Theses & Dissertations

http://open.bu.edu

Boston University Theses & Dissertations

2015

Communicating through motion in dance and animal groups

https://hdl.handle.net/2144/15193 Boston University

BOSTON UNIVERSITY COLLEGE OF ENGINEERING

Dissertation

COMMUNICATING THROUGH MOTION IN DANCE AND ANIMAL GROUPS

by

HASAN KAYHAN ÖZCİMDER

B.S., Gazi University (Ankara), 2007,M.S., Boston University, 2011

Submitted in partial fulfillment of the

requirements for the degree of

Doctor of Philosophy

2015

© Copyright by HASAN KAYHAN ÖZCİMDER 2015

Approved by

First Reader	
	John Baillieul, Ph.D. Professor of Mechanical Engineering, Professor of Electrical and Computer Engineering Professor of Division of Systems Engineering
Second Reader	
	Hua Wang, Ph.D. Associate Professor of Division of Systems Engineering
Third Reader	
	Calin Belta, Ph.D. Associate Professor of Mechanical Engineering
Fourth Reader	
	Mac Schwager, Ph.D. Assistant Professor of Mechanical Engineering

Acknowledgments

I would like to acknowledge my advisor Dr. John Baillieul for his wise advice, guidance and support throughout all of my studies which makes it possible to accomplish this PhD thesis. I am deeply grateful for all his patient help.

I would like to thank Dr. Calin Belta, Dr. Mac Schwager and Dr. Hua Wang for kindly accepting to be in my defense committee.

I am also thankful to the people in Mechatronics Laboratory for sharing their experience and providing feedback about my work. I would like to thank Dimitar Baronov, Lester Mccoy and Dhananjay Raughathan for help to ease my adaptation to the new academic environment in Boston University. I would like to acknowledge Dimitar Baronov for sharing his knowledge and assistance to improve my projects, Ryan Hunter and Shuai Wang for their help to develop the experimental framework, Georgi Gospodinov and Ally Benett for participating to my experiments as Salsa dancers. I would like to thank Zhaodan Kong for his advice and contributions for the project.

Support from Air Force Office Scientific Research and Office of Naval Research and Turkish Ministry of Education is gratefully appreciated. I would like to thank all of family members for their support and my father Dr. Mustafa Ozcimder for his patient advice and guidance.

COMMUNICATING THROUGH MOTION IN DANCE AND ANIMAL GROUPS HASAN KAYHAN ÖZCİMDER

Boston University, College of Engineering, 2015

Major Professor: John Baillieul, PhD, Professor of Mechanical Engineering, Professor of Electrical and Computer Engineering Professor of Division of Systems Engineering

ABSTRACT

This study explores principles of motion based communication in animal and human group behavior. It develops models of cooperative control that involve communication through actions aimed at a shared objective. Moreover, it aims at understanding the collective motion in multi-agent models towards a desired objective which requires interaction with the environment. In conducting a formal study of these problems, first we investigate the leader-follower interaction in a dance performance. Here, the prototype model is salsa. Salsa is of interest because it is a structured interaction between a leader (usually a male dancer) and a follower (usually a female dancer). Success in a salsa performance depends on how effectively the dance partners communicate with each other using hand, arm and body motion. We construct a mathematical framework in terms of a Dance Motion Description Language (DMDL). This provides a way to specify control protocols for dance moves and to represent every performance as sequences of letters and corresponding motion signals. An enhanced form of salsa (intermediate level) is discussed in which the constraints on the motion transitions are described by simple rules suggested by topological knot theory. It is shown that the proficiency hierarchy in dance is effectively captured by proposed complexity metrics.

In order to investigate the group behavior of animals that are reacting to environmental features, we have analyzed a large data set derived from 3-d video recordings of groups of *Myotis velifer* emerging from a cave. A detailed statistical analysis of large numbers of trajectories indicates that within certain bounds of animal diversity, there appear to be common characteristics of the animals' reactions to features in a clearly defined flight corridor near the mouth of the cave. A set of vision-based motion control primitives is proposed and shown to be effective in synthesizing bat-like flight paths near groups of obstacles. A comparison of synthesized paths and actual bat motions culled from our data set suggests that motions are not based purely on reactions to environmental features. Spatial memory and reactions to the movement of other bats may also play a role. It is argued that most bats employ a hybrid navigation strategy that combines reactions to nearby obstacles and other visual features with some combination of spatial memory and reactions to the motions of other bats.

Contents

1	Intr	oduct	ion	1
	1.1	Motiv	ation	1
	1.2	Prior	Work	3
	1.3	Contr	ibutions	8
2	Dar	nce as	a Prototype for Emergent Group Behavior	11
	2.1	Salsa	and Some Definitions	12
	2.2	The C	Complexity and Artistic Merit of a Dance Performance	13
		2.2.1	Artistic Expressivity of Dance Phrases	17
		2.2.2	Energy Expenditure to Measure Artistic Quality	20
	2.3	Gener	ating Advanced Level Dance Based on Topological Constraints	24
		2.3.1	The Rudiments of Knot Theory	24
		2.3.2	Leader's Decision Making and Admissible Move Transitions .	32
		2.3.3	Complexity Merit of Intermediate Level Dance	41
		2.3.4	Transition from Intermediate Level Salsa to Extended Interme-	
			diate Level Salsa	43
	2.4	Finite	State Machine Modeling of Salsa	51
		2.4.1	Transition Model Representation of a Dance Pair	51
		2.4.2	Optimized Transitions for Salsa Choreography	57
	2.5	Measu	uring the Performance of a Leader in Collective Behavior	65
	2.6	A Rob	potic judge to Evaluate Group Motion	68
		2.6.1	A Modified Transition Model for Dance Move Recognition	69

		2.6.2	Measuring the Performance of the Robotic Judge	71				
3	Coll	lective	Group Motion in Bat Cave Emergence	77				
	3.1	The F	light Behavior of <i>Myotis velifer</i> Bats	78				
		3.1.1	Experimental Procedure for Recording Bat Trajectories	79				
	3.2	Analys	sis of the 3D Myotis velifer Trajectories	80				
		3.2.1	Motion Control Using Optical Sensor Feedback	86				
	3.3	Leader	Follower Interactions of a Paired Bat Flight	96				
		3.3.1	Virtual Loom	98				
		3.3.2	Statistical Analysis of the Bat Emergence	102				
	3.4	Genera	rating Synthetic Bat Trajectories					
	3.4.1 Are Bats Pursuing?							
		3.4.2 Steering Law For Following						
		3.4.3	Simulation Results	109				
		3.4.4	Is Pure Following Strategy Sufficient?	109				
		3.4.5	Integrated Strategy: Spatial Memory Fused with Reactions to					
			Environment and Other Bats	110				
4	Con	clusio	ns and Future Work	115				
Aj	ppen	dices		120				
Re	efere	nces		126				
Cı	urric	ulum V	Vitae	133				

List of Figures

$2 \cdot 1$	On the left hand side the initial pose of a salsa pair is illustrated.	
	On the right hand side, leader's (male dancer) and follower's (female	
	dancer) foot prints are shown	13
$2 \cdot 2$	The illustration of the leaders' signals to communicate move B (Right	
	Turn) and move C (Cumbia Step) to the follower. The leader dancer	
	raises his left hand up (S_B) in order to signal a 360° turn for the	
	follower. In Cumbia step, the leader releases the hands and rotates	
	(S_C) to signal diagonal step backward	14
$2 \cdot 3$	The linear regression plot of the average judges' scores as a function	
	of the computed symbol frequency. Blue circles represent symbol fre-	
	quency complexity and Judges' average ranking for each dance sequence	17
$2 \cdot 4$	The illustration of a Khepera III robot in x-y plane. θ is the angle	
	between the robots' direction and horizontal axis. x_0^B and y_0^B are the	
	initial positions and superscript B represents Bob (leader robot). $\ . \ .$	22
$2 \cdot 5$	Initial position of the robotic dance pair. Red and Black coloring are	
	used to distinguish follower and leader robot, respectively. \ldots .	23
$2 \cdot 6$	Over and under pass information in 2-D regular diagrams	25
$2 \cdot 7$	Three Reidemeister moves preserving the link equivalence	26
$2 \cdot 8$	Four possible crossings between the link edges	26
$2 \cdot 9$	Linking numbers for some of the 2-component links	27
$2 \cdot 10$	Whitehead link with $\#(L_1, L_2) = 0$	27

$2 \cdot 11$	Investigating one crossing to calculate Alexander polynomial of a knot	28
$2 \cdot 12$	The enumerated regions and crossings for the Hopf link in order to	
	calculate the Alexander matrix	30
2.13	Unknotting the trefoil: The cutting and splicing operation locally changes	
	the over and under passes at a crossing	31
$2 \cdot 14$	Orientation preserving knot sum operation for the links ${\cal L}_1$ and ${\cal L}_2$	31
2.15	The link diagram of L with its components L_1 and L_2	32
$2 \cdot 16$	The knot sum of the links L and F	33
$2 \cdot 17$	The illustration of the initial and final poses and link diagram repre-	
	sentations for move B in Intermediate Level Salsa $\ldots \ldots \ldots \ldots$	34
2.18	Initial and final link diagrams for eleven moves in Intermediate Level	
	Salsa (ILS)	35
$2 \cdot 19$	10 moves in which there is follower rotation	37
$2 \cdot 20$	The cut-and-splice operation of Proposition 1 that transforms the end-	
	ing to the starting link diagram (black circle in the figure is the crossing	
	at which the cutting operation is applied in order to alter the over and	
	under passes of the link segments).	38
2.21	Transition from move ${\cal B}$ to the move ${\cal M}$ with link diagram representations	39
2.22	The allowable transitions for each move in ILS based on the physical	
	constraints	40
2.23	Enumerated regions to calculate Alexander matrix for two component	
	link diagram	44
2.24	The knot sum of the links L and F	45
2.25	The knot sum of the links L and F in group $3. \ldots \ldots \ldots \ldots$	45
2.26	Adding a crossing to the regions associated to the original knot	46

2.27	Cut-and-splice operation in topological knot theory applied in the black $\hfill \hfill	
	circled crossing in order to switch over and under passes on the final	
	link diagrams of the moves K, B	48
2.28	Cut-and-splice operation applied to the group 3 links	49
2.29	Transition graph for Extended ILS. The red arrows illustrate the tran-	
	sitions to the three new moves	50
2.30	A representation of a humanoid robot as a spatial stick figure with 8	
	links. In the right hand view, the body roll angle α_b , the leg pitch	
	angle β_l and the arm pitch angle β_a are depicted	52
2.31	Finite state machine representation of beginner salsa performance (BSP)	55
2.32	Finite state machine representation of Intermediate Level Salsa (ILS)	56
2.33	Example deconstruction of 20 letter sequence	58
2.34	The energy and phrase complexity values of the all of the possible	
	phrases in agents' look up table. The blue circles represent the calcu-	
	lated phrase complexity and energy values for each 4-letter phrase in	
	BSP. The red circle represents the phrase with highest energy (CCCC)	
	and green circle is with the highest complexity (ABCD)	62
2.35	The table illustrating the energy and phrase complexity values for four	
	sequences generated by the real dancers	64
2.36	Multi-input Erasure Channel designed for the agents in model (2.38)	66
2.37	A Humanoid robot that represents an actual dancer. The red circles	
	represent five critical points that are named as the Body, Left/Right	
	arm and Left/Right Leg. The number of critical points are chosen to	
	distinguish the moves in Beginner Level Salsa and Intermediate Level	
	Salsa	70

