

QUALITATIVE RELATIONAL MAPPING AND
NAVIGATION FOR LONG-TERM ROBOTIC
OPERATION

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Mark J. McClelland

May 2014

© 2014 Mark J. McClelland
ALL RIGHTS RESERVED

QUALITATIVE RELATIONAL MAPPING AND NAVIGATION FOR LONG-TERM ROBOTIC OPERATION

Mark J. McClelland, Ph.D.

Cornell University 2014

The research presented in this work focuses on several aspects of the remote operation of ground vehicles, notably Navigation and Mapping for autonomous robots and the effects of time delay in tele-operated vehicles.

Navigation and mapping of large, unstructured spaces is achieved by accumulating constraints on the geometrical relationships between landmarks. These relationships are tracked using two qualitative representations of space, one based on qualitative angles between landmark triples, and a second which also considers qualitative edge lengths. For the first representation, measurements and graph inference are performed by way of manually computed lookup tables based on geometrical constraints on qualitative states. For the second representation, measurements are generated online using a branch-and-bound algorithm to solve a set of nonlinear feasibility problems, while lookup tables for inference are generated using a similar, offline approach. Estimates of the Relative Neighborhood Graph are extracted from the qualitative map and used to perform long-distance navigation.

The effects of human control of remote vehicles are considered, focusing on the question of how operators are able to compensate for time delays when tele-operating vehicles in continuous motion. Statistical models fit to experimental data using the Least Angle Regression and Sparse Multinomial Regression algorithms show that human operators anticipate future control needs by predicting rover motion forward through time to determine predicted off-track errors. The

relative contributions of environmental features to model predictive power is used to determine how feature 'importance' varies as a function of time delay.

BIOGRAPHICAL SKETCH

Mark McClelland received his M.S. in Mechanical Engineering from Cornell University in 2012, and his B.S. in Physics and B.A. in Classical Studies from the University of Washington in 2006. He received an Honorable Mention from the NSF Graduate Fellowship Research Program in 2010, and a 3-year fellowship from the NASA Graduate Research Program from 2010-2013. His research interests include probabilistic modeling of human-robotic systems, Bayesian approaches to machine learning, and autonomous robotic navigation and mapping.

ACKNOWLEDGEMENTS

The work presented in Chapters 1 and 2 was performed in collaboration with Dr. Mark Campbell and Dr. Tara Estlin, and was supported by National Science Foundation grant IIS-1320490 and a fellowship from the NASA Graduate Student Research Program. This work was performed by Cornell University and by the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration.

The work presented in Chapter 3 was performed in collaboration with Dr. Mark Campbell, and was funded by AFOSR Grant FA9550-09-1-0462 with Dr. Fariba Fahroo as Program Manager.

TABLE OF CONTENTS

Biographical Sketch	iii
Acknowledgements	iv
Table of Contents	v
List of Tables	vii
List of Figures	viii
Introduction	1
1 Qualitative Relational Mapping for Mobile Robots with Minimal Sensing	5
1.1 Introduction	6
1.2 Qualitative Geometric Relations	12
1.2.1 Freksa’s Double Cross	12
1.2.2 FDC Operators	15
1.3 Measuring Qualitative States	16
1.4 Building Qualitative Maps	22
1.4.1 Map Structure	22
1.4.2 Graph Updates	24
1.4.3 Data Association	26
1.5 Mapping Results	26
1.5.1 Map Evaluation Metrics	27
1.5.2 Simulation Results	27
1.5.3 Experimental Results	33
1.6 Conclusion	36
2 Qualitative Mapping and Navigation for Long-Term Exploration	38
2.1 Introduction	39
2.2 Qualitative Relational Geometry	44
2.3 EDC Operators	46
2.3.1 Unary Operators	46
2.3.2 Composition Operator	48
2.3.3 Operators Example	51
2.4 Measuring Qualitative States	51
2.5 Feasibility Detection	55
2.6 Building Qualitative Maps	59
2.6.1 Map Structure	59
2.6.2 Graph Updates	59
2.6.3 Data Association	61
2.7 Qualitative Navigation	62
2.7.1 The Relative Neighborhood Graph	63
2.7.2 Graph-based Navigation	65
2.8 Mapping Results	66

2.8.1	Map Evaluation Metrics	66
2.8.2	Monte Carlo Simulations	70
2.8.3	Data-Driven Simulation	73
2.9	Conclusion	78
3	Probabilistic Modeling of Anticipation in Human Controllers	81
3.1	Introduction	82
3.2	Human-Robotic Experiments With Time Delays	85
3.2.1	Hardware Setup	85
3.2.2	Reference Route	87
3.2.3	Experimental Process	88
3.2.4	Human Subjects	90
3.2.5	Experimental Results	90
3.3	Modeling Approach: Bayesian, with Feature Ordering	92
3.3.1	Feature Selection	93
3.3.2	Overview of Models and Algorithms	97
3.3.3	Model #1: LASSO with LARS	99
3.3.4	Model #2: SMLR	105
3.4	Model Fitting Results	108
3.4.1	Algorithm Fitting Performance	109
3.5	Model Feature Importance	116
3.5.1	Measuring Importance in LASSO and SMLR	116
3.5.2	Relative Importance of Feature Groups	119
3.6	Conclusion	124
	Conclusion	126
	A Sector Measurement Tables	131
	B Double Cross Operator Tables	135
B.1	Unary Operators	136
B.2	Binary Composition Operator	137
	C EDC Unary Operator Table Generation	139
	Bibliography	153

LIST OF TABLES

2.1	EDC Unary Transformations	48
2.2	EDC Composition Expressions	50
2.3	EDC Measurement Expressions	53
3.1	ANOVA Results	90

LIST OF FIGURES

1.1	Martian Crater	7
1.2	Freksa’s Double Cross	14
1.3	FDC Measurement Sectors	18
1.4	FDC Measurement Regions	21
1.5	Simulation Example Map	27
1.6	FDC Simulation Results	28
1.7	FDC Monte-Carlo Results	29
1.8	FIDO Rover	31
1.9	JPL Mars Yard Reconstruction	33
1.10	FDC Mars Yard Results	34
2.1	Example Martian Environment	40
2.2	Freksa’s Double Cross	45
2.3	Extended Double Cross	47
2.4	Measurement Geometry	53
2.5	Lune Example	64
2.6	Flow Field Examples	67
2.7	EDC Monte-Carlo Results	68
2.8	Mars Yard 3D Reconstruction	69
2.9	FIDO Rover	74
2.10	EDC Mars Yard Results	75
2.11	Relative Neighborhood Graph Examples	77
3.1	Time Delay Example View	87
3.2	Time Delay Experimental Route	88
3.3	Human Performance Results	92
3.4	Human-Robot System Diagram	94
3.5	Route Feature Definitions	98
3.6	LARS Trajectory Example	105
3.7	LASSO Performance	110
3.8	SMLR Performance	112
3.9	LASSO and SMLR Confusion Matrices	114
3.10	Feature Group Importance	120
3.11	Feature Importance vs Time Delay	123
A.1	Measurement Geometry for Case 1: $g = 2, A < B $	132
A.2	Measurement Geometry for Case 2: $g = 2, B < A $	133
A.3	Measurement Geometry for Case 3: $g = 4, A < B $	133
A.4	Measurement Geometry for Case 4: $g = 4, B < A $	134
B.1	Lookup Tables for FDC Unary Operators	136
B.2	Lookup Table for FDC Composition Operator	138

INTRODUCTION

The research presented in this work focuses on several aspects of the remote operation of ground vehicles, notably Navigation and Mapping for autonomous robots and the effects of time delay in teleoperated vehicles. Chapters 1 and 2 focus on the landmark-based mapping and navigation of large, unstructured spaces by autonomous and semi-autonomous robots in the absence of global position or heading information. They present methods for doing so in a qualitative fashion, in which geometrical relationships between landmarks are described in terms of discrete labels. The work in Chapter 3 focuses on the question of modeling the ability of Human operators to compensate for the time delays often incurred when controlling remote vehicles.

The work in Chapter 1, currently in review for the AIAA Journal on Aerospace Information Systems[1], presents a novel method for autonomous robotic navigation and mapping of large scale spaces with sparse landmarks and minimal sensing. The proposed algorithm constructs a graph-based map which encodes the relative location of landmarks in the environment. Uncertainty in these locations is captured by imposing qualitative constraints on the relationships between landmarks observed by the robot. These relationships are represented in terms of the relative geometrical layout of landmark triplets. A novel measurement method based on camera imagery is presented which extends previous work from the field of Qualitative Spatial Reasoning, namely Freksa's Double Cross(FDC). Measurements are fused into the map using a deterministic approach based on iterative graph updates. Algorithm performance is evaluated using Monte-Carlo simulations and data-driven simulation results are presented for a robot traversing the Jet Propulsion Laboratory Mars Yard.

The Qualitative Relational Mapping system is extended in Chapter 2, currently in preparation for a submission to the IEEE Transactions on Robotics, by a new

qualitative representation that adds a form of qualitative distance to the FDC. Reasoning with this geometrical representation is recast as a set of nonlinear feasibility problems. In this new context, measurements of qualitative states can be extracted from camera measurements by treating each state as a set of nonlinear constraints on the unknown ranges to observed landmarks. Allowable states are those for which a feasibility solution can be found. Similarly, the composition operator lookup tables necessary for inference can also be set up as the solutions to nonlinear constraints on landmark position. In both cases, the feasibility problems can be quickly solved by a Branch-and-Bound approach, taking advantage of the small dimensionality of the system. Given the new representation, estimates of an additional graph structure, the Relative Neighborhood Graph (RNG), are easily extracted from the map and used for a long-distance navigation strategy based on a sequence of landmark objectives based on proximity. Asymptotic performance of the mapping algorithm is again evaluated using Monte Carlo tests on randomly generated maps, and simulation results are presented for a robot traversing the Mars Yard.

Chapter 3 considers the effects of human control of remote vehicles, focusing on the question of how operators are able to compensate for time delays when tele-operating vehicles in continuous motion. This work, previously published in the IEEE Transactions on Systems, Man, and Cybernetics: Systems[2], presents a methodology for determining whether human operators anticipate future control needs in order to compensate for time delays when controlling remote vehicles. The approach utilizes techniques drawn from the machine learning community in order to learn statistical models of human decision making. Models are fit to an experimental data set generated by remote operations of a robot subjected to time delays between 0 – 2.5s, using the Least Angle Regression (LARS) and

Sparse Multinomial Logistic Regression (SMLR) algorithms. These algorithms make use of regularization to reduce the effects of over-parameterization due to redundant or noisy environmental features. Models learned by LARS achieve an average prediction rate between 81 – 98%, depending on time delay, while those learned by SMLR achieve average rates between 68–86%. A novel metric of feature ‘importance’ is used to evaluate the relative contributions of environmental features to model performance, motivated by the structure of the LARS algorithm. The degree to which human operators rely on anticipation is determined by examining how ‘importance’ scores for features representing different prediction horizons vary with increasing time delay.

CHAPTER 1
QUALITATIVE RELATIONAL MAPPING FOR MOBILE ROBOTS
WITH MINIMAL SENSING

1.1 Introduction

When available, absolute position sensors such as GPS systems can provide excellent estimates of the position and attitude necessary for long-term autonomous robotic operation. Unfortunately, such systems are unavailable for a number of interesting mission classes, including extra-planetary exploration, operation in GPS-denied regions, and operation of extremely small or low-cost robotic platforms.

In the absence of absolute position sensors, existing robot localization systems tend to either rely solely on local sensors of ego-motion (e.g. Inertial Measurement Units, wheel encoders) as in the current GESTALT system for the Mars Exploration Rovers (MER) discussed by Ali et al. [3], or incorporate measurements of the rover's relative position and orientation with respect to certain landmarks in the environment using vision or ranging sensors. The latter case may consist of triangulation from known reference positions as demonstrated by Kuipers and Levitt [4], or the construction of adaptive feature maps as in the Simultaneous Localization and Mapping (SLAM) framework [5]. These methods have definite strengths, including the ability to provide both global position and orientation estimates as well as accurate estimates of the uncertainty in the parameters. They can also provide global localization of environmental features and thus allow the accumulation of information for the assembly of stable maps necessary for long-distance planning. However, these approaches often face a number of limitations, including the need for high quality sensing to determine the exact distance to visible landmarks, a reliance on point estimates of landmarks, and the use of a large number of closely spaced landmarks in order to overcome uncertainty but which do not scale well computationally.

This chapter presents a method for online robotic mapping in the absence of global reference data using qualitative spatial reasoning in a graph structure. The

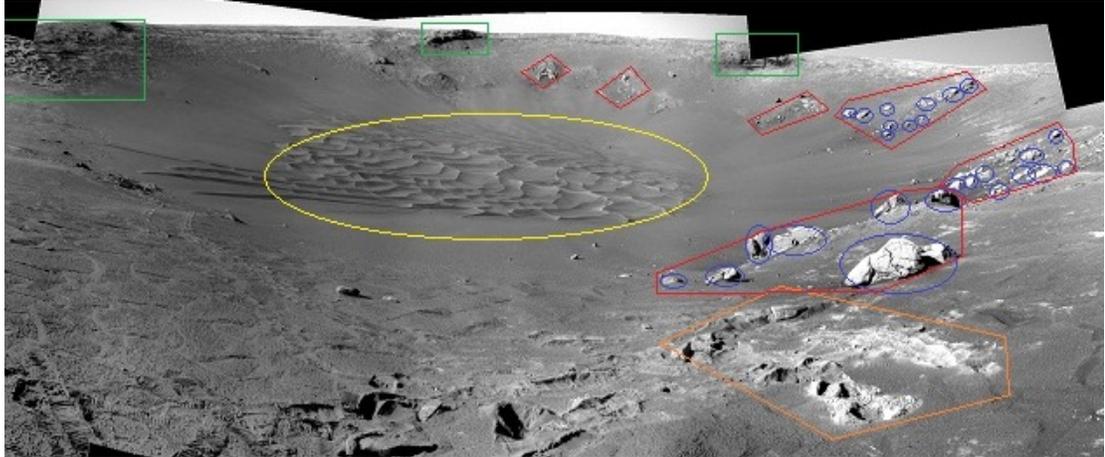


Figure 1.1: Example of objects and groups of objects comprising a crater region on Mars. Blue ellipses indicate distinctive rocks. Red outlines indicate groups of rocks. Green boxes indicate areas of exposed crater wall. The yellow ellipse denotes the distinctive sand pattern of the crater basin. The orange outline highlights an area of exposed outcrop. Image taken by MER Opportunity rover on sol 270

proposed algorithm is an extension of the work developed by the authors and presented in [6]. This process is called a ‘Qualitative Relational Map’ (QRM), in which the relative geometrical relationships between landmarks are tracked using qualitative information inferred from monocular camera images. The key novelties in this work are a greatly improved method for extracting measurements of qualitative states from single-camera images using an optimization approach, and the implementation of the mapping system in a realistic experimental scenario. The use case presented here to evaluate system performance is the exploration and mapping of a Mars-like environment, such as that depicted in Figure 1.1. Such environments are often characterized by the presence of visually distinctive objects, such as the rock clusters, outcrops, and crater walls outlined in Figure 1.1, embedded within large open spaces. Operation in the Martian environment is approximated using the Mars Yard at the Jet Propulsion Laboratory (JPL).

A key characteristic of the proposed QRM is the ability to extract and use as much information about the environment as possible from a minimal set of low-

cost sensors. The QRM implementation detailed in this work relies on a single camera with minimal quality requirements. However, the algorithm is applicable to any sensor system which can provide relative bearing to landmarks as well as an ordering by range. As with all monocular mapping strategies, the resulting map is unable to specify a global scale. Recent work on monocular navigation, in particular monocular-SLAM algorithms such as those discussed by Davison et al. [7], attempts to infer a scale from estimates of ego-motion. However, the qualitative geometries in the QRM operate in a naturally scale-free environment. Critically, algorithm performance does not depend on access to estimates of ego-motion between observations, a situation that is often the case in high-slip environments or with low-cost platforms. While the experimental results in Section 1.5 are shown with images used sequentially, the resulting maps would be the same for any choice of image order, thus making the QRM approach very general.

Previous work on qualitative mapping and navigation for ground robots includes the QUALNAV system described by Lawton [8], which relied on binary relationships inferred from the cyclical ordering of landmarks in a robots view. This representation decomposed the space around landmarks into regions defined by the rays passing through each landmark pair, as crossing those lines swaps landmark position in the view. Cyclical ordering was also used by Wallgrun [9] to learn the topologies of environments made up of hallway junctions, where junctions are labeled according to their qualitative cardinal orientation. The representation was extended by Schlieder [10] to include the directions opposite landmarks in order to eliminate map ambiguities and termed the ‘panorama’. The cyclical order is also revised to include extended objects with occlusions by Fogliaroni et al. [11], in which a topological map of visibility regions is induced by tangent lines from the extrema of convex polygonal obstacles. These regions are then learned either

from an known map of object shapes and locations, or by an exhaustive search of the space.

One major aim of the QRM system described in this paper is to decouple the robot position estimation problem from that of map building as much as possible. This is inspired by the insight that many robot tasks, such as navigation, do not require a fully defined metrical map. Use of qualitative relations between objects allows maps to remain useful in the presence of errors from uncertain ego-motion due to wheel slippage and rate-gyro drift. The proposed QRM encodes the relationships between observed landmarks explicitly in terms of geometrical statements, with qualitative states used to incorporate uncertainty in relative positions. In contrast, the graphical models used in many algorithms from the SLAM community follow the framework summarized by Durrant-Whyte and Bailey in [12] and [13], in which the probabilistic location of landmarks are measured relative to the uncertain robot state and the landmark-landmark relationships are implicitly encoded in a covariance matrix.

Metrical SLAM approaches work extremely well in structured spaces and areas with a high concentration of landmarks, but can have challenged operating in sparse environments. In such environments the reliance on motion-models increases, and errors such as model mismatch, wheel slip, and bias drifts can become a significant limitation to accurate localization. In particular, EKF-SLAM algorithms were proven to be inconsistent over time by Julier and Uhlmann [14], due to the need to linearize about the incorrect ego-motion estimates. As observed by Huang and Dissanayake [15], linearization induced filter inconsistencies are particularly problematic when using the bearing only measurements provided by monocular cameras. Several approaches have been taken in the literature to address the SLAM consistency problem, such as ego-frame approaches with linked

submaps, as described by Castellanos et al. [16]. However, such approaches rely on having a high density of nearby landmarks in order to generate meaningful submaps, and while they can dramatically improve the operation time of SLAM system, the filter will still diverge over long distances. Non-metrical solutions to this problem, such as the topological and place-based methods presented by Angeli et al. [17] and Cummins and Newman [18] respectively, have been more successful at eliminating map drift entirely. However, these solutions rely on environments having either topologies with limited connectivity, such as that seen indoors or in urban environments, or well-defined places identifiable by high feature densities. The qualitative approach detailed in this paper avoids the consistency problem entirely by extracting geometrical constraints on landmark relationships directly from camera measurements, rather than relying on estimated ego-motion. Consequently the system avoids both integration and linearization errors, but does so at the cost of maintaining scale free maps with large uncertainties in exact landmark positions, particularly at the edge of the map. In essence, this can be seen as a trade-off between map precision and map consistency.

Qualitative mapping bears some similarities with topological reasoning, although the underlying representations of map elements are different. Topological mapping places constraints on the geometrical arrangements of a map by specifying the connectivity of extended spaces as discussed by Randell et al. [19]. Both topological and topometric algorithms, such as those presented by Hoiem et al. [20] and Sibley et al. [21], have achieved great success in mapping indoor and urban environments, however they require that the environment be well structured. In large, open environments, the regions represented as nodes in a topological graph become poorly defined, as do the edges representing transitions between such regions. The proposed QRM algorithm takes a similar constraint-based approach,

but operates on the geometries of point sets rather than extended objects as it is aimed at operations in large unstructured spaces, where the definition of discrete areas is challenging or impossible. Just as there are an infinite number of spatial layouts that may satisfy any given topological specification, there are an infinite number of metrical arrangements of landmarks that have equivalent qualitative maps. However, the coordinate sets for all of these point arrangements are constrained to satisfy a set nonlinear inequalities implied by the qualitative statements in the map. Thus, one interpretation of the QRM algorithm is as a form of topological-style reasoning operating on topologically ambiguous spaces.

One approach of how to represent the ‘shape’ of a set of points has been that of statistical shape theory, which defines ‘shape’ to be what remains once scale, rotation, and translation effects have been removed via dimensional reduction. The approach discussed by Dryden and Mardia [22] and Mitteroecker and Gunz [23] uses a QR decomposition to transform a set of high dimensional points to the surface of a hypersphere in a scale, rotation, and translation invariant subspace. Continuous deformation of point sets corresponds to trajectories over the hypersphere, and a statistical similarity metric can be constructed by considering probability distributions over the hypersphere. The relationships encoded in the proposed QRM, although driven by different geometrical concerns, corresponds to defining nonlinear constraints on these point distributions. Landmark arrangements that have the same qualitative map will occupy a bounded, though non-convex, region of the hypersphere defined by the inequality constraints which correspond to the qualitative states encoded in the map edges. Critically, while statistical shape theory requires access to the true landmark locations in some reference frame in order to calculate the ‘shape’ of a point set, the QRM learns the constraints without attempting to estimate the locations themselves.

The remainder of this chapter is laid out as follows. Section 1.2 contains a background on the formalism used to define qualitative spatial relationships used in this work. Section 1.3 presents a novel method for generating measurements of qualitative states using camera images. Section 1.4 defines the map structure and summarizes the algorithm used to update it with new measurements. Section 1.5 presents results using a set of Monte-Carlo simulations to illuminate map properties and limitations as well as the results of testing the algorithm on a rover traversing the JPL Mars Yard.

1.2 Qualitative Geometric Relations

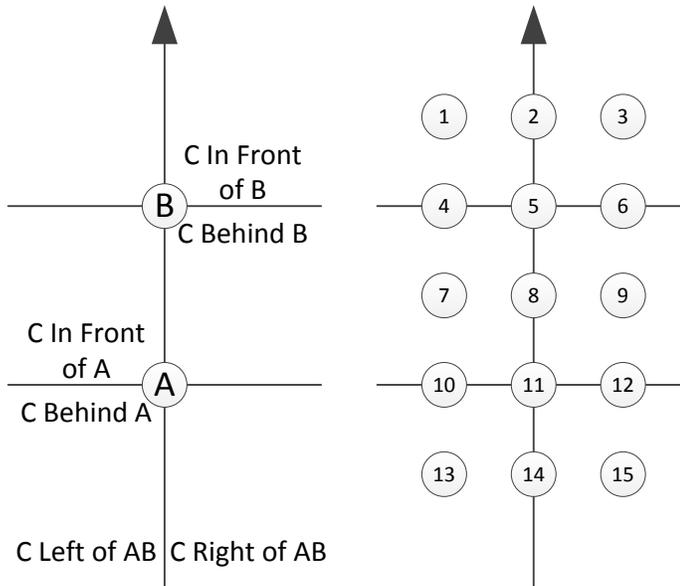
1.2.1 Freksa’s Double Cross

Qualitative statements of geometrical relationships require that the 2D plane around landmarks be segmented into discrete regions. The approach presented in this chapter makes use of a discretization based on triplets of landmarks, proposed in [24] and sometimes termed Freksa’s double cross (FDC). While it is computationally preferable to define relations only in terms of pairs of landmarks, this is impossible in 2D space without specifying an external reference frame, such as the cardinal directions or a fixed origin point.

The FDC representation is based on the observation that humans are generally good at determining if two points are 90° apart, but generally bad at determining finer spacings of angles[25]. An FDC specifies the geometric relationships of a query point C to \overrightarrow{AB} , the vector from point A to point B , by stating that C can be either to the left or right of \overrightarrow{AB} , in front or behind A relative to the direction of \overrightarrow{AB} , and in front or behind B relative to the direction of \overrightarrow{AB} . These three

statements are equivalent to defining the separating boundaries shown in Figure 1.2a. If the boundary lines are also included as possible regions, this definition results in the 15 possible geometrical relationships between C and \overrightarrow{AB} shown in Figure 1.2b. In this chapter the possibility that point C lies within region i of the FDC around \overrightarrow{AB} is denoted as $AB : C_i \in \{\text{TRUE}, \text{FALSE}\}$. If $AB : C_i = \text{TRUE}$, then the case of point C lying in FDC region i with respect to \overrightarrow{AB} , is consistent with the observed measurements; if $AB : C_i = \text{FALSE}$ this case is inconsistent with the measurements. In general there may be insufficient information available to determine exactly which FDC region around \overrightarrow{AB} contains the point C . The 15 possible states $AB : C_i, i \in \{1..15\}$ are collected in the boolean vector $AB : C = [AB : C_1, \dots, AB : C_{15}]$ and termed a ‘qualitative state’. These states can also be rendered graphically using table of the form shown in Figure 1.2c, where a black square for region i indicates that $c_i = \text{TRUE}$. The number of TRUE elements of $AB : C$ is inversely proportional to the number of geometric constraints imposed. For example $[AB : C_i = \text{TRUE}, i \in \{1, 2, 3\}]$ indicates that C is somewhere in front of B , while $[AB : C_i = \text{TRUE}, i \in \{3, 6, 9, 12\}]$ indicates that C is somewhere to the right of \overrightarrow{AB} . The FDC structure can be interpreted as a generalization of the representation imposed by the cardinal directions; if point B is taken to infinity and seen as the north pole, then all query points must occur in regions 7-15 which are equivalent in this limiting case to NE,N,NW,etc.

Although discretizing the plane around \overrightarrow{AB} into a finer spatial resolution can be achieved by adding additional separating boundaries, doing so comes at the cost of increased complexity. Both the storage and computation time of the mapping approach presented in this chapter scale quadratically with the number of constraints/regions. The 15 regions of the FDC representation provide intuitive results and limit the size of measurement lookup tables, described in Section 1.3,to



(a) Region Boundaries

(b) Qualitative Regions.

(c) Example Measurement Table

Figure 1.2: Schematics of a Freksa's Double Cross for two landmarks A and B . (a) shows the three dichotomies which split up the space around the vector AB . (b) shows the 15 qualitative regions in which a third point can lie. (c) shows an example measurement table for the case where C is in one of regions 2, 3, 6, or 9.

the point where they can be computed manually. If greater accuracy is required, the methods detailed in the remainder of this chapter can be readily extended to arbitrary resolutions at the cost of additional off-line computation.

1.2.2 FDC Operators

Making use of the FDC representation requires the definition of five operators that allow manipulation of qualitative states. For any landmark triplet A, B, C there are six possible qualitative states: $AB : C$, $BA : C$, $BC : A$, $CB : A$, $CA : B$, and $AC : B$. Converting between each of these states requires an *inverse* operator and two cyclical permutation operators. The *inverse* operator converts between $AB : C$ and $BA : C$, and is equivalent to reversing the direction of the vector between A and B . The *left cyclical permutation* operator converts between $AB : C$ and $BC : A$, while the *right cyclical permutation* operator converts between $AB : C$ and $CA : B$. The binary *compose* operator uses the information contained in two qualitative states in order to reason about which configurations of a third state are consistent (e.g. which regions in $AC : D$ are consistent with the possible relationships stored in $AB : C$ and $BC : D$). The results of these operators are easily found using lookup tables derived in [25] and [24] and summarized in Appendix B. While the operators are defined over the individual regions of each FDC, they can also be applied to each element of the qualitative state vector by taking the union of the outputs. Finally, the map update procedure discussed in Section 1.4 requires an additional *intersection* operator denoted as \cap that combines the constraints in two estimates of a qualitative state. *Intersection* is defined as follows:

$$\begin{aligned}
(AB : C \cap \widehat{AB} : C)_i &= \text{TRUE} \\
&\text{iff } AB : C_i = \text{TRUE} \wedge \widehat{AB} : C_i = \text{TRUE}
\end{aligned}$$

A formal discussion of these and related operations for a set of spatial representations which includes the FDC is presented by Scivos and Nebel [26].

Unfortunately, while the *inverse* operator is a one-to-one mapping, the two cyclical permutation operators are not. For example, performing a left shift on the qualitative state $[AB : C_i = \text{TRUE}, i = 1]$ returns $[BC : A_i = \text{TRUE}, i = 9]$, while doing so on $[AB : C_i = \text{TRUE}, i = 7]$ returns $[BC : A_i = \text{TRUE}, i \in \{1, 4, 7\}]$. These ambiguities, discussed in detail by Scivos and Nebel [27], have unfortunate consequences for the compactness of the map representation discussed in Section 1.4.

1.3 Measuring Qualitative States

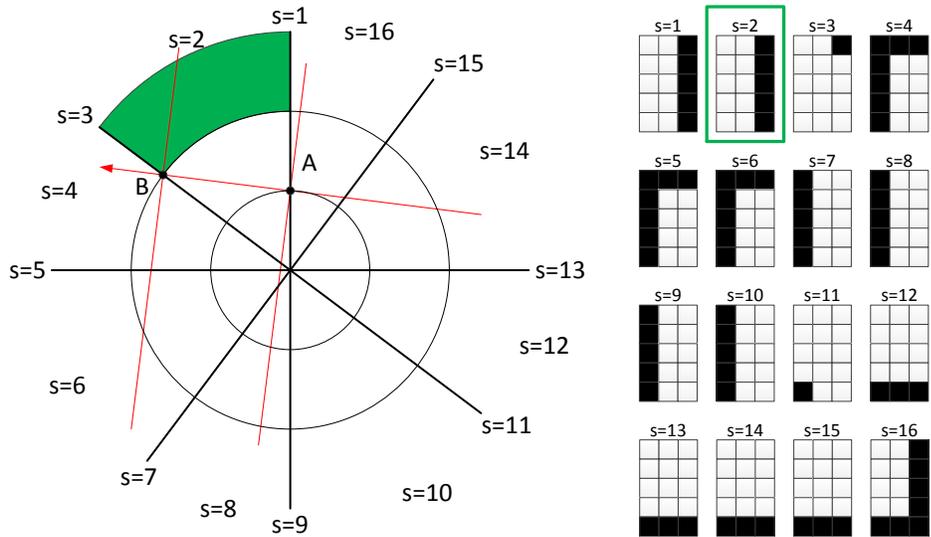
Past work on qualitative mapping, and in particular work using the FDC or similar representations, has characteristically taken a cognitive science approach to the problem. Consequently, the focus has generally been on proving that the underlying representation is sufficient for human navigation, rather than on how it might be practically accomplished in an automatic fashion [28]. The bulk of this research assumes the human building the map is either able to determine exact qualitative states involving all visible landmarks, or is traveling between landmarks and can thus easily determine the states given the angles to landmarks at the beginning and end of each leg. Given the challenges present in computer vision systems, this assumption is insufficient for the general problem of a robot mapping an unknown area. In order to overcome this limitation, a novel method of determining the

possible qualitative relations between a set of landmarks visible in a camera image is presented, which does not require knowledge of any past history or the location of the imaging points.

The measurement function presented in this section uses estimates of the relative bearings to landmarks and an ordering by range in order to give *sectors* for the measurement of landmark C with respect to landmarks A and B . Lookup tables are then used in order to map these *sectors* to FDC regions stored in the qualitative state $AB : C$. This process relies on three assumptions involving information provided by the imaging system:

1. The cyclical order of landmarks can be determined
2. The relative angle between landmarks can be resolved to within a quadrant.
3. There is a low-level algorithm to determine the relative ordering of distances to visible landmarks. In practice, potential methods for accomplishing this ordering include exploiting known sizes of objects, motion parallax, relative changes in object size during approach, and the fact that vertical position in an image is proportional to distance in a flat environment.
4. The landmarks are sufficiently visually distinctive as to be unambiguously identifiable from any orientation. While data association is a challenging and task specific problem that is beyond the scope of this paper, some possible solutions include comparison of high-dimensional visual features such as texture and shape parameters, tracking clusters of objects which may be more distinctive than the individual elements, and the use of previously estimated qualitative states to rule out incorrect associations.

For the purpose of book-keeping, two sets of discrete *sectors* are used to describe the relative angles between landmarks A , B , and C : one relating $\angle A$ and $\angle B$, the



(a) s sectors for $g = 2$

(b) Lookup tables for $g = 2$
and
 $|A| < |B| < |C|$

Figure 1.3: Measurement *sectors* and associated lookup tables for the case $|A| < |B| < |C|$, $0^\circ < B < 90^\circ$. The robot is at the center of the circles. Each *sector* in (a) corresponds to a measurement table element in (b) giving the possible FDC regions for a test point C whose angle lies within the *sector*. For example, $s = 2$, shown as a green region in (a), corresponds with the highlighted table entry in (b). Black squares indicate that the FDC region is consistent with the angle and range ordering. For this configuration of A and B there are additional tables (not shown) for the other possible orderings of $|A|$, $|B|$, and $|C|$.

other relating $\angle C$ to both $\angle A$ and $\angle B$. If $\angle A$ is defined to be 0° then the first of these is denoted as $g \in \{1, 2, \dots, 8\}$ and corresponds to the cases $\angle B = 0^\circ$, $0^\circ < \angle B < 90^\circ$, $\angle B = 90^\circ$, etc. Thus g indicates in which quadrant, or line dividing quadrants, B is located. The second *sector* is denoted as $s \in \{1, 2, \dots, 16\}$ and indicates whether C lies at or between the angles defined by integer multiples of 90° added to $\angle A$ or $\angle B$. So for the case $g = 2$ (i.e. $0^\circ < \angle B < 90^\circ$), $s = 1$ indicates that $\angle C = 0^\circ$, $s = 2$ indicates that $0^\circ < \angle C < \angle B$, $s = 3$ indicates that $\angle C = \angle B$, etc. Labels for s are shown in Figure 1.3a for the case of $g = 2$. The geometries for other common values of g are shown in Appendix A.

Given the relative bearings and range orderings for any three points A , B , and C visible in a camera image, a measurement of the qualitative state $AB : C$ is generated by the following method:

1. Define $\angle A$ to be the 0 angle of the local reference frame, and $|A| = 1$.
2. Given $\angle B$, set the value for g as described above.
3. Given $\angle B$ and $\angle C$, set the value for s as described above.
4. Determine the range ordering $|A| < |B| < |C|$, $|A| < |B| = |C|$, $|A| < |C| < |B|$, etc.
5. Generate a measurement for $AB : C$ using the lookup table associated with g , s , and the range ordering
6. Repeat for all permutations of landmark triplets visible in the image.

This process is best illustrated by considering the example shown in Figure 1.3a where point C lies within the green region. In this case the range ordering is $A < B < C$, $g = 2$ as landmark B lies to the left of A but less than 90° , and $s = 2$ as landmark C lies between A and B . The corresponding lookup table is

outlined in green in Figure 1.3b and indicates that $\overline{AB} : C = \{3, 6, 9, 12, 15\}$ (i.e. C lies somewhere on the right side of \overrightarrow{AB} as shown in Figure 1.2b).

For every combination of landmark ordering and the *sector* g and s there is an associated lookup table which stores the qualitative states consistent with the measurement. These table can be manually constructed by considering the geometrical limits of each FDC region given the constraints on relative ranges and angles. For example, Figure 1.3a shows the case $g = 2$, $|B| > |A|$. In this case, point B is constrained to lie somewhere in the upper left quadrant, outside the circle of radius $|A|$. Lookup tables for the case where $|C| > |B|$, shown in Figure 1.3b, are computed by considering how the FDC boundary lines intersect the region outside the circle of radius $|B|$, as B is moved around within its constraints. For example, the table for $s = 5$ can be constructed as follows: if point B is moved close to the horizontal quadrant boundary, the ray in front of B will intersect the $s = 5$ line outside the $|B|$ circle, so states 1,2,3 have TRUE values in the table. If point B is moved close to the vertical quadrant boundary, both of the lines perpendicular to AB on the left side will cross the $s = 5$ line outside the $|B|$ circle, so states 4,7,10,13 have TRUE values in the table. However, there is no place B can lie within its constraints that will allow the perpendicular line through B on the right of AB to cross the $s = 5$ line, or for any portion of the $s = 5$ line to lie between A and B , or for the line behind A to cross the $s = 5$ line. Consequently, the remaining table elements are FALSE.

