Pentland, A. (2008). Honest signals : how they shape our world. Cambridge, Mass., MIT Press.

- Pentland, A. and A. Liu (1995). Towards Augmented Control Systems. IEEE Intelligent Vehicles '95, Detroit, MI.
- Pentland, A. and A. Liu (1999). "Modeling and prediction of human behavior." Neural Computations **11**(1): 229-242.
- Perrow, C. (1984). Normal Accidents: Living with High-Risk Technologies. Princeton, N.J, Princeton University Press.
- Pinelle, D., C. Gutwin and S. Greenberg (2003). "Task Analysis for Groupware Usability Evaluation: Modeling Shared Workspace Tasks with the Mechanics of Collaboration." ACM Transactions on Computer-Human Interaction **10**(4): pp. 281-311.
- Pomerleau, D. (1993). Knowledge-based Training of Artificial Neural Networks for Autonomous Robot Driving. Robot Learning. J. Connell and S. Mahadevan.
- Poole, A. and J. B. Linden (2005). Eye Tracking in Human-Computer Interaction and Usability Research: Current Status and Future. Encyclopedia of HCI. C. Chaoui. Pennsylvania, Idea Group, Inc.
- Pressing, J. and G. Jolley-Rogers (1997). "Spectral properties of human cognition and skill." Biological Cybernetics **76**(5): 339-347.
- Pretorius, M. C., A. P. Calitz and D. v. Greunen (2005). The added value of eye tracking in the usability evaluation of a network management tool. Proceedings of the 2005 annual research conference of the South African institute of computer scientists and information technologists on IT research in developing countries. White River, South Africa, South African Institute for Computer Scientists and Information Technologists.
- Qian, H., Y. Ou, X. Wu, X. Meng and Y. Xu (2010). "Support Vector Machine for Behavior-Based Driver Identification System." Journal of Robotics **2010**: 11.
- Rabiner, L. and B. Juang (1986). "An introduction to hidden Markov models." ASSP Magazine, IEEE [see also IEEE Signal Processing Magazine] **3**(1): 4-16.
- Rabiner, L. R. (1989). "A tutorial on Hidden Markov Models and selected applications in speech recognition." Proceedings of the IEEE **77**(2): 257-286.
- Rajashekar, U., L. K. Cormack and A. C. Bovik (2002). Visual search: structure from noise. Proceedings of the 2002 symposium on Eye tracking research & applications. New Orleans, Louisiana, ACM.
- Ravich, T. (2009). "The integration of unmanned aerial vehicles into the national airspace." North Dakota Law Review **85**: 597-622.
- Reiser, M. and Y. Lin (1999). "A Goodness-of-Fit Test for the Latent Class Model When Expected Frequencies are Small." Sociological Methodology **29**: 81-111.
- Riley, V. (1996). Operator Reliance on Automation Data: Theory and Data. Automation and Human Performance: Theory and Applications. Mahwah, NJ, Lawrence Erlbaum Associates: 91-115.
- Rogovin, M. (1979). Three Mile Island: a report to the commissioners and the public. Washington, DC, Nuclear Regulatory Commission.

- Roth, E. M., R. J. Mumaw and P. M. Lewis (1994). An empirical investigation of operator performance in cognitively demanding simulated emergencies. Other Information: PBD: Jul 1994: Medium: ED; Size: 127 p.
- Salas, E., T. L. Dickinson, S. A. Converse and S. I. Tannenbaum (1992). Toward an Understanding of Team Performance and Training. Teams: Their Training and Performance. R. W. Swezey and E. Salas. Norwood, NJ, Albex: 3-29.
- Salvucci, D. D. (2005). "A multitasking general executive for compound continuous tasks." Cognitive Science **29**: 457-492.
- Salvucci, D. D. and J. H. Goldberg (2000). Identifying fixations and saccades in eye-tracking protocols. Proceedings of the 2000 symposium on Eye tracking research & applications. Palm Beach Gardens, Florida, United States, ACM.
- Sanches, I. (2000). "Noise-compensated hidden Markov models." Speech and Audio Processing, IEEE Transactions on **8**(5): 533-540.
- Santella, A. and D. DeCarlo (2004). Robust clustering of eye movement recordings for quantification of visual interest. Proceedings of the 2004 symposium on Eye tracking research & applications. San Antonio, Texas, ACM.
- Schnipke, S. K. and M. W. Todd (2000). Trials and tribulations of using an eye-tracking system. CHI '00 extended abstracts on Human factors in computing systems. The Hague, The Netherlands, ACM.
- Schraagen, J. M., S. Chipman and V. E. Shalin (2000). Cognitive Task Analysis. Mahwah, NJ, Erlbaum.
- Scott, S. D., A. E. Rico, C. Y. Furusho and M. L. Cummings (2007). Aiding Team Supervision in Command and Control Operations with Large-Screen Displays. HSIS: ASNE Human Systems Integration Symposium, Annapolis, MD, USA.
- Shannon, C. E. (1948). "A Mathematical Theory of Communication." The Bell System Technical Journal **27**: 379-423.
- Sheridan, T. B. (1992). Telerobotics, Automation and Human Supervisory Control. Cambridge, MA, The MIT Press.
- Shinners, S. M. (1974). "Modeling of Human Operator Performance Utilizing Time Series Analysis." Systems, Man and Cybernetics, IEEE Transactions on **4**(5): 446-458.
- Sibert, L. E. and R. J. K. Jacob (2000). Evaluation of eye gaze interaction. Proceedings of the SIGCHI conference on Human factors in computing systems. The Hague, The Netherlands, ACM.
- Simola, J., J. Salojärvi and I. Kojo (2008). "Using hidden Markov model to uncover processing states from eye movements in information search tasks." Cognitive Systems Research **9**(4): 237-251.
- Singh, S., H. Tu, W. Donat, K. Pattipati and P. Willet (1996). "Anomaly Detection via Feature-Aided Tracking and Hidden Markov Models." IEEE Transactions on Systems, Man, and Cybernetics.
- Stanton, N. A. and C. Baber (2008). "Modelling of human alarm handling response times: a case study of the Ladbroke Grove rail accident in the UK." Ergonomics **51**(4): 423-440.