2.38	A robotic judge scheme to evaluate salsa. The robotic judge involves	
	two components: An observation component and an evaluation com-	
	ponent	72
2.39	The least squares regression plane with coefficients estimated from the	
	previous data (Section 2.2). The x-coordinate is the average phrase	
	complexity of a dance sequence (bits), the y-coordinate is the total	
	energy consumed by the dancers (hectometers) and the z-coordinate is	
	the associated score assigned by the human judges.	74
$2 \cdot 40$	A snapshot of the User Interface of the robotic judge that uses Kinect	
	C++ algorithm to track the critical points on the stick figure represen-	
	tations of the dancers. Recognized dance move is shown on the right	
	bottom corner to the user.	75
2.1	Mustic valifor Bat Credit: photo by Boror W. Barbour	78
3 J	Trajectories of 405 bats along with the the positions of the three com	10
0.7	Trajectories of 405 bats along with the the positions of the three cam-	
	eras, vine and the pole. Black triangles represent the key features in	
	the environment such as the trees and the big tree branches	81
3.3	Smoothed bat trajectories illustrated by the color coding to represent	
	their distinct reactions to the vine and the pole $\ldots \ldots \ldots \ldots \ldots$	82
$3 \cdot 4$	Mean trajectories for six subgroups of bats	84
3.5	The simulation of the variance ellipsoid for four subgroups of bats $\ .$.	85
$3 \cdot 6$	Optical flow of feature images for a sideward-looking imaging system	86
3.7	Maximizing time-to-transit when flying trough two key features \ldots	88
$3 \cdot 8$	Optical flow sensing can be used to guide the idealized vehicle to main-	
	tain a constant distance (dashed line) from three noncolinear obstacles	
	A, B, and C .	92
3.9	The control laws used to produce animal like trajectories	94

- 3.10 Mean bat trajectories (solid curves), synthesized trajectories based on Hypothesis 1 (curves with circles) and synthesized trajectories based on Hypothesis 2 (curves connecting squares). Features used for navigation under Hypothesis 1 are marked by black circles. The subset of features used under Hypothesis 2 are marked by circles with green interior. Red and blue curves correspond to groups of bats passing the vine and the 953.11 Flight path statistics for 39 M. velifer are depicted. The red curve is the mean trajectory, and the blue ellipses (centered on the mean trajectory) represent a dispersion of one standard deviation. Two obstacles, a vine and a pole, are denoted as circles. The triangles are visual features in a wooded area (mostly tree branches) and the dotted lines define the edges of the wooded area. 97 3.12 Frenet frame representations of the leader and the follower together with the follower's side-looking eye. L and F are the center axis points of the eyes of the leader and the follower, respectively. |f| is the focal length distance from the lens to the focal plane (retina). L' is the image point corresponding to L. α is the angle between \boldsymbol{x}_l and \boldsymbol{x}_f . 99 3.13 The sampled rate parameter $\bar{\theta}(t)$ of the subset $\{t_i \in [t, t+T]\}$ with $t \in [0, 450]$ and T = 120 seconds. The time axis corresponds to the

whole duration of the recording period with 0 corresponding to the

3.16	Flight path statistics of G_1 and G_2 are depicted. The red (green) curve	
	is the mean trajectory of G_1 (G_2). The blue (black) ellipses (centered	
	on the mean trajectory) represent a dispesion of one standard deviation	
	of G_1 (G_2).	106
3.17	Analysis results for the pair shown in Fig. 3.14: the baseline direction	
	$m{r}/ m{r} $ (blue) and the angle between the baseline direction $m{r}/ m{r} $ and	
	the follower's heading \boldsymbol{x}_{f} (black). Both are represented as angles. For	
	instance, the blue curve is computed by $\tan^{-1}(p_2/p_1)$ with p_1 and p_2	
	being the first and second component of $r/ r $	107
3.18	Synthesized trajectories with the follower using control law (3.21) .	
	Dots indicate starting locations. The black lines connect the pair's	
	corresponding locations at different time slices	110
3.19	Actual bat trajectories (red: leader bat, green: follower bat) and syn-	
	the sized follower trajectories (purple: based on control law (3.21), blue:	
	based on the integrated strategy)	111
3.20	100 synthesized trajectories based on the new integrated strategy. La-	
	beled features are the ones that are assumed being memorized by the	
	bats	112
3.21	The plot of the mean path and variance ellipse of 100 synthesized	
	trajectories	113
·1	The Foot Work of the Dancers and Performance of the Non-Holonomic	
	Robots	121
$\cdot 2$	The Foot Work of the Dancers and Performance of the Non-Holonomic	
	Robots	122
.3	Initial and final poses for eleven moves in Intermediate Level Salsa (ILS)123
$\cdot 4$	Initial and final poses for eleven moves in Intermediate Level Salsa (ILS)124
$\cdot 4$	Initial and final poses for eleven moves in Intermediate Level Salsa (ILS)124

 $\cdot 5$ $\,$ Dance sequences performed by beginner and expert level dancers $\,$. $\,$. $\,125$

Chapter 1

Introduction

1.1 Motivation

This dissertation explores the dynamics of animal and human collective behavior in order to understand the interactions between individuals as well as their adaptation to the uncertainties in the environment. Emergent behavior of biological models is an increasingly popular topic among the field of biology, robotics and control with a broad range of applications [1, 2, 3]. In the study [1], the authors discuss the formations generated by schools of fish to protect the group from the predators. A colony of *Bacillus subtilis* bacteria is investigated in [2] and it is illustrated that the colony forms closely packed dynamic clusters for cooperative group motion. The study [3] discusses honey bee interactions to communicate the location of the food sources around a hive (for this work, the Nobel Prize in Physiology or Medicine was awarded to Karl von Frish in 1973). These biological models have motivated researchers to develop new techniques which efficiently reveal the rules of local member interactions driven by a shared group objective.

The application of the rules governing biological models has also received a great deal of attention [4, 5, 6]. Moreover, there are ongoing studies experimenting with multi-agent systems, which cooperate with human operators [7]; a proposed improvement to fully autonomous systems. For this class of problems, the main concern is to construct an artificial intelligence that can imitate a biological model in terms of group members' decision-making and their performance in execution. Even then, measuring the quality of a performance is quite challenging since it requires the construction of quantitative metrics to evaluate an execution. These challenges have motivated us to study and resolve the issues associated with such systems.

This dissertation focuses on groups performing shared tasks in which the group members communicate with one another by means of gestures and body motions. One motivating example is maintaining stealthy formations in military applications, where a group of robots must avoid direct channel communication to prevent the enemy from decoding the teams' mission. This phenomenon is also inherent in team athletics such as basketball, soccer or American football. The manager generates offensive/defensive plays in which players communicate through motions (positioning) and gestures (team leader signals) to prevent the opponent team from discerning the hidden strategy. This is achieved through a predefined library of motion signals which are used by players during the game. A similar approach is taken when designing communication protocols between underwater vehicles. The vehicles use transmitted motion signals in lieu of radio signals that are attenuated by water resistance over long distances [8].

In what follows, we investigate human and animal collective behavior in two interesting problem settings: dance and bat cave emergence. Dance is an accessible medium to study communication through motion within groups. There exists the notion of a leader, who is responsible for transmitting motion signals to a follower (his dance partner), and she is in turn responsible for executing a response based on her perceptions and best estimate of the leader's intentions. Besides accomplishing these objectives, a dance pair is also required to execute an artistically appealing performance. Hence, a dance model is perceived as a novel example to study human group behavior in terms of the leader's decision making and the dancers' internal interactions for accomplishing the shared task. We further extend the analysis with a second prototype, which involves *Myotis velifer* bats' group interactions for navigation, and their reactions to environmental features when emerging from a cave. These bats emerge in sequences of small groups that typically contain no more than three or four individuals, and they thus provide ideal subjects for studying leader-follower behaviors in nature.

1.2 Prior Work

Human Motion Analysis

In the 1920s, Rudolf van Laban created a notation system to describe and illustrate the physical motions of the human body [9, 10]. Similar to the musical notes written on a five-line staff, Laban introduced the *Labanotation* as a rich notational system for transcribing and analyzing the moments of dance. Laban's model describes the part of the body that executes the motion, its direction, level, duration and quality. In order to define the 'quality' of a movement, the Labanotation uses four effort parameters that are *Space*, *Time*, *Weight* and *Flow*. For instance, *Weight* parameter identifies the strength of a move and *Time* parameter distinguishes a sudden (quick) move from a sustained one. His studies have become a fundamental source for designing robots that can imitate human motion [11].

The studies [12], [13] and [14] illustrate the uses of Laban's effort analysis as a base to qualify the artistic merit of ballet moves. Distinct motions from a warm up routine are extracted for representing ballet as a transition model [12]. The angles between the dancer's limb joints are defined as the states and the warm up moves are represented by the state transitions. It is shown that certain ballet moves have to be followed by a subset of allowable moves from a library based on the physical constraints of the human body. By integrating the motion constraints as propositions in temporal logic formulas, the authors generated controlled dance sequences executed by a humanoid robot and compared the results with the uncontrolled ones. As a result, the controlled synthetic dance moves are shown to be better representations of humanlike dance moves [13]. The authors have extended the work by incorporating Laban's effort parameters as weight matrices for a linear quadratic optimization problem to generate stylistic robot motions [14].

Similar to the ballet study, in Chapter 2 of this dissertation, a performance art, salsa, is represented with a transition system in which states are defined by initial and final poses of a dance pair and transitions are defined by the physical dance moves between the poses. However, different from ballet, sals involves a pair of dancers who must interact with one another in order to perform. Hence, the transition system must include multiple finite state machines with a communication channel through which the leader transmits his move decisions to the follower. We adapt terminologies from the *Motion description languages* in order to formally define the moves in salsa. A salsa dance sequence is discretized into moves executed in eight musical beats and each move is assigned a distinct letter from an alphabet. This mathematical framework is similarly presented in the work [15] which models puppet motion control and choreography design by using a motion description language in order to discretize human-like motion primitives into less complex submoves. Each submove is derived by a control law, which is a solution of an optimization problem with time and energy scale parameter constraints. The study suggests that the concatenation of simpler submoves generate human-like movements that can be executed by a puppet.