Similar arguments can be made for each of the remaining FDC regions, resulting in the complete lookup tables shown in Figure 1.3b. This process can be performed for each combination of g , s , and range ordering, resulting in a total of 2304 possible measurement tables. Tables for the most common arrangements of landmarks are listed in Appendix A. These tables were constructed manually, then verified against

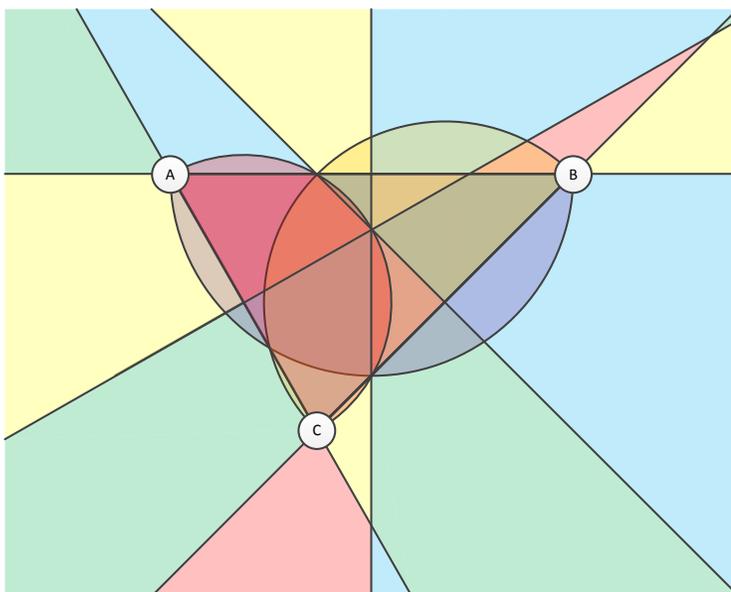


Figure 1.4: Example measurement regions for three arbitrarily placed landmarks. All images taken within a region provide the same qualitative information about the relations $AB : C$, $BC : A$, and $AC : B$. In order to gather new information about the qualitative states, a robot must travel between measurement regions. Region coloring is only for the purpose of visual distinction.

truth for 10,000,000 randomly generate landmark triples.

As in the choice of resolution in the FDC representation, the limitation to 8 *sectors* for B and 16 *sectors* for C in the measurement region is motivated by computational concerns. This resolution results in a set of 2304 measurement tables, although there are a number of symmetries which can be exploited to reduce the amount of calculation required. From a practical point of view, this is near the limit of what can be feasibly computed by hand using the geometrical arguments described above; otherwise one must develop a method to automatically generate measurement tables for arbitrary angular resolutions.

For every set of three landmarks, the measurement function results in a segmentation similar to that shown in Figure 1.4. Intuitively, measurements with differ-

ent graph constraints can only be found by moving between regions, as movement within a region provides no new qualitative information; this spatial breakdown is equivalent to that generated by landmark pair boundaries in [8]. These regions are formed by the lines through each landmark pair, the perpendicular bisectors to those lines, and inward facing semicircles between each pair. In some landmark geometries, adjacent regions may provide the same measurements, however the space is never segmented beyond the degree seen in Figure 1.4.

1.4 Building Qualitative Maps

Repeated measurements of landmarks using the methods in Section 1.3 can be combined to form a consistent map of the environment. This section details the use of a graph structure for map representation and the method by which the map can be updated given new measurements.

1.4.1 Map Structure

The landmark map generated by the QRM approach takes the form of a 3-uniform hypergraph. Formally, the map is defined as a tuple $M = (Q, E)$, where $Q = \{P_1, P_2, \dots, P_N\}$ are the N nodes, and $E = \{E_{ijk}\}$, $i = 3 \dots N, j = 2 \dots i, k = 1 \dots j$ are the edges $E_{ijk} = \{P_i P_j : P_k, P_j P_k : P_i, P_k P_i : P_j\}$. Each node P_i of the graph corresponds to an observed landmark, and each edge E_{ijk} in the graph connects three landmark nodes and contains the qualitative states necessary to define constraints on their geometrical arrangement. There are six possible qualitative states for any three landmarks A, B , and C : $\{AB : C, BA : C, BC : A, CB : A, CA : B, AC : B\}$. As the *inverse* operator is a one-to-one mapping, the states $\{BA : C, CB : A, AC : B\}$ contain redundant information and need not be

explicitly tracked in the map, as they can be easily recreated at any time from the other three. Thus an edge between the three corresponding nodes consists only of the states $\{AB : C, BC : A, CA : B\}$. Unfortunately, the *left cyclical permutation* and *right cyclical permutation* operators which relate these three states to each other are non-unique mappings. Consequently it is not possible to further reduce the edge to a single qualitative state. In online operations, landmark nodes are added to the graph when they are first observed, and need only have edges to other nodes with at least one active constraint.

Maps formed using this graph structure are invariant under a number of potential transformations of the underlying landmark positions. The qualitative presentation used to describe relative landmark positions depends solely on the angles between triplets of landmarks. Consequently, transformations that preserve angles cause no change in the map states. These include rotations, translations, and uniform scaling of the underlying locations. As a result, the mapping system cannot provide any information about the global position and orientation of the map, or about the metrical distances between landmarks. However, transformations which do not preserve angles, such as shearing or non-uniform scaling, can induce changes in the map states when applied to the landmark positions. This means that the map can provide some information about the relative distances between landmarks. For example, if the scale of the landmark distances is known along one axis, the map places constraints on the possible scaling in all other directions as there is a limit to the amount of off-axis scaling that can occur before the constraints embedded in the qualitative states are violated.

1.4.2 Graph Updates

Information provided by measurements is propagated through the graph structure by making use of the operators discussed in Section 1.2. This procedure operates as follows.

1. Given a new measurement $\overline{AB : C}$, find the graph edge linking nodes A , B , and C .
2. Use the *inverse* operator if the edge qualitative states are in the wrong order (e.g. the measurement was $\overline{AB : C}$ but the graph edge stored $\widehat{BA : C}$).
3. Find the intersection of the stored estimate $\widehat{AB : C}$ with the measurement: $\widetilde{AB : C} = \widehat{AB : C} \cap \overline{AB : C}$. The resulting qualitative state contains only those regions consistent with the constraints embedded in both $\widehat{AB : C}$ and $\overline{AB : C}$.
4. If $\widetilde{AB : C} = \widehat{AB : C}$, terminate the update as the measurement contains no new information.
5. Set $\widehat{AB : C} = \widetilde{AB : C}$ to update the estimated qualitative state.
6. Use the *left cyclical permutation* and *right cyclical permutation* operators to generate $\widetilde{BC : A}$ and $\widetilde{AC : B}$ and update the corresponding stored estimates as in steps 3 and 5.
7. For each qualitative state which has changed as a result of the measurement, generate new qualitative state estimates using the *compose* operator on all connected edges.
8. Treat the generated estimates as new measurements and repeat steps 1-6 on each.

The above procedure allows the logical consequences of the observed geometrical constraints to be determined given the existing network of constraints. It is equivalent to the path-consistency algorithm by van Beek [29] discussed in detail by Renz and Nebel [30]. In general, the wavefront of updates may hit any given node a number of times, though the first observation of a new landmark tends to affect most of the graph edges, while subsequent observations tend to only update a few. As portions of the graph become more constrained, they are less likely to be affected by any new measurements. Each iteration either increases the number of constraints or terminates; thus the update is guaranteed to terminate in a finite number of steps. In addition, the update procedure requires only simple set intersections and table lookups, and has a minimal computational cost for each iteration. The main limiting factor in this approach is that the number of edges in a fully connected graph scales as $O(n^3)$.

For any number and configuration of landmarks, there exists some finite sequence of images which provide the measurements necessary to generate a fully constrained graph. However, it is not guaranteed, and in general is not the case, that any given image sequence of that length can generate a fully constrained graph. Unlike standard triangulation, it is possible to construct an infinite sequence of measurements which do not further constrain the graph. For example, all images taken from points in one of the regions in Figure 1.4 provide the same measurements. Given the true positions of landmarks, it is always possible to predict where a fully constraining image sequence should be taken, based on evaluations of the measurement function. However, it is not currently clear if these predictions can be made in a way which enables a robot to automatically plan and decide where to take measurements in order to resolve map ambiguities.

1.4.3 Data Association

The mapping process described in this section relies critically on accurate measurement associations, as an incorrect association can lead to inconsistent state estimates which propagate through the graph. While the issue of consistent data association is highly problem dependent and a full discussion is well beyond the scope of this work, there is one aspect of the mapping process described above that can be used to limit the number of associations that must be considered. When presented with an uncertain assignment, feasibility tests can be performed on all possible qualitative states with regard to visible landmarks with good associations, just as though the landmark in question was a previously unobserved one. The resulting set of qualitative states can then be compared to those for existing map landmarks. Only landmarks with at least one overlapping state for each relationship need be considered for associations, as the remainder are inconsistent with the new measurement. If no possible associations remain after this step, the landmark can be safely added to the graph as a new node. If association remains unclear, the fusion of the measurement can be delayed until the map has converged further, which will lead to fewer possible associations. The order in which measurements are incorporated into the map has no effect on the final map performance, and a delayed fusion will result in the same final map.

1.5 Mapping Results

This section discusses a series of Monte-Carlo simulations designed to test the QRM algorithm, as well as the results of a mapping experiment performed at the JPL Mars Yard.

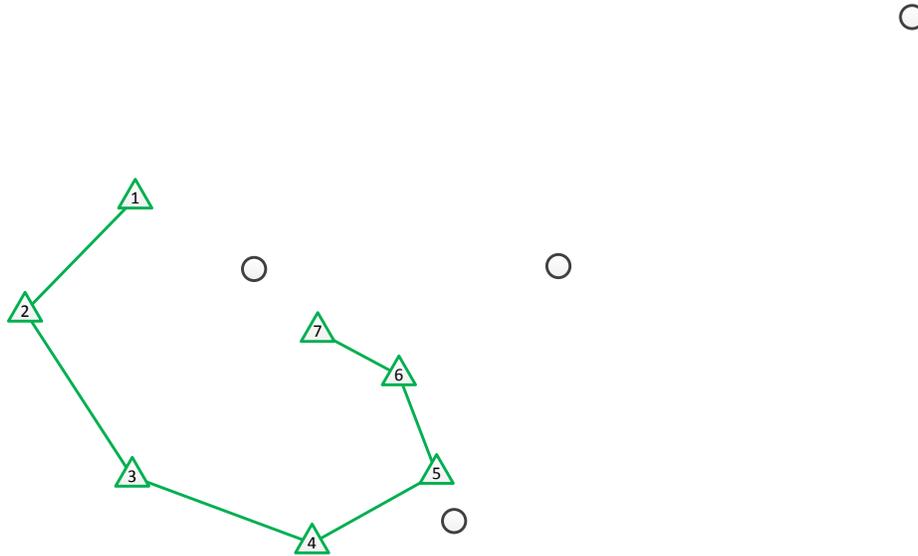


Figure 1.5: Example map layout for simulations, with landmarks as black circles and imaging locations as green triangles.

1.5.1 Map Evaluation Metrics

There are two primary measures for evaluating the quality and convergence of a relational map. The first is the number of open FDC states stored in the graph (i.e. FDC states that are consistent with all the measurements taken thus far). For a fully constrained graph in which the relationships between all landmarks are known, this value is equal to three times the number of edges, as each edge contains the three relationships $\{AB : C, BC : A, CA : B\}$, each with one possible state. The second performance metric is the number of edges which have been fully constrained, compared to the total number of edges in the graph.

1.5.2 Simulation Results

This section presents the results of several test scenarios designed to illuminate some of the properties of the QRM algorithm defined above. The simulation system operates on a set of specified landmark and imaging locations. At each imaging

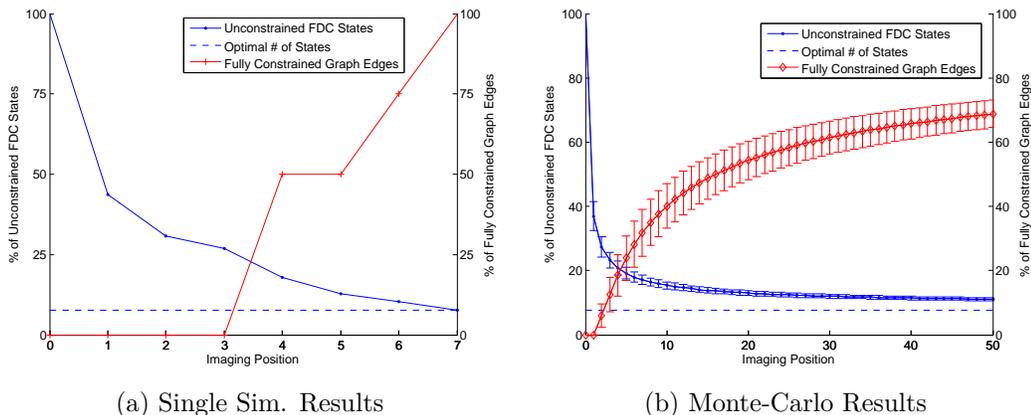


Figure 1.6: Performance of the QRM algorithm as measurements are incorporated into the map. (a) plots results from the simulation shown in Figure 1.5. (b) shows the mean and standard deviations for results from 1000 Monte-Carlo simulations on maps of the same size, each consisting of 49 randomly placed imaging locations and 15 randomly placed landmarks. In both plots the left axis plots the percentage of open FDC states in the map, while the right axis plots the percentage of graph edges which have been fully constrained (reduced to only one FDC state for each of the relations $AB : C$, $BC : A$, and $CA : B$). The dashed line indicates the percentage of FDC states that would remain in a fully converged map.

point the system generates measurements of all detected landmarks using the process described in Section 1.3. Sensor performance can be arbitrarily degraded by limiting the maximum range at which landmarks can be identified. A qualitative map is built by combining measurements from each imaging point sequentially using the approach detailed in Section 1.4.

The first scenario considered is a basic test case, as shown in Figure 1.5. The four landmarks in the map are represented as black circles, and the seven imaging locations are shown as green triangles, numbered by order of incorporation. The imaging points were chosen using the measurement regions shown in Figure 1.4 in order to generate a fully constrained graph by the end of the simulation. Constructing such a sequence is straightforward for small numbers of known landmarks, but is more challenging to do so for large numbers of landmarks without the use of optimization tools. Simulation results are shown in Figure 1.6a. The left axis plots

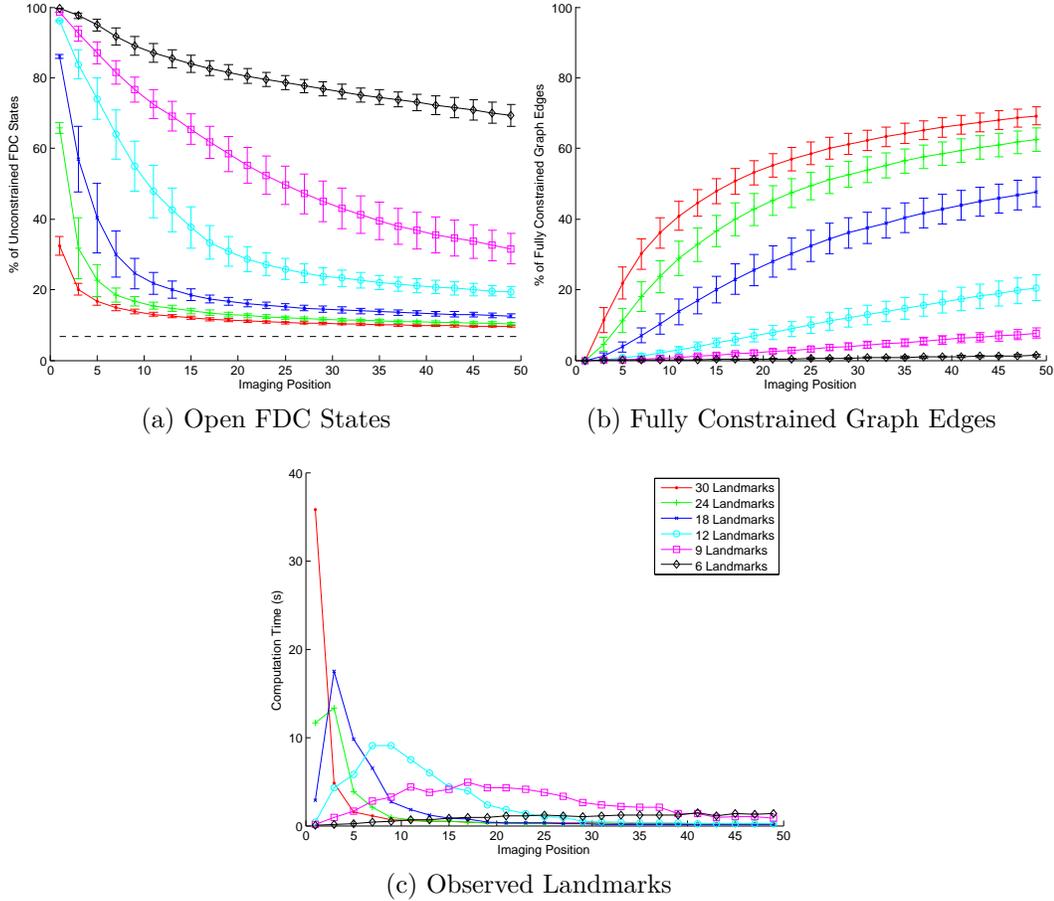


Figure 1.7: Performance of the QRM algorithm as measurements are incorporated into the map for 100 randomly generated maps with 30 landmarks. For each simulation the N closest landmarks to the robot were used to generate measurements, with N varying between 6 and 30. (a) plots mean and standard deviations of the percentage of open FDC states in the map (i.e. states which are consistent with all measurements up to that point), with the dashed line indicating a fully constrained map. (b) plots mean and standard deviations of the percentage of fully constrained graph edges (i.e. edges with only one possible state for all 3 relationships between the associated landmarks). (c) plots the mean computation time on a Pentium Xeon 2.5 GHz processor. Relative errors are omitted for clarity, but range between 30% and 120%.

the number of open FDC states as a percentage of the total number of states in the graph, with the number of states that corresponds to a fully constrained graph shown as a dashed line. The right axis plots the number of fully constrained edges in the graph (edges with only one open state for each of the three relationships stored in the edge: AB:C, BC:A, CA:B) as a percentage of total graph edges. An examination of the lookup tables shown in Figure 1.3b, as well as those listed in Appendix A, shows that most measurements are able to constrain the associated landmark triplet to less than half of the possible qualitative states. This effect manifests as the rapid drop in open states after the first measurement seen in Figure 1.6a. While the initial measurements are able to greatly reduce the number of open states, the system requires the repeated observation of landmarks from different orientations in order to constrain any landmark triplet to a single state. This is seen by the fact that none of the graph edges become fully constrained until after the fourth measurement is incorporated into the map. While the image sequence was processed in the order shown in Figure 1.5, the mapping approach discussed in Section 1.4 makes no assumptions about the order of measurements or about the locations of the imaging points.

The second simulation scenario consists of 1000 Monte-Carlo runs designed to examine the general trends of the QRM algorithm for arbitrary map configurations. For each run 15 landmarks were randomly generated from an area of similar dimensions to that seen in Figure 1.5. Simulated measurements were then taken from 49 randomly chosen imaging locations and combined into a qualitative map using the method described in Section 1.4. Means and standard deviations for both the percentage of open FDC states and the percentage of fully constrained graph edges are plotted in Figure 1.6b, with the percentage of open states for a perfect map again shown as a blue dashed line. These results show that the initial

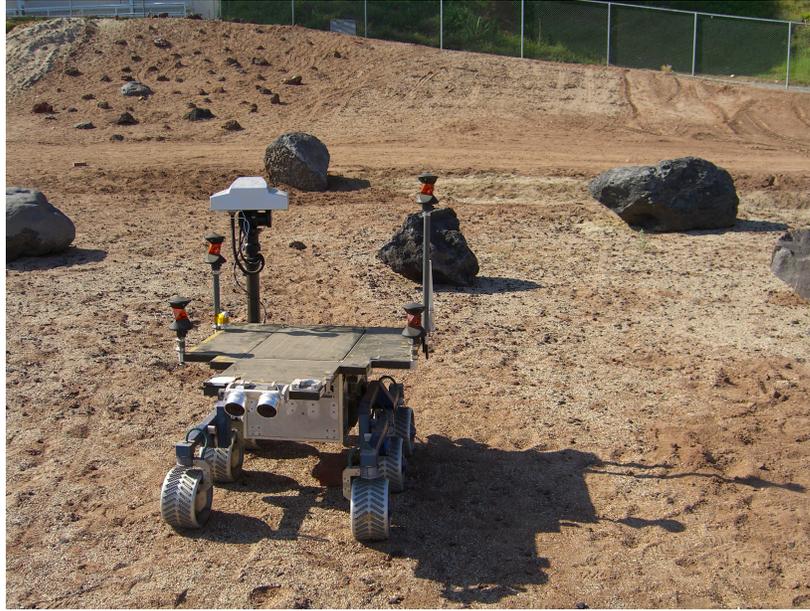


Figure 1.8: The FIDO research rover operating in the JPL Mars yard. The 3D reconstruction of the area shown in Figure 1.9 was performed using images taken by a stereo camera pair located on the sensor mast.

drop-off in open states seen in Figure 1.6a is a characteristic of the QRM algorithm and is independent on the choice and ordering of imaging locations. The slow convergence towards a fully constrained map is to be expected: after the first few measurements are incorporated in the map, new information can only be gained by visiting specific regions. Consequently, it becomes progressively more unlikely that a randomly selected imaging location will provide additional constraints on more than a few landmark relationships.

The QRM algorithm performs best when the robot is able to see all of the landmarks in each image. Doing so allows measurements extracted from each image to potentially add new constraints between all landmarks without needing to rely on information propagating through the graph. This situation will not generally be true in practice, as landmarks visibility will be reduced by both range and occlusions. The effects of reduced landmark visibility can be determined in a straightforward manner by including only the N closest landmarks to the robot at

each time step. In the case of uniformly distributed landmarks this is equivalent to varying the sensor range, but avoids the need to specify a particular scale factor. For this test a set of 100 Monte-Carlo simulations were performed on randomly generated maps with 30 landmarks and 50 imaging positions, each uniformly distributed in a square area. For each map, simulations were performed with the number of closest landmarks measured at each point varying from 6 to 30.

Figure 1.7a plots the mean and standard deviation of the percentage of open FDC states after each update. The results for $N=30$ landmarks used show the same convergence rate seen in Figure 1.6b, while the remaining curves show that convergence remains rapid for $N=24$ and $N=18$, but drops rapidly after $N=12$. Figure 1.7b plots the mean and standard deviations of the percentage of fully constrained graph edges, and shows a more regular drop in performance as N is decreased. This suggests that the open states that are not constrained due to limited sensor range are uniformly distributed in the graph. These results are consistent with tests performed on maps of different sizes, in which performance remains high so long as half of the landmarks are observed at each point.

Figure 1.7c plots the mean computation time required for an unoptimized C# implementation of the mapping algorithm running on a 2.5Ghz Pentium Xeon processor. Relative errors are omitted for clarity, but range between 30% for the peak times and 120% for the minimum times. These results suggest that if limited computation is available, it may be useful to initialize the map using only a subset of the visible landmarks, then increase the number used as the map becomes more constrained. Alternatively, the fusion of measurements for less important landmarks may be delayed until additional computing resources are available. Although this will reduce the accuracy of the map initially, the final performance will be the same regardless of the order in which measurements are

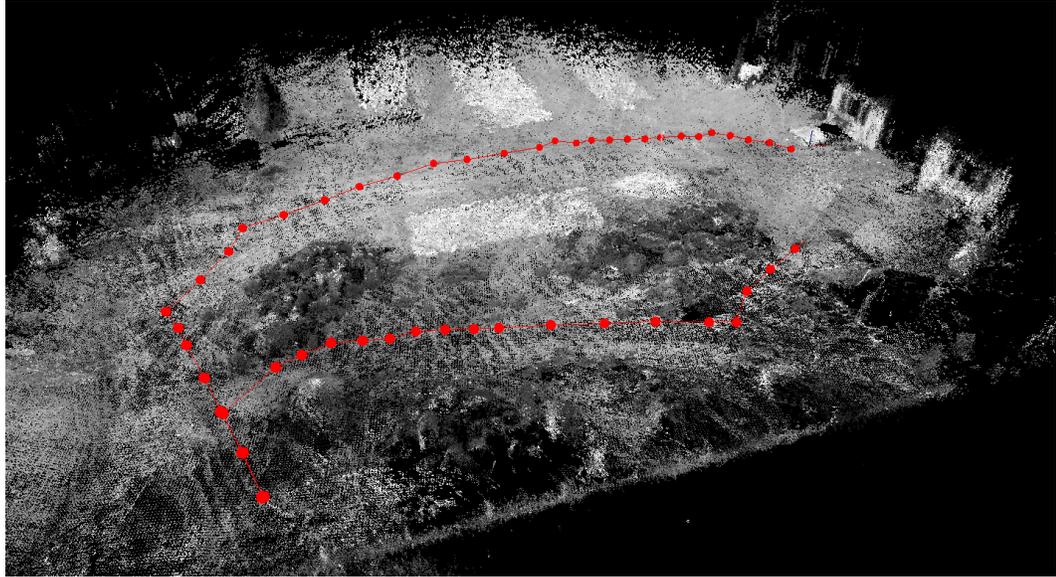


Figure 1.9: 3D reconstruction of the JPL Mars Yard. The pointcloud was generated from stereo panoramas taken at the imaging points denoted by red circles, stitched together using the NDT and LUM algorithms implemented in the PointCloud Library. Landmarks include medium sized rocks such as those in the image center as well as similarly sized objects such as the generators in the upper left and right corners.

fused. An empirical power-law analysis of varying the number of landmarks in the map showed that both the peak time and the time at update 20 go as $O(n^5)$ with the number of landmarks when using all landmarks in each measurement. This improvement relative to the worst case scenario where each of the n^3 measurements affects each of the n^3 graph edges is a result of the increasing level of constraint imposed by each update. After a few measurements have been included, updates cease propagating through the entire graph and only effect edges within a few steps of their origin.

1.5.3 Experimental Results

This section presents an experimental scenario designed to show the algorithm performing in a more realistic environment. The robotic platform used was the JPL

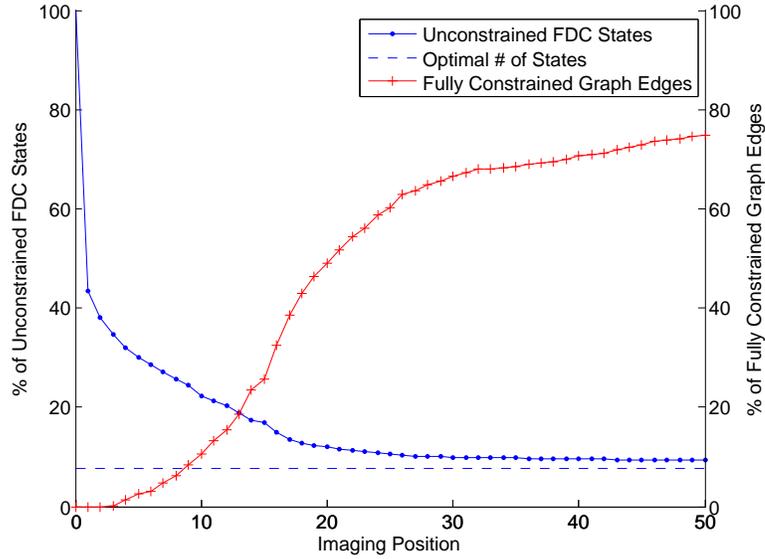


Figure 1.10: Performance of the QRM algorithm for mapping the area shown in Figure 1.9. The left axis plots the percentage of open FDC states in the map, while the right axis plots the percentage of graph edges which have been fully constrained (reduced to only one FDC state for each of the relations $AB : C$, $BC : A$, and $CA : B$). The dashed line indicates the percentage of FDC states that would remain in a fully converged map.

‘FIDO’ research rover shown in Figure 1.8[31]. This vehicle is a 6-wheeled rocker-bogey frame with a mast-mounted stereo camera, and is functionally equivalent to the two Mars Exploration Rovers(MER), Spirit and Opportunity. The experiment objective was to construct a qualitative map of a set of rock fields in a Mars-like environment, in this case the JPL Mars Yard. The rover was driven through the yard, stopping to take panoramic images every 1-2 meters of travel. Landmarks measurements were extracted from these images using the method presented in Section 1.3 and combined using the QRM algorithm described in Section 1.4.

Figure 1.9 shows a 3D reconstruction of the Mars Yard overlaid with the rover trajectory and imaging locations in red; note that this reconstruction was used only for data association and visualization purposes. The pointcloud was built using tools drawn from the JPL Vision group and the PointCloud Library [32]. Specifi-

cally, stereo range data was computed using the approach presented by Goldberg et al. [33] and converted into robot-centered pointclouds containing position and intensity data. The set of clouds from images taken at a single position were aligned using mast attitude measurements, then refined using the Normal Distribution Transform approach of Magnusson et al. [34]. These panoramic clouds were then aligned using position estimates from rover odometry and fused into a final map using the batch alignment method described by Lu and Milios [35]. The large overlap in points between the first and last imaging position was exploited to construct a circular graph of correspondences in order to minimize position drift over the trajectory.

The 30 most visually distinct objects of appropriate size in the environment were manually selected as landmarks for the mapping algorithm. These primarily consisted of medium sized rocks in the clusters seen in the center of Figure 1.9, but also included a few man-made objects such as the generators seen in the upper left, upper right, and lower right corners of the field. At every imaging location, the rover stopped and captured a panoramic set of images using the mast-mounted cameras. Landmarks were extracted from the left camera images and compared against the reconstruction for data association. While the QRM algorithm described previously was run on this data set offline, a desktop computer was able to construct the map in what would have been real-time for the experiment.

Figure 1.10 shows the QRM performance evaluated as a function of imaging position for the experiment described above. The final map produced is somewhat better than the mean results from Monte-Carlo simulations seen in Figure 1.6b, but not dramatically so. The slow initial convergence, compared to the simulation results, is likely due to the fact that the experimental image locations are close together on the rover trajectory, rather than being randomly dispersed throughout

the map. This spatial correlation reduces the likelihood that an image will produce novel measurements when compared to the previous image taken. In general a realistic rover trajectory, which has not been chosen to optimize information gathering, would be expected to perform somewhat worse than a random image sequence of the same length, as the trajectory will have effectively sampled less of the total workspace.

1.6 Conclusion

This chapter presents a novel method for generating qualitative maps for autonomous robotic navigation of large scale spaces with sparse landmarks and minimal sensing. The Qualitative Relational Mapping (QRM) algorithm constructs a constraint network in graph form which restricts the qualitative geometrical relations between landmarks in the map. The underlying representation of these relationships is based on Freksa’s Double Cross (FDC), with the addition of a novel method for generating estimates of allowable qualitative relations based on camera images. Intuitively, this mapping approach performs qualitative triangulation based on angle measurements and estimates of the relative range orderings of visually distinctive landmarks. The graph constructed by the QRM algorithm provides a description of the landmark geometries which is invariant under translation, rotation, and some scaling transformations. Robot navigation objectives which can be expressed in terms of the intersecting half planes formed by the FDCs associated with the landmarks (e.g. ‘stay to the right of points A and B’) can be re-expressed in terms of desired qualitative states with respect to the map graph. Although the graph structure used in this work is fully connected, computational performance could be improved by considering a hierarchical mapping system which eliminates redundant long-distance connections. For a set of land-

marks with known locations it is possible to determine an optimal trajectory for generating a qualitative map; methods for doing so using only the qualitative information in the map are under investigation. Algorithm performance is evaluated using Monte Carlo simulations and shows consistent map convergence as the number of imaging locations is increased. Additional simulations with restrictions on sensor range show that the QRM algorithm continues to perform well so long as at least half of the landmarks are, on average, visible in each image. Experimental data from a traversal of the JPL Mars Yard is used to show that realistic robot trajectories produce similar results, though in general more images will be required for a given level of performance if the imaging are not uniformly distributed.

CHAPTER 2
QUALITATIVE MAPPING AND NAVIGATION FOR LONG-TERM
EXPLORATION

2.1 Introduction

When available, absolute position sensors such as GPS systems provide high quality measurements for generating the position and heading estimates necessary for long-distance autonomous robotic operation. Unfortunately, such systems are unavailable for a number of applications, including extra-planetary exploration, operation in GPS-denied regions, and operation of extremely small or low-cost robotic platforms.

In the absence of absolute position sensors, existing robot localization systems tend to either rely solely on local sensors of ego-motion (such as IMUs and wheel encoders) as in the current GESTALT system for the Mars Exploration Rovers (MER) discussed by Ali et al. [3], or incorporate measurements of the rover's relative position and orientation with respect to certain landmarks in the environment using vision or ranging sensors. This may consist of triangulation from known reference positions as demonstrated by Kuipers and Levitt [4], or the construction of adaptive feature maps as in the Simultaneous Localization and Mapping (SLAM) framework [5]. These methods have definite strengths, including the ability to provide both global position and orientation estimates as well as accurate estimates of the uncertainty in the parameters. They can also provide localization of environmental features in the global reference frame and thus allow the accumulation of information for the assembly of the stable maps necessary for long-distance planning. However, these approaches often face a number of limitations, including computational expense, a reliance on point estimates of landmarks, and the need for high quality sensing to determine metrical distance measurements to visible landmarks. In contrast, the motivation behind this work is to extract information about objects of interest from a minimal set of low-cost sensors, in this case a single camera without any estimates of ego-motion.

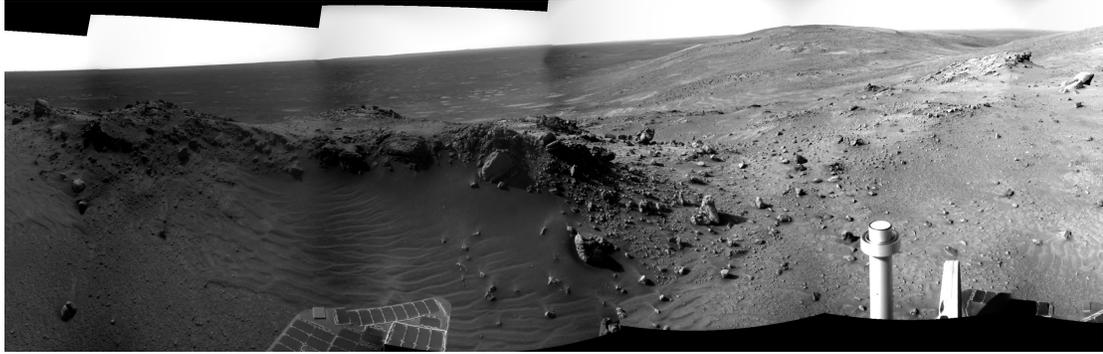


Figure 2.1: Example of a Martian landscape, taken by the Spirit rover on Sol 476. The area shown in this figure is characterized by large open spaces with scattered landmarks.

The solution to the problem of long-term autonomy in the absence of global reference data proposed in this paper is process called the ‘Qualitative Relational Map’ (QRM), in which the relative geometrical relationships between landmarks are tracked using qualitative information inferred from camera images. The key novelties in this work are a representation of geometrical relationships that defines both qualitative orientations and distances, a method for extracting and fusing measurements of qualitative states using global nonlinear optimization, and the implementation of the mapping system in a realistic experimental scenario with data gathered in the JPL Mars Yard. The test case used to evaluate system performance is the exploration and mapping of a Mars-like environment; this application is characterized by large open areas with clusters of interesting features such as the region shown in Figure 2.1. The QRM system developed in this work is an extension of the online mapping process presented in [36].