- Sulis, W. H. and A. Combs (1996). Nonlinear Dynamics in Human Behaviour, World Scientific Pub Co Inc.
- Swezey, R. W. and E. Salas (1992). Teams : their training and performance. Norwood, N.J., Ablex Pub. Corp.
- Terran, L. (1999). Hidden Markov Models for Human Computer Interface Modeling International Joint Conferences on Artificial Intelligence, Workshop on Learning About Users, Stockholm, Sweden.
- Thede, S. M. and M. P. Harper (1999). A second-order Hidden Markov Model for part-of-speech tagging. Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics. College Park, Maryland, Association for Computational Linguistics.
- Trager, E. A. (1988). Special study report: Significant events involving procedures. Washington, DC, Office for Analysis and Evaluation of Operational Data, Nuclear Regulatory Commission.
- Tsay, R. S. (2005). Analysis of financial time series. Hoboken, N.J., Wiley.
- Tsochantaridis, I., T. Joachims, T. Hofmann and Y. Altun (2005). "Large Margin Methods for Structured and Interdependent Output Variables." J. Mach. Learn. Res. **6**: 1453-1484.
- Tversky, A. and D. Kahneman (1974). "Judgment under Uncertainty: Heuristics and Biases." Science **185**(4157): 1124-1131.
- Tversky, A. and D. Kahneman (1981). "The framing of decisions and the psychology of choice." Science(211): 453-458.
- van Gompel, R. P. G., M. H. Fischer, W. S. Murray and R. L. Hill, Eds. (2007). Eye Movements: A Window on Mind and Brain Elsevier Science.
- Vapnik, V. N. (2000). The nature of statistical learning theory. New York, Springer.
- Varga, A. P. and R. K. Moore (1990). Hidden Markov model decomposition of speech and noise. Acoustics, Speech, and Signal Processing, 1990. ICASSP-90., 1990 International Conference on.
- Wagenmakers, E.-J., S. Farrell and R. Ratcliff (2005). "Human Cognition and a Pile of Sand: A Discussion on Serial Correlations and Self-Organized Criticality." Journal of Experimental Psychology: General **134**(1): 108-116.
- Watson, B. and A. Chunk Tsoi (1992). Second-order Hidden Markov Models for speech recognition. 4th Australian International Conference on Speech Science and Technology, Brisbane, Australia.
- Wayne, D. G., E. J. Bonnie and E. A. Michael (1992). The precis of Project Ernestine or an overview of a validation of GOMS. SIGCHI Conference on Human Factors in Computing Systems. Monterey, California, United States, ACM.
- Weber, E. U. and O. Coskunoglu (1990). "Descriptive and prescriptive models of decision-making: implications for the development of decision aids." Systems, Man and Cybernetics, IEEE Transactions on **20**(2): 310-317.
- Weibel, R. E. and J. R. Hansman (2005). Safety Considerations for Operation of Unmanned Aerial Vehicles in the National Airspace System. ICAT Reports. Cambridge, MA, MIT.

Weick, K. E. (1990). "The vulnerable system: an analysis of the Tenerife air disaster." Journal of Management **16**(3): 571-593.

Welford, A. T. (1952). "The psychological refractory period and the timing of high-speed performance - a review and a theory." British Journal of Psychology **43**: 2-19.

Wooding, D. S. (2002). Fixation maps: quantifying eye-movement traces. Proceedings of the 2002 symposium on Eye tracking research & applications. New Orleans, Louisiana, ACM.

Xiangwei, K., Z. Kun and D. Naiyang (2008). Unsupervised Support Vector Machines with Perturbations. Second International Symposium on Intelligent Information Technology Application, 2008. IITA '08. .

Xiao, Y. and C. F. Mackenzie (1995). "Decision Making in Dynamic Environments: Fixation Errors and their Causes." Human Factors and Ergonomics Society Annual Meeting Proceedings **39**: 469-473.

Yeung, D., Z.-Q. Liu, X.-Z. Wang, H. Yan, Z. Zhang, F. Vanderhaegen and P. Millot (2006). Prediction of Human Behaviour Using Artificial Neural Networks. Advances in Machine Learning and Cybernetics, Springer Berlin / Heidelberg. **3930**: 770-779.

Yogameena, B., E. Komagal, M. Archana and S. R. Abhaikumar (2010). "Support Vector Machine-Based Human Behavior Classification in Crowd through Projection and Star Skeletonization." Journal of Computer Science **6**(9): 1008-1013.