In [16] and [17], human flocking behavior is modeled with a graph theoretic approach. In an experimental set up, a group of dancers are asked to walk inside a room while applying a simple set of rules (following at most two dancers and keeping an arm distance with each other) in order to generate group cohesion and repulsion.

The video recordings are used to compute the corresponding graph representation of the group in each instant of time. The features of the graph are extracted in order to study the local interactions between the dancers and the group's emergent behavior. One relevant observation was the emergence of natural leaderships in the group even though there was no leader/follower assignments.

Animal Group Behavior

A part of this document discusses navigation strategies and leader-follower interactions in bat cave emergence. Similar to the human individual/group motion analysis, animals are also widely studied in the literature [1, 3, 18]. In [3], it is shown that honey bees perform dance-like motions for communicating the location and the distance of a food source. The Waggle Dance of honey bees includes straight and oscillatory motions. The orientation of a straight motion carries information about the position of a food source relative to the sun, while the duration of the oscillation transmits information about its distance. In [1], fish school behaviors in the presence of predators are studied. The authors explain how individual interactions between neighbors drive group formations to protect the school from predators. Similarly in[18], group interactions in a flock (murmuration) of starlings are studied. It is revealed that reacting to six or seven neighbors is optimal for a starling to maintain the formation while executing simultaneous individual motions.

These results have inspired biologists and control engineers to study and understand how animals use sensory feedback to react to stationary environmental features and other animals in nearby airspace. In their work [19], the authors study the behavior of a group of Brazillian free-tailed bats, *Tadarida brasiliensis*, as they emerge from a cave. From the statistical analysis of 3D trajectories, the authors reported the number of bats in emergence through time, bats' average flight speed and average distance to the neighbors. The results suggest that on average, bats typically do not fly directly above or below each other. In Chapter 3, we seek to answer two questions inspired by these studies:

- What sensory information do the bats use for navigation or alignment?

- Is there any emergence of natural leadership in the group?

We have conducted experiments in order to record the 3D trajectories of *Myotis velifer* bats as they emerge from a cave. This particular species flies at lower altitudes and tends to navigate through feature rich environments such as woods and bushes. The recorded trajectories are analyzed to determine bats' navigation strategies and their leader-follower interactions.

Other studies-e.g. [20, 21], focus on the navigation strategies of bees and ants which use optical flow sensing while traveling. In [20], authors investigate the visual control of honey bees for height and speed of flight. In an experimental study, trained honey bees are flown through a tunnel in which the visual features are controlled. The study has two major conclusions; 1. The bees regulate their flight speed by keeping the velocity of the image in their visual cortex constant. 2. They use visual cues from the ground to control their height during flight. In a similar study [21], the authors conducted experiments in which trained ants use a wall as a landmark for navigating to a food source. The height of the wall is manipulated through the experiment. From the recorded ant trajectories, it is concluded that the ants tend to maintain a desired distance to the wall by keeping the image of the top of the wall in a particular retinal elevation.

Robotic Applications

Lessons learned from animal and human collective motion have inspired applications to build bio-inspired robotic systems [4, 12, 22, 23]. In [12], the constructed framework (the framework is explained in the Human Motion Analysis section) for the ballet automation is implemented on a simulated humanoid robot. During the robot's ballet performance, its decision making is investigated using a given set of predefined motions. The primary variable is whether the robot performs ballet moves with/without previously defined physical or artistic motion constraints. The soft and hard specifications are used to define physical and artistic dance constraints. The constraints are then incorporated into the model by using temporal logic formulas. Without any soft or hard specifications the authors observed that the robot executes physically unfeasible motion sequences. Using only soft specifications results as generating feasible but not artistically appealing moves. However, the robot performs more human-like motions when soft and hard specifications are used simultaneously.

The study [22] analyzes communication through motion in case of wheeled robots with non-holonomic motion constraints. As individual objectives, two mobile robots are required to navigate from an initial position to a desired location. A leader robot transmits signals by using periodic motions. The follower robot adjusts its speed based on the perceived signal in order to keep its distance constant to the leader. The authors propose control laws to achieve these individual and shared group objectives. In another application [4], multiple quadcopters perform a choreographed dance. Each quadcopter executes a periodic trajectory from a library that is synchronized with the musical beats. Soccer formations of robotic teams are other popular application for studying the communication through motion in emergent behavior. In [23], the implicit communication protocols driven by the soccer game objectives are explored. The authors discuss the relative positioning of each robot in a formation for ball movement.

This thesis summarizes a study of motion-based communication in the context of dance and *Myotis velifer* emergence analyses. In Chapter 2, mobile and humanoid robots are used to study the rules that drive the leader-follower behavior in salsa. To evaluate the success of a robotic dance execution, we propose two metrics for measur-

ing the artistic merit of the moves. These metrics evaluate the energy expenditure and the artistic expressiveness of dance phrases. This approach is different from Laban's effort parameters [9] since our metrics evaluate a phrase (sequence of moves) in lieu of a single move. By incorporating these metrics into a transition system representation of salsa, we solve a forward and an inverse problem. The forward problem involves generating a robotic dance that is optimal in terms of the proposed metrics. This is similar to multi-agent robotic applications that minimize the energy consumption when performing a mission [24]. The inverse problem is a third party robotic evaluation of a group execution to determine the 'perceived artistic merit'. By using the results from the bat group behavior, in Chapter 3, we present generated synthetic bat-like trajectories in a simulated feature rich environment by a robot with nonholonomic motion constraints. The robot is driven by the rules extracted from the bats optical flow sensing that uses the land marks (key features) in the environment for navigation.

1.3 Contributions

This dissertation is organized into four chapters. Chapter 2 discusses the construction of the mathematical framework for studying the dance pair interactions in salsa. One contribution of the thesis is the formal exploration of physically admissible move transitions based on rules derived from topological knot theory. The interrelation between dance moves and the link diagram representation of a dance pair is discussed. Moreover, the *Alexander polynomial* of each link diagram is computed to describe a dance move by polynomial function manipulation. We propose two mathematical metrics that measure dancers' energy expenditures and the quality of the executed dance phrases. We generate robotic dance sequences by solving an optimization problem that minimizes robots' energy consumption subject to the artistic constraints. We discuss what constitutes a 'better' leader by mapping various skill level leaders' motion signals to a physical channel to compute channel capacity. We observe that channel capacity is strongly correlated to the expertise of the leader. The studies are extended by building an automated judge that tracks the moves of a dance pair in order to deconstruct a dance performance into move sequences and evaluates a score with respect to the predefined metrics. Finally, we discuss the results of an experiment in which a salsa pair performs in the view of the robotic judge. The video recordings of the dances are also shown to human judges. A strong correlation is observed between robotic and human judges' evaluations which implies our autonomous robot performs well in evaluating a group execution.

In Chapter 3, emergence of *Myotis velifer* bats from a cave is discussed in terms of their individual and group behaviors. We explain the details of the experimental procedure for recording the three dimensional bat trajectories in their natural habitat. Smoothing and filtering of the raw bat trajectories are carried out using cubic spline smoothing. The smoothed trajectories are classified into the subgroups with respect to the bats' reactions to obstacles in the flight path. In order to understand bats' decision making, a mobile robot is used to navigate through a simulated environment which is identical to the bats' flight corridor. Robots' trajectory is driven by an optical flow based control law that use a new concept *time-to-transit* which is a quantity computed by the animals' visual cortex. The synthetic trajectories generated by the robot suggest that the bats use environmental features as well as the special memory for navigation.

For the analysis of leader-follower interactions, we define a leader as the bat which flies at a distance in the range of follower bats' field of view or echolocation calls. The data shows that the flight behavior of a follower bat is influenced by the flight behavior of a leader bat. Thus, we modify the concept time-to-transit to capture the geometrical configuration of the leader-follower pair and introduce *virtual loom*. The mobile robot's trajectory is driven through the simulated environment by optical flow based control laws that incorporates time-to-transit, virtual loom and special memory. It is concluded that the generated synthetic bat trajectories resemble to the original bat flight paths.

In Chapter 4, we will conclude the thesis and propose future research directions. Parts of the discussion that follow are based on our previously published work [25, 26, 27, 28, 29, 30, 31].

Chapter 2

Dance as a Prototype for Emergent Group Behavior

This chapter summarizes work to understand various aspects of the communication that occur through the movements of partners in dance. By adopting terminology from motion description languages, we deconstruct an elementary form of the wellknown popular dance, salsa, in terms of four motion primitives (dance steps). We introduce two metrics in order to measure the energy expenditure of dancers as well as the artistic merit of a dance performance. The analysis is extended to an enhanced form of salsa in which the upper body motions play a major role in the steps of the dance. The dance move transitions are described by rules inspired by topological knot theory. We present the solutions for a forward and an inverse problem. A forward problem is defined as generating autonomous dance sequences by humanoid robots based on the energy and artistic constraints. An inverse problem formalizes the evaluation of a group execution by a third party autonomous judge. At the end of the chapter, we present metrics that measure the proficiency level of a leader and his 'success' in transmitting motion signals to the other group members for accomplishing the shared objective of artistic expression.

2.1 Salsa and Some Definitions

Salsa is a Latin dance form which is popular around the World. Different from other dances, which are generally the result of years of practice, two dancers without any prior practice can perform and enjoy salsa. This is achieved by a universal set of moves and communication signals that can be easily learned by both the leader and the follower dancers. Hence, equipped with the prior knowledge of the moves, the dancers can perform salsa as long as the leader as a decision maker executes the correct gestures to communicate and the follower correctly estimates the upcoming moves during the dance. Salsa can be seen as a particular type of collective motion in which the collective goal is to perform an artistically appealing dance while each individual has to fulfill his/her role as a leader or a follower.

In order to study salsa formally, we are going to build mathematical models by using two key features of salsa. The first feature is that every distinct move in salsa has to be performed in eight musical beats $(t_i, i = 1, ..., 8)$. This enables us to discretize a salsa performance into moves of eight beat intervals (starting from an initial position Fig. 2.1) and to assign a letter to each move from a finite-sized alphabet, $\mathcal{M} := \{A, B, ...\}$. By this method, a salsa performance can be represented as a concatenation of letters (one might think this as similar to a DNA sequence in biology).

The second feature of salsa is the characteristics of leader-follower interaction. The leader (generally a male dancer) is responsible to communicate with the follower (generally a female dancer) by using gestures in order to signal his move decisions. We use $S := \{S_A, S_B, S_C, ...\}$ to represent the collection of the signals communicated by the leader to the follower to signal the corresponding moves from the set \mathcal{M} (Fig. 2·2).

As is always the case in performing arts, there are distinct levels of proficiency in



Figure 2.1: On the left hand side the initial pose of a salsa pair is illustrated. On the right hand side, leader's (male dancer) and follower's (female dancer) foot prints are shown.

salsa. In order to conduct the analysis of collective behavior in dance, we begin with investigating a simpler version of salsa, *Beginner Level Salsa* (BLS), that uses only four basic dance steps.