A key aim of the proposed qualitative framework is to decouple the robot position estimation problem from that of map building. This is inspired by the insight that many robot tasks, such as navigation, do not require a fully defined metrical map. Use of qualitative relations between landmarks allows maps to remain useful in the presence of the distortion that may occur in traditional metrical mapping

approaches due to wheel slippage, rate gyro biases, etc. These sensing errors lead to uncertain estimates of robot ego-motion, which can induce filter inconsistencies in traditional metrical SLAM, as observed by Julier and Uhlmann [14] and Huang and Dissanayake [15], particularly when using the bearing only measurements provided by monocular cameras. SLAM inconsistencies have been addressed in a number of ways in the literature; such as the ego-frame approach with linked submaps presented by Castellanos et al. [16], the topological methods presented by Randell et al. [19] and Angeli et al. [17], the topometric mapping discussed by Sibley et al. [21], and the place-base mapping discussed by Cummins and Newman [18]. These approaches are often successful at limiting filter inconsistencies and map drift in indoors or in urban environments, however, they face a number of challenges in large, unstructured environments. Such areas lack the high feature densities necessary for generating well-defined places or submaps, and do not have the limited connectivities between areas required for topological reasoning.

The qualitative approach detailed in this paper avoids the consistency problem entirely by extracting geometrical constraints on landmark relationships directly from camera measurements, rather than relying on estimated ego-motion. Navigation objectives can then be expressed in terms of these relationships. For example, ‘stay to the right of points A and B’ can be re-expressed in terms of a sequence of desired qualitative states with respect to the map graph. Representing landmark relationships qualitatively avoids both integration and linearization errors, but does so at the cost of maintaining scale free maps with large uncertainties in exact landmark positions, particularly at the edge of the map. In essence, this can be seen as a trade-off between map precision and map consistency. Just as there are an infinite number of spatial layouts that may satisfy any given topological specification, there are an infinite number of metrical arrangements of landmarks

that have equivalent qualitative maps. However, the coordinate sets for all of these point arrangements are constrained to satisfy a set nonlinear inequalities implied by the qualitative statements in the map. Thus, one interpretation of the QRM algorithm is as a form of topological-style reasoning operating on topologically ambiguous spaces.

One approach of how to represent the ‘shape’ of a set of points has been that of statistical shape theory, which defines ‘shape’ to be what remains once scale, rotation, and translation effects have been removed via dimensional reduction. The approach discussed by Dryden and Mardia [22] and Mitteroecker and Gunz [23] uses a QR decomposition to transform a set of high dimensional points to the surface of a hypersphere in a scale, rotation, and translation invariant subspace. Continuous deformation of point sets correspond to trajectories over the hypersphere, and a statistical similarity metric can be constructed by considering probability distributions over the hypersphere. The relationships encoded in the proposed QRM, although driven by different geometrical concerns, correspond to defining nonlinear constraints on these point distributions. Landmark arrangements that have the same qualitative will occupy a bounded, though non-convex, region of the hypersphere defined by the inequality constraints which correspond to the qualitative states encoded in the map edges. Critically, while statistical shape theory requires access to the true landmark locations in some reference frame in order to calculate the ‘shape’ of a point set, the QRM learns the constraints without attempting to estimate the locations themselves.

Previous work on qualitative mapping and navigation for ground robots includes the QUALNAV system described by Lawton [8], which relied on binary relationships inferred from the cyclical ordering of landmarks in a robots view. This representation decomposed the space around landmarks into regions defined

by the rays passing through each landmark pair, as crossing those lines swaps landmark position in the view. Cyclical ordering was also used by Wallgrun [9] to learn the topologies of environments made up of hallway junctions, where junctions are labeled according to their qualitative cardinal orientation. The representation was extended by Schlieder [10] to include the directions opposite landmarks in order to eliminate map ambiguities and termed the ‘panorama’. The cyclical order is also revised to include extended objects with occlusions by Fogliaroni et al. [11], in which a topological map of visibility regions is induced by tangent lines from the extrema of convex polygonal obstacles. These regions are then learned either from an known map of object shapes and locations, or by an exhaustive search of the space.

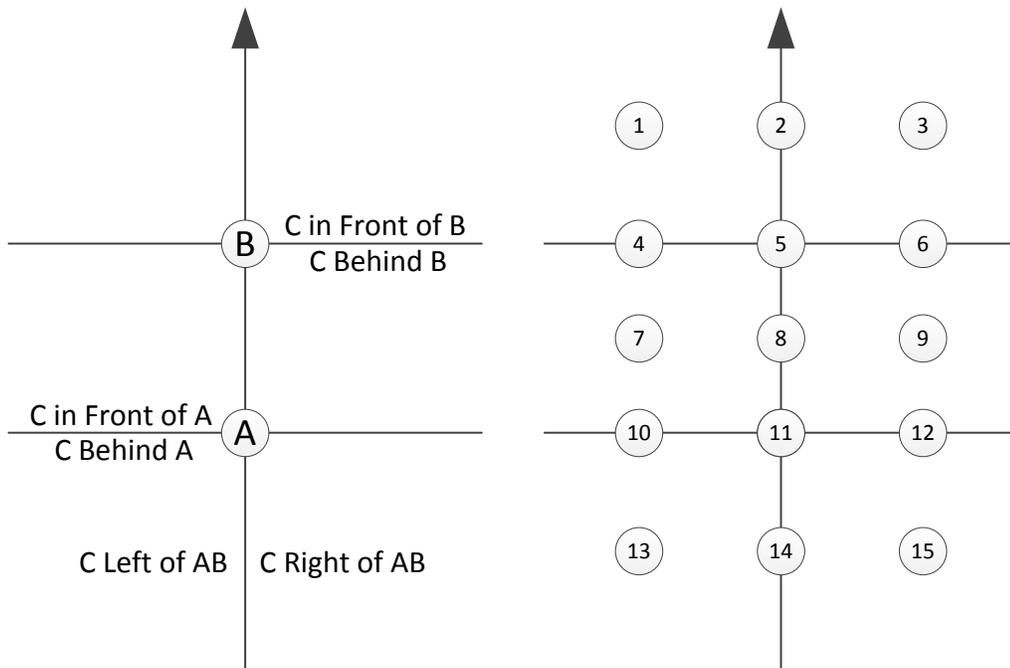
The remainder of this chapter is laid out as follows. Section 2.2 contains a background on the formalism used to define qualitative spatial relationships used in this work. Section 2.3 discusses the generation of lookup tables for operators used to manipulate qualitative relationships. Section 2.4 presents a novel method for generating measurements of qualitative states using camera images. Section 2.5 presents a Branch-and-Bound algorithm for solving the non-convex quadratic feasibility problems required to generate measurements and operator tables. Section 2.6 defines the map structure and summarizes the measurement update algorithm. Section 2.7 presents a method for extracting estimates of the Relative Neighborhood Graph from qualitative maps, as well as a long-distance navigation strategy based on Voronoi regions. Section 2.8 presents the results of a set of Monte-Carlo tests used to evaluate average and asymptotic performance of the mapping algorithm as a function of the number of landmarks simultaneously observed. The results of a data-driven simulation are also presented for a robot moving through a Mars-like environment.

2.2 Qualitative Relational Geometry

Qualitative statements of geometrical relationships require that the 2D plane around landmarks be segmented into discrete regions. The approach presented in this chapter makes use of an extension of the double cross discretization based on triplets of landmarks proposed by Freksa [24]. Freksa’s Double Cross (FDC) specifies the position of a query point C to \overline{AB} , the vector from point A to point B , by stating that C can be either to the left or right of \overline{AB} , in front or behind A relative to the direction of \overline{AB} , and in front or behind B relative to the direction of \overline{AB} . These three statements are equivalent to defining the separating boundaries shown in Figure 2.2a. If the boundary lines are also included as states, this results in the 15 possible geometrical relationships between C and \overline{AB} shown in Figure 2.2b.

The work in this chapter defines an Extended Double Cross (EDC), which adds the additional statements that compare the distance from C to A against that from C to B , the distance from C to A with that between A and B , and the distance from C to B with that between A and B . The separating boundaries associated with the EDC representation are shown in Figure 2.3a, and the 20 possible regions between boundaries are labeled in Figure 2.3b. The FDC representation can be interpreted as qualitatively specifying the angles in the triangle ABC , while the EDC adds explicit qualitative statements about the edge lengths $|AB|$, $|BC|$, and $|CA|$.

The relationship between the point C and \overline{AB} is denoted as the ‘qualitative state’ $AB : C$, which can be one of 20 regions, as shown in Figure 2.3b. In general, there may be insufficient information available to determine exactly which EDC region around \overline{AB} contains the point C , in which case the state $AB : C$ will indicate a set of possible EDC regions. For the sake of clarity, the qualitative



(a) Region Boundaries

(b) Qualitative Regions.

Figure 2.2: Schematics of Freksa's Double Cross for 2 landmarks A and B . (a) shows the three dichotomies which split up the space around the vector AB . (b) shows the 15 qualitative regions in which a third landmark C can lie that result from these dichotomies.

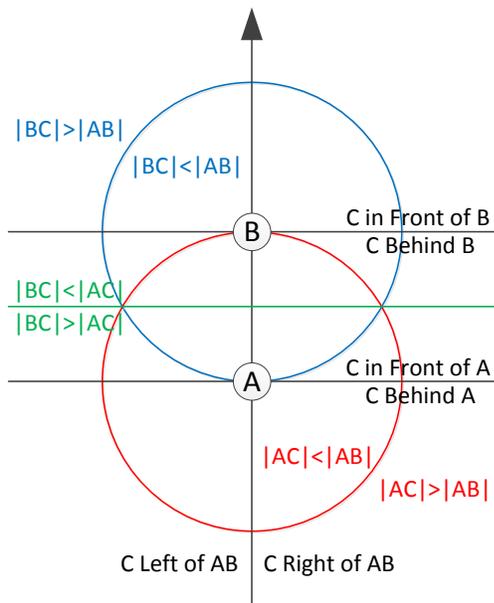
representation used in this chapter is restricted to considering only the relationships defined by the 20 regions defined by the separating lines shown in Figure 2.3a, which can be expressed in terms of linear and quadratic inequalities. In most practical implementations, this is sufficient because physical landmarks are unlikely to lie exactly on a boundary line. If necessary, the optimization approach detailed in the following sections can be easily extended to equality constraints in order to include the lines and line intersections as additional states, or the lines may be incorporated into neighboring regions.

2.3 EDC Operators

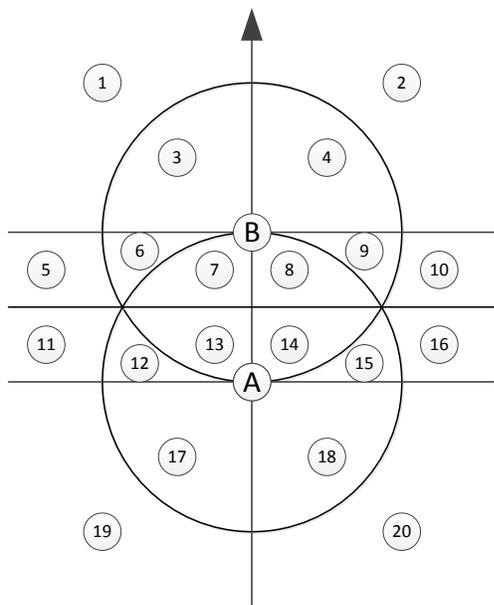
Building a cohesive map of landmark relationships from disparate camera measurements requires the ability to infer how observed relationships can restrict the states of unobserved ones. Doing so requires three unary operators that convert between different representations of a given landmark triple and a composition operator that uses known information about two qualitative states in order to reason about a third. These operators enable transformations between EDC states in the same manner as the FDC operators discussed by Scivos and Nebel [26].

2.3.1 Unary Operators

The qualitative relationships between points A , B , and C can be stored in the EDC states for any of $\{AB : C, BC : A, CA : B, BA : C, AC : B, CB : A\}$. These states are highly interdependent; conversion between the triples is straightforward using two cyclical permutation operations to generate $BC : A$ and $CA : B$ given $AB : C$, and an inversion operation to determine $BA : C$ given $AB : C$. The left shift operator is denoted as $LEFT(AB : C) = BC : A$, while the right shift



(a) Region Boundaries



(b) Qualitative Regions.

Figure 2.3: Schematics of the Extended Double Cross (EDC) for two landmarks A and B . (a) six dichotomies splitting the space around the vector AB . (b) 20 qualitative regions where a third landmark C can lie.

Table 2.1: EDC Unary Transformations

$AB : C$	$BC : A$	$CA : B$	$BA : C$
1	17	7	20
2	18	8	19
3	19	13	18
4	20	14	17
5	12	7	16
6	11	13	15
7	{1,5}	{12,17}	14
8	{2,10}	{15,18}	13
9	16	14	12
10	15	8	11
11	13	6	10
12	7	5	9
13	{3,6}	{11,19}	8
14	{4,9}	{16,20}	7
15	8	10	6
16	14	9	5
17	7	1	4
18	8	2	3
19	13	3	2
20	14	4	1

operator is denoted as $RIGHT(AB : C) = CA : B$, and the inverse operator is denoted as $INVERSE(AB : C) = BA : C$. The results of these unary operations are listed in Table 2.1, and derived algebraically in Appendix C. Unfortunately, while the inversion operator provides a one-to-one mapping, there are four states which are ambiguous under the cyclical transforms. The ambiguities introduced by these operators are similar to those discussed by Scivos and Nebel [27].

2.3.2 Composition Operator

The composition operator determines which EDC states for $AB : D$ are consistent given observed states for $AB : C$ and $BC : D$. While determining the composition rule for any given pair of EDC states for $AB : C$ and $BC : D$ by inspection is

a straightforward process, the number of combinations required to fully populate the operator table renders accurate manual calculation impractical. Instead, the problem can be formulated as determining the feasibility of a set of inequality constraints that can be automatically defined and solved offline for each state combination. Let the points A, B, C, D be generally defined as $A = (0, 0)$, $B = (0, 1)$, $C = (\alpha, \beta)$, and $D = (\gamma, \delta)$. Specifying a state for $AB : C$ is equivalent to defining a set of inequalities drawn from the upper third of Table 2.2 that point C must satisfy. For example, $AB : C = 2$ is equivalent to requiring that

$$\begin{aligned}\alpha &> 0 \\ \beta - 1 &> 0 \\ \alpha^2 + \beta^2 - 2\beta &> 0\end{aligned}$$

Similarly, the EDC states for $BC : D$ are equivalent to inequality sets drawn from the middle block of Table 2.2, while those for $AB : D$ are drawn from the lower third of Table 2.2. The problem of determining if the composition table entry for a pair of states $AB : C$ and $BC : D$ should include a given state for $AB : D$ is accomplished by searching for a point $(\alpha, \beta, \gamma, \delta)$ that jointly satisfies the associated inequality constraints. An efficient Branch-and-Bound algorithm for solving these problems offline is detailed in Section 2.5. An examination of the EDC geometry indicates that any feasible region for this problem will both occupy a non-zero volume of the search space and extend close to the origin. Thus, it is reasonable to also include upper and lower bounds on $(\alpha, \beta, \gamma, \delta)$, so long as those bounds are large compared to $|AB| = 1$.

Table 2.2: EDC boundary expressions for $A = (0, 0)$, $B = (0, 1)$, $C = (\alpha, \beta)$, $D = (\gamma, \delta)$

Expression	Interpretation of Expression < 0
$ \begin{aligned} & -\alpha \\ & -\beta \\ & 1 - \beta \\ & 1 - 2\beta \\ & 1 - (\alpha^2 + \beta^2) \\ & 2\beta - (\alpha^2 + \beta^2) \end{aligned} $	$ \begin{aligned} & C \text{ is to the right of } \overline{AB} \\ & C \text{ is in front of } A \text{ wrt } \overline{AB} \\ & C \text{ is in front of } B \text{ wrt } \overline{AB} \\ & AC > BC \\ & AC > AB \\ & BC > AB \end{aligned} $
$ \begin{aligned} & (\alpha\delta + \gamma) - (\alpha + \beta\gamma) \\ & (\beta + \delta) - (\beta\delta + \alpha\gamma + 1) \\ & (\alpha^2 + \beta^2 + \delta) - (\beta\delta + \alpha\gamma + \beta) \\ & (\alpha^2 + \beta^2 + 2\delta) - (2\beta\delta + 2\alpha\gamma + 1) \\ & (\alpha^2 + \beta^2 + 2\delta) - (\gamma^2 + \delta^2 + 2\beta) \\ & (2\alpha\gamma + 2\beta\delta + 1) - (\gamma^2 + \delta^2 + 2\beta) \end{aligned} $	$ \begin{aligned} & D \text{ is to the right of } \overline{BC} \\ & D \text{ is in front of } B \text{ wrt } \overline{BC} \\ & D \text{ is in front of } C \text{ wrt } \overline{BC} \\ & BD > CD \\ & BD > BC \\ & CD > BC \end{aligned} $
$ \begin{aligned} & -\gamma \\ & -\delta \\ & 1 - \delta \\ & 1 - 2\delta \\ & 1 - (\gamma^2 + \delta^2) \\ & 2\delta - (\gamma^2 + \delta^2) \end{aligned} $	$ \begin{aligned} & D \text{ is to the right of } \overline{AB} \\ & D \text{ is in front of } A \text{ wrt } \overline{AB} \\ & D \text{ is in front of } B \text{ wrt } \overline{AB} \\ & AD > BD \\ & AD > AB \\ & BD > AB \end{aligned} $

2.3.3 Operators Example

The use of these operators on EDC states is best illustrated by a simple example. Consider the case of four landmarks, A , B , C , and D . Let X represent the set of qualitative states $AB : C = \{6, 7\}$, Y the state $AC : D = \{16\}$, and Z the states $BC : D = \{1, 5, 11, 12, 17, 18, 19, 20\}$. The EDC operators can be used to show that X and Y imply Z , or more specifically that $Z = \text{COMPOSE}(\text{LEFT}(X), \text{INVERSE}(Y))$. Evaluation of this expression is done as follows. Performing a left shift on X is done by finding the mappings from states in $AB : C$ to states in $BC : A$ for each state in X using Table 2.1: 6 maps to 11 and 7 maps to $\{1, 5\}$. Consequently, $\text{LEFT}(X)$ results in $BC : A = \{1, 5, 11\}$. The inverse operator applied to Y uses the mappings from $AB : C$ to $BA : C$ given in Table 2.1, so $\text{INVERSE}(Y)$ results in $CA : D = \{5\}$. The composition operator results in the union of the composition of each pairwise combination of states in its arguments. Evaluation of the compose look-up table gives the identities

$$\text{COMPOSE}(1, 5) = \{1, 5, 11, 12, 17, 19\}$$

$$\text{COMPOSE}(5, 5) = \{12, 17, 18, 19, 20\}$$

$$\text{COMPOSE}(11, 5) = \{17, 18, 19, 20\}.$$

Therefore

$$\begin{aligned} \text{COMPOSE}(\text{LEFT}(X), \text{INVERSE}(Y)) = \\ \{1, 5, 11, 12, 17, 18, 19, 20\} = Z \end{aligned}$$

2.4 Measuring Qualitative States

Past work on qualitative mapping, particularly that using the FDC or similar representations, has characteristically taken a cognitive science approach to the

problem in which the focus has been on proving the representation to be sufficient for human navigation, rather than for autonomous robotics [28]. In particular, past work on the FDC has relied on the human building the map to be able to determine exact qualitative states involving all visible landmarks, but has not discussed how this might be achieved by a robot mapping an unknown area.

This section presents a novel method of determining the possible qualitative states for landmarks visible in a camera image, without requiring knowledge of any past history or the location of the imaging point. The measurement function relies on three assumptions involving information provided by the imaging system.

1. Given the image, the angles to the centroids of all visible landmarks can be determined. This is equivalent to having either point-like landmarks, or landmarks with known geometries. The requirement on angle is only in the local camera frame, and there is no need for the robot to know its global orientation.
2. There is a low-level algorithm that determines the relative range ordering of visible landmarks relative to the robot. Possible methods for accomplishing this in practice include exploiting known sizes of objects, motion parallax, relative changes in object size during approach, and the fact that vertical position in an image is proportional to distance in a flat environment.
3. Landmarks are sufficiently distinctive as to be unambiguously identifiable from any orientation. Section 2.6 discusses some aspects of the data association problem and how the map structure can limit the number of associations that must be considered.

Given the bearings and range ordering for each set of three points A , B , and C visible in a camera image, a measurement can be generated which consists of a list of all of the possible EDC states for $AB : C$ consistent with the observation.

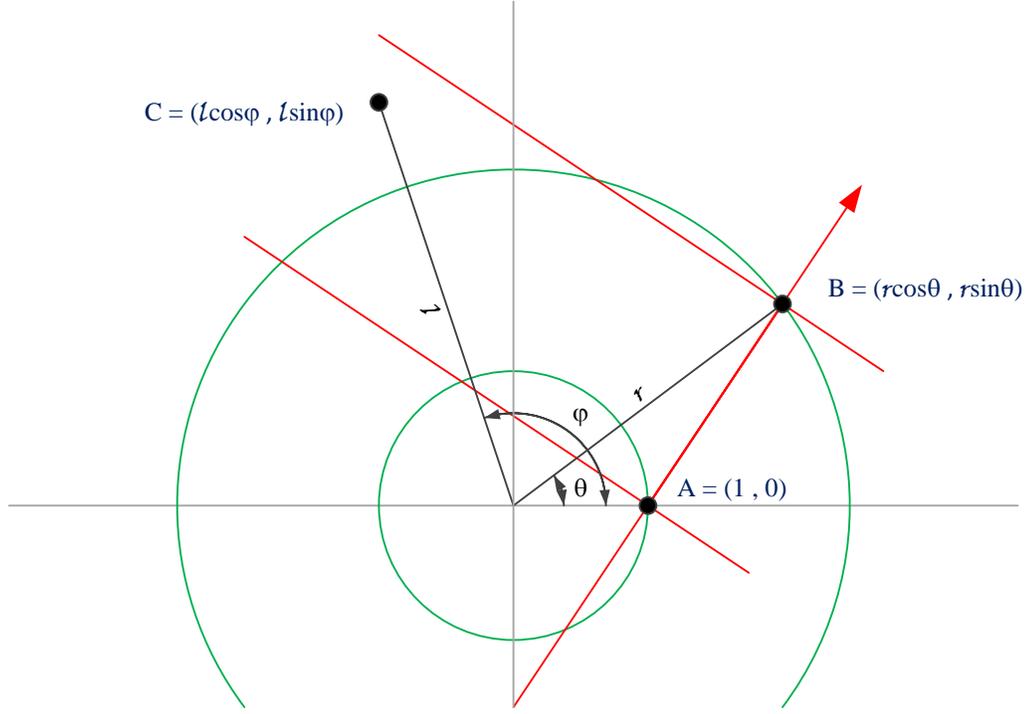


Figure 2.4: Geometrical formulation for the problem of determining the qualitative state $AB : C$ given measurements of θ , ϕ , and the relative order of l , r , and 1. A can be freely defined to lie at $(1, 0)$ as camera measurements provide only relative angle and a scaleless ordering of distances.

Table 2.3: EDC boundary expressions for $A = (0, 1)$, $B = (r \cdot \cos(\theta), r \cdot \sin(\theta))$, $C = (l \cdot \cos(\phi), l \cdot \sin(\phi))$

Expression	Interpretation of Expression < 0
$(\sin(\phi) \cos(\theta) - \cos(\phi) \sin(\theta))lr - \sin(\phi)l + \sin(\theta)r$	C is to the right of \overline{AB}
$-(\sin(\phi) \sin(\theta) + \cos(\phi) \cos(\theta))lr + \cos(\phi)l + \cos(\theta)r - 1$	C is in front of A wrt \overline{AB}
$r^2 - (\sin(\phi) \sin(\theta) + \cos(\phi) \cos(\theta))lr + \cos(\phi)l - \cos(\theta)r$	C is in front of B wrt \overline{AB}
$r^2 - 2(\sin(\phi) \sin(\theta) + \cos(\phi) \cos(\theta))lr + 2 \cos(\phi)l - 1$	$ BC < AC $
$l^2 - r^2 - 2 \cos(\phi)l + 2 \cos(\theta)r$	$ AC < AB $
$l^2 - 2(\sin(\phi) \sin(\theta) + \cos(\phi) \cos(\theta))lr + 2 \cos(\theta)r - 1$	$ BC < AB $

In a camera-centered local reference frame, the point A can be defined to lie at $A = (1, 0)$. Points B and C can then be specified as lying at $B = (r \cdot \cos(\theta), r \cdot \sin(\theta))$ and $C = (l \cdot \cos(\phi), l \cdot \sin(\phi))$, as shown in Figure 2.4, where θ and ϕ are the bearings measured relative to the direction of A , and l and r are the unknown ranges. The boundary conditions of EDC states may then be expressed as a series of equalities, as listed in Figure 2.3b. These equalities are composed of the linear and quadratic expressions listed in Table 2.3. Each EDC state corresponds to a set of three or four expressions being less than or greater than zero. For example, EDC state 7 corresponds to the inequalities

$$\begin{aligned} (\sin(\phi) \cos(\theta) - \cos(\phi) \sin(\theta))lr - \sin(\phi)l + \sin(\theta)r &> 0 \\ r^2 - 2(\sin(\phi) \sin(\theta) + \cos(\phi) \cos(\theta))lr + 2 \cos(\phi)l - 1 &< 0 \\ l^2 - r^2 - 2 \cos(\phi)l + 2 \cos(\theta)r &< 0 \end{aligned}$$

Determining which EDC states are consistent with camera observations can then be achieved by solving the feasibility problem of finding a point $(l, r) > 0$ that satisfies both the EDC state inequalities as well as the observed ordering of l , r , and 1. These ordering constraints are: $1 - l < 0$ if A is closer than C , $1 - r < 0$ if A is closer than B , and $r - l$ if B is closer than C . This feasibility evaluation must be performed for each EDC state (1 – 20). Solving these problems requires the Branch-and-Bound strategy detailed in Section 2.5, as well as a problem-specific upper bound on l and r which more practically bounds the search space. Lists of EDC states consistent with the camera observation for each observed landmark triplet are passed as measurements to the graph update algorithm discussed in Section 2.6.

2.5 Feasibility Detection

Both the calculation of measurements detailed in Section 2.4 and the generation of the lookup table for Composition operator described in Section 2.3 require the solution of feasibility problems in either two or four variables. These problems can be formalized as determining whether there is an x that satisfies a set of quadratic inequalities

$$x^T A_j x + b_j^T x + c_j < 0 \quad j = 1, \dots, M \quad (2.1)$$

subject to the bound constraints

$$l_b \leq x \leq u_b$$

where A_j are N -by- N symmetric matrices, b_j , l_b , u_b are N -by-1 matrices, and c_j are scalars. As A_j may be indefinite for some j , this problem is equivalent to a non-convex global minimization and can be shown to be NP-Hard in general. Fortunately, the small number of variables and the exploitation of the underlying geometry allows the problem to be rapidly solved using a Branch-and-Bound strategy based on that used by Maranas and Floudas [37]. This approach, summarized in Algorithm 1, proceeds by iteratively splitting the search space into sub-rectangles, then finding a lower bound for each constraint inequality over those rectangles. If any lower bound is non-negative, then the rectangle cannot contain a feasible sub-region and is removed from the search. If all lower bounds over a rectangle are negative, then the rectangle is split in half along its longest edge and the new sub-rectangles are evaluated.

An examination of the constraint expressions listed in Tables 2.3 and 2.2 shows that the constraints are, in order of increasing complexity: linear, bi-linear, quadratic with no cross terms, quadratic with only one cross-term, or general quadratic. The exact minimum value in a rectangle can be easily found for the

first four forms, while a tight lower bound can be found for the fifth. Methods for doing this are as follows:

1. The exact minimum value of linear and bi-linear constraints can be found by simply finding the smallest value of the constraint evaluated at each of the rectangle corners [38].
2. The exact minimum value of constraints with no cross-terms can be found by independently optimizing over each variable. The minimum for x_i occurs at either the upper bound on x_i , the lower bound on x_i , or at $x_i = \frac{-b_{ij}}{2A_{ij}}$ if this point lies within the rectangle.
3. The exact minimum value of constraints with only one non-zero diagonal element in A can be found by one-dimensional optimizations over the corresponding variable with all other variables set to each permutation of their extreme values[39].
4. Tight lower bounds can be found for general quadratic constraints by finding the minima of relaxed linear approximations as discussed by Sherali and Tuncbilek [40] and summarized as follows. Dropping the subscript j , let $z = x - l_b$, $\tilde{b} = (2l_b^T A + b)$, and $\tilde{c} = (l_b^T A l_b + b^T l_b + c)$, then the minimization of the right hand side of equation 2.1 becomes

$$\min z^T A z + \tilde{b}^T z + \tilde{c} \quad s.t. \quad 0 \leq z \leq u_b - l_b \quad (2.2)$$

The problem can be augmented by adding the nonlinear implied constraints

$$(g_i - G_i z)(g_j - G_j z) \geq 0 \quad \forall 1 \leq i \leq j \leq 2n \quad (2.3)$$

where g_i and G_i are found by re-writing the original bound constraints

$$\begin{bmatrix} u_b - l_b - z \geq 0 \\ z \geq 0 \end{bmatrix} \equiv \begin{bmatrix} g_i - G_i z \geq 0 \\ i = 1, \dots, 2n \end{bmatrix} \quad (2.4)$$

The augmented problem is linearized by the substitution

$$w_{kl} \equiv z_k z_l \quad \forall 1 \leq k \leq l \leq n \quad (2.5)$$

The resulting linear problem is

$$\begin{aligned} \min \tilde{b}^T z + \sum_{k=1}^n A_{kk} w_{kk} + 2 \sum_{k=1}^{n-1} \sum_{l=k+1}^n A_{kl} w_{kl} + \tilde{c} \quad (2.6) \\ \text{subject to } (g_i - G_i z)(g_j - G_j z) \geq 0 \\ \forall 1 \leq i \leq j \leq 2n \end{aligned}$$

which can be easily solved using any off-the-shelf linear optimization routine.

This branch-and-bound approach is guaranteed to either find a feasible solution to the constrained inequalities in Equation 2.1, or to prove that any such solution must lie within the remaining rectangles of volume less than ϵ , where the value of ϵ is dependent upon the maximum iteration count. The latter case generally indicates that either there is no solution, or the the solution lies on a manifold of lower-dimensionality than the search space and thus a randomly selected point within a rectangle would be unlikely to ever exactly satisfy the constraint equations. For the feasibility problems considered in this chapter, if there is a solution, it must occupy a finite volume of the search space, and an examination of the geometries involved suggests that the necessary value of ϵ should be within a few orders of magnitude of 1. In practice, a maximum depth of 30 with an initial search rectangle of length 1,000 gives error free results for the measurement problem in section 2.4 on trials of 100,000,000 randomly selected point combinations. Generation of the composition tables is an offline function, so a depth of 60 was chosen to minimize the possibility of errors. A copy of this table can be found at <https://campbell.mae.cornell.edu/content/edc-composition-table>.

Algorithm 1: Feasibility Detection

```
1 add rectangle  $r_0 = [l_b, u_b]$  to search queue  $S$ ;  
2 while  $S \neq \emptyset$  do  
3   pop rectangle  $r$  from  $S$ ;  
4   if  $VOLUME(r) < \epsilon$  then  
5     return FALSE;  
6   else  
7     choose random  $x^* \in r$ ;  
8     evaluate constraints  $q(x)_j = x^T A_j x + c_j^T x + d_j$ ;  
9     if  $q(x^*)_j < 0, \forall j \in \{1, M\}$  then  
10      return TRUE;  
11    else  
12      for  $j \leftarrow 1$  to  $M$  do  
13        find  $q_j$  which lowerbounds  $q(x)_j$  on  $r$ ;  
14        if  $q_j < 0, \forall j \in \{1, M\}$  then  
15          split  $r$  into  $r_l$  and  $r_u$ ;  
16          add  $r_l$  and  $r_u$  to  $S$ ;  
17        else  
18          continue;  
19 return FALSE;
```

2.6 Building Qualitative Maps

2.6.1 Map Structure

The qualitative map generated by the algorithm presented in this section takes the form of a 3-uniform hypergraph. Each node of the graph corresponds to an observed landmark, and each edge in the graph connects three nodes and contains estimates of the qualitative states for the geometrical relationships which define their arrangement. There are six possible relationships for any three landmarks A , B , and C : $\{AB : C, BA : C, BC : A, CB : A, CA : B, AC : B\}$. As the inversion operator is a one-to-one mapping, the relationships $\{BA : C, CB : A, AC : B\}$ contain redundant information given $\{AB : C, BC : A, CA : B\}$, and need not be explicitly tracked in the map. Thus, an edge between the three corresponding nodes only stores estimates of $\{AB : C, BC : A, CA : B\}$. The cyclical operators which relate states in these relationships to each other are non-unique mappings, so reduction of the edge to a single relationship is not possible. Formally, the map is defined as a tuple $M = (P, E)$, where $P = \{p_1, p_2, \dots, p_n\}$ are the nodes, and $E = \{e_{ijk}\}$, $i = \{3 \dots n\}$, $j = \{2 \dots i\}$, $k = \{1 \dots j\}$ are the edges $e_{ijk} = \{p_i p_j : p_k, p_j p_k : p_i, p_k p_i : p_j\}$.

2.6.2 Graph Updates

For the following discussion, let $\overline{AB : C}$ indicate a set of states for relationship $AB : C$ stored in the graph, $\widehat{AB : C}$ indicate a measurement of the qualitative states for $AB : C$, and $\widetilde{AB : C}$ indicate temporary estimates of $AB : C$ used for intermediate steps. Information provided by measurements is propagated through the graph structure by making use of the operators discussed in Section 2.2. This procedure, equivalent to the path-consistency algorithm by van Beek [29] discussed

in detail by Renz and Nebel [30], operates as follows.

1. Given a new measurement of relationship $AB : C$, labeled as $\widehat{AB : C}$, check that nodes for landmarks A , B , and C are already in the map; if not add new nodes and create new edges to all existing nodes.
2. Find the graph edge e_{ABC} linking nodes A , B , and C .
3. Invert the states in e_{ABC} if the nodes are stored in the wrong order (e.g. the measurement was $\widehat{AB : C}$ but the graph edge stored $\overline{BA : C}$).
4. Update the stored set of qualitative states $\overline{AB : C}$ by finding the set intersection with the measured states: $\widetilde{AB : C} = \widehat{AB : C} \cap \overline{AB : C}$. The resulting qualitative state contains only those regions consistent with the constraints embedded in both the original value stored in the map, $\overline{AB : C}$, and the the measurement, $\widehat{AB : C}$.
5. If the intersection resulted in the set of states already stored in the map, i.e. $\widetilde{AB : C} = \overline{AB : C}$, terminate the update as the measurement contains no new information.
6. Otherwise, store the reduced set of states in the map by setting $\overline{AB : C} = \widetilde{AB : C}$.
7. Use the cyclical operators to generate pseudo-measurements $\widehat{BC : A}$ and $\widehat{CA : B}$ and update the corresponding edge states as in step 3.
8. For each qualitative state which has changed as a result of the measurement, generate new qualitative state estimates using the composition operator on all connected edges. For example, if $\overline{AB : C}$ has been updated, find all nodes X which have an edge with the stored state $\overline{BC : X}$ and generate $\widehat{AB : X} = \text{COMPOSE}(\overline{AB : C}, \overline{BC : X})$

9. Treat the generated estimates $\widehat{AB : X}$ as pseudo-measurements and repeat steps 1-6 for each X .

For any number and configuration of landmarks, it is guaranteed that there exists a finite image sequence which generates a fully constrained graph. Given the 2D positions, of the landmarks the imaging position of such a sequence can be predicted from evaluations of the measurement function.