Definition 2.1.1 The move set $\mathcal{M}_{BLS} := \{A, B, C, D\}$ involves four distinct moves: A-Basic Salsa Move, B-Right Turn, C-Cumbia Step and D-Cross Body Lead.

Moves in \mathcal{M}_{BLS} are referred to as the *fundamental salsa moves* that are assumed to be the foundations of the advanced level moves. Leader and follower motion primitives to generate the moves in \mathcal{M}_{BLS} are illustrated in Appendix A.

2.2 The Complexity and Artistic Merit of a Dance Performance

The artistic content of formalized movements that occur in dance is central to what must be expressed in the motion-based language associated with each dance vernacular. It would seem natural, then, to develop a formal means of transcribing basic motion primitives for dance, but attempts to do this have not led to widespread use



Figure 2.2: The illustration of the leaders' signals to communicate move B (Right Turn) and move C (Cumbia Step) to the follower. The leader dancer raises his left hand up (S_B) in order to signal a 360° turn for the follower. In Cumbia step, the leader releases the hands and rotates (S_C) to signal diagonal step backward.

among dance professionals. Perhaps the best known effort in this direction was the development in the 1920's of *labanonation* [9]. Rolf Von Laban attempted to develop a scripting language that was sufficiently expressive that all human movement could be described and recorded on paper. This has never been widely used, probably because in its attempt to be universal, it became complex and nonintuitive (This is supported by noting the "more than 700 symbols that indicate parts of the body, direction, levels, and types of movement and the durations of each action". Quoted from the web page [32]). We avoid dealing with such expressive complexity by restricting our attention to BLS.

In an attempt to understand how people perceive the artistic merit of a dance performance, two dancers were asked to perform a number of short salsa segments using the four basic dance primitives in \mathcal{M}_{BLS} . Digital video recordings of the salsa segments were shown to twenty "judges" who were asked to rank the performances in order of artistic merit. The judges included both trained dancers as well as people with no formal training in dance. All judges were instructed to use standard criteria in their rankings, including artistic content, dance routine difficulty, partner synchronization, and complexity of the choreography. Ten dance sequences, each comprised of 23 basic dance primitives were selected to be ranked by each of the judges. Using the motion primitives (dance steps) in \mathcal{M}_{BLS} , the ten performances are given in Table 1.

Dance 1	DDADBBBBACCCDDDDDBDAAAA
Dance 2	AAAAAAADDDDDDDDDDDDBDBB
Dance 3	ADBCDACBDADBCDABACDACBD
Dance 4	DBCADBCADBCADBCADBCADBC
Dance 5	ACBDACBDACBDACBDACBDACB
Dance 6	ABCDBCDACDABDABCABADBCD
Dance 7	DBADACBDDBABDDAACDBBDAD
Dance 8	AAAABAAADAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
Dance 9	DBCDCBBDCBDDDBDDDAABCCC
Dance 10	DBDCCBDDBBDDDCCCCABDDDB

Table 1

The average scores of the twenty judges are given in the second row of Table 2. Dance sequence 9 was preferred, while almost no one liked dance number 2. It is noted that although the judges were in substantial agreement regarding dance number 2 (rated as poor) there was comparatively high variance in the judges scores on other dances. Having thus tabulated the judges' rankings, we were led to the question of whether the artistic qualities in terms of which the performances were differentiated could be identified in a precise and even quantitative way.

Dance no.	1	2	3	4	5	6	7	8	9	10
Average										
score by	3.6	1.9	5.1	5.7	7.3	6	6	4.2	7.8	7.3
judges	(2.0)	(1.5)	(3.0)	(2.1)	(1.8)	(2.6)	(2.2)	(3.1)	(2.4)	(2.2)
Symbol										
frequency	1.897	1.403	1.985	1.996	1.996	1.996	1.848	0.927	1.848	1.731
complexity										
Average										
phrase	0.625	0.162	1.8	2	2	1.9	1.5	0.487	1.362	1.362
complexity										
Number										
of phrases	2.322	1.522	2.322	0	0	2.322	2.322	1.922	2.322	2.322
complexity										
Robot										
dance	13727	12945	14326	14567	14547	14248	13349	13181	14627	14647
energy										

Table 2

The late Dennis Dutton identified complexity as one of the four central characteristics of great art [33]. To evaluate the complexity of a sequence of symbols such as those of Table 1, we considered metrics suggested by the well-known *Shannon Entropy*. The simplest possible metric may be arrived at by recording the number of occurrences of each of the symbols in the symbol set $\mathcal{M}_{BLS} := \{A, B, C, D\}$. Each dance is exactly 23 symbols in length, and thus the relative frequency of occurrence of the k-th symbol is $f_k = (\# \text{of occurrences of k-th symbol})/23$. The metric

$$\begin{pmatrix} \text{symbol} \\ \text{frequency} \\ \text{complexity} \end{pmatrix} = -\sum_{k=1}^{4} f_k \log_2 f_k \tag{2.1}$$

is then a measure of the variability of the component steps that make up the dance. Because there are only four symbols involved, the maximum value this measure could take is $\log_2 4 = 2$, which would be attained if each symbol appeared in the sequence equally often. (Since the sequence lengths are all 23, this bound is never achieved). On the other hand, if any single symbol were to appear in all 23 places in the sequence, the complexity (2.1) would have the value 0. When the complexity metric (2.1) is evaluated on the ten dance sequences of Table 1, the values are strictly between the two extremes, and they are given in row three of Table 2. A simple linear regression in which the average judges' scores were regressed on the computed symbol frequency suggests only a modest correlation (See Fig. 2.3). Indeed, the value of the coefficient of correlation for the sequences is only 0.48, indicating a weak correlation. The following section describes some refined notions of complexity that may more faithfully reflect the artistic quality of the sequences.



Figure 2.3: The linear regression plot of the average judges' scores as a function of the computed symbol frequency. Blue circles represent symbol frequency complexity and Judges' average ranking for each dance sequence

2.2.1 Artistic Expressivity of Dance Phrases

It is an interesting exercise to attempt to fit four-state Markov chain models to the symbol sequences of Table 1. While the sequences are long enough and the sets of transitions are rich enough in some cases to construct such models, any model of the dance in which the next step depends only on its immediate predecessor step will probably seem a bit aimless and not reflective of the artistic quality of the sequence of steps that the dance actually contained. As has been noted in the computer music literature, [34], higher order Markov chain models can be used to capture the phrasal nature of music.

As noted above, each of the four motion primitives in \mathcal{M}_{BLS} is executed over a period of eight beats of music. Each phrase is thus eight musical measures in length. Since there are four beats to a measure, it is natural to group the letters in the sequences into four letter phrases (Table 3).

		# phrases
Dance 1	(DDAD)(BBBB)(ACCC)(DDDD)(DBDA)AAA	5
Dance 2	(AAAA)(AAAA)(DDDD)(DDDD)(DDDB)DBB	3
Dance 3	(ADBC)(DACB)(DADB)(CDAB)(ACDA)CBD	5
Dance 4	(DBCA)(DBCA)(DBCA)(DBCA)(DBCA)DBC	1
Dance 5	(ACBD)(ACBD)(ACBD)(ACBD)(ACBD)ACB	1
Dance 6	(ABCD)(BCDA)(CDAB)(DABC)(ABAD)BCD	5
Dance 7	(DBAD)(ACBD)(DBAB)(DDAA)(CDBB)DAD	5
Dance 8	(AAAA)(BAAA)(DAAA)(AAAA)(CAAA)DAA	4
Dance 9	(DBCD)(CBBD)(CBDD)(DBDD)(DAAB)CCC	5
Dance 10	(DBDC)(CBDD)(BBDD)(DCCC)(CABD)DDB	5

Table 3

Several phrase centric complexity metrics can then be considered. One such metric is based on viewing each four symbol phrase as a complete dance sequence in its own right. In terms of the symbol set \mathcal{M}_{BLS} every four letter phrase has a complexity given by equation (2.1) where now $f_k = (\#\text{of occurrences of k-th symbol})/4$. Clearly, there are five possible values that this phrase complexity metric can take on phrases made up of the four letters in \mathcal{M}_{BLS} . They are $0, -\frac{1}{4}\log\frac{1}{4} - \frac{3}{4}\log\frac{3}{4} = 0.811278$, $-\log\frac{1}{2} = 1, -\frac{1}{2}\log\frac{1}{4} - \frac{1}{2}\log\frac{1}{2} = 1.5$ and $\log 4 = 2$ in the respective cases that all letters in the phrase are equal, three letters in the phrase are equal, there are two distinct pairs of equal letters, there are exactly three letters in the sequence, and finally in the case that there are four distinct letters in the sequence. Based on this phrase metric, we prescribe an *average phrase complexity* (W_{ave}) metric for each of the twenty-three letter sequences. Ignoring the final three letters in each sequence, the right hand column in Table 3 lists the number of distinct four letter phrases that make up the dance. The fourth row of Table 2 lists the average phrase complexity of the dance.

A further metric in terms of which to evaluate dance complexity is what we shall call the *number-of-phrases complexity*. This metric is based on the number of distinct phrases and their frequency of occurrence among the first twenty letters in each dance sequence (a number between 1 and 5). The possible values of the number-of-phrases complexity in terms of the appropriately restated formula (2.1) range between 0 and $\log_2 5 \approx 2.344$. The values taken on by this metric for our ten dances are listed in row 5 in Table 2. Note that while dances 4 and 5 have the highest average phrase complexities (being comprised of four distinct letters), they also have the lowest complexity measured in terms of number-of-phrases.

Comparing the average judges' scores with the average phrase complexity showed a discernible correlation, with the coefficient of correlation being 0.75. On the other hand, the number of phrases complexity had no meaningful correlation with the judges rankings (correlation coefficient -0.099). It is interesting to note, however, that a convex combination of these complexity metrics in which the relative weightings are 90% average-phrase complexity and 10% number-of-phrases complexity has a slightly higher value of 0.764 coefficient of correlation with the judges rankings. This metric slightly discounts dance routines that repeat the same four steps over and over. It is also interesting to note that both these complexity metrics are identical on and do not discriminate between dances 4 and 5, and yet the judges had a clear preference for dance 5. There is clearly some aspect of artistic merit that is not captured by the complexity metrics.

2.2.2 Energy Expenditure to Measure Artistic Quality

A major theme in recent work of Wong and Baillieul [35, 36, 37] is understanding the complexity of communicating through a control system in terms of the required control energy. As dance requires physical exertion, it seems natural to compare the dance sequences of Section 2.2 (Table 1) in terms of the amount of energy required to perform them. While energy data was not recorded for the human dancers who performed the ten dance routines, we have done energy calculations on wheeled robots in our lab doing appropriately stylized versions of the same beginner salsa routines. This is achieved by converting the motion description language (MDL) created for the dance into a hybrid system representation for a robotic dance pair.

General MDL is defined as the sequences of control protocols (atoms) to describe a hybrid system in the form [38],

$$\dot{x} = f(x, u), x \in X, u \in U, \tag{2.2}$$

$$\dot{y} = h(x), y \in Y, \tag{2.3}$$

where x is the state, u is the control input and y is the observation and every control law steers the system until a switching condition occurs. MDLe is an extended version of general MDL which provides switching constraints based on sensory feedback [38]. In an MDLe the atoms are in the form (u_i, ϵ_i, t_i) where the index *i* represents the mode the system currently running, u_i is the control to steer the system in the mode i, ϵ_i is the interrupt function $\epsilon : Y \to \{0, 1\}$ and t_i is the timer.

In the rest of the analysis, we refer to the leader and the follower robot as Bob and Alice, respectively. For a robotic dance performance, we discretize each dance move in \mathcal{M}_{BLS} into finite number of *submoves* which are defined as follows.
Definition 2.2.1 η_i (for Alice) and κ_i (for Bob) are the submoves generated by partitioning an eight-beat dance move into distinct motion primitives, where i = 1...m, and m is the number of sub moves.

In this particular case, the index i given in MDLe definition represents the particular submove that the robots are currently executing. Thus, the atom for a submove becomes,

$$(u^A_{\eta_i}, \xi^A(t,\eta)), \tag{2.4}$$

for Alice (follower robot) and,

$$(u^B_{\kappa_i}, \xi^B(t, \kappa)), \tag{2.5}$$

for Bob (leader robot) where $u_{\eta_i}^A$ and $u_{\kappa_i}^B$ are the inputs which may be either open loop or feedback control commands for executing each sub move η_i and κ_i . The interruption function $\xi_i(.)$ is a boolean function of time and the submove being performed.

$$\xi_i^A : R^+ \times \eta_i \to \{0, 1\}, \text{ for Alice,}$$
(2.6)

$$\xi_i^B : R^+ \times \kappa_i \to \{0, 1\}$$
 for Bob. (2.7)

Thus, $\eta, \kappa \in \{A, B, C, D\}$ are represented as the finite set of atoms in the form $(u_{\eta_i}^A, \xi^A(t, \eta))$ for Alice and $(u_{\kappa_i}^B, \xi^B(t, \kappa))$ for Bob, where $i = 1, \ldots, m$.

This framework is used to generate a robotic dance performance by Khepera III robots with non-holonomic motion constraints. If one agent is assumed to be the leader (Bob) and second agent is the follower (Alice) the equations of motion are

$$\dot{x}^B = v^B \cos\theta^B, \qquad \dot{x}^A = v^A \cos\theta^A, \qquad (2.8)$$

$$\dot{y}^B = v^B sin\theta^B, \qquad \dot{y}^A = v^A sin\theta^A, \qquad (2.9)$$

$$\dot{\theta}^B = \omega^B, \qquad \qquad \dot{\theta}^A = \omega^A.$$
 (2.10)

Superscripts B and A represent Bob and Alice, respectively. In the system (2.8)-

(2.10), x and y are the positions of the robots in a 2-D plane. The input pairs $\{v^A, \omega^A\}$ and $\{v^B, \omega^B\}$ are the linear and the angular velocities and θ is the angle between the direction of the robot and the horizontal axis (Fig. 2.4).



Figure 2.4: The illustration of a Khepera III robot in x-y plane. θ is the angle between the robots' direction and horizontal axis. x_0^B and y_0^B are the initial positions and superscript B represents Bob (leader robot).

In this set up, the initial positions of the robotic pair to perform the moves A, B, C, D are illustrated in Fig. 2.5.



Figure 2.5: Initial position of the robotic dance pair. Red and Black coloring are used to distinguish follower and leader robot, respectively.

It is assumed that the robots start from the same initial position with a fixed distance. Robotic dance motion primitives that correspond to the executions of a real dance pair are shown in Appendix A. The distance covered by the robots for each dance move is computed and used as an *energy metric* to measure dancers' energy expenditures in salsa. In the last row of Table 2, the total energy values (as distances in cm) are illustrated for each dance sequence. A strong correlation coefficient (R = 0.8) is observed between the energy consumption and judges rankings. This result suggests that judges evaluate the sequences by the artistic quality of the dance phrases as well as the energy reflected by the dancers in the execution. This phenomenon will be used in the rest of the analysis to understand the proficiency hierarchy in dance. The proposed energy and complexity metrics will be incorporated into a robotic performance for generating 'optimal' dance phrases.

2.3 Generating Advanced Level Dance Based on Topological Constraints

In this section, we consider an enhanced form of salsa, which we refer to as intermediate level salsa (ILS) and in which there are additional motion primitives (dance steps) as well as a physical constraint imposed by requiring the dance partners to maintain hand contact. The size of the alphabet is extended to capture more advanced dance performances in the analysis. In the performing arts, the body poses as well as the arm positions have relevant influence on the artistic expressivity of an execution. Hence, we incorporate rules from topological knot theory to investigate the role of dancers' body motions in leader-follower interactions and leader's decision making for move transitions. We begin with introducing some basic concepts from the knot theory.

2.3.1 The Rudiments of Knot Theory

We recall that *knots* are simple closed curves in \mathbb{R}^3 , and *links* are finite sets of knots that may be entangled with one another. Although interest in knots dates to antiquity, the formal study of knots as mathematical objects may be traced to Vandermonde's 1771 paper "Remarques sur les problèmes de situation", [39]. The modern introduction of polynomial invariants and other algebraic tools has significantly deepened the theoretical foundations, while the simultaneous proliferation of applications to statistical mechanics, molecular biology, and chemistry has secured the place of knot theory as an important mathematical discipline. Knot theory is of interest in the enhanced version of salsa that we shall examine below where the dancers' joined hands both enable artistic expression and constrain the grammar of motion sequences in the dance. For the purpose of our discussion, we shall provide a rudimentary conceptual introduction to the language of knot theory. For any two knots, the linking number specifies how many times each curve (knot) winds around the other. The *linking number* is always an integer, and since the curves are oriented, the linking number may be positive or negative. A simple way to determine the linking number is to "project" the curves onto the plane by an *immersion* $f : \mathbb{R}^3 \to \mathbb{R}^2$. Under this mapping a knot (or link) is in one-to-one correspondence with its image except at double points (called *crossings*) where a distinction needs to be made between the top and bottom segments of the knot (link). (Note, that we do not allow image points of multiplicity higher than two [no triple points, for instance], and all crossings are assumed to be transverse.) To keep track of which segment crosses over and which crosses under, we represent the segment that is under by means of a break, as illustrated in Fig. 2.6. The planar image (with respect to f) of the knot (or link) together with the labeling of "over" and "under" segments at crossings constitutes the *link diagram*.



Figure 2.6: Over and under pass information in 2-D regular diagrams

Two knots (or links) are *equivalent* if there exists an orientation preserving homeomorphism of a neighborhood of the first in \mathbb{R}^3 onto a neighborhood of the second (also in \mathbb{R}^3) such that the second knot (link) is the image of the first.

Well known results of K. Reidemeister [40] have shown that equivalent links may be transformed into one another by a finite sequence of three elementary moves—the



Figure 2.7: Three Reidemeister moves preserving the link equivalence.

so-called *Reidemeister moves*. These are depicted in Fig. 2.7. A type 1 move simply removes or adds a kink, a type 2 move is a separation and a type 3 move is preserving the number of crossings.

Consider two links—colored, say, black and red. Assuming that each is oriented, there are precisely four possible crossings, as depicted in Fig. 2.8.



Figure 2.8: Four possible crossings between the link edges

Let the numbers of each type of crossing be n_1, n_2, n_3, n_4 respectively. The linking

number is then defined to be

$$\#(L_1, L_2) = \frac{n_1 + n_2 - n_3 - n_4}{2}.$$
(2.11)

If L is a link with n components $L_1, L_2, \ldots L_n$ then the linking number of L is [41],



Figure $2 \cdot 9$: Linking numbers for some of the 2-component links

Any two unlinked knots have linking number zero (Fig. 2.9.). However, linking number zero for two knots does not necessarily mean that the knots are unlinked. One well known example is the Whitehead link which has linking number zero but for which the two components are linked (Fig. 2.10). This example is important because



Figure 2.10: Whitehead link with $\#(L_1, L_2) = 0$

it shows that linking numbers do not determine the topology of knots and links. The

related concept of an *unknot* will be important in what follows.

Definition 2.3.1 A knot is called an unknot (trivial knot) if it bounds an embedded disc. (That is to say, it is equivalent to a circle.)

Another knot(link) invariant Alexander polynomial is introduced by James Waddell Alexander [42]. Let us assume that D is a regular diagram representation of a knot (link) L which has v crossings. Then by Euler's theorem [42], every arc of the knot (link) D divides the plane into v + 2 regions including the region around the knot (link) which are represented by $\sigma_1, \sigma_2, \ldots, \sigma_{v+2}$. By investigating one crossing cr_1 locally,



Figure 2.11: Investigating one crossing to calculate Alexander polynomail of a knot

the regions that lie on the left of the underpass arrow will be marked with 'dots' as illustrated in Fig. 2.11. Without loss of generality it can be assumed that each crossing can be rotated to match the orientation as in Fig. 2.11 above which then will be used to calculate the *Alexander matrix* of a knot *L*. The regions σ_j , σ_k are called "dotted top" and "dotted bottom" regions and regions σ_m , σ_l are called "top" and "bottom" regions respectively. In this set up, the linear equation for crossing one (cr_1) ,

$$cr_1(t) = t\sigma_j - t\sigma_k + \sigma_l - \sigma_m, \qquad (2.13)$$

is defined by multiplying the "dotted top" and "dotted bottom" regions by t (where t is a dummy variable without any physical meaning) and -t and top and bottom

regions by 1 and -1 respectively. Calculating the linear equations for each crossing, $cr_1(t), cr_2(t), \ldots, cr_v(t)$, will result as a system of v + 2 variables with v equations. These set of v equations will then be represented by a $v \times (v+2)$ matrix \hat{M}_L (Equation 2.14) such that each row will correspond to a crossing and each column will correspond to a region around the knot(link) L.

$$\hat{M}_L = \begin{array}{c} cr_1 & \cdots & \sigma_{v+2} \\ \vdots & \ddots & \vdots \\ cr_v & \cdots & \cdots \end{array} \right).$$

$$(2.14)$$

By definition, the entries of the matrix \hat{M}_L are ± 1 , 0 and $\pm t$. Furthermore, the Alexander matrix M_L is defined to be the $v \times v$ square matrix obtained from \hat{M}_L by deleting two columns that correspond to any two neighboring regions σ_p and σ_q in the link diagram [42].

Definition 2.3.2 [42] The Alexander polynomial $\triangle(t)$ of a knot (link) L is defined to be the determinant of the Alexander matrix $\triangle(t) = det(M_L)$ which is a function of t and has integer coefficients.