2.6.3 Data Association

The mapping process described in this section relies critically on accurate measurement associations, as an incorrect association can lead to inconsistent state estimates which propagate through the graph. While the issue of consistent data association is highly problem dependent and a full discussion is beyond the scope of this work, there is one aspect of the mapping process described above that can be used to limit the number of associations to be considered. When presented with an uncertain assignment, feasibility tests can be performed on all possible qualitative states with regard to visible landmarks with good associations, just as though the landmark in question was previously unobserved. The resulting set of qualitative states can then be compared to those for existing map landmarks. Only landmarks with at least one overlapping state for each relationship need be considered for associations, as the remainder are inconsistent with the new measurement. If no possible associations remain after this step, the landmark can be safely added to the graph as a new node. If association remains unclear, the fusion of the measurement can be delayed until the map has converged further, which leads to fewer possible associations. The order in which measurements are incorporated into the map has no effect on the final map performance; the delayed fusion results in the same final map.

2.7 Qualitative Navigation

Landmark based robotic navigation can be intuitively decomposed into two distinct sub-problems: long-distance navigation between landmarks widely separated in the map, and short distance navigation between landmarks and nearby points of interest. This section focuses upon the first of these sub-problems, as there are a number of vision-based solutions to the second extant in the literature, such as visual homing [41], place recognition [42], etc. Given the ability to reliably travel between a landmark and nearby points, long distance navigation can be achieved provided a strategy can be found to travel between the regions around any two arbitrary landmarks.

Algorithm 2: RNG Estimation

```
1 Given qualitative map  $M = (P, E)$ ;  
2  $N = \text{sizeof}(P)$ ;  
3 Initialize  $D = \{\}$ ,  $W = \{\}$ ;  
4 for  $i = 1 \cdots N$ ,  $j = 1 \cdots N$ ,  $i \neq j$  do  
5     add  $d_{ij}$  to  $D$ ;  
6     add  $w_{ij} = 0$  to  $W$ ;  
7     for  $k = 1 \cdots N$ ,  $k \neq i, j$  do  
8          $conflicts = 0$ ;  
9          $openstates = 0$ ;  
10        for all states  $s \in e_{ijk}$  do  
11             $openstates = openstates + 1$ ;  
12            if  $s \in \{7, 8, 13, 14\}$  then  
13                 $conflicts = conflicts + 1$ ;  
14            if  $openStates = conflicts$  then  
15                remove  $d_{ij}$  from  $D$  remove  $w_{ij}$  from  $W$ ;  
16                BREAK;  
17            else  
18                 $w_{ij} = w_{ij} + conflicts/openstates$ ;  
19         $w_{ij} = w_{ij}/N$ 
```

2.7.1 The Relative Neighborhood Graph

The navigation approach presented in this section makes use of a Relative Neighborhood Graph (RNG) of the landmarks. The RNG, as discussed by [43], is a connected sub-graph of the well known Delaunay triangulation often used in computer vision, as it generates point clusters similar to those produced by humans. Landmarks are neighbors in the RNG if no third landmark appears in the lune between them. As Figure 2.5 shows, points A and C are neighbors, as are C and B , but A and B are not neighbors as point C lies within the green lune between them.

Formally, the RNG is defined as the tuple $R = (P, D, W)$, where $P = \{p_1, p_2, \dots, p_n\}$ are the landmark nodes used in section 2.6, $D = \{d_{ij}\}$ are edges connecting pairs of nodes p_i and p_j , and $W = \{w_{ij}\}$ are edge costs for each edge in D . Estimates of the RNG can be easily extracted from the qualitative map described in section 2.6 by making use of the fact that EDC states $AB : C = \{7, 8, 13, 14\}$ correspond exactly to the lune between A and B . An RNG edge d_{ij} only exists between nodes P_i and P_j for which there is no third landmark in the map in any of these four states.

In the case of an incompletely converged map, estimates of the RNG have to be realized. This is common in cases such as limited exploration of the area, reduced sensor range, and landmark occlusions during exploration. In an incomplete map, there will generally be edges that have some open states indicating that there is a landmark in the lune, and others that indicate there is not. In order to accommodate these cases, candidate RNG edges can be assigned a cost based on the number of potentially conflicting landmarks, each weighted by the fraction of open states within the lune and normalized by the total number of nodes. Edges with no conflicts have a cost of zero and are guaranteed to be true RNG edges, while

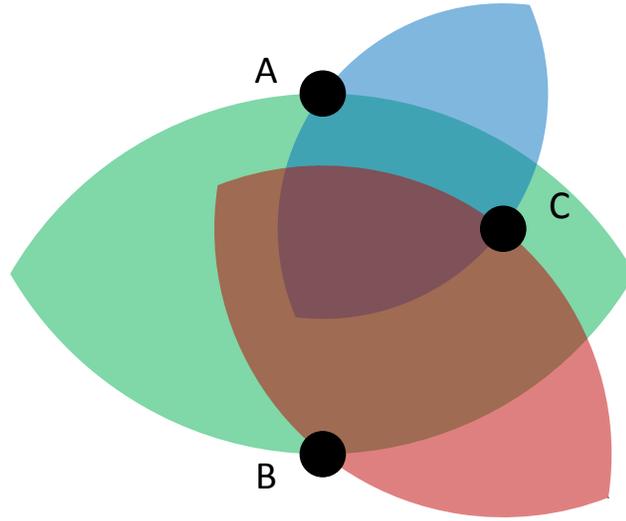


Figure 2.5: Lunes for 3 points A , B , and C which govern their neighbors in the RNG. A and C are neighbors, as are C and B , but A and B are not neighbors as point C lies within the green lune between them.

edges with at least one landmark whose only open states are in the lune can be pruned from the graph. This process, summarized in Algorithm 2, can be cheaply performed after each measurement update, adding potential RNG edges between new landmarks and all existing landmarks in the map, then pruning them away based on the graph updates. As RNG edge estimates depend only on determining if the lunes of landmark pairs contain other landmarks, the convergence rate is bounded above by that of the qualitative map. However, in practice, close approximations to the true RNG are often found early in the mapping process, long before the qualitative map is fully constrained.

2.7.2 Graph-based Navigation

The navigation approach presented in this section assumes that local planners can reliably operate with the Voronoi cells of each landmark (the locus of points closer to the selected landmark than to any other), and focuses on traveling between the Voronoi cells of the landmark closest to the start and goal points. This can be achieved using the limited sensors used in Section 2.4 and the RNG described above, assuming that landmarks are visible from adjacent cells. A simple, but effective, navigation strategy is as follows:

1. Given the start and goal points START and END
2. Find p_s and p_e , the closest landmarks to START and END respectively
3. Use a graph search algorithm to find the shortest sequence of intermediate landmarks connected by RNG edges between q_s and q_e . If the map is well constrained (i.e. the RNG estimate is close to the true RNG), then Dijkstra's algorithm is sufficient. In less constrained cases a weighted approach, such as A^* or D^* is likely to be more effective, using the RNG edge costs as a heuristic to be added to a fixed separation-based distance costs. This biases the search towards paths along the RNG edges least likely to be pruned away by new measurements and towards edges that are most likely to be correct.
4. Drive towards the first landmark in the search path until the rover enters its Voronoi cell, as detected by the relative range orderings of observed landmarks.
5. Remove the current landmark from the search path and drive towards the second landmark until you reach its Voronoi cell.
6. Repeat steps 4 and 5 until the rover has entered the cell around p_e
7. Use a local planner to drive to END

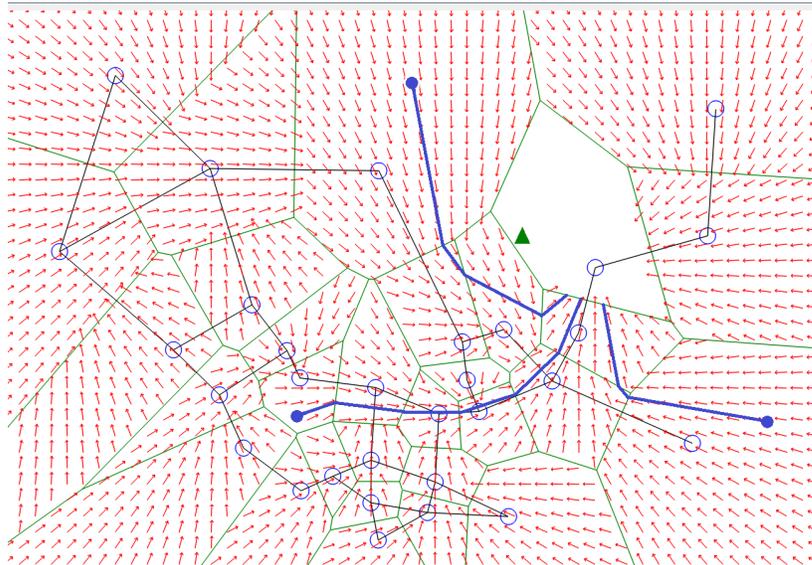
Figure 2.6 shows flow fields for two randomly selected END locations, along with examples of rover trajectories from random START locations. These trajectories show that the navigation approach generates paths that trade distance optimality for a guarantee on reaching the goal, given the assumptions on landmark visibility states in Section 2.4. As the landmark distribution approaches uniformity, distance in the RNG becomes a better proxy for metrical distance, and the difference between trajectories generated by the above strategy and optimal paths decreases. Critically, the navigation approach has no control requirements other than that the robot can always make forward progress towards visible landmarks, and it only requires sensing of local landmarks in order to achieve long-distance objectives.

2.8 Mapping Results

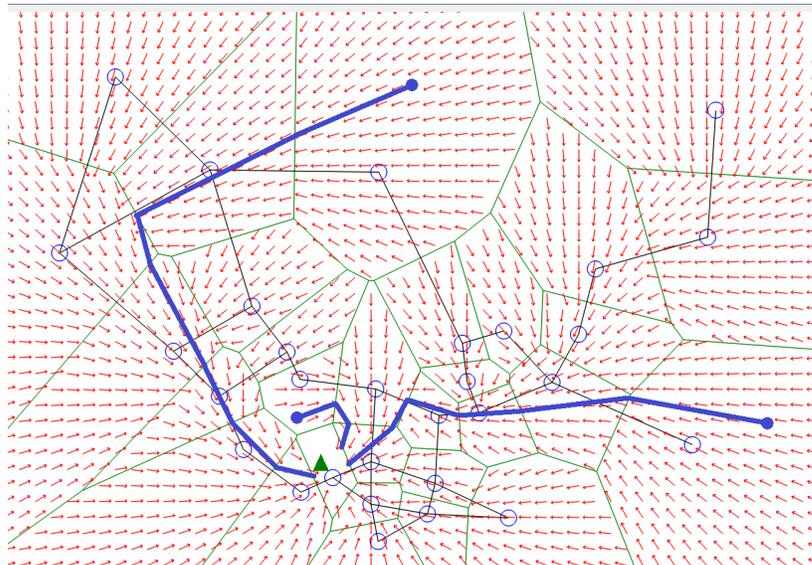
This section discusses a series of Monte-Carlo simulations designed to test the QRM algorithm, as well as experimental results of a mapping the JPL Mars Yard.

2.8.1 Map Evaluation Metrics

There are three primary measures for evaluating the quality and convergence of a relational map. The first is the number of incorrect EDC states that have been removed from the graph by measurement updates, expressed as a percentage of the total number of possible states. The second performance metric is the number of map edges that have been fully constrained, i.e. have only one remaining open state for each of the relationships $AB : C$, $BC : A$, and $CA : B$, again expressed as a percentage of the number of edges in the final map. The third metric is the sum of the edge costs in the RNG, $\sum_i \sum_j w_{ij}$, which indicates the degree of confidence



(a)



(b)

Figure 2.6: Example flow fields along which a robot would travel using the navigation strategy discussed in section 2.7.2. Open circles indicate landmark positions, green lines indicate the borders of Voronoi cells for each landmark, black lines indicate the RNG estimate used for navigation. Red arrows indicate the direction of motion calculated at each point for a robot traveling to the Voronoi region containing the green triangle. Blue lines indicate trajectory of a robot starting from closed circles at three random starting locations.

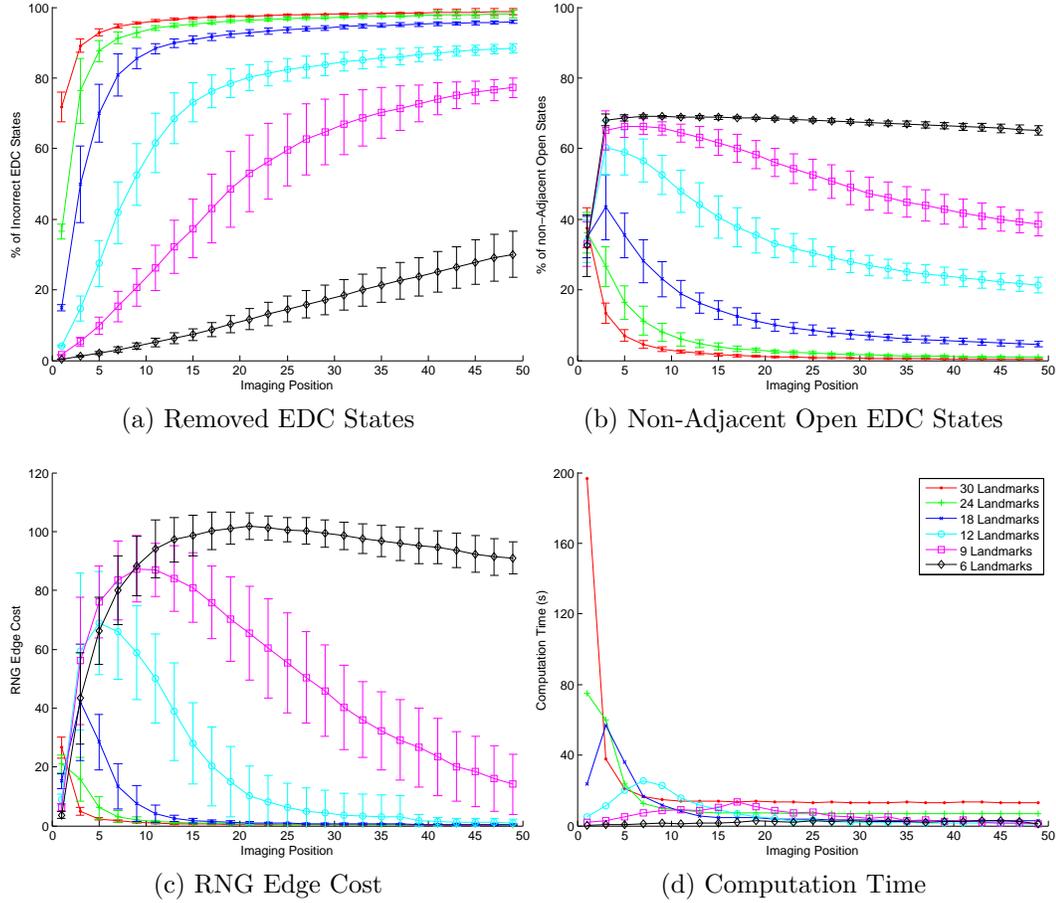


Figure 2.7: Monte-Carlo performance of the QRM algorithm as measurements are incorporated into the map, as a function of the number of n closest landmarks used at each imaging position. The legend in (d) applies to all plots. (a) means and standard deviations of the cumulative percentage of incorrect EDC states that have been removed from the map due to being inconsistent with the observed measurements. (b) means and standard deviations of the percentage of open EDC states which are not adjacent to the true state. (c) means and standard deviations of the total cost of all remaining RNG edges that have potentially conflicting nodes. (d) mean computation times for each measurement update. The relative deviations are not shown for the sake of clarity, but averaged between 15% and 30%.

in the RNG estimate after each measurement.

2.8.2 Monte Carlo Simulations

This section presents the results of a set of Monte Carlo tests designed to illuminate properties of the QRM algorithm defined above. The simulation operates on a set of specified landmark and imaging locations. At each imaging point, the simulation generates measurements of all detected landmarks using the process described in Section 2.4. A qualitative map is built by combining measurements from each imaging point sequentially using the approach detailed in Section 2.6. A total of 100 Monte-Carlo simulations were run to examine the general trends of the QRM algorithm for arbitrary map configurations. For each run, 30 landmarks were randomly generated from a uniform distribution in a square map. Simulated measurements were then taken from 50 randomly chosen imaging locations and combined into a qualitative map using the method described in Section 2.6.

The QRM algorithm performs best when the robot is able to see all of the landmarks in each image, as this allows measurements extracted from each image to potentially add new constraints between all landmarks directly, without needing to rely on less precise information propagated through the graph. This situation is not generally true in practice, as landmark visibility is reduced by both range and occlusions. In addition, for computational reasons the number of landmarks used at each location may be limited. The effect of sensor limitations was tested by evaluating algorithm performance using only closest n landmarks to the rover at each imaging point. For uniformly distributed landmark maps, this measurement restriction is equivalent to imposing a maximum sensor range. Results for these simulations are shown in Figure 2.7 for values of n ranging from 6 to 30.

Figure 2.7a shows the means and standard deviations for the number of EDC

states removed from the graph as a percentage of the total number of possible incorrect states for 30 landmarks. Each plotted line corresponds to a different number of landmarks used for measurement updates, as indicated in the legend in Figure 2.7d. The overall trend is for a rapid initial pruning of the incorrect states, followed by a slow tapering as the remaining states are removed. The initial measurement of a landmark triple is always able to remove at least half of the potential states, as seen by the jump at image position 1. While the initial measurements are able to greatly reduce the number of unconstrained states, the system requires the repeated observation of landmarks from different orientations in order to constrain any landmark triplet to a single state. Consequently, a randomly selected imaging location becomes progressively more unlikely to provide additional constraints on more than a few landmark relationships, manifesting as a slow convergence towards the fully constrained case. Convergence is fastest for the case in which all landmarks are measured (the red 30 landmark line); however the cases using the nearest 18 and 24 landmarks perform nearly as well by the end of the simulations, despite a slower start. In contrast, convergence slows dramatically when less than half of the landmarks, i.e. ($n < 15$), are used in each imaging measurement.

Figure 2.7b plots the mean and standard deviations of the percentage of open EDC states in the map which are not adjacent (sharing a boundary edge or vertex) to the true state of the associated landmark triple. If open EDC states are uniformly distributed, this value ranges between 45% and 90%, depending on the exact geometry of the map. Values lower than 45% indicate that the open states are clustered around the true states, i.e. the map ambiguities are primarily between adjacent states, while values above 75% indicate that the remaining ambiguities are between distant states. The results in Figure 2.7b show that for the 18, 24,

and 30 visible landmark cases, the map ambiguities are quickly reduced to states close to the true state, while the 6 and 9 landmark cases stay within the uniform range.

Figure 2.7c plots the mean and standard deviations of the total edge cost of the RNG estimates after each update. The initial peak in total edge costs followed by a convergent tail corresponds to adding landmarks to the graph, and consequently additional edges, faster than there is enough data to remove existing incorrect edges. For the cases where more than half of the landmarks are used at each step, the RNG estimates clearly converge faster than either the map qualitative states or edges. This trend is unsurprising, as the RNG estimates depend on determining only whether landmarks lie within one of the four lune states, so a great deal of ambiguity can still be present in the map as a whole even after the RNG has converged.

Figure 2.7d shows the mean computation time required for each measurement update; the relative deviations are not shown for the sake of clarity, but were typically 15 – 30%. Simulations were performed using unoptimized C# code running on a Pentium Xeon at 2.5GHz. At each step, computation costs are dominated by the number of feasibility tests that must be performed in order to generate measurements, which depends on the number of landmarks observed and the number of open EDC states in the map. When all, or nearly all, landmarks are seen in every image, the peak computation time occurs in the initial measurements, as every landmark triplet must be checked, and every EDC state is open. A power-law analysis shows that that this peak cost scales as $O(n^{3.5})$ with the number of landmarks. When a small number of landmarks are seen in each image, the initial cost is greatly reduced, as only a few triplets need to be checked, and the map itself contains fewer edges. Results suggest that if limited computation is available, the

map may be initialized using only a subset of the visible landmarks, and then the number of landmarks used increased as the map becomes more constrained. Alternatively, the fusion of measurements for less important landmarks may be delayed until additional computing resources are available. Although this may reduce the accuracy of the map initially, the final performance will be the same regardless of the order in which measurements are fused.

2.8.3 Data-Driven Simulation

This section presents a scenario designed to illuminate some of the properties of the Qualitative Mapping and Navigation algorithms developed in this paper. The platform used was a 6-wheeled rocker-bogey frame with a mast-mounted stereo camera functionally equivalent to the two Mars Exploration Rovers (MER), Spirit and Opportunity. The experiment objective was to construct a qualitative map of a set of rock fields in a Mars-like environment, with Mars-like hardware and operations. The rover was driven through the field, stopping to take panoramic images every 1-2 meters of travel. Landmarks measurements were extracted from these images using the method presented in Section 2.4 and combined using the mapping algorithm described in Section 2.6.

As the Mars Yard data did not include the necessary parallax information for range ordering of landmarks in each image, these measurements were generated using the true position of the rover and landmarks at each imaging point. This process relied on extracting landmark and rover positions from a 3D reconstruction of the environment overlaid with the rover trajectory and imaging locations, as shown in Figure 2.8. Stereo ranges were computed using the approach presented in [33] and converted into robot-centered pointclouds containing position and intensity data. The set of clouds from images taken at a single position were

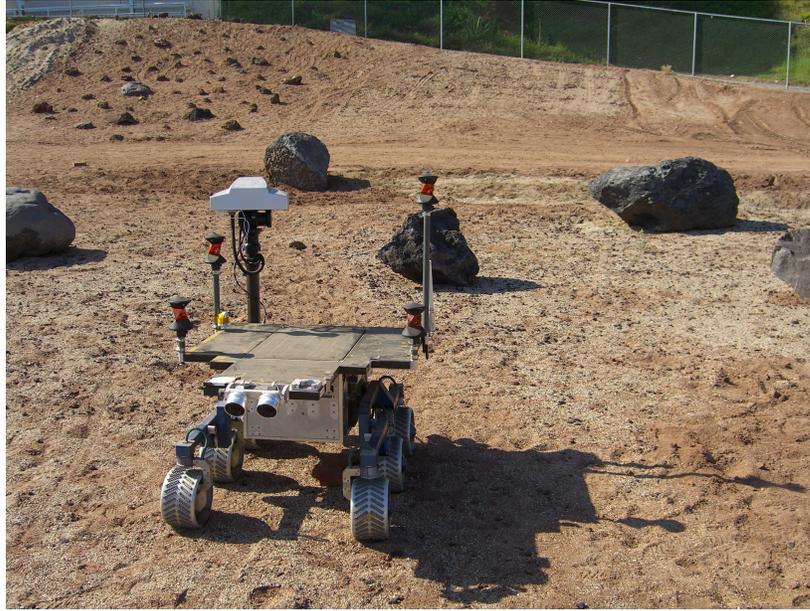
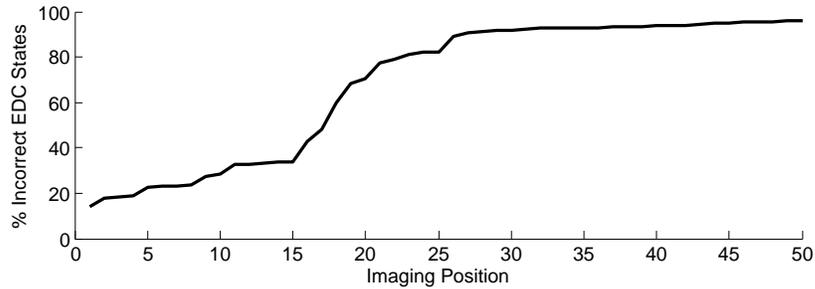


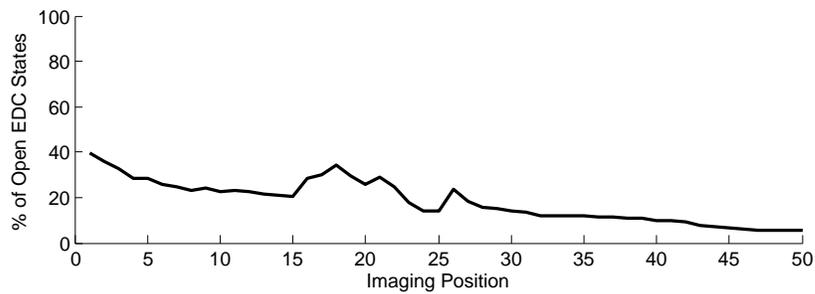
Figure 2.9: The FIDO research rover operating in the JPL Mars yard. The 3D reconstruction of the area shown in Figure 2.8 was performed using images taken by a stereo camera pair located on the sensor mast.

aligned using mast attitude measurements, then refined using the Normal Distribution Transform approach of Magnusson et al. [34]. These panoramic clouds were initially aligned using position estimates from rover odometry, and fused into the final map using the batch alignment method described by Lu and Milios [35]. The rover traversal formed a loop through the Mars yard, which created a significant overlap in points between the first and last imaging position that was exploited to construct a circular graph of correspondences in order to minimize position drift.

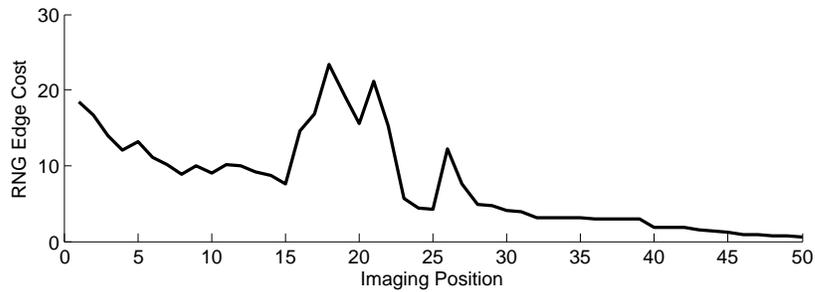
The 30 most visually distinct objects of appropriate size in the environment were manually selected as landmarks for the mapping algorithm. These primarily consist of medium sized rocks in one of the clusters seen in the center of Figure 2.8, but also a few man-made objects such as the generators seen in the corners of the field. At every imaging location the rover stopped and captured a panorama using the mast-mounted cameras. Landmarks were manually extracted from the left camera images and compared against the reconstruction for data association.



(a) Removed EDC States



(b) Non-Adjacent Open EDC States



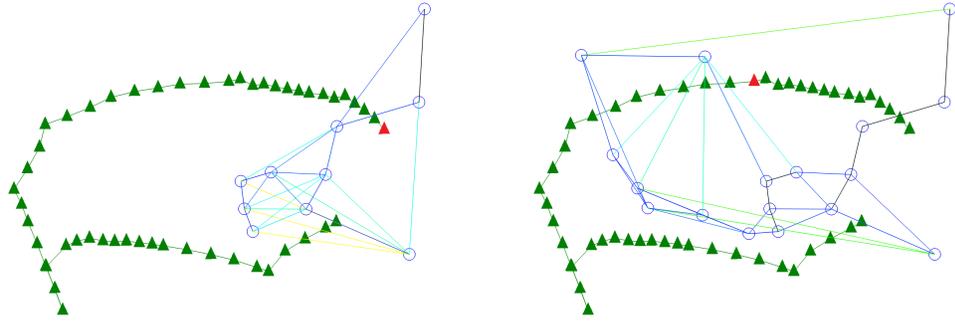
(c) RNG Edge Cost

Figure 2.10: Mapping performance of the QRM algorithm for the FIDO rover traversal of the JPL mars yard described in section 2.8.3 and shown in Figure 2.8. At each step, measurements of the 18 landmarks nearest the rover were taken. (a) the cumulative number of EDC states removed from the map after each imaging point, as a percentage of the number of EDC states for a 30 landmark map. (b) the percentage of open EDC states that are not adjacent to the true state. (c) the total cost of the RNG edges extracted from the map at each step, where cost is equal to the number of conflicting states as a fraction of the total number of open states.

The nearest 18 landmarks were then used to perform map updates. While the mapping algorithm described previously was run on this data set offline, a desktop computer was able to construct the map in real-time.

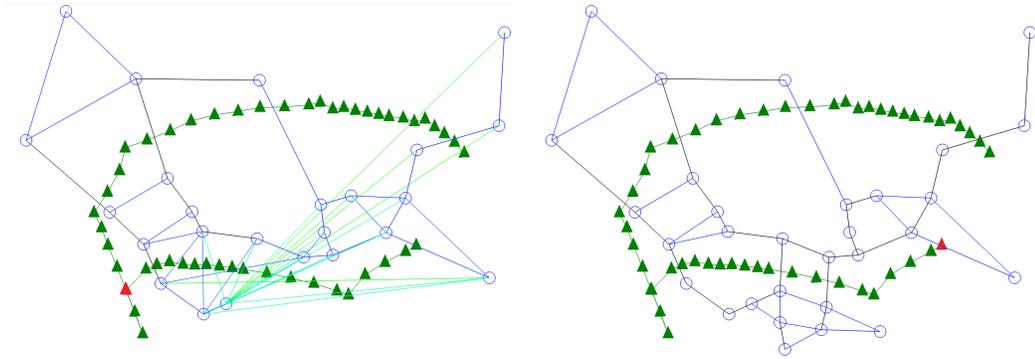
Mapping results for this experiment are shown in Figure 2.10; as a comparison to the prior simulation results, the Mars yard experiment results can be compared to the blue lines in Figure 2.7(a-c). The most striking features when compared against the Monte-Carlo results is the slower convergence of map states and the distinct sigmoid shape of the plot in Figure 2.10a. This can be attributed to two distinguishing characteristics of a realistic traversal. The first is that the FIDO rover explored a smaller fraction of the total map, measured in terms of rover footprint, than a randomly selected set of imaging locations does. This means that the range of novel viewpoints is more limited than the total waypoint count alone suggests. The second, related, characteristic is that sequential images are highly correlated in this experiment, and thus provide much less additional information than would a new image taken from a random point on the map. These effects are particularly apparent in the first few images, as the random points used in the Monte-Carlo tests are likely to be far apart, and thus provide very different views of the scene, while the first few waypoints in the FIDO traverse are close together. The effects of these differences are less apparent toward the end of the simulations, as in both cases, new images are taken near previously visited locations and provide a limited amount of new information regarding landmark qualitative states.

The RNG performance shown in 2.10c shows equivalent final performance, reduced peak error, but a higher average error in the middle of the traverse than seen in the Monte-Carlo results. This is a direct result of the limited sensor range and sequential measurements. As the traverse imaging points are close spatially, the system tends to acquire the measurements necessary to constrain the RNG



(a) After 1st Measurement

(b) After 15th Measurement



(c) After 30th Measurement

(d) After 50th Measurement

Figure 2.11: Evolution of the Relative Neighborhood Graph (RNG) extracted from the qualitative map generated for the rover traversal described in section 2.8.3 and shown in Figure 2.8. Blue circles indicate the 2D locations of landmarks already observed and included in the map, green triangles indicate the imaging locations, with the rover starting in the upper right corner, the red triangle indicates the most recent imaging location, and lines indicate RNG edges. RNG edges are colored according to the edge weights w_{ij} discussed in Algorithm 2, with black indicating that $w_{ij} = 0$ and then ranging from dark blue to red as w_{ij} ranges from 0 to 1.

faster than than observing new landmarks and adding them to the graph. This results in a slowly growing, but well constrained RNG estimate at each step. This progression can be best seen through the snapshots of the RNG shown in Figure 2.11, taken after the 1st, 15th, 30th, and 50th (final) measurements are incorporated into the graph. In these plots, RNG edges are removed from consideration if any landmark is confirmed to lie within the lune (i.e. there is some landmark C s.t. $AB : C$ has no components except for members of the set $\{7, 8, 13, 14\}$). These figures illustrate that as the RNG grows, only a few long-distance (and thus, for this case, incorrect) edges are maintained, and that these edges tend to have a high edge cost. When a new landmark is observed, as occurs in 2.11c, there is sufficient information stored in the map to restrict its incorrect connections to only a few nodes, and these erroneous edges are quickly pruned away by subsequent measurements.

Total computation time for generating the qualitative map and updating the RNG estimate at each step was 250 seconds. For comparison, the FIDO rover required approximately six hours of continuous operation to perform the traverse, stopping every 1–2 meters to collect a panoramic image. The speed of this process was primarily limited by the inefficient method used to gather panoramic images and the rovers low top speed of 9cm/s.

2.9 Conclusion

A novel landmark-based mapping and long-distance navigation approach using qualitative geometry has been presented. The problem of long-distance operation of robots in sparse, unconstrained environments is considered, using the robotic exploration of Mars as an example applications. The algorithms generate and operate on graphical networks which store constraints on qualitative geometrical

relationships between triplets of landmarks in the map based on limited sensor measurements. The underlying qualitative representation of these relations, termed the Extended Double Cross (EDC), defines constraints on the qualitative distances between landmark triplets as well as their qualitative angles. This mapping approach performs a form of qualitative triangulation based on angle measurements and estimates of the relative range orderings of visually distinctive landmarks; these measurements are consistent with current Mars-rover sensing technology. Both the measurements and the offline generation of lookup tables for converting between states make use of a Branch-and-Bound approach to determining the feasibility of sets of non-convex quadratic inequalities. The hypergraph constructed by this algorithm provides a description of the landmark geometries which is invariant under translation, rotation, and uniform scaling transformations. Robot navigation objectives can be expressed in terms of the intersecting regions formed by the EDC state boundaries associated with the landmarks; for example ‘stay to the right of points A and B’, can be re-expressed in terms of desired qualitative states with respect to the map graph. An example navigation strategy was presented which uses estimates of the landmark Relative Neighborhood Graph (RNG) extracted from the qualitative map in order to find paths between the Voronoi regions of arbitrary landmarks.

The asymptotic behavior of the mapping system was evaluated using Monte-Carlo simulations of randomly generated maps with 30 landmarks, and simulated rovers which utilize a varying number of the closest landmarks at uniformly distributed random imaging positions. Results show that while map convergence rates are closely linked to the number of landmarks simultaneously observed, the system has similar asymptotic performance when at least half the landmarks are used at each step, and with greatly reduce computational requirements compared

to simultaneous observations of all 30 landmarks. The results also demonstrate that computational cost is strongly tied to the presence of new landmarks in a measurement, so that an incremental map-building strategy is preferred to a batch approach. The qualitative mapping system was also evaluated using data-driven simulations based on a traversal of the Jet Propulsion Laboratory Mars Yard by the FIDO research rover. A 3D reconstruction of the yard was used to determine the true rover trajectory and landmark locations. Results show that although overall map convergence was slower than the Monte-Carlo results with random imaging locations, due to correlations between sequential measurements, the system was able to reach a comparable performance level by the end of the traverse, and that the RNG extracted at each step tended to remain more constrained than that seen in the Monte-Carlo results.

CHAPTER 3
PROBABILISTIC MODELING OF ANTICIPATION IN HUMAN
CONTROLLERS

3.1 Introduction

Substantial time delays are an endemic problem in many tele-robotic applications, ranging from extraterrestrial exploration to remote surgery. Delays are often introduced into a feedback loop by a combination of signal transmission time and processing requirements at either end of the remote link. An extreme example of this is the robotic exploration of Mars, in which one-way delays range from 3 to 22 minutes, and communication occurs only once or twice a day [44]. Delays may also be a product of the dynamics of the control system, as in the case of underwater vehicles controlled by a long tether, where delays may be as long as 5 minutes [45]. The effects of delay on mission performance varies with task complexity [46] and the consequences of failure; results can include increased time to complete tasks [47], reduced mission performance [44], and unacceptable or catastrophic failures [48].