Let us calculate the Alexander polynomial of Hopf Link with the orientations given in Fig. 2.12, as an example.

The linear equations for the two crossings is found as, $cr_1(t) = t\sigma_1 - t\sigma_2 + \sigma_3 - \sigma_0$ and $cr_2(t) = t\sigma_1 - t\sigma_0 + \sigma_3 - \sigma_2$. The matrix \hat{M}_L becomes,

$$\hat{M}_L = \begin{pmatrix} -1 & t & -t & 1 \\ -t & t & -1 & 1 \end{pmatrix}.$$
(2.15)

Two columns correspond to the two neighboring regions are deleted to come up with the 2 × 2 Alexander matrix. If we delete column 2 and 3 (since σ_2 and σ_3 are



Figure $2 \cdot 12$: The enumerated regions and crossings for the Hopf link in order to calculate the Alexander matrix

neighboring regions) the resulting square matrix becomes

$$M_L = \begin{pmatrix} -1 & 1\\ -t & 1 \end{pmatrix}.$$
 (2.16)

Hence, the Alexander polynomial for the Hopf Link L, is the determinant of the matrix M_L which is found as,

$$\Delta(t) = \det(M_L) = t - 1. \tag{2.17}$$

As it is noted above that Alexander polynomial is a knot(link) invariant which is stated in the next theorem.

Theorem 2.3.3 [42] The Alexander polynomial $\triangle(t)$ calculated by using the method presented above differs only by the power of $\pm t^k$ for some integer k for equivalent knots.

We conclude this section by introducing two operations that change the topology of a knot (link). An important operation in knot theory is that of cutting and splicing (Fig. 2.13). This operation can be used to alter the local over and under passing at crossings, and thus, it can be used to change the topology of a knot or link.

The final basic concept of elementary knot theory that we shall make use of is the *knot sum*. Two oriented knots/links L_1 and L_2 can be added together to form



Figure 2.13: Unknotting the trefoil: The cutting and splicing operation locally changes the over and under passes at a crossing.

their sum $L_1 \# L_2$ by placing them side by side and cutting each one once and splicing in two line segments such that the orientation is preserved [9]. This is depicted in Fig. 2.14.



Figure 2.14: Orientation preserving knot sum operation for the links L_1 and L_2

In the following section, these topological concepts will be incorporated into salsa dance model for formal exploration of the dance pair interactions with artistic constraints.

2.3.2 Leader's Decision Making and Admissible Move Transitions

We start our analysis by assuming that in an abstract model a dancer and his/her arms will be represented by a two component link L. The components are an infinitely long cylinder (or line) L_1 (representing the body or torso of the dancer) and a trivial knot L_2 (representing the arms) with the orientations given as in the Fig. 2.15.



Figure 2.15: The link diagram of L with its components L_1 and L_2

It is assumed that the link representing the initial pose of a dance couple (Fig. 2.17.) is a knot sum of the links representing each dancer. In Fig. 2.16, one can see the knot sum L#F of the links L and F with the orientations preserved and its three components, L_1 , L_2 and L_3 . We retain the color coding, blue and red, of the link segments to represent the male (leader) and female (follower) dance partners respectively.

It is easy to conclude from Fig. 2.16 that linking number of the link sun L#F is #(L, F) = 2 and the number of crossings cr = 4. This link diagram above is assumed to be the starting link since it represents the link diagram for the initial pose of the dancers when they start to dance (Fig. 2.1).

We shall address several questions: (1) How many non equivalent links are required to represent the poses that occur in a salsa performance? (2) What are the changes of link attributes corresponding to the physical movements that define the dance steps?



Figure 2.16: The knot sum of the links L and F.

To answer these questions, we define the set of dance moves (motion primitives) that will be the alphabet from which dance sequences of intermediate level salsa (ILS) are constructed. The key distinction that will be drawn with respect to BLS treated above is that in ILS, the dancers never break hand contact. Hence their arms and bodies remain knotted in the sense described above such that the link diagram always has three components. To the basic moves A, B, C, D that we have already defined (and which are pictorially represented in Appendix A) we add seven additional moves $\{J, K, M, N, O, P, T\}$, each of which continues to be performed over the course of eight musical beats. Hence, $\mathcal{M}_{ILS} = \{A, B, C, D, J, K, M, N, O, P, T\}.$ We do not describe the foot movement used to execute these, but rather identify the distinguishing characteristic as simply the beginning and ending poses (Appendix B). It will turn out that the knottedness of the dancers' arms will constrain the sequencing of dance steps (The arm positions that occur in move B are shown as an example in Fig. 2.17). Hence, unlike the beginners' salsa considered in Section 2.1, ILS does not allow complete freedom in the choice of letter sequences. This will, of course, affect the information theoretic metrics that we have used to discuss complexity of dance routines.



Figure 2.17: The illustration of the initial and final poses and link diagram representations for move B in Intermediate Level Salsa

The focus here is understanding how dance movement transitions are constrained by the arm positions. The link diagrams corresponding to the poses illustrated in Appendix B are shown in Fig. 2.18. The dancers' arms will cross in different patterns, and these patterns are shown in the usual way as under- and over-crossings.

Remark 2.3.4 Let q_{ji} and q_{jf} be the initial and final link diagrams representing the initial and final poses p_{ji} and p_{jf} respectively for the moves $j \in \mathcal{M}_{ILS}$. We can classify the links into 3 groups with respect to both the linking number (lk) and number of crossings (cr) (Fig. 2.18.).

- lk = #(L, F) = 2 and cr = 4 (e.g. $q_{Ai}, q_{Nf},...$) : 14 link diagrams (including starting link),
- lk = #(L, F) = 2 and cr = 6 (e.g. $q_{Bf}, q_{Mi},...$): 4 link diagrams,
- lk = #(L, F) = 3 and cr = 7 (e.g. $q_{Jf}, q_{Ni},...$): 4 link diagrams.

Remark 2.3.5 The purpose of the discussion here is to illuminate the constraints on motion sequences that are imposed by the topologies of the links depicted in Fig. 2.18. These constraints are completely determined by the beginning and ending poses of the



Figure 2.18: Initial and final link diagrams for eleven moves in Intermediate Level Salsa (ILS)

dance steps. The significance of the constraints in terms of our complexity metrics will be discussed in the next section. We point out that for motion primitives A, B, C, D, the foot motions of the dancers are shown in Appendix A.

We note that the crossing number (cr) is not a knot (link) invariant, but it will be shown to usefully characterize the poses corresponding to the link diagrams of Fig. 2.18. To understand this correspondence, we define a motion operator that describes the movement of the dancers in the transition from their initial to final pose in each of the eleven steps. As apparent from Fig. 2.18, the motions of the male (blue) and female (red) partners are qualitatively different from one another. The link crossings and uncrossings arise from the female rotating her body through angles of π or 2π with respect to the initial pose. The dance motions from beginning to ending poses in our eleven dance primitive can be labeled as follows in terms of a motion operator, *(.,.):

*(0,0)	\sim	the female partner					
		begins and ends facing the					
		male partner in the move					
		(A, C, D).					
$*(\pi, CW)$	\sim	the female partner rotates					
		by π in the clockwise					
		$\operatorname{direction}(\mathbf{J}, \mathbf{O}).$					
$*(\pi, CCW)$	\sim	the female partner rotates					
		by π in the counterclockwise					
		$\operatorname{direction}(\mathbf{N}, \mathbf{T}).$					
$*(2\pi, CW)$	\sim	the female partner rotates					
		by 2π in the clockwise					
		direction (B, P) .					
$*(2\pi, CCW)$	\sim	the female partner rotates					
		by 2π in the counterclockwise					
		$\operatorname{direction}(\mathbf{K}, \mathbf{M}).$					

Consider move B. This has the follower rotating 2π in the CW direction. The change from the initial pose p_{Bi} to the ending pose p_{Bf} in move B is represented by the notation,

$$p_{Bi} * (2\pi, CW) \to p_{Bf}. \tag{2.18}$$

The *(.,.) operator will be used to describe the rotations in ILS. The move descriptions are listed in Fig. 2.19.

$$\begin{array}{rcl} p_{Ai}*(0,0) & \rightarrow p_{Af}, \\ p_{Ci}*(0,0) & \rightarrow p_{Cf}, \\ p_{Di}*(0,0) & \rightarrow p_{Df}, \\ p_{Bi}*(2\pi,CW) & \rightarrow p_{Bf}, \\ p_{Ji}*(\pi,CW) & \rightarrow p_{Jf}, \\ p_{Ki}*(2\pi,CCW) & \rightarrow p_{Kf}, \\ p_{Ti}*(\pi,CCW) & \rightarrow p_{Tf}, \\ p_{Mi}*(2\pi,CCW) & \rightarrow p_{Mf}, \\ p_{Ni}*(\pi,CCW) & \rightarrow p_{Nf}, \\ p_{Oi}*(\pi,CW) & \rightarrow p_{Of}, \\ p_{Pi}*(2\pi,CW) & \rightarrow p_{Pf}. \end{array}$$

Figure 2.19: 10 moves in which there is follower rotation.

From this figure, it is clear that the physical actions of the follower is one of the four $*(2\pi, CW)$, $*(2\pi, CCW)$, $*(\pi, CW)$, $*(\pi, CCW)$ together with the null rotation (of move A etc.). The link transformations associated with the five motion operations are characterized as follows.

Proposition 2.3.6 In ILS, the physical transformations $*(2\pi, CW)$, $*(2\pi, CCW)$ will result in topologically equivalent (under Reidemeister transformations) initial and final link representations.

$$p_{\ell i} * (2\pi, .) \rightarrow p_{\ell f}, \quad q_{\ell i} \rightarrow q_{\ell f}, \quad q_{\ell i} \approx q_{\ell f}, \quad \ell = B, K, M, P.$$
 (2.19)

However, the physical transformations $*(\pi, CW)$, $*(\pi, CCW)$ will have non equivalent initial and final links.

$$p_{\ell i} * (\pi, .) \to p_{\ell f}, \quad q_{\ell i} \to q_{\ell f}, \quad q_{\ell i} \not\approx q_{\ell f}, \quad \ell = J, N, O, T.$$
 (2.20)

Proof: For the proof we begin by proving the second part of our result. Links whose linking numbers differ cannot be topologically equivalent. Hence in Fig. 2.18, one may observe the moves J, N, O, T, which involve applications of the $*(\pi, \cdot)$ operator are such that the linking number changes by 1, proving that the links are not equivalent. In order to make the link diagrams of the beginning and ending pose equal, one would need to cut need to cut and splice one of the link components as illustrated in Fig. 2.20—the black circle showing the location of the cutting operation.



Figure 2.20: The cut-and-splice operation of Proposition 1 that transforms the ending to the starting link diagram (black circle in the figure is the crossing at which the cutting operation is applied in order to alter the over and under passes of the link segments).

In order to prove the equivalence of the links representing the poses after the $*(2\pi, .)$ operator is applied, it is enough to show the proper elementary Reidemeister transformation between the initial and final links. One can apply a Reidemeister *Type 1* move (Fig. 2.7.) (inverting the right or left vertical link segment *L* or *F*) to decrease the number of crossings in the connecting link while preserving the link invariants, and showing the equivalence with the link diagram corresponding to move

А.

Proposition 2.3.6 characterizes the change of the link topology produced by the

corresponding motions in each dance step. These changes and the fact that ending and beginning poses may have topologically distinct link diagrams places constraints on assembling admissible sequences of moves in a dance. These may be understood in terms of differences the among the moves illustrated in Fig. 2.18. Moves in the set $\{A, C, D\}$ start and end with the starting link. Moves in the set $\{T, J, K, B\}$ only start with the starting link, and $\{O, N, P, M\}$ only end with the starting link. The move transition pairs BM, JN, KP, TO also start and end with starting link. Move M can be thought as an "inverse" move of move B—one that transforms its final link back to the starting link (Fig. 2.21). This raises the question of finding all possible move transitions in ILS. We shall describe this feasible set in terms of the link diagrams of the beginning and ending poses.



Figure 2.21: Transition from move B to the move M with link diagram representations

Proposition 2.3.7 Assume that q_{ji} and q_{jf} are the initial and final link diagram representations for the initial and final poses p_{ji} and p_{jf} , $j \in \mathcal{M}_{ILS}$, respectively. Then any admissible dance sequence can be represented by a finite concatenation of brackets $[q_{ji}, q_{jf}]$ in which any pair of successive brackets has the final link diagram of the first bracket equal to the initial link diagram of the second bracket.

Discussion: It is obvious that with respect to the physical constraints on the dance, the final pose of a move performed and the initial pose of the next must be exactly the same for the dancers to transition from one to another. Because of these physical constraints, move transitions in a dance sequence $\dots [q_{ji}, q_{jf}][q_{ki}, q_{kf}] \dots$ are allowed only when the corresponding final link diagram q_{jf} of the previous move and the initial link diagram of the next move q_{ki} where $k, j \in \{A, B, C, \dots T\}$ are the same.

By looking at the link diagram representations of the poses given in the figure 2.18, one may conclude that there are 3 types of moves in an admissible dance sequence. We can summarize the allowable transitions $\dots [q_{ji}, q_{jf}][q_{ki}, q_{kf}] \dots$ as follows.

- if $j \in \{A, C, D\}$ then, either $k \in \{A, C, D\}$ or $k \in \{T, J, K, B\}$.
- if j ∈ {T, J, K, B} then the transitions are deterministic such that, BM, JN, KP, TO must appear in the sequence. (This is similar to the need to have the letter u follow the letter q in English.)
- if $j \in \{O, N, P, M\}$ then either $k \in \{A, C, D, \}$ or $k \in \{T, J, K, B\}$.

This may be summarized in the state transition diagram of Fig. 2.22.



Figure 2.22: The allowable transitions for each move in ILS based on the physical constraints

2.3.3 Complexity Merit of Intermediate Level Dance

The transition constraints imposed by the knotting and unknotting of the dancers' arms will affect the complexity metrics introduced in Section 2.2. The intrinsic complexity of the enhanced (ILS) dance sequences can be discussed in terms of a Markov model of the step transitions, however, and this complexity can be compared with the corresponding model of BLS discussed above. Assuming no particular biases in the choice of step sequences allowed according to Fig. 2.22, we let the probabilities of steps A, C, D, B, J, K, T following step A in a sequence be 1/7. Assigning transition probabilities in a similar way in accordance with the given transition constraints, we can model dance sequences as a Markov chain with transition matrix

It is easy to see that this is a doubly stochastic matrix, and since it is aperiodic and irreducible, the associated invariant distribution is the uniform distribution on the 11 dance steps. I.e. the invariant probabilities are $\mu_i = 1/11$ for $i \in \{A, B, C, D, J, K, M, N, O, P, T\}$. The entropy of this is $\log_2 11 = 3.46$, whereas the entropy of the fourletter sequences of beginner's salsa is $\log_2 4 = 2$. What is perhaps more revealing is to compare the entropy rates of the unconstrained beginner's salsa and the intermediate level salsa. Recall [43] that given a random walk X_i on a graph described by a Markov transition matrix P having stationary distribution μ_1, \ldots, μ_{11} , the entropy rate is given by

$$-\sum_{i} \mu_i, \sum_{j} P_{ij} \log_2 P_{ij}.$$

$$(2.22)$$

For the systems described by (2.21) this is 1.7865, which is less than $\log_2 4 = 2$ for the unconstrained four state system of beginner's salsa.

In thinking about the complexity metrics of Section 2.2, we note that some differences can be expected with intermediate level salsa, but some metrics will be similar. For the intermediate level case, we have

$$\begin{cases} \text{symbol} \\ \text{frequency} \\ \text{complexity} \end{cases} = -\sum_{k=1}^{11} f_k \log_2 f_k.$$
 (2.23)

Since there are eleven symbols in ILS the maximum value of this measure is $\log_2 11 = 3.45$ which is simply performing each move equally often in a given *n* symbols length dance sequence. On the other hand this value remains 0 when only one move appears the sequence. The *average phrase complexity* metric can be calculated by the equation 2.23 where

$$f_k = (\# \text{of occurrences of the k-th symbol})/4.$$

Thus, the complexity value for the whole sequence is calculated by taking the average of the total number of phrases. There are four distinct possibilities of phrase complexities, 0, 0.811278, 1, 1.5, 2 when all four symbols are equal, three symbols are equal, two symbols are equal and when there are four distinct symbols in a phrase respectively. These cases are exactly the same as in beginner's salsa where there were a total of four distinct symbols. The *number-of-phrases complexity* which takes values that range between 0 to $\log_2(\text{total number of phrases})$ will also remain roughly the same in ILS, although the set of possible distinct phrases is much larger.

2.3.4 Transition from Intermediate Level Salsa to Extended Intermediate Level Salsa

In a regular salsa performance, leader has the freedom to choose whether to break the hand contact or keep the hands holding. Thus, in this section, we extend enhanced level salsa analysis to the case when the hand constraint is relaxed in a structured way: A dance pair is allowed to break hand contact only to return to the starting dance pose (Fig. 2.1). Since there exist no arm crossings in the starting pose, it is physically the most comfortable pose for a dance pair and it is illustrated in Fig. 2.22 that starting link has the largest set of admissible move transitions.

For the formal analysis of relaxing the hand constraint, we use algebraic representations of the link diagrams by calculating their Alexander polynomial invariants. In Section 2.3.2, the links in ILS are classified into three groups based on their linking and crossing numbers, Group 1: lk = #(L, F) = 2 and cr = 4 (e.g. $q_{Ai}, q_{Nf},...)$, 14 links (including starting link), Group 2: lk = #(L, F) = 2 and cr = 6 (e.g. $q_{Bf},$ $q_{Mi},...)$, 4 links, Group 3: lk = #(L, F) = 3 and cr = 7 (e.g. $q_{Jf}, q_{Ni},...)$, 4 links, where *starting link* is the link diagram representation of the dancers' initial pose (starting pose) in salsa. The Alexander polynomial invariants for the groups 1 (*Gr*1), 2 (*Gr*2) and 3 (*Gr*3) are given in what follows.

Proposition 2.3.8 The Alexander polynomial values for the links in group 1 and 2 are calculated as,

$$\Delta_{Gr1}(t) = \Delta_{Gr2}(t) = t^2 - 2t + 1, \qquad (2.24)$$

and for the group 3,

$$\Delta_{Gr3}(t) = -t^4 + 2t^3 - 2t^2 + 2t - 1.$$
(2.25)

Proof: We begin the proof for the two component link given in Fig 2.23.



Figure 2.23: Enumerated regions to calculate Alexander matrix for two component link diagram

Using Alexander's Algorithm the matrix \hat{M}_L is calculated as,

$$\hat{M}_L = \begin{pmatrix} -1 & -t & t & 1 \\ -t & -1 & t & 1 \end{pmatrix}.$$
(2.26)

Two neighboring regions σ_1 and σ_2 are deleted which results in the Alexander matrix such that

$$M_L = \begin{pmatrix} -1 & 1\\ -t & 1 \end{pmatrix}, \tag{2.27}$$

and the Alexander polynomial is computed as $\triangle(t) = t - 1$. By definition the starting link is the composition of two identical knots (Fig. 2.24) by the knot sum operation.

Since the Alexander polynomial of connected sum of the two links is the multiplication of the polynomials computed for each link [44], $\triangle_{L\#F}(t) = \triangle_L(t).\triangle_F(t)$, the polynomial for the starting link (group 1) is calculated as

$$\Delta_{Gr1}(t) = (t-1).(t-1) = t^2 - 2t + 1.$$
(2.28)

Computation for the links in group 3 is similar. As the knot sum of the links is



Figure 2.24: The knot sum of the links L and F.

illustrated in Fig. 2.25,



Figure 2.25: The knot sum of the links L and F in group 3.

with some tedious algebra, the Alexander polynomial for the link F (Fig. 2.25) is calculated as,

$$\Delta_F(t) = -t^3 + t^2 - t + 1. \tag{2.29}$$

The polynomial for the sum of the links L#F in Fig. 2.25, is given by,

$$\Delta_{L\#F}(t) = (t-1).(-t^3 + t^2 - t + 1) = -t^4 + 2t^3 - 2t^2 + 2t - 1.$$
(2.30)

In order to prove that group 1 and group 2 have the same Alexander polynomial, we will show the part of the proof for theorem 2.3.3 which shows the Alexander polynomial invariance for knots(links) under Reidemeister Type 1 transformation. We begin with introducing ϵ – equivalence of the matrices which will be used in the proof.

Definition 2.3.9 [42] Two matrices M_1 and M_2 with entries that are integer coefficients of polynomial in (t) are ϵ -equivalent if one can be transformed to another by sequence of the following operations: 1. Multiplying a row or column by -1, 2. Swapping two rows or columns, 3. Adding one row or column to another, 4. Adding or removing a border where the corner element is 1 and the rest of the elements are zeros, 5. Multiplying or dividing a column by t.

It is shown in [42] that the determinants of these two matrices M_1 , M_2 differ by the power of $\pm t^k$. The transformation of the links in group 1 to the links in group 2 is a Reidemeister *Type 1* operation that adds a crossing as shown in Fig. 2.26.