The use of onboard autonomy can improve robustness to time delays, and can even replace a time-delayed remote operator. However, the required increase in local processing power often presents serious limits to the application of autonomy [49]. Indeed, even were fully autonomous robots to be in common use, it is likely that there would remain tasks that were sufficiently complex or sensitive as to require human control, such as remote surgery [48] or handling of sensitive material. In such cases, systems must be constructed to mitigate the instabilities that are introduced by time delay in the control loop. Accurate models of human adaptation and behavior when presented with time delays are essential to the design process of such systems.

The bulk of the research over the past 60 years in the area of human compensation for time delays has focused on remote control of manipulators and haptic force-feedback control systems. The work in [47] and [50] found that the time to

complete manipulative tasks in the presence of delay can be consistently predicted. Other studies have focused on how time delay effects can be mitigated, for example by the use of predictive displays that provide the operator with a preview of the results of control actions [51], or by adjusting the viewpoint of the operator [52]. There are two consistent limitations present in these studies. The first is that they allowed the human operator complete control over robot velocities. This leads to so called 'move and wait' strategies, in which operators perform open loop maneuvers and then wait for visual feedback before performing the next maneuver. Obviously, the utility of such systems is limited to applications in which intermittent control is allowed. The second limitation is that they generally avoid determining how the operator compensates for the delay and focus on measuring and predicting the degradation of performance, generally in terms of task completion time[53] or tracking performance [54].

In contrast, the experiment conducted by the authors and discussed in [36], was designed to show that when operating in a continuously closed-loop manner, human operators rely upon *anticipation* in order to compensate for system time delays. *Anticipation* is defined here as the ability of human operators to determine future control needs based on internal system models, estimates of the current system state, and visual cues from the environment. It is hypothesized that a more detailed understanding of this *anticipation* can be accomplished by finding good models of the input/output response of human subjects, and then examining how the parameters of these models vary with time delay.

This chapter presents a statistical modeling paradigm designed to formally study how humans anticipate in general, and how they compensate for time delays in command and control systems. More specifically, the experimental data in [36] is used to train models of the human input/output response when controlling

a remote vehicle with time delayed feedback. The proposed modeling approach naturally utilizes information gleaned from the local environment in order to evaluate how humans *anticipate* future control needs. Machine learning techniques are used to probabilistically model the relationship between human responses and the measured set of environmental data. This process is represented as a model selection problem, in which models are fit to the data using two currently popular approaches: the ‘Least Angle Regression’ (LARS) and ‘Sparse Multinomial Logistic Regression’ (SMLR) algorithms. These methods are commonly used to either find a model of minimum size achieving a desired performance level, or to find a model with the maximum performance for a given size [55]. Unfortunately, these two strategies provide little insight into the relative contribution of features within the model to the final performance. This insight is critical to a deeper analysis of the system, such as the discovery of the underlying causes of a particular decision. In addition, the model parameter values cannot, in general, be compared between subjects, as these can be very sensitive to the choice of model size as well as system noise.

In order to overcome these limitations, this work defines a novel ‘importance’ metric in order to gain deeper insight into the environmental features most relied upon by the human subjects in their decision making. This approach presumes that if both modeling approaches produce models that are capable of predicting human responses in a similar manner, then these models must share key characteristics with the true input/output relationship of the system. Furthermore, if models produced by two different algorithms on the same data show similar characteristic patterns in feature scorings on the basis of importance, it is reasonable to conclude that the underlying truth also exhibits these patterns. In the case presented here, if the hypothesis that humans *anticipate* in command and control systems is correct,

then the ‘importance’ of key features in both models is expected to vary with time delay in the same manner.

The remainder of this chapter is structured as follows. Section 3.2 presents the design of the experiment, and a brief discussion of the performance of human subjects. The modeling strategies used to fit the resulting data are discussed in Section 3.3, as are the motivations behind feature and algorithm selection. This chapter makes use of two models with roots in the machine learning community: regularized linear regression and regularized multiclass logistic classification. These models are fit to the data using the LARS and SMLR algorithms respectively. The resulting data fits and subsequent analysis of human compensation mechanisms is presented in Section 3.4. Section 3.5 presents a method for determining feature importance based on how they add to the predictive power of the models.

3.2 Human-Robotic Experiments With Time Delays

In August of 2009, a series of remote driving tests were conducted at Cornell University, in which human subjects drove a wirelessly controlled robot around an indoor route with varying amounts of time delay introduced into the control loop. The goal of this experiment was to probe the degree to which human operators rely on the ability to anticipate future control needs, as opposed to acting as a purely reactive controller. The experimental hardware, procedures, and human trial performance are presented in this section.

3.2.1 Hardware Setup

The robotic platform used for these experiments was a Mobile Robots Pioneer 3-DX differential drive robot mounted with a custom micro-ITX computer and a

video camera. Subjects controlled the system by means of a joystick and robot-mounted camera system; subjects had no view of the test area other than the video feed. In order to eliminate move-and-wait strategies from the study, the robot was programmed to move at a constant translational velocity of **0.2m/s**, leaving the user to command desired rotational velocities using the joystick interface. Continuous rotation rates were limited to lie between ± 0.5 rad/s. These speed limits were chosen so that subjects were able to follow the route with relative ease when no delays were present, and were able to avoid hitting the walls of the test area at high time delays. By allowing the user to vary only the rotation rate, it was hypothesized that the effects of time delay on the human controller would be clearer in the resulting data, and more amenable to modeling. While compensation for time delay can be expected to vary with translational velocity, an examination of this relationship is outside the scope of this study.

The workspace was outfitted with a 24 camera, near-IR motion capture system produced by Vicon, Inc operating at 100Hz, downsampled to 20Hz to match the frame rate of the onboard camera. This system is able to localize the position of the robot to within approximately 1cm in position and 2 degrees in angle. The resulting data was stored and used to fit the models used in subsequent analysis.

While the robot-mounted camera recorded at a resolution of 640x480 pixels, compressed 160x120 pixel jpeg images were broadcast at a rate of ~ 20 Hz to a video monitor placed so that the subject could not directly observe the workspace in which the robot operated. This reduction in quality was necessary due to processing and bandwidth limitations, but the work in [56] indicates that this decision should have had no effect on subject performance. The camera had a **50°** field of view and could see objects lying between 0.4m and 2m from the kinematic center of the robot; an example camera frame showing the green carpet



Figure 3.1: Example of a camera image seen by subject at the beginning of a lap. The yellow line is the reference route the subject has been instructed to follow. Note that a future portion of the path is shown in the upper left.

environment and the desired route, is shown in Figure 3.1. Symmetric time delays were introduced into the system between the joystick and the robot and between the camera and the monitor.

3.2.2 Reference Route

The reference route, shown in Figure 3.2 as a black line, consisted of five curved segments of varying complexity, separated by short straight segments inserted to assist subjects in returning to the route between turns. The curved segments included:

1. an initial slow turn to the left
2. a five turn switchback
3. a medium turn to the left
4. a medium turn to the right

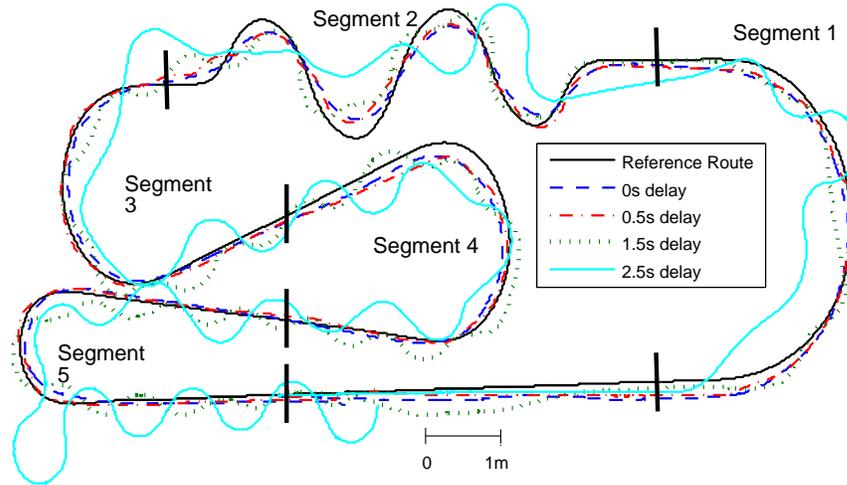


Figure 3.2: Reference route (black solid), along with four characteristic experimental trajectories for a single subject at differing time delays. Black bars indicate breaks between segments used in performance analysis.

5. a sharp turn to the left

This layout was chosen to provide a range of driving challenges, from the easier left turn (1) to the far more difficult switchback (2). It was observed that when the robot passed through the tight turns of the switchback, the camera was only able to see a small portion of the route ahead of the robot. It is hypothesized that this challenging segment was likely to induce noticeable errors even under no delay.

3.2.3 Experimental Process

The experiment objective for each human subject was to steer the robot along the reference route, keeping the robot center as close to the route as possible. Each subject was first asked to drive a single lap at no delay in order to familiarize them self with the control system and robot dynamics. The good performance at no time delay seen in Figure 3.3 suggests that this training was sufficient for

providing subjects with an internal model of the robot response to commands. Subjects then conducted three laps at each delay, pausing after every lap to allow the system to be reset. The round trip delay was varied in 0.5s intervals from 0 to 2.5s, which was the maximum delay at which the robot could be driven safely given the space constraints of the experiment. Subjects were informed of the current time delay, and the same sequence of increasing delays was used for each subject. As all subjects were given the same order of delays, the presence of long-term adaptation to time delay, or other order effects, cannot be ruled out. However, the presence or absence of such effects has no effect on the overall aim of this chapter, namely the use of model fitting algorithms to determine the relative ‘importance’ of environmental features. For the remainder of this chapter, all time delays discussed are stated in terms of the round trip delay imposed by the software. While the effects of human reaction time can be significant when presented with sudden visual stimuli, this work assumes that the variations in the reference route are smooth enough that additional delay due to reaction time is small enough compared to the imposed delays as to have negligible effect.

The following data was collected for each trial at a rate of 20Hz:

1. \mathbf{p} : the 2D position of the robot kinematic center with regards to the fixed coordinate system of the workspace.
2. θ : the robot heading with regards to the fixed coordinate system.
3. \mathbf{u} : the control action of the human operator expressed as a desired angular velocity, measured by the joystick commands.

Given this history of robot position and heading, and the known geometry of the system, it was possible to determine the visible portion of the reference route (as seen by camera) at each timestep of the experiment. This approach was used to

Table 3.1: Results of the 6 x 5 x 3 ANOVA test for the effects of time delay, route segment, and lap number on human RMS off-route error.

Independent Variable	F	p
Time Delay	$F(5, 1518) = 238$	$p < 0.001$
Segment	$F(4, 1518) = 8.9$	$p < 0.001$
Lap #	$F(2, 1518) = 1.2$	$p = 0.30$
Time Delay + Segment	$F(20, 1518) = 1.2$	$p = 0.26$
Time Delay + Lap #	$F(10, 1518) = 0.9$	$p = 0.49$
Route Segment + Lap #	$F(8, 1518) = 0.5$	$p = 0.88$

generate the extended feature set discussed in Section 3.3, and subsequently used for modeling.

3.2.4 Human Subjects

A total of 20 subjects took part in the experiments, consisting of 16 students in the Department of Mechanical and Aerospace Engineering at Cornell University and 4 other members of the Cornell community. The gender breakdown was 7 women and 13 men between the ages of 20 and 40. None of the subjects had prior experience using only a joystick and camera system to control a robot, with or without delay added to the system.

3.2.5 Experimental Results

This section summarizes the performance of the human subjects in completing the robot steering task. For the following analysis, subject performance is defined as the root mean square (RMS) off-route error over each segment, defined at each time step as the Euclidean distance between the robot position and the nearest point on the reference route in the same segment. The effects on off-route error of time delay, route segment, and which lap the subject was on are shown via

the results of a 6 (Time Delay) x 5 (Route Segment) x 3 (Lap Number) ANOVA test in Table 3.1. As expected, the effect of time delay meets the criterion for statistical significance ($p < 0.001$). Likewise, the effect of route segment had a significant effect on performance ($p < 0.001$), indicating that different maneuvers had significantly different difficulties. The effect of lap number on RMS off-route error is seen to be statistically insignificant ($p > 0.05$), which suggests that there is little to no short term learning demonstrated between laps at the same time delay. Consequently, for the rest of the analysis the results of different laps within the same time delay, for one subject, are considered to be a single joint data set. Finally, the ANOVA results show that there are no significant second order interactions between variables, indicating that the relative difficulty of route segments was independent of time delay.

The means and standard deviations for RMS error across subjects are shown in Figure 3.3, for each combination of time delay and route segment. These results show that, in addition to the average human performance decreasing with increased time delay, the variation between subjects also increases. Consequently, models fit to the human trial data are expected to have a more difficult time predicting human responses to known inputs at higher delays. Note that, while subjects reported the initial slow left turn (segment 1) to be the easiest segment, the mean RMS errors for this segment were the highest for delays of **0.5s** and above. This is likely due to subjects' attempts to center the reference route in the camera view, rather than center the robot, leading to systematic over-steering and a large associated error.

Examples of typical paths taken at four different time delays are shown in Figure 3.2. The growing oscillations with increasing time delay are characteristic of all subjects. It is hypothesized that the primary cause of performance loss was due to the over-correction of small errors, with larger deviations occurring due to

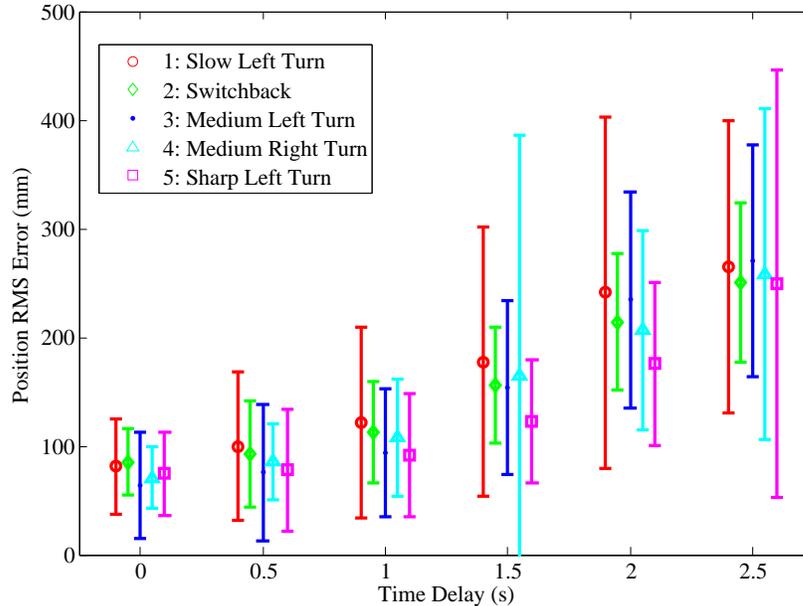


Figure 3.3: Human performance as a function of time delay for each route segment. Performance was measured by finding the RMS off-route error over each segment at each time delay. Error bars indicate 1σ bounds.

subject disorientation when over-corrections lead to the camera losing sight of the route for extended periods of time. These oscillations indicate that, while human operators are clearly able to compensate for time delay to a degree, the manner in which they do so is less than ideal. Likely sources of error include a difficulty with keeping track of the true delay magnitude, an inaccurate internal model of the robot, and insufficient memory of the control history.

3.3 Modeling Approach: Bayesian, with Feature Ordering

This section presents a strategy for generating models of the input/output response of the human subjects. The goal is to find interpretable models of human decisions, and their dependencies. A Bayesian approach is used, in which the human control response \mathbf{u} is modeled by the probability distribution $p(\mathbf{u}|\mathbf{x})$, where conditioning occurs based on a set of features \mathbf{x} extracted from the information provided to the

operator. This approach is chosen because all available data can be used, including *a priori* knowledge of the system, and because the representation is both compact and intuitive. In addition, a probabilistic approach has an intuitive appeal when faced with the vagaries of human response data, as the associated uncertainty can be captured directly.

A key challenge in finding good models to explain the data set is that the video signal is potentially very feature rich, and there is little *a priori* information as to which features are important to operators. This implies that the data provides little information on ‘over-fitting’, or the observability of model parameters. In order to solve this problem, a feature set \mathbf{x} is selected which deliberately over-parameterizes the problem so that there is a high degree of confidence that the system response can be accurately predicted. Models of the dependence of \mathbf{u} on this feature set are fit using two state of the art statistical learning techniques. These two methods have the key characteristic that the features can be ‘ordered’ based on their relative importance to the model. Thus, it is hypothesized that these methods can yield important insight into the decision making process of humans, most notably as a function of time delays and anticipation. The remainder of this section presents the chosen feature set followed by the implementation details of both algorithms.

3.3.1 Feature Selection

Let \mathbf{I}_k indicate the camera image displayed to the human operator at timestep k , \mathbf{p}_k and $\boldsymbol{\theta}_k$ be the position and orientation of the robot in the workspace reference frame at the time \mathbf{I}_k was taken, and $\mathbf{r}(\mathbf{p}_k)$ be the closest point on the reference route to point \mathbf{p}_k . The definitions for \mathbf{p}_k and $\boldsymbol{\theta}_k$ account for the fact that the operator has no direct measurements of the robots current position and heading

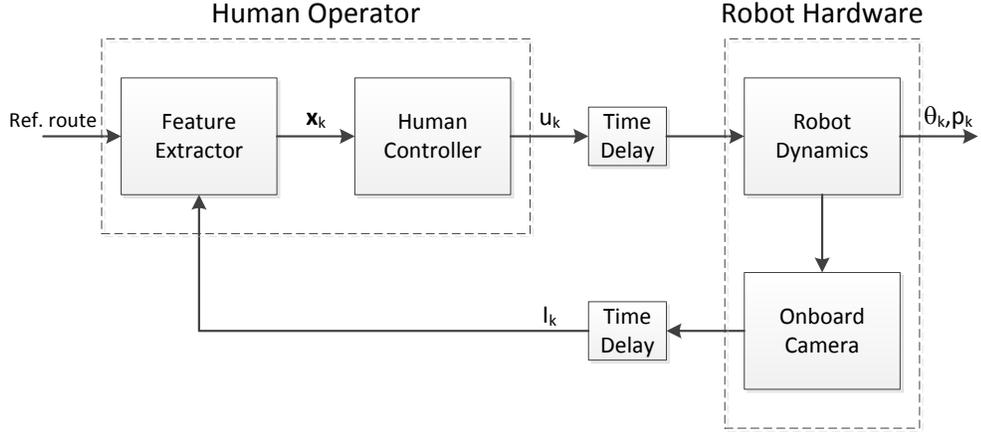


Figure 3.4: Schematic of combined human-robotic system. Note the symmetric time delay blocks between the human controller and the robot.

relative to the reference route, only what is provided by the time-delayed image I_k . The robot rotational rate commanded by the subject at timestep k is given by u_k , and lies in the range ± 0.5 rad/s. A block diagram of the resulting human-robotic system is shown in Figure 3.4.

In principle, u_k could depend on any number of nonlinear functions defined over all the camera frames already observed, $p(u_k | f_1(I_0, \dots, I_k), f_2(I_0, \dots, I_k), \dots)$. However, it is hypothesized that the human subjects rely on a smaller set of features extracted from the images nearest the current timestep. As such, a set of environmental features was chosen to include those that have an intuitive appeal to the problem, with the goal that the true set of features used by human operators is either contained within, or at the least highly correlated with, the chosen set. This approach naturally leads to an over-parameterization of the system, a challenge that requires the use of data fitting algorithms that have the potential to ignore uninformative features.

In order to generate the chosen set of environmental features, the following assumptions are made: the operator estimates p_k and θ_k ; given I_k and previous

knowledge of the route the operator can find $\mathbf{r}(\mathbf{q})$ for any point \mathbf{q} in the vicinity of \mathbf{p}_k ; and the operator has a course model of the internal mapping of the robot response to input commands, represented mathematically as a kinematic model. Knowledge of these elements even approximately enables an operator to predict the consequences of their choice of the command \mathbf{u}_k .

The features generated at time k are shown in Figure 3.5, and defined as follows:

1. $\boldsymbol{\eta}_k$: the past 3 seconds of control inputs \mathbf{u}_k , averaged over $\Delta t = 0.2$ s bins. These features are designed to capture a form of control memory in the system. It is assumed that $\boldsymbol{\eta}$ constitutes the only dependence on past information and all other features can be generated knowing only \mathbf{I}_k , \mathbf{p}_k , $\boldsymbol{\theta}_k$, and the reference route.
2. $\mathbf{e}_k^\phi = (e^\phi(\boldsymbol{\theta}_k, \mathbf{r}(\mathbf{p}_k)), \dots, e^\phi(\boldsymbol{\theta}_k, \mathbf{r}(\mathbf{p}_k) + 8\Delta\mathbf{r}))$: route heading errors, defined as the difference between the current robot heading $\boldsymbol{\theta}_k$ and the tangent to the route at each point over the visible portion of the reference route less than 2m from the robot, averaged over $\Delta\mathbf{r} = 0.2$ m intervals.
3. $\boldsymbol{\kappa}_k = (\kappa(\mathbf{I}_k, \mathbf{r}(\mathbf{p}_k)), \dots, \kappa(\mathbf{I}_k, \mathbf{r}(\mathbf{p}_k) + 8\Delta\mathbf{r}))$: the visible route curvatures, defined as the signed curvature of the route over the visible portion of the reference route less than 2m from the robot, averaged over $\Delta\mathbf{r} = 0.2$ m intervals.
4. $\mathbf{e}_k^\psi = (e^\psi(\mathbf{I}_k, \mathbf{r}(\mathbf{p}_k)), \dots, e^\psi(\mathbf{I}_k, \mathbf{r}(\mathbf{p}_k) + 8\Delta\mathbf{r}))$: route camera angles, defined as the angle between the center of the camera frame and the visible portion of the reference route less than 2m from the robot, averaged over $\Delta\mathbf{r} = 0.2$ m intervals. This is included because several subject reported after trials that, rather than trying to center the robot body on the route, they focused on centering the route in the camera view.

5. $\mathbf{e}_k^{ps} = (e^{ps}(\mathbf{p}_k, \mathbf{r}(\mathbf{p}_k)), \dots, e^{ps}(\mathbf{p}_k, \mathbf{r}(\mathbf{p}_k + 10\Delta\mathbf{p})))$: predicted off-track error for the next 2m assuming the robot continues along the current heading θ_k , averaged over $\Delta\mathbf{p} = 0.2\text{m}$ intervals. Defined as the Euclidean distance between the center of the robot and the point on the reference route nearest the predicted robot position.
6. $\mathbf{e}_k^{\theta s} = (e^{\theta s}(\theta_k, \mathbf{r}(\mathbf{p}_k)), \dots, e^{\theta s}(\theta_k, \mathbf{r}(\mathbf{p}_k + 10\Delta\mathbf{p})))$: predicted heading error for the next 2m assuming the robot continues along the current heading θ_k , averaged over $\Delta\mathbf{p} = 0.2\text{m}$ intervals. Defined as the difference between θ_k and the tangent line to the route at the point nearest the predicted robot position.
7. $\mathbf{e}_k^{pt} = (e^{pt}(\mathbf{p}_k, \mathbf{r}(\mathbf{p}_k)), \dots, e^{pt}(\mathbf{p}_k, \mathbf{r}(\mathbf{p}_k + 10\Delta\mathbf{p})))$: predicted off-track error for the next 2m assuming the robot continues to rotate at the last commanded rate \mathbf{u}_{k-1} , averaged over $\Delta\mathbf{p} = 0.2\text{m}$ intervals. Defined as the Euclidean distance between the center of the robot and the point on the reference route nearest the predicted robot position.
8. $\mathbf{e}_k^{\theta t} = (e^{\theta t}(\theta_k, \mathbf{r}(\mathbf{p}_k)), \dots, e^{\theta t}(\theta_k, \mathbf{r}(\mathbf{p}_k + 10\Delta\mathbf{p})))$: predicted heading error for the next 2m assuming the robot continues to rotate at the last commanded rate \mathbf{u}_{k-1} , averaged over $\Delta\mathbf{p} = 0.2\text{m}$ intervals. Defined as the difference between the predicted robot heading and the tangent line to the route at the point nearest the predicted robot position.

Items 5-8 are a set of features based on predicting robot motion over a short horizon into the future; this approach has worked well for model predictive control (MPC) [57]. These predicted features assume that the subjects maintain a working memory of the recently observed route that is sufficient to allow determination of position and heading errors over the 0.4m blind spot directly in front of the robot. Note that many of the above features are highly correlated, and determining which,

if any, are relevant is a key element of the proposed modeling approach. Figure 3.5 shows schematics for \mathbf{e}_k^ϕ , \mathbf{e}_k^ψ , \mathbf{e}_k^{ps} , \mathbf{e}_k^{pt} , $\mathbf{e}_k^{\theta s}$, and $\mathbf{e}_k^{\theta t}$.

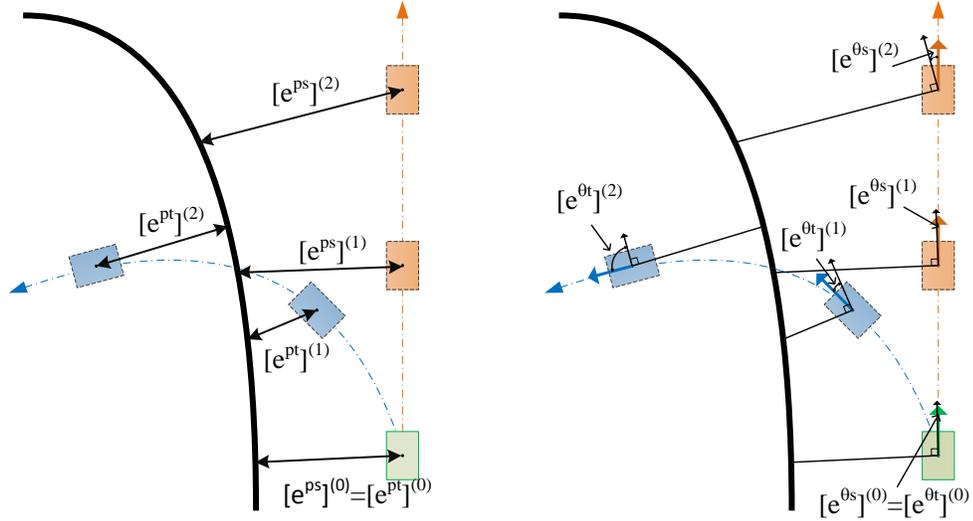
The feature vector at timestep \mathbf{k} is defined to be

$$\begin{aligned} \mathbf{x}_k &= (\boldsymbol{\eta}_k, \mathbf{e}_k^\phi, \boldsymbol{\kappa}_k, \mathbf{e}_k^\psi, \mathbf{e}_k^{ps}, \mathbf{e}_k^{\theta s}, \mathbf{e}_k^{pt}, \mathbf{e}_k^{\theta t})^T \\ &= (\mathbf{x}_k^{(1)}, \mathbf{x}_k^{(2)}, \dots, \mathbf{x}_k^{(m)})^T, \end{aligned} \quad (3.1)$$

where m indicates the number of features. The full data set from $\mathbf{k} = \mathbf{0}$ to $\mathbf{k} = \mathbf{K}$ can then be written as $\mathbf{X}_K = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_K)$. If the selected features form a sufficiently rich representation that the control \mathbf{u}_k is conditionally independent given the feature values, the distribution of \mathbf{u}_k can be expressed as $\mathbf{p}(\mathbf{u}_k | \mathbf{x}_k)$. Finding a model for the input/output behavior of the human operators then consists of defining a form for $\mathbf{p}(\mathbf{u}_k | \mathbf{x}_k)$ and choosing an algorithm to estimate all associated parameters.

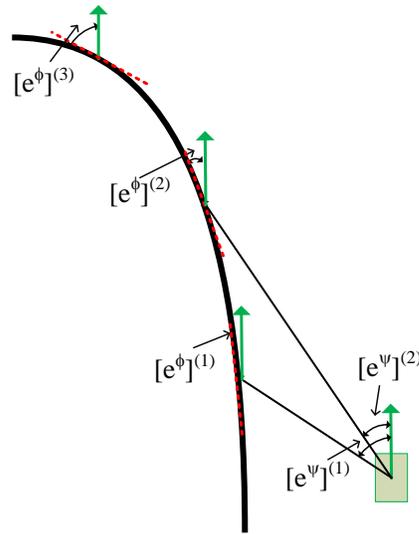
3.3.2 Overview of Models and Algorithms

The remainder of this section summarizes the details of two popular statistical modeling approaches. The first model considered, termed ‘least absolute shrinkage and selection operator’ (LASSO), predicts the continuous control response of the human subjects using a linear combination of feature values, with a constraint on the absolute sum of the weighting parameters. LASSO models for all values of the weight constraint are fit to the data using the Least Angle Regression (LARS) algorithm. The second model used, termed ‘sparse multinomial logistic regression’ (SMLR), finds the probability of a discretized control response using a normalized exponential of a linear combination of features, with one weighting vector for each discrete choice. As in LASSO, there is again a constraint on the absolute sum of the weighting parameters. SMLR models are fit to the data at specified values of



(a) Position Errors \mathbf{e}^{ps} and \mathbf{e}^{pt}

(b) Heading Errors $\mathbf{e}^{\theta s}$ and $\mathbf{e}^{\theta t}$



(c) Route Features \mathbf{e}^{ϕ} and \mathbf{e}^{ψ}

Figure 3.5: Schematics for defining the features which depend upon the robot position and heading. The reference route is shown in black. (a) Components of the predicted position errors \mathbf{e}^{ps} and \mathbf{e}^{pt} . (b) Components of the predicted heading errors $\mathbf{e}^{\theta s}$ and $\mathbf{e}^{\theta t}$. (c) Components of the route heading errors \mathbf{e}^{ϕ} and route camera angles \mathbf{e}^{ψ} .

the constraint using an iterative algorithm that is also commonly called SMLR. The purpose of examining the results of multiple modeling approaches is not to suggest that one is superior to the other. Rather, the goal of this analysis is to show that the ‘importance’ metric detailed in Section 3.5 generates results which are consistent between models.

3.3.3 Model #1: LASSO with LARS

The LASSO approach attempts to find a predictive model of the human responses where the control action of the human subjects is a linear combination of the observed feature values. The exact form of this model is derived using a probabilistic argument as follows. Assume that \mathbf{u}_k is linearly dependent on the feature vector \mathbf{x}_k ,

$$\mathbf{u}_k = \sum_{j=1}^m \mathbf{w}^{(j)} \mathbf{x}_k^{(j)} + \epsilon_k = \mathbf{w}^T \mathbf{x}_k + \epsilon_k \quad (3.2)$$

where $\epsilon_k \sim \mathcal{N}(\mathbf{0}, \beta^{-1})$ is an error modeled as zero-mean Gaussian noise with precision β , \mathbf{w} is a column vector of model parameters, and the m^{th} feature at timestep k is defined to be $\mathbf{x}_k^{(m)} = \mathbf{1}$ in order to include a bias term in the parameter vector. The conditional distribution of \mathbf{u}_k given the data \mathbf{x}_k and the model parameter vector \mathbf{w} is then written as

$$p(\mathbf{u}_k | \mathbf{x}_k, \mathbf{w}, \beta) = \mathcal{N}(\mathbf{u}_k | \mathbf{w}^T \mathbf{x}_k, \beta^{-1}) \quad (3.3)$$

Let the total control history be defined as $\mathbf{u}_K = (\mathbf{u}_1, \dots, \mathbf{u}_K)$. Assuming the data to be identically distributed and conditionally independent given the features, the conditional distribution for \mathbf{u}_K is given as

$$p(\mathbf{u}_K | \mathbf{X}_K, \mathbf{w}, \beta) = \prod_{k=0}^K \mathcal{N}(\mathbf{u}_k | \mathbf{w}^T \mathbf{x}_k, \beta^{-1}) \quad (3.4)$$

The maximum likelihood estimate for the model parameters \mathbf{w}_{ML} can be found by

maximizing the log likelihood function,

$$\begin{aligned} \ell(\mathbf{w}, \beta) &= \ln p(\mathbf{u}_K | \mathbf{X}_K, \mathbf{w}, \beta) \\ &= \frac{K}{2} \ln \beta - \frac{K}{2} \ln(2\pi) - \frac{\beta}{2} \sum_{k=0}^K (\mathbf{u}_k - \mathbf{w}^T \mathbf{x}_k)^2 \end{aligned} \quad (3.5)$$

Given that the best estimate for \mathbf{u}_k is based only on the peak of the distribution $p(\mathbf{u}_k | \mathbf{x}_k, \mathbf{w}, \beta)$, the value of the width parameter β will have no effect on the predicted control outputs and can be neglected. The maximization of Eqn. (3.5) is equivalent to the standard Ordinary Least Squares (OLS) solution for the parameter vector \mathbf{w} [55].

While Eqn. (3.4) serves as a good starting point in the modeling process, the OLS solution has an unfortunate characteristic that prevents it from being a good candidate for the purpose of this work. The OLS approach is prone to overfitting errors when presented with highly correlated feature sets, or when some features contain only noise [55]. Given that the features were deliberately chosen to maximize the chance of selecting the unknown ‘true’ features used by the subjects, the feature set is likely to contain highly noisy and redundant components; this is confirmed by the analysis results presented in Section 3.5. As a consequence of this overfitting, unique values for \mathbf{w} cannot be estimated, and the relative contribution of each feature to the regression accuracy is unobservable. The resulting high variance in parameter estimates also leads to difficulties in finding consistent trends between the parameters and feature ordering for different subjects, a key goal of this work.

The challenge of redundant model features is commonly addressed in the statistics and machine learning literature by using some form of ‘regularization’, where the values for the parameters are constrained by adding a mathematically convenient prior distribution on \mathbf{w} to the model [55]. Two of the most popular regularization strategies are to add a term to the cost in Eqn. (3.5) proportional to

either the \mathbf{L}_2 or \mathbf{L}_1 norm of the parameter vector; these are often termed ‘ridge regression’ and ‘the LASSO’ respectively. Ridge regression is equivalent to adding a Gaussian prior over \mathbf{w} ; LASSO corresponds to adding a Laplacian prior over \mathbf{w} . While ridge regression does serve to reduce overfitting errors by minimizing the parameters associated with uninformative features, the values of such parameters remain noticeably non-zero even as the \mathbf{L}_2 term grows large [58]. In order to compare the relative contributions of features across multiple subjects, the analysis in this work makes use of an ‘importance’ metric which relies on the order in which features are added to the model. Therefore the LASSO approach is used, as it is known to drive the weights for noisy or redundant feature parameters identically to zero without any hard thresholding required, as the prior distribution weight is increased [58].

Formally: a posterior distribution $p(\mathbf{u}_K, \mathbf{w} | \mathbf{X}_K, \beta)$ is constructed by multiplying the likelihood defined in Eqn. (3.4) by a Laplacian prior on \mathbf{w} of the form

$$p(\mathbf{w}) = \prod_{j=1}^m \frac{\lambda}{2} \exp\left(-\frac{\lambda}{2} |\mathbf{w}^{(j)}|\right), \quad (3.6)$$

where λ is a scaling constant. This produces a new log likelihood

$$\begin{aligned} L(\mathbf{w}, \beta) = \frac{K}{2} (\ln \beta - \ln(2\pi)) - \frac{\beta}{2} \sum_{k=0}^K (\mathbf{u}_k - \mathbf{w}^T \mathbf{x}_k)^2 \\ + m \ln\left(\frac{\lambda}{2}\right) - \frac{\lambda}{2} \sum_{j=1}^m |\mathbf{w}^{(j)}| \end{aligned} \quad (3.7)$$

Minimizing Eqn. (3.7) is equivalent to the constrained minimization

$$\mathbf{w}_{\text{LASSO}} = \underset{\mathbf{w}}{\text{argmin}} \sum_{k=0}^K (\mathbf{u}_k - \mathbf{w}^T \mathbf{x}_k)^2 \quad \text{s.t.} \quad \sum_{j=1}^m |\mathbf{w}^{(j)}| \leq t \quad (3.8)$$

where t is a constraint that varies inversely with λ [58].

The LASSO approach has several convenient properties with regards to the problem of interest. As $t \rightarrow \sum_j \mathbf{w}_{\text{OLS}}^{(j)}$, where \mathbf{w}_{OLS} is the OLS solution, the LASSO solution $\mathbf{w}_{\text{LASSO}}$ converges to that of the OLS solution and thereafter

remains constant, as \mathbf{t} ceases to constrain the elements of $\mathbf{w}_{\text{LASSO}}$. Conversely, as $\mathbf{t} \rightarrow \mathbf{0}$, the parameter values $\mathbf{w}^{(j)}$ decay to identically zero one at a time until $\mathbf{w}^{(j)} = \mathbf{0}$, $\forall j \in [1, \dots, m]$ at $\mathbf{t} = \mathbf{0}$. The varying of \mathbf{t} thus determines an *ordering* over the features, in terms of their importance in describing the data. Additionally, estimator variance and bias errors vary as \mathbf{t} changes, so that it is common for the accuracy of the predictor, measured by the size of the residual, to increase as uninformative features are removed from the model, then steadily worsen as useful features are removed [58].

Since first proposed by Tibshirani [58], several algorithms have been developed to efficiently solve the LASSO problem, including those in [59] and [60]. The modified least angle regression (LARS) algorithm presented in [61] is used here due to its useful computational properties and readily analyzable structure. Solutions to the LASSO problem, the parameter values $\mathbf{w}_{\text{LASSO}}^{(j)}$, are piecewise linear functions of the constraint \mathbf{t} . The set of these solutions for all values of \mathbf{t} can be interpreted as a trajectory in parameter space, the shape of which is dependent upon the data. The LARS algorithm uses an iterative approach to find the vertices of this trajectory; linear interpolation is then used to calculate parameter values for any value of \mathbf{t} . The LARS algorithm is outlined as follows (with a summary in Algorithm 3).

At iteration α , let $\mathbf{w}(\alpha)$ be the current parameter vector, $\mathcal{A}(\alpha)$ be the set of features whose weights are non-zero, $\hat{\mathbf{u}}(\alpha) = \mathbf{X}\mathbf{w}(\alpha)$ be the predicted control history given $\mathbf{w}(\alpha)$, and $\boldsymbol{\mu}(\alpha) = \mathbf{u} - \hat{\mathbf{u}}(\alpha)$ be the current residual vector. The correlations between the feature vectors and elements of the residual is then $\boldsymbol{\rho} = \mathbf{X}^T \boldsymbol{\mu}(\alpha)$. Note that the subscript has been dropped from \mathbf{X} and \mathbf{u} for the sake of clarity, and that for the remainder of this chapter they will represent the full data set through time $\mathbf{k} = \mathbf{K}$. These variables are initialized with $\mathbf{w}(\mathbf{0}) = \mathbf{0}$, $\hat{\mathbf{u}}(\mathbf{0}) =$

$\mathbf{0}$, $\boldsymbol{\mu}(\mathbf{0}) = \mathbf{u}$, $\boldsymbol{\rho} = \mathbf{X}^T \boldsymbol{\mu}(\mathbf{0})$, and $\mathcal{A} = \{\hat{j}\}$, where $\hat{j} = \mathbf{argmax}_{j \notin \mathcal{A}}(|\boldsymbol{\rho}^{(j)}|)$.

The objective at each iteration of the LARS algorithm is to determine the value of the parameters \mathbf{w} at the next vertex of the solution trajectory. These vertices occur either when a feature is found to be as correlated with the current residual as the active features, in which case it is added to the model, or when a feature is removed from the model. Parameter values are found by taking an appropriately sized step in a direction in parameter space that ensures that all active features remain equally correlated with the residual. At iteration $\boldsymbol{\alpha}$ of the algorithm the following steps are performed. Define

$$\mathbf{X}_{\mathcal{A}} = (\cdots, \text{sign}(\boldsymbol{\rho}^{(j)})(\mathbf{x}^{(j)})^T, \cdots)_{j \in \mathcal{A}}^T \quad (3.9)$$

$$\mathbf{w}_{\mathcal{A}} = (\mathbf{1}^T (\mathbf{X}_{\mathcal{A}} \mathbf{X}_{\mathcal{A}}^T)^{-1} \mathbf{1})^{-1/2} (\mathbf{X}_{\mathcal{A}} \mathbf{X}_{\mathcal{A}}^T)^{-1} \mathbf{1} \quad (3.10)$$

where $\mathbf{x}^{(j)}$ is the j^{th} feature vector, ie the j^{th} row of \mathbf{X} . Let the direction in feature space that is equiangular between all active feature vectors be \mathbf{d} , the elements of which are $\mathbf{d}^{(j)} = \text{sign}(\boldsymbol{\rho}^{(j)}) \mathbf{w}_{\mathcal{A}}^{(j)}$ for $j \in \mathcal{A}$, and $\mathbf{d}^{(j)} = \mathbf{0}$ for $j \notin \mathcal{A}$. Taking a step along \mathbf{d} ensures that all active features are equally correlated with the residual at iteration $(\boldsymbol{\alpha} + 1)$. Let

$$\gamma = \min_{j \notin \mathcal{A}}^+ \left[\frac{P - \boldsymbol{\rho}^{(j)}}{b - \mathbf{a}^{(j)}}, \frac{P + \boldsymbol{\rho}^{(j)}}{b + \mathbf{a}^{(j)}} \right] \quad (3.11)$$

$$\tilde{\gamma} = \min_{j \in \mathcal{A}} (-\mathbf{w}^{(j)} / \mathbf{d}^{(j)} > \mathbf{0}) \quad (3.12)$$

where $P = \max_j |\boldsymbol{\rho}^{(j)}|$, $\mathbf{a}^{(j)}$ is the j^{th} element of $\mathbf{a} = \mathbf{X} \mathbf{X}_{\mathcal{A}}^T \mathbf{w}_{\mathcal{A}}$, $b = (\mathbf{1}^T (\mathbf{X}_{\mathcal{A}}^T \mathbf{X}_{\mathcal{A}})^{-1} \mathbf{1})^{-1/2}$, and the $+$ indicates finding a minimum only over positive components. If $\tilde{\gamma} < \gamma$, remove $\hat{j} = \mathbf{argmin}_{j \in \mathcal{A}}(\tilde{\gamma})$ from \mathcal{A} and set $\gamma = \tilde{\gamma}$, as this indicates that the value of parameter $\mathbf{w}^{(\hat{j})}$ is zero, otherwise add $\hat{j} = \mathbf{argmax}_{j \notin \mathcal{A}}(\gamma)$ to \mathcal{A} . The new LASSO model and control history estimate

at iteration $(\boldsymbol{\alpha} + \mathbf{1})$ is

$$\mathbf{w}(\boldsymbol{\alpha} + \mathbf{1}) = \mathbf{w}(\boldsymbol{\alpha}) + \gamma \mathbf{d} \quad (3.13)$$

$$\hat{\mathbf{u}}(\boldsymbol{\alpha} + \mathbf{1}) = \hat{\mathbf{u}}(\boldsymbol{\alpha}) + \gamma \mathbf{X}_{\mathcal{A}}^T \mathbf{w}_{\mathcal{A}} \quad (3.14)$$

Algorithm 3: Least Angle Regression (LARS)

```

1 Set  $\mathbf{w}(\mathbf{0}) \leftarrow \mathbf{0}$ ,  $\hat{\mathbf{u}}(\mathbf{0}) \leftarrow \mathbf{0}$ ,  $\boldsymbol{\mu}(\mathbf{0}) \leftarrow \mathbf{u}$ ,  $\boldsymbol{\alpha} \leftarrow \mathbf{0}$ ;
2  $\boldsymbol{\rho} \leftarrow \mathbf{X}^T \boldsymbol{\mu}(\mathbf{0})$ ;
3  $\mathcal{A}(\mathbf{0}) \leftarrow \{\text{argmax}_j (|\boldsymbol{\rho}^{(j)}|)\}$ ;
4 while  $\text{size}(\mathcal{A}(\boldsymbol{\alpha})) < m$  do
5    $\boldsymbol{\mu}(\boldsymbol{\alpha}) \leftarrow \mathbf{u} - \hat{\mathbf{u}}(\boldsymbol{\alpha})$ ;
6    $\boldsymbol{\rho} \leftarrow \mathbf{X}^T \boldsymbol{\mu}(\boldsymbol{\alpha})$ ;
7    $P \leftarrow \max_{\rho^{(j)}} (|\rho^{(j)}|)$ ;
8    $\mathbf{X}_{\mathcal{A}} \leftarrow$  result of Eqn. (3.9);
9    $\mathbf{w}_{\mathcal{A}} \leftarrow$  result of Eqn. (3.10);
10  for  $j \leftarrow 1$  to  $m$  do
11    if  $j \in \mathcal{A}(\boldsymbol{\alpha})$ ,  $\mathbf{d}_j \leftarrow \text{sign}(\boldsymbol{\rho}^{(j)}) \mathbf{w}_{\mathcal{A}}^{(j)}$ ;
12    else,  $\mathbf{d}^{(j)} \leftarrow \mathbf{0}$ ;
13   $\gamma \leftarrow$  result of Eqn. (3.11);
14   $\tilde{\gamma} \leftarrow$  result of Eqn. (3.12);
15  if  $\tilde{\gamma} < \gamma$  then
16     $\gamma \leftarrow \tilde{\gamma}$ ;
17     $\mathcal{A}(\boldsymbol{\alpha} + \mathbf{1}) \leftarrow \mathcal{A}(\boldsymbol{\alpha}) \setminus \text{argmin}_{j \in \mathcal{A}(\boldsymbol{\alpha})} (\tilde{\gamma})$ ;
18  else
19     $\mathcal{A}(\boldsymbol{\alpha} + \mathbf{1}) \leftarrow \mathcal{A}(\boldsymbol{\alpha}) \cup \text{argmax}_{j \notin \mathcal{A}(\boldsymbol{\alpha})} (\tilde{\gamma})$ ;
20  update  $\mathbf{w}(\boldsymbol{\alpha} + \mathbf{1})$  and  $\hat{\mathbf{u}}(\boldsymbol{\alpha} + \mathbf{1})$  with Eqns. (3.13) and (3.14);
21   $\boldsymbol{\alpha} \leftarrow \boldsymbol{\alpha} + \mathbf{1}$ ;

```

Figure 3.6 plots trajectories characteristic of LASSO models for a subset of five model features, for one subject at no time delay. This example shows that parameter values can vary as the constraint t is increased by at least an order of magnitude, even after they are included in the model and become non-zero. Also note that one feature switches sign and is removed from the model from $t = 0.3 - 0.5$. Finally, any ordering of features based on magnitude of parameter size leads to a dependence on the choice of t , as the parameter trajectories for

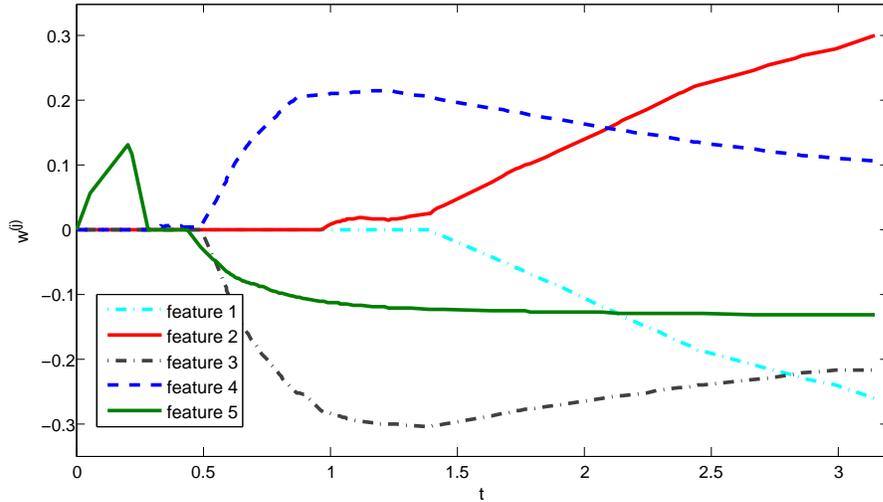


Figure 3.6: Example of 5 LASSO parameter trajectories for a random subset of the features for one subject at no time delay as a function of constraint t . When $t = 0$, the fully constrained case results in no features present in the model; when $t = 3.14$, the unconstrained OLS solution results in all features present in the model.

different features tend to cross. These variations make it impossible to use the parameter values at any given value of t as a metric for determining the overall contribution of a feature to the model. However, the order in which features are added to the model will motivate a definition of relative feature importance, described in Section 3.5.

3.3.4 Model #2: SMLR

The second approach to modeling the human controller uses a classification method, where the control \mathbf{u}_k is assumed to be drawn from one of n discrete classes defined by Gaussian class conditional densities. This approach is equivalent to partitioning the feature space into regions separated by hyper planes, with each region containing the feature values that are most likely to predict the associated discrete class of control output. The class label associated with feature vector

\mathbf{x}_k can be encoded using a ‘1-of- n ’ encoding so that $\mathbf{c}_k = [\mathbf{c}_k^{(1)}, \mathbf{c}_k^{(2)}, \dots, \mathbf{c}_k^{(n)}]^T$ where $\mathbf{c}^{(i)} = \mathbf{1}$ if and only if \mathbf{x}_k is drawn from class i , and k is the time index. Let $\mathbf{W} = [\mathbf{w}^{(1)}, \mathbf{w}^{(2)}, \dots, \mathbf{w}^{(n)}]$ be a model parameter matrix, where $\mathbf{w}^{(i)}$ is the vector of parameters associated with class i . As before, m is the number of features in \mathbf{x}_k , and $\mathbf{x}_k^{(m)} \equiv \mathbf{1}$ in order to include a bias term in the model parameters. Given this representation, the probability that \mathbf{x}_k belongs to class i is described by the multinomial logistic (also known as softmax) model,

$$P(\mathbf{c}_k^{(i)} = \mathbf{1} | \mathbf{x}_k, \mathbf{W}) = \frac{\exp((\mathbf{w}^{(i)})^T \mathbf{x}_k)}{\sum_{h=1}^n \exp((\mathbf{w}^{(h)})^T \mathbf{x}_k)} \quad (3.15)$$

The probability distribution over the full set of class identities $\mathbf{C} = (\mathbf{c}_1, \dots, \mathbf{c}_K)$ is then,

$$P(\mathbf{C} | \mathbf{X}, \mathbf{W}) = \prod_{k=0}^K \prod_{i=1}^n P(\mathbf{c}_k^{(i)} = \mathbf{1} | \mathbf{x}_k, \mathbf{W}) \quad (3.16)$$

Similar to the OLS approach to linear regression, a unique maximum likelihood solution for \mathbf{W} cannot be found if noisy or redundant features are present in the model [55]. As in the LASSO derivation earlier, the parameters associated with uninformative features can be driven to zero by the addition of an \mathbf{L}_1 penalty term. The likelihood model in Eqn. (3.16) is multiplied by a Laplacian prior over \mathbf{W} of the form

$$p(\mathbf{W}) \propto \exp(-\lambda \sum_{i=1}^n \sum_{j=1}^m |\mathbf{w}^{(i,j)}|), \quad (3.17)$$

where $\mathbf{w}^{(i,j)}$ is the j^{th} component of the i^{th} class vector $\mathbf{w}^{(i)}$. This sets the log likelihood to be

$$\begin{aligned} L(\mathbf{W}) = \sum_{k=0}^K \left[\sum_{i=1}^n \mathbf{c}_k^{(i)} (\mathbf{w}^{(i)})^T \mathbf{x}_k - \ln \sum_{i=1}^n \exp((\mathbf{w}^{(i)})^T \mathbf{x}_k) \right] \\ - \lambda \sum_{i=1}^n \sum_{j=1}^m |\mathbf{w}^{(i,j)}| \quad (3.18) \end{aligned}$$

Solutions for the parameter matrix \mathbf{W} are found using the iterative approach of the sparse multinomial logistic regression (SMLR) algorithm presented in [62]

and summarized here. The SMLR algorithm finds an estimate \mathbf{W}_{SMLR} maximizing the log likelihood in Eqn. (3.18), for some choice of the tuning parameter λ , by means of the bound optimization approach presented in [63]. To begin, define \mathbf{v} to be the vectorized form of \mathbf{W} , with $\mathbf{v}^{(l)} = \mathbf{w}^{(i,j)}$ for appropriate choices of the indices l , i , and j . The function $L(\mathbf{v}) = L(\mathbf{W})$ can be maximized by finding a surrogate function $Q(\mathbf{v}|\hat{\mathbf{v}}(\alpha))$ such that $[L(\mathbf{v}) - Q(\mathbf{v}|\hat{\mathbf{v}}(\alpha))]$ achieves a minimum at $\mathbf{v} = \hat{\mathbf{v}}(\alpha)$, where $\hat{\mathbf{v}}(\alpha)$ is the estimated parameter matrix at iteration α . Letting

$$\hat{\mathbf{v}}(\alpha + 1) = \underset{\mathbf{v}}{\text{argmax}} Q(\mathbf{v}|\hat{\mathbf{v}}(\alpha)) \quad (3.19)$$

and iterating over α is guaranteed to monotonically increase the original function $L(\mathbf{v})$. For $L(\mathbf{v})$ equivalent to Eqn. (3.18), a useful surrogate function is

$$Q(\mathbf{v}|\hat{\mathbf{v}}(\alpha)) = \mathbf{v}^T (\mathbf{g}(\hat{\mathbf{v}}(\alpha)) - \underline{\mathbf{H}}\hat{\mathbf{v}}(\alpha)) + \frac{1}{2}\mathbf{v}^T \underline{\mathbf{H}}\mathbf{v} - \frac{1}{2}\lambda \sum_{l=1}^{m \times n} \hat{\mathbf{v}}^{(l)}, \quad (3.20)$$

where $\underline{\mathbf{H}}$ is a negative definite matrix that lowerbounds the Hessian of $L(\mathbf{v})$, and $\mathbf{g}(\hat{\mathbf{v}}(\alpha))$ is the gradient of $L(\mathbf{v})$ evaluated at $\hat{\mathbf{v}}(\alpha)$. The definitions of $\underline{\mathbf{H}}$ and $\mathbf{g}(\mathbf{v})$ are

$$\underline{\mathbf{H}} = -\frac{1}{2} \left[\mathbf{I} - \frac{1}{m} \mathbf{1}\mathbf{1}^T \right] \otimes \sum_{k=0}^K \mathbf{x}_k \mathbf{x}_k^T \quad (3.21)$$

$$\mathbf{g}(\mathbf{v}) = \sum_{k=0}^K \left[(\mathbf{c}_k^T - \mathbf{p}_k(\mathbf{v})) \otimes \mathbf{x}_k \right] \quad (3.22)$$

where $\mathbf{p}_k(\mathbf{v}) = [\mathbf{p}_k^{(1)}(\mathbf{v}), \mathbf{p}_k^{(2)}(\mathbf{v}), \dots, \mathbf{p}_k^{(m-1)}(\mathbf{v})]^T$, $\mathbf{p}_k^{(i)}(\mathbf{v}) = P(\mathbf{c}_k^{(i)} = \mathbf{1}|\mathbf{x}_k, \mathbf{v})$, $\mathbf{1} = [\mathbf{1}, \mathbf{1}, \dots, \mathbf{1}]^T$, and \otimes is the Kroneker matrix product.

The surrogate function $Q(\mathbf{v}|\hat{\mathbf{v}}(\alpha))$ is maximized in a component-wise fashion by letting

$$\hat{\mathbf{v}}^{(l)}(\alpha + 1) = \text{soft} \left(\hat{\mathbf{v}}^{(l)}(\alpha) - \frac{\mathbf{g}^{(l)}(\hat{\mathbf{v}}(\alpha))}{\underline{\mathbf{H}}^{(ll)}}; \frac{-\lambda}{\underline{\mathbf{H}}^{(ll)}} \right), \quad (3.23)$$

where $\underline{\mathbf{H}}^{(l)}$ is the l^{th} diagonal component of $\underline{\mathbf{H}}$, $\mathbf{g}^{(l)}$ is the l^{th} component of the gradient in Eqn. (3.22), and $\text{soft}(\mathbf{a}; \delta) \equiv \text{sign}(\mathbf{a}) \cdot \max(\mathbf{0}, |\mathbf{a}| - \delta)$, $\forall(\mathbf{a}, \delta) \in \mathfrak{R}$. The estimate of $\hat{\mathbf{v}}(\boldsymbol{\alpha})$ is generated by cycling through every component l . Iteration continues until the log likelihood in Eqn. (3.18) converges to within a desired threshold Δ_L .

Algorithm 4: Sparse Multinomial Logistic Regression (SMLR)

```

1 begin
2   Set  $\underline{\mathbf{H}} \leftarrow$  Eqn. (3.21),  $\boldsymbol{\alpha} \leftarrow \mathbf{0}$ ,  $\hat{\mathbf{v}}(\boldsymbol{\alpha}) \leftarrow$  random init;
3   while  $\Delta_L \geq \Delta_{min}$  do
4      $\mathbf{g} \leftarrow$  result of Eqn. (3.22);
5     for  $l \leftarrow 1$  to  $n \times m$  do
6        $\hat{\mathbf{v}}^{(l)}(\boldsymbol{\alpha} + \mathbf{1}) \leftarrow$  result of Eqn. (3.23);
7        $\Delta_L \leftarrow L(\hat{\mathbf{v}}(\boldsymbol{\alpha} + \mathbf{1})) - L(\hat{\mathbf{v}}(\boldsymbol{\alpha}))$ ;
8        $\boldsymbol{\alpha} \leftarrow \boldsymbol{\alpha} + \mathbf{1}$ ;

```

3.4 Model Fitting Results

This section presents the results of fitting the proposed models detailed in Section 3.3 to the data generated by the experiments discussed in Section 3.2. The primary intent is to demonstrate that the SMLR and LARS algorithms *both* provide solutions that accurately predict the control response of the human subjects, and to provide a relative importance score for each of the model features. If the resulting feature scores of the two approaches are similar, it is reasonable to conclude that the set of features with high scores in *both* approaches is consistent with the features utilized by the subjects in controlling the robot. In the process of accomplishing this objective, the relationship between predictive power and model complexity is shown to be consistent with the hypothesis that only a subset of the features are used by the subjects. For the remainder of this work LASSO will refer

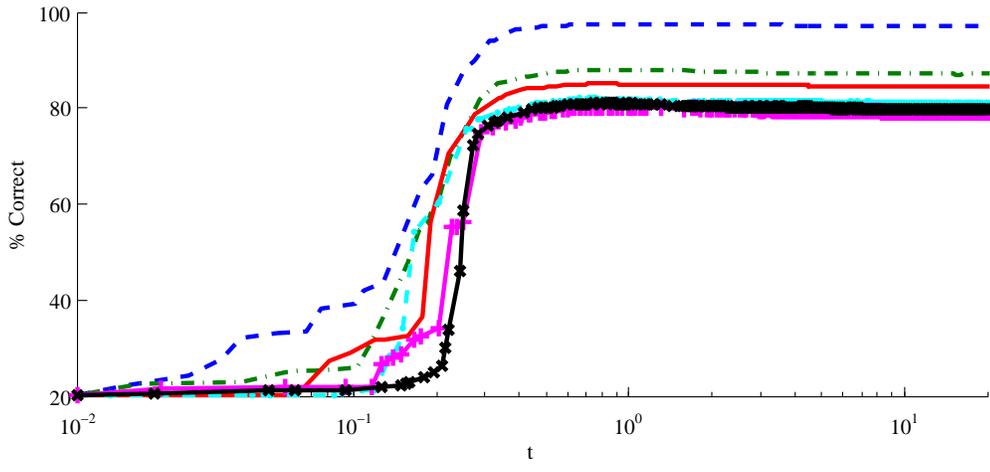
to regularized linear models with a Laplacian prior, and LARS will refer to the algorithm used to find the parameter values that best predict the experimental data. SMLR will indicate both regularized multinomial logistic models with Laplacian priors and the algorithm used to learn such models from the data.

The computational cost of the LARS algorithm is $O(m^3 + K \cdot m^2)$, where m is the total feature count and K is the length of the data set, and plus an $O(m^2)$ downdate for every iteration in which a feature is removed from the model [61]. While in principle there is no limit to the number of downdates that can occur, in practice none of the data sets required more than ten additional iterations. The cost of the SMLR algorithm is more variable, as the number of iterations required to minimize of the log likelihood function below a selected threshold is highly sensitive to the initialization. Each iteration requires $O(K \cdot m \cdot n)$ computations, with between **500** – **2000** iterations per data set for the choice of K , m , n , and likelihood threshold used in this work [62]. The computational burden of the SMLR algorithm is exacerbated by the need to fit a new model for every desired value of λ/N , while the LARS algorithm finds all possible solutions at once.

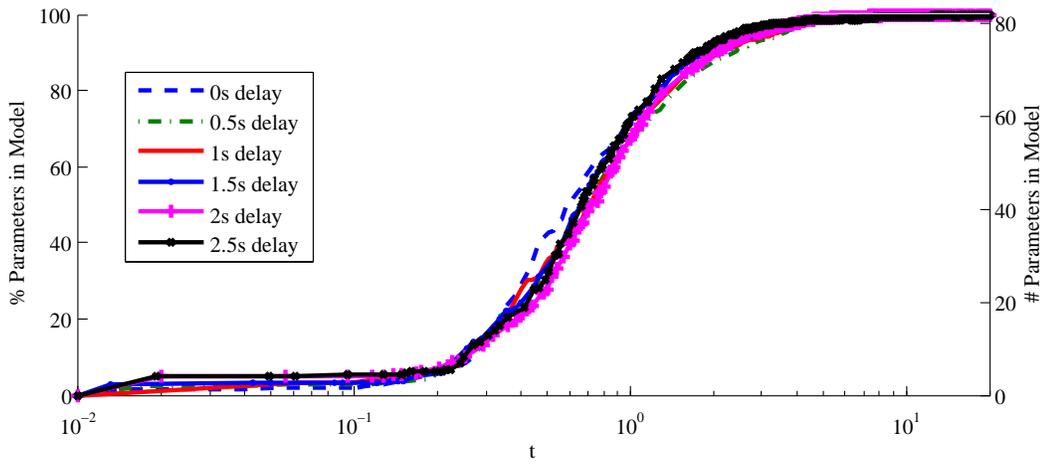
3.4.1 Algorithm Fitting Performance

In order to learn SMLR models, the control values \mathbf{u}_k were binned into $n = 15$ discrete classes. These classes correspond to sequential angle ranges of the joystick used as an input device. The values of the control sequence, $\hat{\mathbf{u}}_k$, predicted by LASSO models are also segmented into the same bins. This additional binning is done so that a single scoring metric may be used to evaluate and compare both modeling approaches.

The LASSO and SMLR models were both fit to the data as described in Section 3.3 and evaluated as follows. The features \mathbf{X} and associated control histories \mathbf{u} and



(a) LASSO Performance



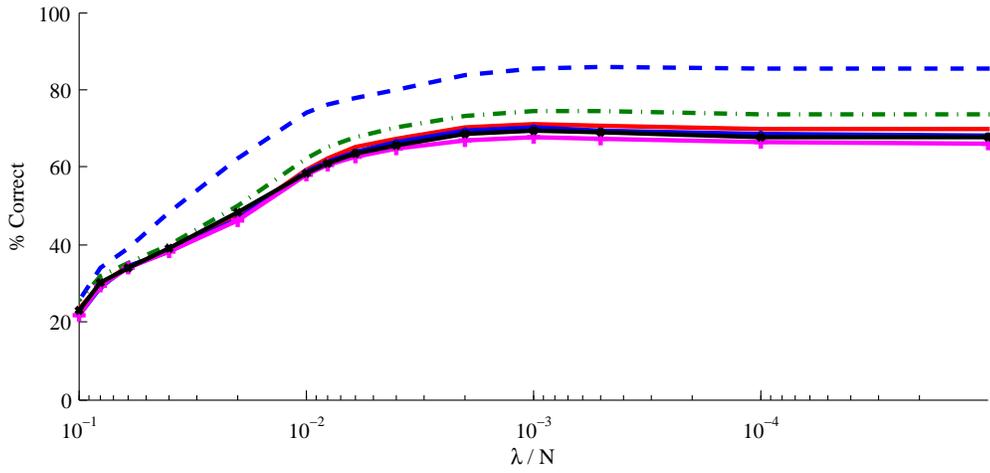
(b) LASSO Parameters

Figure 3.7: Average performance of LASSO models, and the percentage of non-zero parameters in the model as a function of the constraint $\sum |\mathbf{w}^{(j)}| \leq t$. Standard deviations, not shown for the sake of clarity, varied between 10-20% for performance and 6-12% for parameters.

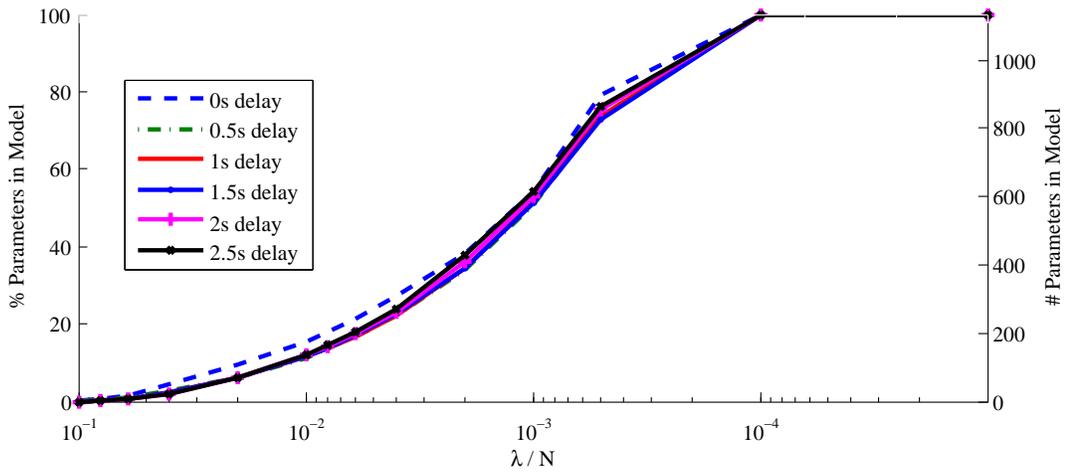
class labels \mathbf{C} for each lap of the route (of a total of 3) were combined into a single data set for each time delay, for each subject (of a total of 20). This resulted in 120 data sets, the length (\mathbf{K}) of which varied based on the amount of time required for the subject to complete all three laps at a given time delay. Data points belonging to each of the n discrete control classes were sampled randomly from each data set in order to ensure that the models were trained on an equal number of examples from every class. This resulted in final data sets with between 1200-1500 data points, or about 80-100 examples of each control class. The LASSO and SMLR models were then fit and evaluated using 5-fold cross-validation, with both models using the same data folds.

Training of LASSO models was performed once for each fold, giving solutions for the vertices of the parameter trajectories, as described in Section 3.3.3. Linear interpolation was used to generate exact solutions for intermediate values of \mathbf{t} , as the parameter trajectories are piecewise linear. One consequence of the class discretization required for comparison of LASSO and SMLR results is that if a control value is close to a class boundary, even a good LASSO prediction may be placed into the adjacent class. For the sake of compactness, rather than plotting both correct labels and a deviation, the performance score used in this work is defined to be the percentage of class labels in the validation identified as either the correct class or an adjacent class. In order to compare results between subjects, the performance is evaluated at approximately **2000** points between $\mathbf{t} = \mathbf{0}$ and $\mathbf{t} = \mathbf{t}_{OLS}$, and averaged over the 5 folds of cross-validation. This gives a score for each subject, at each time delay, as a function of the constraint \mathbf{t} . The mean performance as a function of constraint \mathbf{t} across all subjects is shown in Figure 3.7, as well as the percentage of non-Zero parameters at each value of \mathbf{t} .

The LASSO models have a peak performance (Figure 3.7 top) of between 81%



(a) SMLR Performance



(b) SMLR Parameters

Figure 3.8: Average performance of SMLR models, and the percentage of non-zero parameters in the model as a function of the constraint $\sum |\mathbf{w}^{(j)}| \leq t$. Standard deviations, not shown for the sake of clarity, varied between 5-15% for performance and 1-10% for parameters.

and 98%, depending on time delay, when averaged over all subjects. For all time delays this occurs when the percentage of features included in the model (Figure 3.7 bottom) is $\sim 25\%$. The sharp change in performance at $t \sim 0.3$ indicates the remaining features can be removed from the model with little effect on performance. The consistency between subjects of feature membership in this subset is examined in the next section. The fact that peak performance does not occur at the unconstrained end of the plot is a result of tradeoffs between classifier variance and bias errors, and is characteristic of most regularization methods [58].

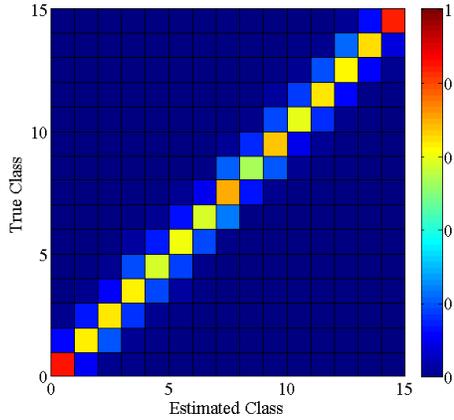
A total 15 values of λ/N were chosen for evaluation of the SMLR models, because of the high computational load of the SMLR algorithm. These values are spread over the full range between fully constrained ($\lambda = \mathbf{inf}$) and unconstrained ($\lambda = \mathbf{0}$) models. Statistics across folds and between subjects were generated in the same manner as described for the LASSO results and are shown in Figure 3.8. The SMLR models have a peak performance (Figure 3.8 top) of between 68% and 86%, depending on time delay, when averaged over all subjects. For all time delays this occurs when the percentage of non-zero parameters included in the model (Figure 3.8 bottom) is $\sim 55\%$.

Another approach to visualizing the performance of the LARS and SMLR models is to plot the pattern of classification errors across all subjects. For a given value of the constraint t , a confusion matrix is generated for each time delay by plotting, for each class i_1 , the percentage of validation examples classified into class i_2 for $i_1, i_2 \in [1, \dots, 15]$,

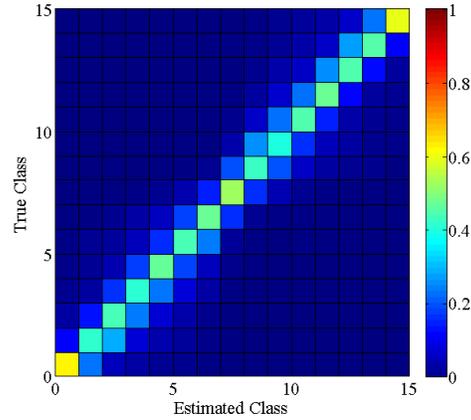
$$\text{CON}(i_1, i_2) = \frac{\sum_s \xi_s(i_1, i_2)}{\sum_j \sum_s \xi_s(i_1, i_2)} \quad (3.24)$$

$$\xi_s(i_1, i_2) = \sum_k c_{s, \text{TRUE}}^{(i_1)}(\mathbf{k}) \cdot c_{s, \text{PRED}}^{(i_2)}(\mathbf{k}) \quad (3.25)$$

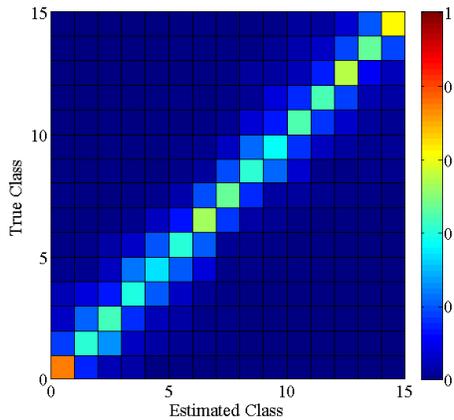
where $\mathbf{c}_{s, \text{TRUE}}(\mathbf{k})$ is the class label vector for data set \mathbf{s} at time \mathbf{k} , and $\mathbf{c}_{s, \text{PRED}}(\mathbf{k})$ is the class label vector predicted by either LASSO or SMLR models. As in Section



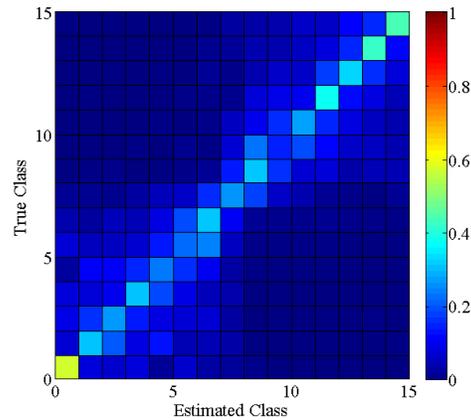
(a) LASSO at 0s delay



(b) LASSO at 2.5s delay



(c) SMLR at 0s delay



(d) SMLR at 2.5s delay

Figure 3.9: Confusion Matrices for the LASSO and SMLR models at 0s and 2.5s of time delay. Elements were summed across all subjects at the value of t which gave best average performance, then normalized to lie in the range $[0,1]$. By way of comparison, a perfect predictor would have only a diagonal of ones in its confusion matrix.

3.3.4, class labels use the ‘1-of-n’ encoding: $\mathbf{c}_s^{(i)} = \mathbf{1}$ if and only if \mathbf{u}_k is an example of class i . Confusion matrices for 0s and 2.5s of delay are shown in Figure 3.9, as these results bound the error values for the other delay values. Intuitively, perfect classification results in an identity matrix; consequently, a strong diagonal element reflects good performance. A qualitative examination of the confusion matrices can provide meaningful insight into the types of classification errors made by the two models. For example, the diffusion of weight in the upper right and lower left quadrants in Figure 3.9d shows that the SMLR models have difficulty with distinguishing between similar control actions, but do not predict turns of the wrong direction.

The difference in performance between LASSO and SMLR models of $\sim 10\%$ is likely due to the fact that the SMLR algorithm must determine the location and orientation of each hyperplane segmenting the data into classes individually with no knowledge of the underlying structure of the problem. In contrast, binning the output $\hat{\mathbf{u}}$ of LASSO models implicitly imposes the known condition that the classes are intrinsically ordered. Consequently, the SMLR algorithm requires more data than is available in this study in order to fit a model as accurately as the LARS algorithm does.

Neither the number of nonzero parameters in the SMLR and LASSO models, nor their values, can be directly compared. The parameters for SMLR models indicate the contribution of features to each class definition independently, while those for LASSO models are overall weights on the features themselves. A method for feature comparison that does not depend on the parameter values directly, and can thus be successfully applied to both models, is proposed in the next section.

3.5 Model Feature Importance

The primary aims of this section are to develop a method for determining the relative ‘importance’ of a set of features to models produced by the LARS and SMLR algorithms, and to use this method to analyze the experimental data described in Section 3.2 in order to understand how humans anticipate control changes in response to time delays. Further, if the importance analysis of both the LASSO and SMLR models yields common patterns, then these patterns can yield important insight into the true input/output function of human operators. The ‘importance’ of a feature is defined in this work to be the fraction of the predictive power of a model that can be directly attributed to the inclusion of the feature. To begin, Section 3.5.1 presents a novel procedure for calculating a measure of the relative ‘importance’ of a given feature to a particular model. Section 3.5.2 examines how the ‘importance’ of groups of related features changes as a function of time delay.

3.5.1 Measuring Importance in LASSO and SMLR

This section presents a novel ‘importance’ scoring metric which allows a comparison of the relative degree to which subjects make use of features at each time delay in both the LASSO and SMLR models. This goal is in contrast to the traditional model selection approach commonly used with LASSO, as the primary aim is to examine the manner in which subjects directed their attention during the control task, as opposed to finding a minimal model for a given level of performance. This metric is inspired by the correlation structure underlying the LARS algorithm described in Section 3.3.3, but can be generalized to the more complex SMLR models.

Recall from Eqns. (3.11) and (3.13) that the LARS algorithm adds features to the active set \mathcal{A} , and adjusts the parameters at each iteration such that the active

features in the set are equally correlated with the current residual and all inactive features are less correlated. Thus, over one iteration of LARS, any increase in classification performance can be equally attributed to all active features. A score can then be generated for each feature by the following approach. Let $\tau_\alpha \in [0, 1]$ be the performance of the LASSO model produced by LARS at iteration α , \mathcal{A}_α be the set of parameters whose values are nonzero at iteration α , n_α be the size of \mathcal{A}_α , and $b_\alpha^{(j)} = 1 \forall j \in \mathcal{A}_\alpha$. The importance score for feature j is then given as

$$\mathcal{I}_{\text{LASSO}}^{(j)} = \frac{\sum_{\alpha} b_\alpha^{(j)} \max(0, \tau_\alpha - \tau_{(\alpha-1)}) / n_\alpha}{\mathbf{max}_{\alpha} \tau_\alpha} \quad (3.26)$$

where the normalization by the maximum classification rate, $\mathbf{max}_{\alpha} \tau_\alpha$, is taken in order to allow meaningful comparison between subjects with varying performance. Scores are averaged over the 5 cross-validation folds to get a single estimate of the importance of each feature, for each subject. Feature importance scores can range between 0 and 1, and must sum to 1. An importance score for a feature of 1 occurs only if all of the performance can be ascribed to that feature alone, in which case all other features would have scores of 0. A score of 0 occurs whenever the inclusion of a feature provides no improvement in performance, and generally indicates that the feature is either a linear combination of other features already in the model, or is entirely uncorrelated with the output.

If a feature with a nonzero score is removed from the set feature set being considered, the scores of all remaining features can be expected to increase in proportion to their original score, so that the sum remains 1. The ratios of feature importance scores are stable when low-importance features are removed from the model. However, if a high-scoring feature is removed from the model, the ratios of other feature scores will only be constant if all remaining features are orthogonal to the removed feature. If a low-scoring feature still present in the model is highly

correlated with the removed high-scoring feature, its score may be dramatically increased. This effect is most clearly seen in the following example. Let \mathbf{x}_1 and \mathbf{x}_2 be two features such that $\mathbf{x}_1 = 2\mathbf{x}_2 + \epsilon$, where ϵ is a small random noise. If by chance \mathbf{x}_1 is slightly more correlated with the output than \mathbf{x}_2 , both the SMLR and LARS algorithms will assign zero weights to \mathbf{x}_2 at all iterations. As a result, \mathbf{x}_2 will have contributed nothing to the model and will be given a feature score of 0. If \mathbf{x}_1 is removed from the feature set, \mathbf{x}_2 will be added to the model in its place, will have essentially the same contribution to the performance as \mathbf{x}_1 did, and will end up with the same importance score previously obtained by \mathbf{x}_1 . Consequently, while the proposed importance metric indicates which features provide the best predictive power, it does not imply that there is no subset of features which could have performed *almost* as well as those selected.

Although it is based on the correlation structure of the LARS algorithm, importance scores can be generated for any iterative algorithm that makes use of an active set of features that is updated at each step. For example, while SMLR models contain many more parameters than LASSO models, a performance score can be constructed for each feature by increasing the importance score at iteration α for all features which have any nonzero parameters.

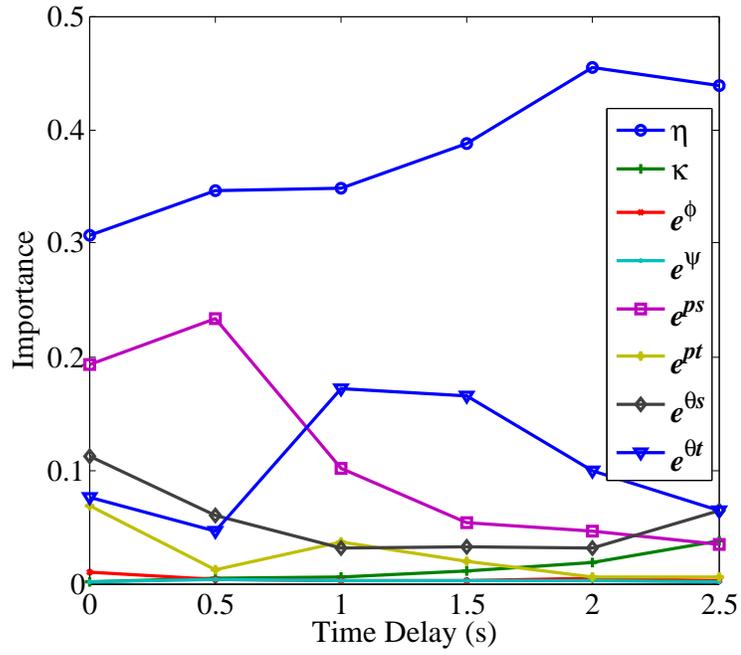
$$\mathcal{I}_{\text{SMLR}}^{(j)} = \frac{\sum_{\alpha} \left[\sum_{i=1}^n \mathbf{b}_{\alpha}^{(i,j)} \max(0, \tau_{\alpha} - \tau_{(\alpha-1)}) / n_{\alpha} \right]}{\max_{\alpha} \tau_{\alpha}} \quad (3.27)$$

where $\mathbf{b}_{\alpha}^{(i,j)} = \mathbf{1} \forall (i, j) \in \mathcal{A}_{\alpha}$ indicates that the parameter $\mathbf{w}^{(i,j)}$ associated with class i and feature j is present in the model at iteration α , and α indicates sequential values of λ/N for which a SMLR model is learned. Due to the restriction to only 15 values of λ/N because of computational limits, the importance results for the SMLR models are more uncertain than those for LASSO.

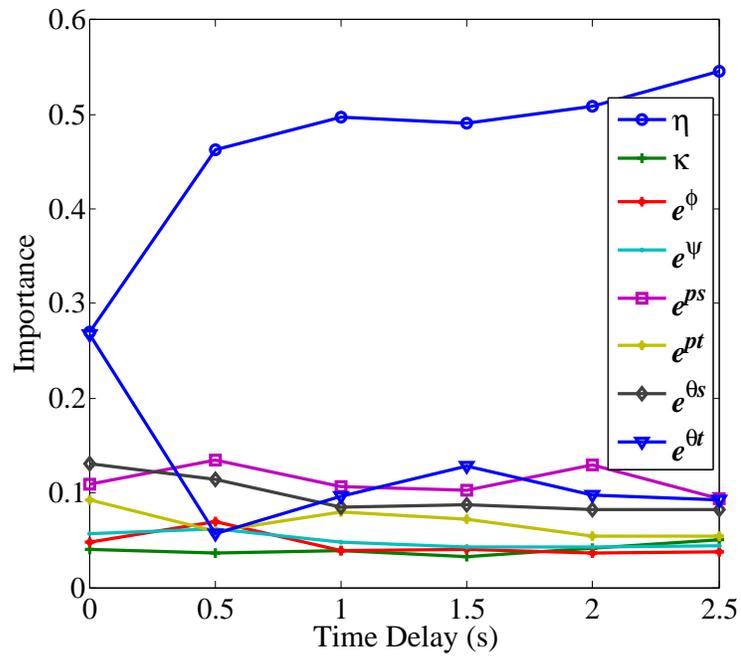
3.5.2 Relative Importance of Feature Groups

As described in Section 3.3, features in the data set are grouped based on type and a measure of temporal or spatial extent. Importance scores are averaged between subjects for each feature at each time delay, which allows for a deeper analysis of the two models. More specifically, insight can be attained by examining the relative importance of feature groups as function of delay, as well as the relative importance of features within each group. Figure 3.10 plots importance scores for feature groups as a function of time delay, found by summing over the importance scores for the features in each group. From these results it is apparent that both LASSO and SMLR models agree that the past control history $\boldsymbol{\eta}$ is the most significant feature group. The two models also agree that the predicted errors \mathbf{e}^{ps} , \mathbf{e}^{pt} , $\mathbf{e}^{\theta s}$, and $\mathbf{e}^{\theta t}$ comprise the next four most significant groups at low delays, and that the route curvature $\boldsymbol{\kappa}$ increases in importance at 2s of delay and above. The models disagree at 0.5s delay, where SMLR places \mathbf{e}^{ϕ} in the top four, and at 2-2.5s, where LASSO drops \mathbf{e}^{pt} towards zero.

Insight into the meaning of the large importance score attached to the past control history group $\boldsymbol{\eta}$ can be gained by looking at how the scores of the component features are distributed. In both the LASSO and SMLR models, more than 90% of the group score comes from the first element $\boldsymbol{\eta}^{(1)}$, which corresponds to the average control over the last 0.2s. This suggests that conditioning the model on past control outputs, $\mathbf{p}(\mathbf{u}_k | \mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_{k-1})$, simply captures a smoothness in the control history; this conclusion is not surprising given that subjects generally tried to avoid rapid motions of the joystick, as these would result in excessive oscillations. This pattern does not suggest that subjects make direct use of their past control actions other than the most recent one. If past memory plays a role in the decision making of human subjects, then clearly it must be captured in some other



(a) LASSO



(b) SMLR

Figure 3.10: Relative Importance of feature groups for both LASSO and SMLR models. Standard deviations are omitted for clarity but range between **0.003** – **0.17** for LASSO, and between **0.02** – **0.19** for SMLR.

way. However, the high performance of both models seen in Figures 3.7 and 3.8 suggests that any other use of past memory by the subjects must play a relatively small role, if any.

Support for the *anticipation* hypothesis discussed in Section 3.1 can be gained by considering a breakdown of scores for the rest of the features that have significant ‘importance’ across all of the time delays. Figure 3.11 plots the importance scores for components of the predicted errors \mathbf{e}^{ps} , \mathbf{e}^{pt} , $\mathbf{e}^{\theta s}$, and $\mathbf{e}^{\theta t}$ as functions of time delay for both LASSO and SMLR models. Each component corresponds to a different look-ahead distance over which the robot’s trajectory is predicted. As the LASSO and SMLR models assign different scores to each of the feature groups, the relative changes in magnitude within a group as a time delay varies are more important for this discussion than the absolute height of the peaks.

Figures 3.11a and 3.11b plot the scores for the predicted position errors assuming the robot continues in a straight line, and show a fairly linear relationship between look-ahead distance and time delay. The peak score at no delay is 0.6m and increases to 1.6m at 2.5s delay for LASSO, and to 1.2m at 2.5s delay for SMLR. Figures 3.11c and 3.11d plot the scores for the predicted heading errors assuming the robot continues in a straight line, and show very little dependence on time delay. Figures 3.11e and 3.11f plot the scores for the predicted position errors assuming the robot turns at its current rate, and show the roughly the same dependence of peak score on time delay as in 3.11a and 3.11b, except that the peak changes from 0.4m to 1.2m for LASSO and between 0.2m and 1m for SMLR. Figures 3.11g and 3.11h plot the scores for the predicted heading errors assuming the robot turns at its current rate, and show a second peak around 1.2m at no time delay in addition to the main peak that shifts from 0.2m to 0.8m for both models. A key feature of Figure 3.11 is that the scores for \mathbf{e}^{ps} , \mathbf{e}^{pt} , and $\mathbf{e}^{\theta t}$ all show

the same characteristics of a roughly linear increase in importance at distance as a function of time delay.

The SMLR plots in Figure 3.11 for all four feature groups have a flatter distribution of importance scores at all time delays when compared with the LASSO plots. This can be attributed to the coarser method of evaluating the SMLR models. As SMLR models were only fit at 15 different values of λ/N , parameters were added to the model in blocks, rather than in the one at a time pattern of LARS. This leads to more important features that would have entered the model earlier having reduced scores, while less important features have increased scores.

The feature groups in Figure 3.11 show a marked pattern of increasing importance further into the future as time delay increases, except for e^{pt} , which is generally the lowest scoring of the four. Averaged between the remaining 3 feature groups, the mean of the score distribution shifted $0.73 \pm 0.15m$ between 0s and 2.5s of time delay. This transference of ‘importance’ to further look-ahead distances supports the hypothesis that human subjects compensate for increasing amounts of delay by increasing a prediction horizon in order to *anticipate* future control needs. Even at no delay the importance scores for e^{ps} and $e^{\theta s}$ peak at 0.6m from the robot center. This distance is well inside the camera field of view and suggests that subjects make use some form of this *anticipation* even when no delay is present. An interpretation of the the feature e^{pt} , associated with predicting robot motion forward at the current heading, is that subjects make use of the off-track error of an ‘aim point’ in front of the robot, at a distance which increases with added time delay. This view is consistent with the results presented in [64], which found that driver behavior could be well predicted using such an input in the case of undelayed control. The importance scores for e^{pt} seen in Figures 3.11a and 3.11b suggest that subjects make use of such aiming points, and that the

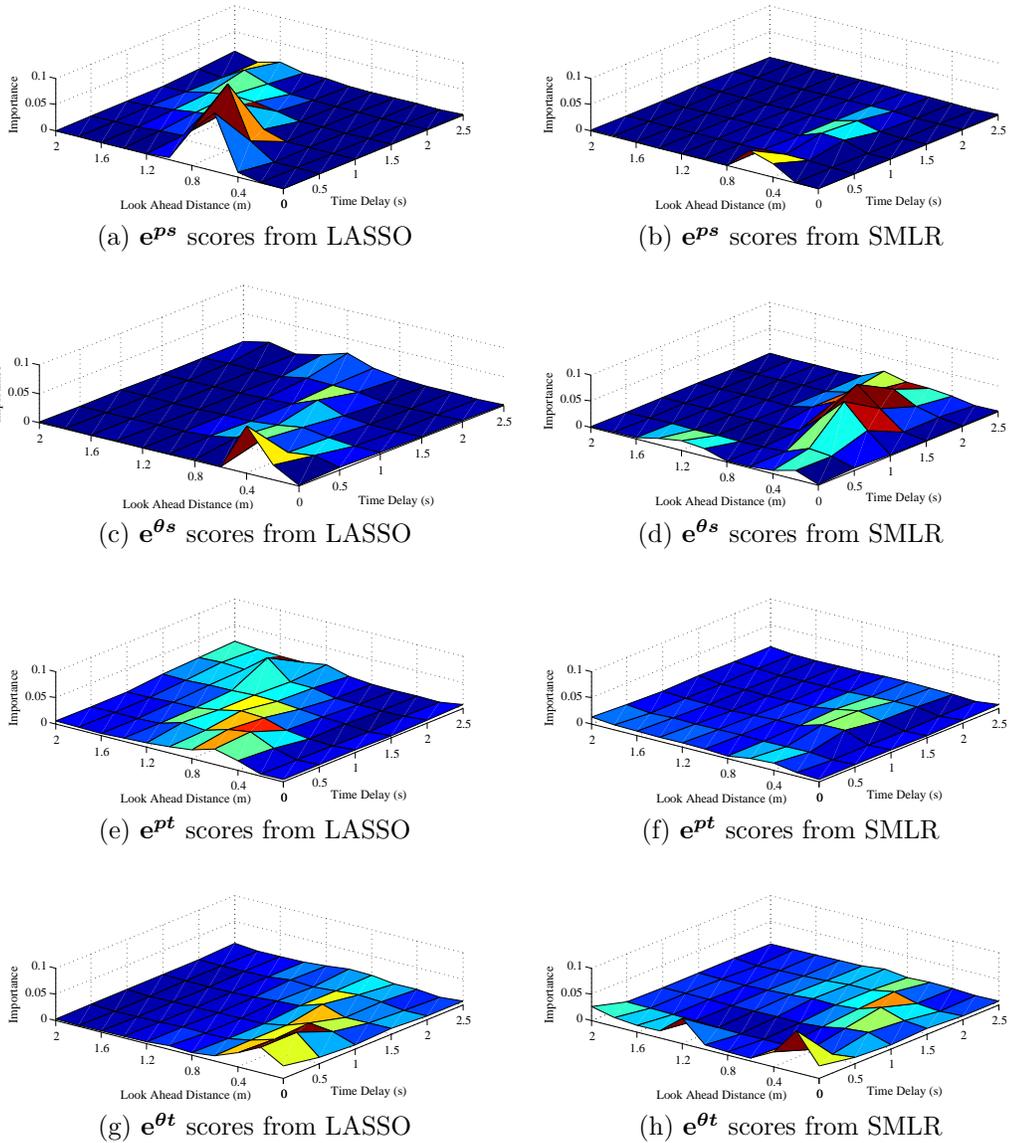


Figure 3.11: Comparison of average importance scores for four feature groups between the LASSO and SMLR models. As described in Section 3.3.3, e^{ps} and $e^{\theta s}$ are predicted position and heading errors assuming the robot continues in a straight line; e^{pt} and $e^{\theta t}$ are predicted position and heading errors assuming the robot continues turning at its current rate. The components of these groups are based on the robot's position and heading errors at set points on its predicted trajectory. Importance scores are plotted as a function of time delay and look-ahead distance.

lead-distance increases with time delay

An examination of the variation of feature predictive power with time delay shows that $e^{\theta t}(\theta_k, r(p_k + 1.2m))$ is a far better predictor of control response in the switchback region than other related features. This effect only occurs at 0s of time delay, and is possibly due to the fact that 1.2m is the approximate size of the physical features present in that segment of the route. As a consequence, $e^{\theta t}(\theta_k, r(p_k + 1.2m))$ provides a signal to turn at just the right moment in the most challenging of the areas to traverse. This is likely the cause of the secondary peak at no delay seen in Figures 3.11g and 3.11h.

3.6 Conclusion

A series of experiments and predictive models examining human compensation for time delays in robotic control was presented. The experiment measured the ability of human subjects to drive a remotely operated robot around a closed track when subjected to varied amounts of time delay. Position and heading data from the robot collected during the experiment showed that subjects' off-track error increased as time delay was increased, and that no short-term learning effects were apparent. Environmental features observable to the human subjects were combined with the known responses to generate regularized linear and logistic models. These models were trained on the data set using the 'Least Angle Regression' (LARS) and 'Sparse Multinomial Logistic Regression' (SMLR) algorithms respectively, and were able to successfully predict the human responses with a high degree of accuracy (up to 98% for models generated by LARS and 85% for those generated by SMLR). A novel 'importance' metric was defined in order to determine the relative contributions of each model feature to the model performance. This metric is potentially useful for applications in which the objective is to rank features in

order of their contribution to the model prediction performance, as opposed to simply finding a minimal model. ‘Importance’ scores can be found for any iterative model fitting algorithm that maintains a limited active set of features. An analysis of the feature scores showed that both models agree as to the relative importance of feature groups: with the most important being the past control history $\boldsymbol{\eta}$, followed by the predicted errors \mathbf{e}^{ps} , \mathbf{e}^{pt} , $\mathbf{e}^{\theta s}$, and $\mathbf{e}^{\theta t}$. This ordering shows that subjects attempt to maintain a smooth control output, and rely primarily on predicted errors based on an internal motion model in order to correct the robot trajectory. The subject’s attention, as measured by importance scores, appears to follow a window which moves forward approximately 0.73m between 0s and 2.5s of time delay. The emphasis placed on predictive features, and the shift in subjects’ attention in response to increased delays, supports the hypothesis that subjects are using a predictive model of the robot motion in order to *anticipate* upcoming control needs.

CONCLUSION

The research presented in this work has focused on several aspects of the remote operation of ground vehicles, notably Navigation and Mapping for autonomous robots and the effects of time delay in teleoperated vehicles.

Chapter 1 presents a qualitative method for robotic mapping of large scale spaces with sparse landmarks and minimal sensing. The Qualitative Relational Mapping (QRM) algorithm constructs a constraint graph which tracks which qualitative geometrical relations between landmarks are consistent with the measurements. The geometrical representation of these relationships is based on Freksa's Double Cross (FDC), in which the space around a pair of landmarks is discretized into 15 regions by front/back and left/right splits. This mapping approach can be seen as a form of qualitative triangulation based on angle measurements and estimates of the relative range orderings of visually distinctive landmarks. Algorithm performance is evaluated using Monte Carlo simulations and shows consistent map convergence as the number of imaging locations is increased. Additional simulations with restrictions on sensor range show that convergence rates are dependent upon the number of landmarks simultaneously observed, with good performance so long as approximately half of the landmarks seen from any given point. Although the graph structure used in this work is fully connected, computational performance could be improved by considering a hierarchical mapping system which eliminates redundant long-distance connections. Experimental data from a traversal of the JPL Mars Yard shows that realistic robot trajectories produce similar results, though in general more images are required for a given level of performance if the imaging points are not uniformly distributed. A method for determining new areas to take measurements is presented, given the metrical positions of the landmarks. A practical mapping implementation would benefit from a strategy for determining the optimal position for new measurements using only information

in the qualitative map. This would allow a rover to dynamically plan an active mapping strategy based on the environment.

The FDC representation is expanded in Chapter 2 to include splits based on relative distances as well as angles, resulting in the Extended Double Cross (EDC). The EDC states can be written as nonlinear inequalities in either range, for the measurement problem, or landmark position, for the graph inference problem. Both the measurements and the offline generation of lookup tables for converting between EDC states make use of a Branch-and-Bound approach to determining the feasibility of sets of these non-convex quadratic inequalities. An example navigation strategy was presented which uses estimates of the landmark Relative Neighborhood Graph (RNG) extracted from the qualitative map in order to find paths between the Voronoi regions of arbitrary landmarks. The asymptotic behavior of the mapping system was again evaluated using Monte-Carlo simulations of randomly generated maps and a data driven simulation of traversing the JPL Mars Yard, with similar overall results to those seen in Chapter 2 despite the increased complexity of the EDC representation. However, computation time was significantly increased. Additionally, the RNG estimates were seen to converge significantly faster than the map itself, indicating that navigation objectives can be achieved even when significant map ambiguities remain. A major limitation of the QRM system is a reliance on perfect data association in order to maintain map consistency. Data association failures cause inconsistencies to rapidly propagate through the graph, leading to edge estimates with no open states. This limitation could be addressed by moving to a probabilistic representation, where measurements updated discrete probability distributions over EDC states, rather than the current hard ‘yes/no’ decisions about open states.

Chapter 3 examined the ability of human operators to compensate for time delays when tele-operating remote vehicles. An experiment measured the ability of human subjects to drive a remotely operated robot around a closed track when subjected to varied amounts of time delay. Statistical models were trained on the data set using the ‘Least Angle Regression’ (LARS) and ‘Sparse Multinomial Logistic Regression’ (SMLR) algorithms, and were able to successfully predict the human responses with a high degree of accuracy (up to 98% for models generated by LARS and 85% for those generated by SMLR). A novel ‘importance’ metric was defined in order to determine the relative contributions of each model feature to the model performance. An analysis of the feature scores showed that both models agree as to the relative importance of feature groups: with the most important being the past control history, followed by the predicted off-track and heading errors based on simple models of the rovers future trajectory. The subject’s attention, as measured by importance scores, appears to follow a window which moves forward approximately 0.73m between 0s and 2.5s of time delay. The emphasis placed on predictive features, and the shift in subjects’ attention in response to increased delays, supports the hypothesis that subjects are using a predictive model of the robot motion in order to *anticipate* upcoming control needs.

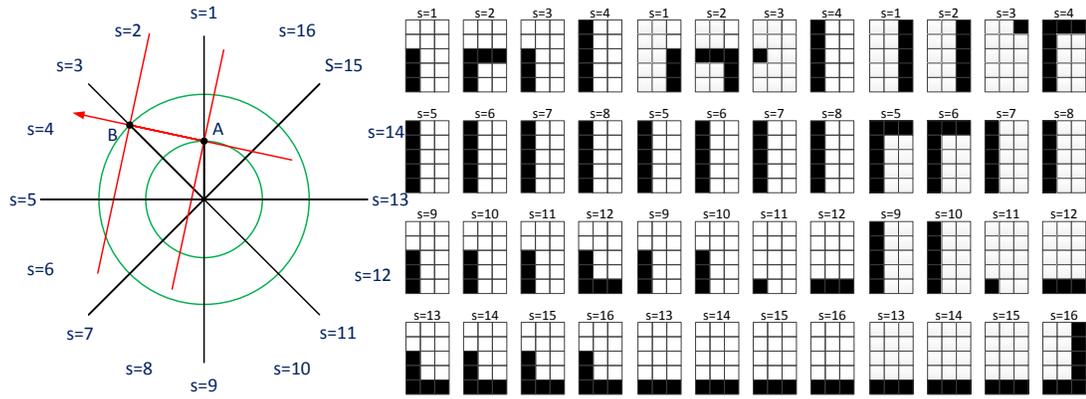
In summary, this thesis has presented the following novel contributions:

- A method for generating estimates of FDC states from camera images using geometrically constructed look-up tables
- Evaluation of mapping performance using the new FDC measurement function, using both Monte-Carlo and data-driven simulations
- A qualitative representation of landmark geometrical relationships, the EDC, which incorporates both qualitative angles and edge lengths
- Unary and Binary operators for manipulating EDC states

- An optimization approach for generating offline look-up table necessary for graph inference in the new representation, using algebraic inequalities derived from the geometrical regions
- An optimization approach for generating online measurements of qualitative states from camera images
- A branch-and-bound algorithm for solving the quadratic feasibility problems required for the above approaches
- A method for extracting estimates of the landmark RNG from the EDC qualitative map
- A long-distance navigation approach using the RNG to find intermediate landmarks
- Evaluation of mapping performance using the new EDC representation, using both Monte-Carlo and data-driven simulations
- A method for evaluating human anticipation when remote controlling time-delayed vehicles using linear regression and classification algorithms
- An ‘importance’ metric for evaluating feature contributions to linear regressions and classifiers
- Analysis of a set of human trials testing the anticipation hypothesis

APPENDIX A
SECTOR MEASUREMENT TABLES

For the vast majority of configurations for a robot and the landmarks A, B, C , landmark B will be either less than or greater than 90° from A . Listed below are measurement tables covering the case in which B is to the left of A . The symmetric case where B is to the right of A can be found by simply horizontally mirroring each table element. While unlikely to occur outside of simulation, measurement tables can be easily generated for the cases in which B is an exact multiple of 90° from A . These tables are far more constrained and are omitted for the sake of compactness. Likewise, additional tables must be generated if two or more landmarks are exactly equidistant from the robot.



(a) Geometry of A and B

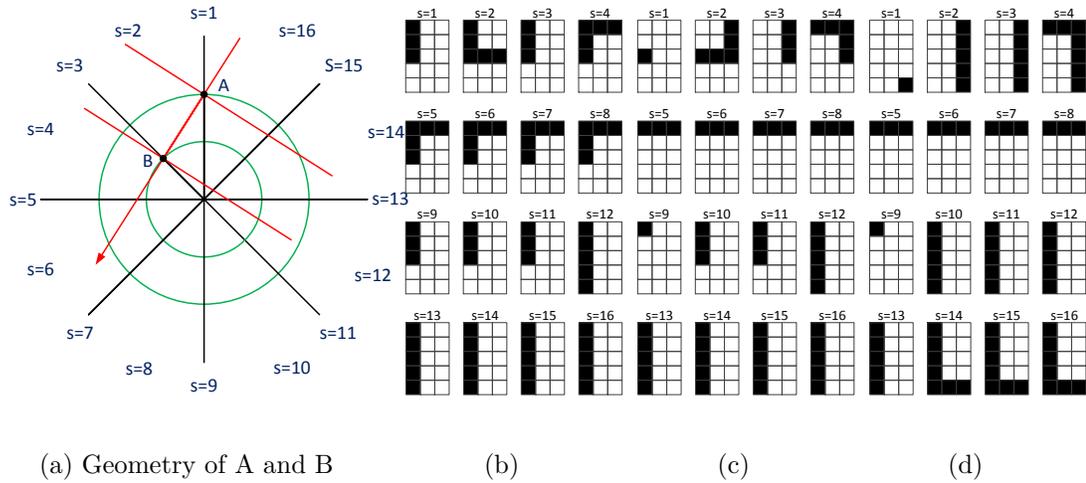
(b)

(c)

(d)

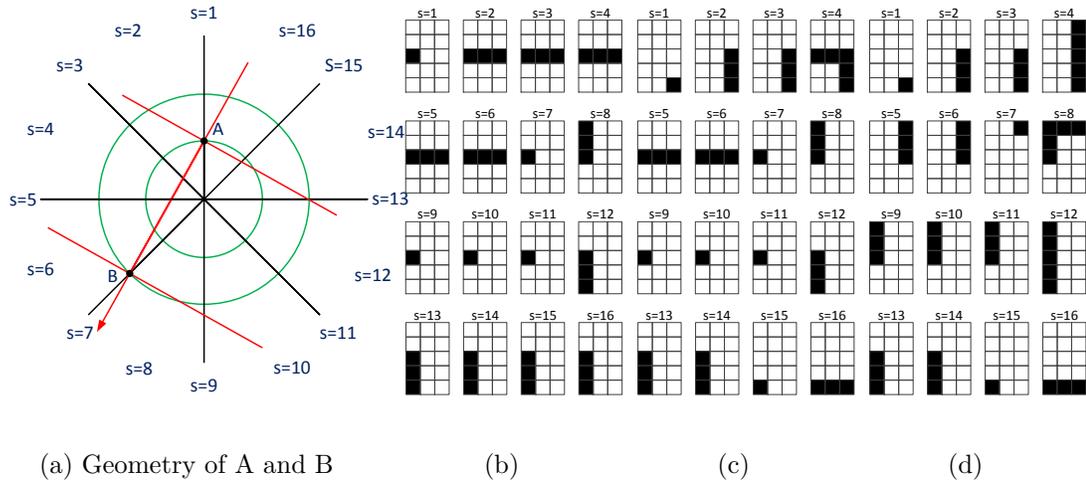
$$|C| < |A| < |B| \quad |A| < |C| < |B| \quad |A| < |B| < |C|$$

Figure A.1: Case 1: $g = 2$ (B lies less than 90° from A), $|A| < |B|$. (a) shows one possible arrangement of points A and B , superimposed with the corresponding double cross in red. Black lines indicate 90° intervals with respect to A and B . The numbers label angle sectors, s , for point C . The tables in (b-d) show the possible qualitative states for $AB : C$ for each sector as black squares. (b) corresponds to the case that C is closer than both A and B , (c) to the case that C lies between A and B , and (d) to the case that C lies further than A or B .



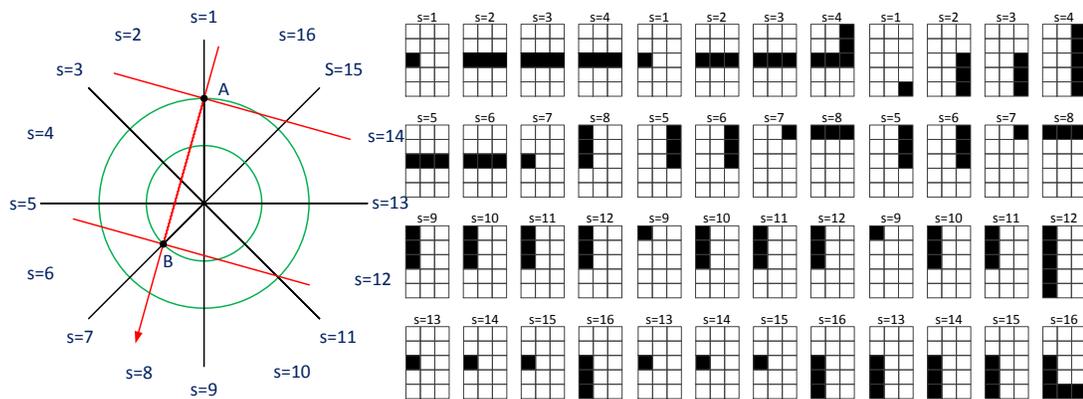
$$|C| < |B| < |A| \quad |B| < |C| < |A| \quad |B| < |A| < |C|$$

Figure A.2: Case 2: $g = 2$ (B lies less than 90° from A), $|B| < |A|$. The figure labeling and tables are constructed as described in Figure A.1



$$|C| < |A| < |B| \quad |A| < |C| < |B| \quad |A| < |B| < |C|$$

Figure A.3: Case 3: $g = 4$ (B lies more than 90° from A but less than 180°), $|A| < |B|$. The figure labeling and tables are constructed as described in Figure A.1



(a) Geometry of A and B

(b)

(c)

(d)

$$|C| < |B| < |A| \quad |B| < |C| < |A| \quad |B| < |A| < |C|$$

Figure A.4: Case 4: $g = 4$ (B lies more than 90° from A but less than 180°), $|B| < |A|$. The figure labeling and tables are constructed as described in Figure A.1

APPENDIX B
DOUBLE CROSS OPERATOR TABLES

B.1 Unary Operators

Zimmermann and Freksa [25] discusses three unary operators for the FDC model. Equivalent operators, *inverse*, *left cyclical*, and *right cyclical*, are used in this paper and are associated with the lookup tables shown in Figure B.1.

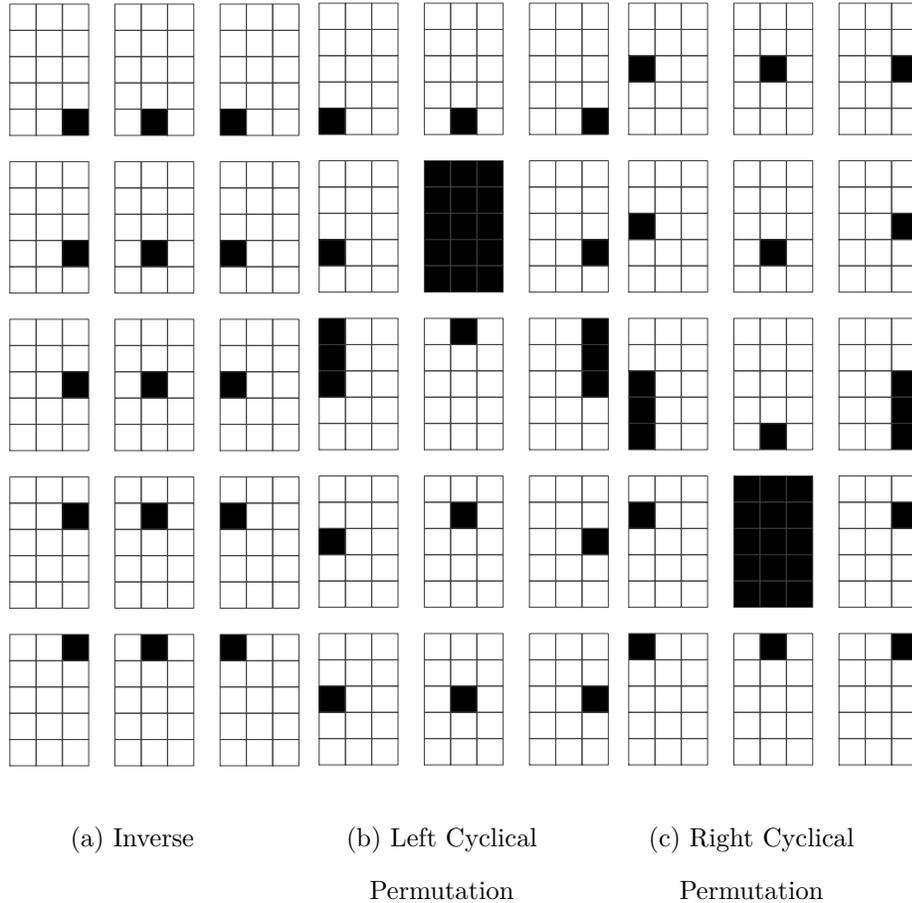


Figure B.1: Lookup tables for the unary operators. (a) shows the effects of converting $AB : C$ to $BA : C$. (b) shows the effects of converting $AB : C$ to $BC : A$. (c) shows the effects of converting $AB : C$ to $CA : B$. Each table is organized according to the principle of self-similarity: for example, the effect of the left shift operation on the right-back state is found in the right-back table position (row 5, column 3).

B.2 Binary Composition Operator

Freksa [24] presents the binary composition operator which allows inference about the qualitative relationships between landmarks not directly observed together. The lookup table for compositions is recreated in Figure B.2, with some reordering and omission of the elements corresponding to two landmarks being co-located. Given a state for $AB : C$, shown in red on the vertical, and a state for $BC : D$, shown in blue on the horizontal, the corresponding table element lists the possible states for the relationship $AB : D$ in black.

Figure B.2: FDC composition operator look-up table

APPENDIX C
EDC UNARY OPERATOR TABLE GENERATION

The EDC unary operators are defined by the state equivalences listed in Table 2.1. Table elements can be proven by showing that the listed states have the same constraint equations for their boundaries. These constraints can be written as follows for any three arbitrary points $\mathbf{X} = (x_0, x_1)$, $\mathbf{Y} = (y_0, y_1)$, and $\mathbf{Z} = (z_0, z_1)$:

$$c_1(\mathbf{X}, \mathbf{Y}, \mathbf{Z}) = (y_1 - x_1)(z_0 - x_0) - (y_0 - x_0)(z_1 - x_1)$$

$$c_2(\mathbf{X}, \mathbf{Y}, \mathbf{Z}) = (x_0 - y_0)(z_0 + y_1 - x_1 - x_0) - (y_1 - x_1)(z_1 - y_0 - x_1 + x_0)$$

$$c_3(\mathbf{X}, \mathbf{Y}, \mathbf{Z}) = (x_0 - y_0)(z_0 + y_1 - y_0 - x_1) - (y_1 - x_1)(z_1 - y_1 - y_0 + x_0)$$

$$c_4(\mathbf{X}, \mathbf{Y}, \mathbf{Z}) = (z_1 - y_1)^2 - (z_1 - x_1)^2 + (z_0 - y_0)^2 - (z_0 - x_0)^2$$

$$c_5(\mathbf{X}, \mathbf{Y}, \mathbf{Z}) = (x_1 - y_1)^2 + (x_0 - y_0)^2 - (z_1 - x_1)^2 - (z_0 - x_0)^2$$

$$c_6(\mathbf{X}, \mathbf{Y}, \mathbf{Z}) = (x_1 - y_1)^2 + (x_0 - y_0)^2 - (z_1 - y_1)^2 - (z_0 - y_0)^2$$

Values of these expressions correspond to the EDC region boundaries as follows:

- if $c_1(\mathbf{X}, \mathbf{Y}, \mathbf{Z}) < 0$, \mathbf{Z} is to the left of $\overline{\mathbf{XY}}$
- if $c_3(\mathbf{X}, \mathbf{Y}, \mathbf{Z}) < 0$, \mathbf{Z} is in front of \mathbf{X}
- if $c_2(\mathbf{X}, \mathbf{Y}, \mathbf{Z}) < 0$, \mathbf{Z} is in front of \mathbf{Y}
- if $c_4(\mathbf{X}, \mathbf{Y}, \mathbf{Z}) < 0$, \mathbf{Z} is in closer to \mathbf{Y} than it is to \mathbf{X}
- if $c_5(\mathbf{X}, \mathbf{Y}, \mathbf{Z}) < 0$, \mathbf{Z} is further from \mathbf{X} than \mathbf{Y} is from \mathbf{X}
- if $c_6(\mathbf{X}, \mathbf{Y}, \mathbf{Z}) < 0$, \mathbf{Z} is further from \mathbf{Y} than \mathbf{Y} is from \mathbf{X}

If $\mathbf{A} = (a_x, a_y)$, $\mathbf{B} = (b_x, b_y)$, $\mathbf{C} = (c_x, c_y)$, then the state equivalences in the second column of Table 2.1 for the left-side EDC regions are proven as follows:

- State $\mathbf{AB} : \mathbf{C} = 1$ is defined by the constraints $c_1(\mathbf{A}, \mathbf{B}, \mathbf{C}) < 0$,

$c_6(\mathbf{A}, \mathbf{B}, \mathbf{C}) < 0$, and $c_3(\mathbf{A}, \mathbf{B}, \mathbf{C}) < 0$. These can be expanded into

$$-b_x c_y + a_x c_y + b_y c_x - a_y c_x - a_x b_y + a_y b_x < 0 \quad (\text{C.1})$$

$$-c_y^2 + 2b_y c_y - c_x^2 + 2b_x c_x - 2a_y b_y - 2a_x b_x + a_y^2 + a_x^2 < 0 \quad (\text{C.2})$$

$$-b_y c_y + a_y c_y - b_x c_x + a_x c_x + b_y^2 - a_y b_y + b_x^2 - a_x b_x < 0 \quad (\text{C.3})$$

State $\mathbf{BC} : \mathbf{A} = 17$ is defined by the constraints $c_1(\mathbf{B}, \mathbf{C}, \mathbf{A}) < 0$, $-c_5(\mathbf{B}, \mathbf{C}, \mathbf{A}) < 0$, and $-c_2(\mathbf{B}, \mathbf{C}, \mathbf{A}) < 0$. These can be expanded into

$$-b_x c_y + a_x c_y + b_y c_x - a_y c_x - a_x b_y + a_y b_x < 0 \quad (\text{C.4})$$

$$-c_y^2 + 2b_y c_y - c_x^2 + 2b_x c_x - 2a_y b_y - 2a_x b_x + a_y^2 + a_x^2 < 0 \quad (\text{C.5})$$

$$-b_y c_y + a_y c_y - b_x c_x + a_x c_x + b_y^2 - a_y b_y + b_x^2 - a_x b_x < 0 \quad (\text{C.6})$$

(C.1) = (C.4), (C.2) = (C.5), and (C.3) = (C.6), therefore $\mathbf{AB} : \mathbf{C} = 1$ implies $\mathbf{BC} : \mathbf{A} = 17$.

- State $\mathbf{AB} : \mathbf{C} = 3$ is defined by the constraints $c_1(\mathbf{A}, \mathbf{B}, \mathbf{C}) < 0$, $-c_6(\mathbf{A}, \mathbf{B}, \mathbf{C}) < 0$, and $c_3(\mathbf{A}, \mathbf{B}, \mathbf{C}) < 0$. These can be expanded into

$$-b_x c_y + a_x c_y + b_y c_x - a_y c_x - a_x b_y + a_y b_x < 0 \quad (\text{C.7})$$

$$c_y^2 - 2b_y c_y + c_x^2 - 2b_x c_x + 2a_y b_y + 2a_x b_x - a_y^2 - a_x^2 < 0 \quad (\text{C.8})$$

$$-b_y c_y + a_y c_y - b_x c_x + a_x c_x + b_y^2 - a_y b_y + b_x^2 - a_x b_x < 0 \quad (\text{C.9})$$

State $\mathbf{BC} : \mathbf{A} = 19$ is defined by the constraints $c_1(\mathbf{B}, \mathbf{C}, \mathbf{A}) < 0$, $c_5(\mathbf{B}, \mathbf{C}, \mathbf{A}) < 0$, and $-c_2(\mathbf{B}, \mathbf{C}, \mathbf{A}) < 0$. These can be expanded into

$$-b_x c_y + a_x c_y + b_y c_x - a_y c_x - a_x b_y + a_y b_x < 0 \quad (\text{C.10})$$

$$c_y^2 - 2b_y c_y + c_x^2 - 2b_x c_x + 2a_y b_y + 2a_x b_x - a_y^2 - a_x^2 < 0 \quad (\text{C.11})$$

$$-b_y c_y + a_y c_y - b_x c_x + a_x c_x + b_y^2 - a_y b_y + b_x^2 - a_x b_x < 0 \quad (\text{C.12})$$

(C.7) = (C.10), (C.8) = (C.11), and (C.9) = (C.12), therefore $AB : C = 2$ implies $BC : A = 19$.

- State $AB : C = 5$ is defined by the constraints $c_6(A, B, C) < 0$, $-c_3(A, B, C) < 0$, and $c_4(A, B, C) < 0$. These can be expanded into

$$-c_y^2 + 2b_y c_y - c_x^2 + 2b_x c_x - 2a_y b_y - 2a_x b_x + a_y^2 + a_x^2 < 0 \quad (C.13)$$

$$b_y c_y - a_y c_y + b_x c_x - a_x c_x - b_y^2 + a_y b_y - b_x^2 + a_x b_x < 0 \quad (C.14)$$

$$-2b_y c_y + 2a_y c_y - 2b_x c_x + 2a_x c_x + b_y^2 + b_x^2 - a_y^2 - a_x^2 < 0 \quad (C.15)$$

State $BC : A = 12$ is defined by the constraints $-c_5(B, C, A) < 0$, $c_2(B, C, A) < 0$, and $c_6(B, C, A) < 0$. These can be expanded into

$$-c_y^2 + 2b_y c_y - c_x^2 + 2b_x c_x - 2a_y b_y - 2a_x b_x + a_y^2 + a_x^2 < 0 \quad (C.16)$$

$$b_y c_y - a_y c_y + b_x c_x - a_x c_x - b_y^2 + a_y b_y - b_x^2 + a_x b_x < 0 \quad (C.17)$$

$$-2b_y c_y + 2a_y c_y - 2b_x c_x + 2a_x c_x + b_y^2 + b_x^2 - a_y^2 - a_x^2 < 0 \quad (C.18)$$

(C.13) = (C.16), (C.14) = (C.17), and (C.15) = (C.18), therefore $AB : C = 5$ implies $BC : A = 12$.

- State $AB : C = 6$ is defined by the constraints $-c_6(A, B, C) < 0$, $-c_3(A, B, C) < 0$, and $c_5(A, B, C) < 0$. These can be expanded into

$$c_y^2 - 2b_y c_y + c_x^2 - 2b_x c_x + 2a_y b_y + 2a_x b_x - a_y^2 - a_x^2 < 0 \quad (C.19)$$

$$b_y c_y - a_y c_y + b_x c_x - a_x c_x - b_y^2 + a_y b_y - b_x^2 + a_x b_x < 0 \quad (C.20)$$

$$-c_y^2 + 2a_y c_y - c_x^2 + 2a_x c_x + b_y^2 - 2a_y b_y + b_x^2 - 2a_x b_x < 0 \quad (C.21)$$

State $BC : A = 11$ is defined by the constraints $c_5(B, C, A) < 0$,

$c_2(B, C, A) < 0$, and $-c_4(B, C, A) < 0$. These can be expanded into

$$c_y^2 - 2b_y c_y + c_x^2 - 2b_x c_x + 2a_y b_y + 2a_x b_x - a_y^2 - a_x^2 < 0 \quad (C.22)$$

$$b_y c_y - a_y c_y + b_x c_x - a_x c_x - b_y^2 + a_y b_y - b_x^2 + a_x b_x < 0 \quad (C.23)$$

$$-c_y^2 + 2a_y c_y - c_x^2 + 2a_x c_x + b_y^2 - 2a_y b_y + b_x^2 - 2a_x b_x < 0 \quad (C.24)$$

(C.19) = (C.22), (C.20) = (C.23), and (C.21) = (C.24), therefore

$AB : C = 6$ implies $BC : A = 11$.

- State $AB : C = 7$ is defined by the constraints $c_4(A, B, C) < 0$, $-c_5(A, B, C) < 0$, and $c_1(A, B, C) < 0$. These can be expanded into

$$-2b_y c_y + 2a_y c_y - 2b_x c_x + 2a_x c_x + b_y^2 + b_x^2 - a_y^2 - a_x^2 < 0 \quad (C.25)$$

$$c_y^2 - 2a_y c_y + c_x^2 - 2a_x c_x - b_y^2 + 2a_y b_y - b_x^2 + 2a_x b_x < 0 \quad (C.26)$$

$$-b_x c_y + a_x c_y + b_y c_x - a_y c_x - a_x b_y + a_y b_x < 0 \quad (C.27)$$

States $BC : A = \{1, 5\}$ are defined by the constraints $c_6(B, C, A) < 0$,

$c_4(B, C, A) < 0$, and $c_1(B, C, A) < 0$. These can be expanded into

$$-2b_y c_y + 2a_y c_y - 2b_x c_x + 2a_x c_x + b_y^2 + b_x^2 - a_y^2 - a_x^2 < 0 \quad (C.28)$$

$$c_y^2 - 2a_y c_y + c_x^2 - 2a_x c_x - b_y^2 + 2a_y b_y - b_x^2 + 2a_x b_x < 0 \quad (C.29)$$

$$-b_x c_y + a_x c_y + b_y c_x - a_y c_x - a_x b_y + a_y b_x < 0 \quad (C.30)$$

(C.25) = (C.28), (C.26) = (C.29), and (C.27) = (C.30), therefore

$AB : C = 7$ implies $BC : A = \{1, 5\}$.

- States $AB : C = \{11, 19\}$ are defined by the constraints $-c_4(A, B, C) < 0$, $c_5(A, B, C) < 0$, and $c_1(A, B, C) < 0$. These can be expanded into

$$2b_y c_y - 2a_y c_y + 2b_x c_x - 2a_x c_x - b_y^2 - b_x^2 + a_y^2 + a_x^2 < 0 \quad (C.31)$$

$$-c_y^2 + 2a_y c_y - c_x^2 + 2a_x c_x + b_y^2 - 2a_y b_y + b_x^2 - 2a_x b_x < 0 \quad (C.32)$$

$$-b_x c_y + a_x c_y + b_y c_x - a_y c_x - a_x b_y + a_y b_x < 0 \quad (C.33)$$

State $BC : A = 13$ is defined by the constraints $-c_6(B, C, A) < 0$, $-c_4(B, C, A) < 0$, and $c_1(B, C, A) < 0$. These can be expanded into

$$2b_y c_y - 2a_y c_y + 2b_x c_x - 2a_x c_x - b_y^2 - b_x^2 + a_y^2 + a_x^2 < 0 \quad (C.34)$$

$$-c_y^2 + 2a_y c_y - c_x^2 + 2a_x c_x + b_y^2 - 2a_y b_y + b_x^2 - 2a_x b_x < 0 \quad (C.35)$$

$$-b_x c_y + a_x c_y + b_y c_x - a_y c_x - a_x b_y + a_y b_x < 0 \quad (C.36)$$

(C.31) = (C.34), (C.32) = (C.35), and (C.33) = (C.36), therefore $AB : C = \{11, 19\}$ implies $BC : A = 13$.

- States $AB : C = \{12, 17\}$ are defined by the constraints $c_6(A, B, C) < 0$, $-c_5(A, B, C) < 0$, and $c_1(A, B, C) < 0$. These can be expanded into

$$-c_y^2 + 2b_y c_y - c_x^2 + 2b_x c_x - 2a_y b_y - 2a_x b_x + a_y^2 + a_x^2 < 0 \quad (C.37)$$

$$c_y^2 - 2a_y c_y + c_x^2 - 2a_x c_x - b_y^2 + 2a_y b_y - b_x^2 + 2a_x b_x < 0 \quad (C.38)$$

$$-b_x c_y + a_x c_y + b_y c_x - a_y c_x - a_x b_y + a_y b_x < 0 \quad (C.39)$$

State $BC : A = 7$ is defined by the constraints $-c_5(B, C, A) < 0$, $c_4(B, C, A) < 0$, and $c_1(B, C, A) < 0$. These can be expanded into

$$-c_y^2 + 2b_y c_y - c_x^2 + 2b_x c_x - 2a_y b_y - 2a_x b_x + a_y^2 + a_x^2 < 0 \quad (C.40)$$

$$c_y^2 - 2a_y c_y + c_x^2 - 2a_x c_x - b_y^2 + 2a_y b_y - b_x^2 + 2a_x b_x < 0 \quad (C.41)$$

$$-b_x c_y + a_x c_y + b_y c_x - a_y c_x - a_x b_y + a_y b_x < 0 \quad (C.42)$$

(C.37) = (C.40), (C.38) = (C.41), and (C.39) = (C.42), therefore $AB : C = \{12, 17\}$ implies $BC : A = 7$.

- State $AB : C = 13$ is defined by the constraints $-c_6(A, B, C) < 0$,

$-c_4(\mathbf{A}, \mathbf{B}, \mathbf{C}) < 0$, and $c_1(\mathbf{A}, \mathbf{B}, \mathbf{C}) < 0$. These can be expanded into

$$c_y^2 - 2b_y c_y + c_x^2 - 2b_x c_x + 2a_y b_y + 2a_x b_x - a_y^2 - a_x^2 < 0 \quad (\text{C.43})$$

$$2b_y c_y - 2a_y c_y + 2b_x c_x - 2a_x c_x - b_y^2 - b_x^2 + a_y^2 + a_x^2 < 0 \quad (\text{C.44})$$

$$-b_x c_y + a_x c_y + b_y c_x - a_y c_x - a_x b_y + a_y b_x < 0 \quad (\text{C.45})$$

States $\mathbf{BC} : \mathbf{A} = \{3, 6\}$ are defined by the constraints $c_5(\mathbf{B}, \mathbf{C}, \mathbf{A}) < 0$,

$-c_6(\mathbf{B}, \mathbf{C}, \mathbf{A}) < 0$, and $c_1(\mathbf{B}, \mathbf{C}, \mathbf{A}) < 0$. These can be expanded into

$$c_y^2 - 2b_y c_y + c_x^2 - 2b_x c_x + 2a_y b_y + 2a_x b_x - a_y^2 - a_x^2 < 0 \quad (\text{C.46})$$

$$2b_y c_y - 2a_y c_y + 2b_x c_x - 2a_x c_x - b_y^2 - b_x^2 + a_y^2 + a_x^2 < 0 \quad (\text{C.47})$$

$$-b_x c_y + a_x c_y + b_y c_x - a_y c_x - a_x b_y + a_y b_x < 0 \quad (\text{C.48})$$

(C.43) = (C.46), (C.44) = (C.47), and (C.45) = (C.48), therefore

$\mathbf{AB} : \mathbf{C} = 13$ implies $\mathbf{BC} : \mathbf{A} = \{3, 6\}$.

Given the correct table elements for the left-side EDC states, the equivalences for the right-side EDC states listed in the second column of Table 2.1 must be true by symmetry. Given the state equivalences between $\mathbf{AB} : \mathbf{C}$ and $\mathbf{BC} : \mathbf{A}$ and between $\mathbf{AB} : \mathbf{C}$ and $\mathbf{BA} : \mathbf{C}$, we can safely use the LEFT and INVERSE operators to generate the equivalent states for $\mathbf{CA} : \mathbf{B}$ listed in the third column of Table 2.1, exploiting the identity $\text{INVERSE}(\text{LEFT}(\text{INVERSE}(\mathbf{AB} : \mathbf{C}))) = \mathbf{CA} : \mathbf{B}$.

BIBLIOGRAPHY

- [1] Mark McClelland, Mark Campbell, and Tara Estlin. Qualitative relational mapping for mobile robots with minimal sensing. *AIAA Journal of Aerospace Information Systems*, In Review, 2014.
- [2] Mark McClelland and Mark Campbell. Probabilistic modeling of anticipation in human controllers. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 43(4):886–900, 2013. ISSN 2168-2216. doi: 10.1109/TSMCA.2012.2220541.
- [3] K. Ali, C. Vanelli, J. Biesiadecki, M. Maimone, Y. Cheng, A. San Martin, and J. Alexander. Attitude and position estimation on the mars exploration rovers. In *Proceedings of the IEEE Conference on Systems, Man, and Cybernetics*, 2005.
- [4] Benjamin J. Kuipers and Tod S. Levitt. Navigation and mapping in large scale space. *AI Magazine*, 9(2):25–43, 1988.
- [5] S. Thrun, W. Burgard, and D. Fox. *Probabalistic Robotics*. MIT Press, 2006.
- [6] Mark McClelland, Mark Campbell, and Tara Estlin. Qualitative relational mapping for robotic navigation. In *AIAA Infotech@Aerospace*, 2012.
- [7] Andrew J. Davison, Ian D. Reid, Nicholas D. Molton, and Olivier Stasse. Monoslam: Real-time single camera slam. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26:1052–1067, 2007.
- [8] Tod Levitt Daryl Lawton. Qualitative navigation for mobile robots. *Artificial intelligence*, 44(3):305–360, 1990.
- [9] Jan Oliver Wallgrun. Qualitative spatial reasoning for topological map learning. *Spatial Cognition & Computation*, 10(4):207–246, 2010.

- [10] Christoph Schlieder. *Representing Visible Locations for Qualitative Navigation*, pages 423–532. CIMNE, 1993.
- [11] Paolo Fogliaroni, Jan Oliver Wallgrun, Eliseo Clementini, Francesco Tarquini, and Diedrich Wolter. A qualitative approach to localization and navigation based on visibility information. In *Spatial Information Theory*, volume 5756 of *Lecture Notes in Computer Science*, pages 312–329. Springer Berlin Heidelberg, 2009. ISBN 978-3-642-03831-0.
- [12] Hugh Durrant-Whyte and Tim Bailey. Simultaneous localization and mapping: part i. *IEEE Robotics Automation Magazine*, 13(2):99–110, 2006. ISSN 1070-9932. doi: 10.1109/MRA.2006.1638022.
- [13] Tim Bailey and Hugh Durrant-Whyte. Simultaneous localization and mapping (slam): part ii. *IEEE Robotics Automation Magazine*, 13(3):108–117, 2006. ISSN 1070-9932. doi: 10.1109/MRA.2006.1678144.
- [14] Simon Julier and Jeffrey Uhlmann. A counter example to the theory of simultaneous localization and map building. In *IEEE International Conference Robotics and Automation*, 2001.
- [15] Shoudong Huang and Gamini Dissanayake. Convergence and consistency analysis for extended kalman filter based slam. *IEEE Transactions on Robotics*, 23(5):1036–1049, 2007. ISSN 1552-3098.
- [16] J.A. Castellanos, R. Martinez-Cantin, J.D. Tardos, and J. Neira. Robocentric map joining: Improving the consistency of ekf-slam. *Robotics and Autonomous Systems*, 55(1):21 – 29, 2007. ISSN 0921-8890.
- [17] A. Angeli, S. Doncieux, J.A. Meyer, and D. Filliat. Incremental vision-based

- topological slam. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1031–1036, 2008.
- [18] Mark Cummins and Paul Newman. Fab-map: Probabilistic localization and mapping in the space of appearance. *The International Journal of Robotics Research*, 27:647–665, 2008.
- [19] David A. Randell, Zhan Cui, and Anthony G. Cohn. A spatial logic based on regions and connection. In B. Nebel, C. Rich, and W.R. Swartout, editors, *Proceedings of the third National Conference on Principles of Knowledge Representation and Reasoning*. Morgan Kaufmann, 1992.
- [20] Derek Hoiem, Alexei A. Efros, and Martial Hebert. Recovering surface layout from an image. *International Journal of Computer Vision*, 75(1):151–172, October 2007.
- [21] Gabe Sibley, Christopher Mei, Ian Reid, and Paul Newman. Vast-scale outdoor navigation using adaptive relative bundle adjustment. *The International Journal of Robotics Research*, 29(8):958–980, July 2010.
- [22] I. L. Dryden and K. V. Mardia. Multivariate shape analysis. *Sankhya: The Indian Journal of Statistics, Series A*, 55(3):460–480, 1993. ISSN 0581572X.
- [23] Philipp Mitteroecker and Philipp Gunz. Advances in geometric morphometrics. *Evolutionary Biology*, 36(2):235–247, 2009. ISSN 0071-3260.
- [24] Christian Freksa. Using orientation information for qualitative spatial reasoning. *Theories and Methods of Spatio-Temporal Reasoning in Geographic Space (LNCS)*, 639:162–178, 1992.
- [25] Kai Zimmermann and Christian Freksa. Qualitative spatial reasoning using orientation, distance, and path knowledge. *Applied Intelligence*, 6:49–58, 1996.

- [26] Alexander Scivos and Bernhard Nebel. The finest of its class: The natural point-based ternary calculus Irr for qualitative spatial reasoning. In Christian Freksa, Markus Knauff, Bernd Krieg-Brückner, Bernhard Nebel, and Thomas Barkowsky, editors, *Spatial Cognition IV. Reasoning, Action, Interaction*, volume 3343 of *Lecture Notes in Computer Science*, pages 283–303. Springer Berlin Heidelberg, 2005. ISBN 978-3-540-25048-7.
- [27] Alexander Scivos and Bernhard Nebel. Double-crossing: Decidability and computational complexity of a qualitative calculus for navigation. In Daniel R. Montello, editor, *Spatial Information Theory*, volume 2205 of *Lecture Notes in Computer Science*, pages 431–446. Springer Berlin Heidelberg, 2001. ISBN 978-3-540-42613-4.
- [28] A.G. Cohn. Calculi for qualitative spatial reasoning. *Artificial Intelligence and Symbolic Mathematical Computation (LNCS)*, 1138:124–143, 1996.
- [29] Peter van Beek. Reasoning about qualitative temporal information. *Artificial intelligence*, 58(1):297–326, 1992.
- [30] Jochen Renz and Bernhard Nebel. Qualitative spatial reasoning using constraint calculi. In *Handbook of spatial logics*, pages 161–215. Springer, 2007.
- [31] P. S. Schenker, E. T. Baumgartner, P. G. Backes, H. Aghazarian, L.I. Dorsky, J. S. Norris, T. L. Huntsberger, Y. Cheng, A. Trebi-Ollennu, M. S. Garrett, B. A. Kennedy, A. J. Ganino, R. E. Arvidson, and Cornell Univ. S.W. Squyres. Fido: a field integrated design & operations rover for mars surface exploration. In *International Symposium on Artificial Intelligence, Robotics and Autonomous for Space*, 2001.

- [32] Radu Bogdan Rusu and Steve Cousins. 3d is here: Point cloud library (pcl). In *IEEE International Conference on Robotics and Automation (ICRA)*, 2011.
- [33] Steven B. Goldberg, Mark W. Maimone, and Larry Matthies. Stereo vision and rover navigation software for planetary exploration. In *IEEE Aerospace Conference*, 2002.
- [34] Martin Magnusson, Achim Lilienthal, and Tom Duckett. Scan registration for autonomous mining vehicles using 3d-ndt. *Journal of Field Robotics*, 24: 803–827, 2007.
- [35] F. Lu and E. Miliotis. Globally consistent range scan alignment for environment mapping. *Autonomous Robots*, 4:333–349, 1997. ISSN 0929-5593.
- [36] M. McClelland and M. Campbell. Anticipation as a method for overcoming time delay in control of remote systems. In *AIAA Guidance, Navigation and Control Conference*, 2010.
- [37] Costas Maranas and Christodoulos Floudas. Global optimization in generalized geometric programming. *Computers & Chemical Engineering*, 21:351–369, 1997.
- [38] Pasquale DeAngelis, Panos Pardalos, and Gerardo Toraldo. Quadratic programming with box constraints. *Nonconvex Optimization and Its Applications*, 18:73–95, 1997.
- [39] Dieter Vandembussche and George Nemhauser. A branch-and-cut algorithm for nonconvex quadratic programs with box constraints. *Mathematical Programming*, 102:559–575, 2005.
- [40] Hanif Sherali and Cihan Tuncbilek. A reformulation-convexification approach

- for solving nonconvex quadratic programming problems. *Journal of Global Optimization*, 7:1–31, 1995.
- [41] P. Gaussier, C. Joulain, J.P. Banquet, S. Lepretre, and A. Revel. The visual homing problem: An example of robotics/biology cross fertilization. *Robotics and Autonomous Systems*, 30:155 – 180, 2000. ISSN 0921-8890.
- [42] A. Torralba, K.P. Murphy, W.T. Freeman, and M.A. Rubin. Context-based vision system for place and object recognition. In *IEEE International Conference on Computer Vision*, 2003.
- [43] J.W. Jaromczyk and G.T. Toussaint. Relative neighborhood graphs and their relatives. *Proceedings of the IEEE*, 80:1502–1517, 1992. ISSN 0018-9219.
- [44] E.T. Baumgartner, J.J. Biesiadecki, R.G. Bonitz, B.K. Cooper, F.R. Hartman, P.C. Leger, M.W. Maimone, S.A. Maxwell, A. Trebi-Ollenu, and J.R. Wright. Mars exploration rover surface operations: Driving opportunity at meridiani planum. In *Proceedings of 13th Annual Conference on Manual Control*, 2005.
- [45] M.S. Triantafyllou and M.A. Grosenbaugh. Robust control for underwater vehicle systems with time delays. *IEEE Journal of Oceanic Engineering*, 16:146–151, 1991.
- [46] D.A. Thompson. The development of a six degree of freedom robot evaluation test. In *Proceedings of 13th Annual Conference on Manual Control*, 1997.
- [47] T.B. Sheridan and W.R. Ferrell. Remote manipulative control with transmission delay. *IEEE Transactions on Human Factors in Electronics*, 4:25–29, 1963.

- [48] M.D. Fabrizio, B.R. Lee, D.Y. Chan, D. Stoianovici, T.W. Jarrett, C. Yang, and L.R. Kavoussi. Effect of time delay on surgical performance during telesurgical manipulation. *Journal of Endourology*, 14:133–138, 2000.
- [49] L. Matthies, M. Maimone, A. Johnson, Y. Cheng, R. Willson, C. Villalpando, S. Goldberg, A. Huertas, A. Steing, and A. Angelova. Computer vision on mars. *International Journal of Computer Vision*, 75:67–92, 2007.
- [50] W.R. Ferrell. Delayed force feedback. *Human Factors*, 8:449–455, 1966.
- [51] T.B. Sheridan. Space teleoperation through time delay: Review and prognosis. *IEEE Transactions on Robotics and Automation*, 9:592–606, 1993.
- [52] M. Chomet, N. Freeberg, and A. Swanson. A simulation of operator capability in robot vehicle control. In *IRE International Convention Record*, volume 9, 1962.
- [53] J. Corde Lane, C.R. Carignan, B. Sullivan, D. Akin, T. Hunt, and R. Cohen. Effects of time delay on telerobotic control of neutral buoyancy vehicles. In *Proceedings of ICRA*, 2002.
- [54] J.M. Leslie. Effects of time delay in the visual feedback loop of a man-machine system. Technical Report TN-1351, NASA, 1962.
- [55] C.M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.
- [56] D.E. McGovern. *Pictorial Communication in Virtual and Real Environments*, chapter Experience and Results in Teleoperation of Land Vehicles, pages 182–195. CRC Press, 1991.
- [57] Gregor Klančar and Igor Skrjanc. Tracking-error model-based predictive con-

- trol for mobile robots in real time. *Robotics and Automation*, 55:460–469, 2007.
- [58] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistics Society B*, 58:267–288, 1995.
- [59] H. Zou. The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, 101:1418–1429, 2006.
- [60] C. Fraley and T. Hesterberg. Least angle regression and lasso for large datasets. Technical report, Insightful Corporation, 2007.
- [61] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *The Annals of Statistics*, 32:407–451, 2004.
- [62] B. Krishnapuram, L. Carin, M. Figueiredo, and A. Hartemink. Sparse multinomial logistics regression: Fast algorithms and generalization bounds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27:957–968, 2005.
- [63] K. Lange, D. Hunter, and I. Yang. Optimization transfer using surrogate objective functions. *Journal of Computational and Graphical Statistics*, 9: 1–20, 2000.
- [64] D. McRuer, D. Weir, H. Jex, R. Magdaleno, and R. Allen. Measurement of driver-vehicle multiloop response properties with a single disturbance input. *IEEE Transactions on Systems, Man and Cybernetics*, 5:490–497, 1975.