Figure 2.26: Adding a crossing to the regions associated to the original knot

Let us assume that the link (in group 1) without the new crossing has original matrix \hat{M} . The added crossing will create a new region marked as σ^* which adds one row and one column to the original matrix \hat{M} as in the following,

$$M_{1}^{*} = \begin{array}{c} \sigma^{*} & \sigma_{1} & \sigma_{2} & \dots \\ cr_{1} \begin{pmatrix} 1 & t & -(1+t) & 0 \\ & & & \\ \vdots & & & \\ 0 & \hat{M} & \\ \vdots & & & \\ \end{array} \right).$$

Since the regions σ_1 and σ_2 are adjacent they can be deleted from the new matrix M^* which also means deleting them from the matrix \hat{M} which results as the square Alexander matrix for the original knot(link).

$$M_{2}^{*} = \begin{array}{c} cr_{1} \\ cr_{2} \\ \vdots \\ cr_{n} \end{array} \begin{pmatrix} 1 & 0 & 0 & \dots \\ & & & \\ 0 & & M \\ \vdots & & & \end{pmatrix}$$

From the definition of the ϵ -equivalence (definition 2.3.9.) of the matrices, applying operation 4 to M_2^* will result as $M^* = M$ which proves the equivalence $\triangle_{Gr_1} = \triangle_{Gr_2} = t^2 - 2t + 1$ given in the proposition.

Our goal is to investigate the physical dance motion primitives required to break the hand contact for returning to the starting pose based on the links' Alexander polynomial representations. In order to represent the physical moves such as breaking leaders' right or left hand contact, we introduce |(.) operator. We denote |(l) and |(r), in order to represent breaking the leader's left and right hand contact and holding again after altering the arm crossings, respectively. Hence, the alphabet of ILS is extended by introducing three new moves $\{U, V, Z\}$ (Appendix B).

Proposition 2.3.10 In a transition system $\ldots [q_{ji}, q_{jf}] [q_{ki}, q_{kf}] \ldots$, moves with final link q_{jf} that have $\triangle_{Gr_1} = \triangle_{Gr_2} = t^2 - 2t + 1$, is followed by q_{ki} where k = U such that

$$p_{ui} \quad |(l) \quad \to p_{uf}, \tag{2.31}$$

will transform the final link into the starting link. Moves with final link q_{jf} that have $\triangle_{Gr3}(t) = -t^4 + 2t^3 - 2t^2 + 2t - 1$, is followed by either k = V or k = Z such that,

$$p_{vi} \quad * (\pi, CW). | (l) \quad \to p_{vf}. \tag{2.32}$$

$$p_{zi} \quad * (\pi, CCW). | (r) \quad \to p_{zf}. \tag{2.33}$$

will transform the final link into the starting link where the "." represents the composition of the two operations (first breaking the hand contact, followed by the rotation of the follower and holding the hands again).

Proof: Moves $\{K, B\}$ end with the final links classified in group 2 which correspond to the dancers facing to each other with arms crossed (Appendix B). Hence, breaking the hand contact from one hand of the leader (right or left) and switching under and over passes will result as the initial pose which is defined as move U. This can be illustrated in the link diagrams by cut-and-splice operation in topological knot theory (Fig. 2.27).



Figure 2.27: Cut-and-splice operation in topological knot theory applied in the black circled crossing in order to switch over and under passes on the final link diagrams of the moves K, B

However, the final link diagrams of the moves $\{J, T\}$ are in group 3. Thus, breaking the hand contact has to be concatenated with followers rotation in order to reach to the starting pose. The reason is that the links in group three correspond to poses with followers' rotation of π . Hence, |(l) operation has to be followed by a $*(\pi, CW)$ operation and |(r) has to be followed by $*(\pi, CCW)$ and holding the arms again as in the moves V and Z. Corresponding cut-and-splice operation that results with the starting link is illustrated in Fig. 2.28.



Figure 2.28: Cut-and-splice operation applied to the group 3 links.

Definition 2.3.11 Extended ILS is the set of 14 Salsa moves labeled $\{A, B, C, D, J, K, M, N, O, P, T, U, V, Z\}$ such that each move is achieved in eight musical beats.

A New graph illustrating admissible move transitions is given in Fig. 2.29. It is assumed that the leaders have no bias when they execute any possible dance sequence, so that for their n choices, the probability of choosing a move will be 1/n. The transition matrix, when the Extended ILS is modeled as a Markov chain, is given as the new P matrix.



Figure $2 \cdot 29$: Transition graph for Extended ILS. The red arrows illustrate the transitions to the three new moves

	(1/7)	1/7	1/7	1/7	1/7	1/7	1/7	0	0	0	0	0	0	0)	
	1/7	1/7	1/7	1/7	1/7	1/7	1/7	0	0	0	0	0	0	0	
	1/7	1/7	1/7	1/7	1/7	1/7	1/7	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	1/2	1/2	0	0	0	0	0	
	0	0	0	0	0	0	0	0	1/2	1/2	0	0	0	0	
	0	0	0	0	0	0	0	0	0	1/2	1/2	0	0	0	
D	0	0	0	0	0	0	0	0	0	0	0	1/2	1/2	0	
<i>P</i> =	0	0	0	0	0	0	0	0	0	0	0	0	1/2	1/2	•
	1/7	1/7	1/7	1/7	1/7	1/7	1/7	0	0	0	0	0	0	0	
	1/7	1/7	1/7	1/7	1/7	1/7	1/7	0	0	0	0	0	0	0	
	1/7	1/7	1/7	1/7	1/7	1/7	1/7	0	0	0	0	0	0	0	
	1/7	1/7	1/7	1/7	1/7	1/7	1/7	0	0	0	0	0	0	0	
	1/7	1/7	1/7	1/7	1/7	1/7	1/7	0	0	0	0	0	0	0	
	1/7	1/7	1/7	1/7	1/7	1/7	1/7	0	0	0	0	0	0	0 /	
														(2.34))

The associated stationary distribution of the model given with the transition matrix P is given by, $\mu = (1/11, 1/11, 1/11, 1/11, 1/11, 1/11, 1/11, 1/22,$

spectively.

The entropy rate (Equation 2.22) of the Extended ILS model (2.34) is found as $H_{EILS} = 2.15$ which is larger than the calculated values for BSP ($H_{BSP} = 2$) and ILS ($H_{ILS} = 1.78$). We conclude that extending the alphabet size with 3 new moves, which extracts out the deterministic transitions in the ILS, increases the entropy rate and the complexity of the dance.

2.4 Finite State Machine Modeling of Salsa

In Section 2.2.2, we present the hybrid system model constructed for the elementary form of salsa performed by mobile robots with non-holonomic motion constraints. However, it is discussed in Section 2.3.2 that the body poses and arm positions have a major influence in artistic appeal of a performance and also in leader's decision making in move transitions. Hence, mobile robots with two degrees of freedom would be an over abstraction of a real dance pair when performing complex dance moves in Intermediate Level Salsa. Thus, in this section we describe the details of converting the salsa dance model explained in the preceding sections to a finite state machine by introducing humanoid robots to represent a dance pair.

2.4.1 Transition Model Representation of a Dance Pair

We begin with introducing the details of humanoid robot representation of a dancer as a spatial stick figure with eight links (Fig. 2.30) where α , β and γ denote roll, pitch and yaw angles (in radians) with respect to the joint's coordinate frame. We introduce notations a, b, l to represent arm, the body and leg of a dancer and R and L to represent right and left arm/leg, respectively. For instance, α_{Ra} represents roll angle (α) of the dancer's right (R) arm (a). Using this notation we define a state in dance in the following way.



Figure 2.30: A representation of a humanoid robot as a spatial stick figure with 8 links. In the right hand view, the body roll angle α_b , the leg pitch angle β_l and the arm pitch angle β_a are depicted.

Definition 2.4.1 A state q that represents a dancer's pose p is defined as a 15-tuple,

$$q = (\alpha_b, \beta_b, \gamma_b, \alpha_{Ra}, \beta_{Ra}, \gamma_{Ra}, \alpha_{La}, \beta_{La}, \gamma_{La}, \alpha_{Rl}, \beta_{Rl}, \gamma_{Rl}, \alpha_{Ll}, \beta_{Ll}, \gamma_{Ll}),$$
(2.35)

such that each element in the vector q is the corresponding roll, pitch, yaw angles (with respect to the coordinate frame) of the body and limbs for the initial and final poses of the dancer.

We call the dance leader *Bob* and the follower *Alice*. The notations q_{Al} and q_{Bo} represent the discrete states of the leader and the follower in a pose, respectively.

This framework is similar to the work presented in [12] which shows the construction of a transition model for the poses that occur in a ballet warm up routine. Our study differs from [12] in that we include multiple transition models which requires a communication protocol to execute the performance. In our finite state machine representation of paired dance salsa, there will be two finite state machines representing Bob (leader dancer) and Alice (follower dancer) with an information channel between them. We begin with the definition of a state transition model for an abstract representation of a leader dancer. by

$$G_{Bo} = (Q_{Bo}, Act_{Bo}, \to_{Bo}, q_{Bo}^0, !, W, E),$$
(2.36)

where Q_{Bo} is the set of all possible states (2.35) that represent the initial p_i and final p_f dance poses; Act_{Bo} is the set of all possible actions (set of physical moves that are represented by *(.,.) operator in Section 2.3.2); $q_{Bo}^0 \subseteq Q_{Bo}$ is the set of initial states; ! is the formal representation of signal transmitted by the leader to the follower, and W and E are vectors of complexity metrics (phrase complexity and phrase energy) which will be used to achieve structured automated dance sequences. Formal definitions of these complexity vectors (W, E) and their integration in the transition system will be presented in Section 2.4.2. The symbol \rightarrow_{Bo} denotes a transition relation (based on the topological constraints given in Section2.3.2 and complexity vectors W and E). Symbolically, $\rightarrow_{Bo} \subseteq Q_{Bo} \times (W \times E) \times Act_{Bo} \times Q_{Bo}$. In this expression, $Q_{Bo} \times (W \times E) \times Act_{Bo}$ is considered as the input such that from an initial state in the set Q_{Bo} , leader executes an action Act_{Bo} with respect to the constraints E and W. The execution results another state in Q_{Bo} which is considered as the output.

Similarly we can represent the follower dancer (Alice) by a transition system,

$$G_{Al} = (Q_{Al}, Act_{Al}, \to_{Al}, q_{Al}^0, ?, W, E), \qquad (2.37)$$

where definitions of the components Q_{Al} , Act_{Al} , q_{Al}^{0} , W and E in Equation (2.37) are identical to those of the Equation (2.36) but the subscripts are replaced with Alin order to represent Alice. ? is the formal representation of signal received by the follower such that she has the corresponding transition based on the received signal ?, W and E. Hence, symbolically, the definition of the transition relation for Alice can be shown as, $\rightarrow_{Al} \subseteq Q_{Al} \times (W \times ? \times E) \times Act_{Al} \times Q_{Al}$. In order to understand leader-follower interactions, the overall system that represents salsa as a transition model is defined as follows.

Definition 2.4.3 The product of the two transition systems given in Equations (2.36)

and (2.37) is defined as

$$G_{Al,Bo} = (Q, Act, \to, q^0, !, ?, W, E),$$
(2.38)

where (i) $Q \subseteq Q_{Bo} \times Q_{Al}$, (ii) $Act \subseteq Act_{Bo} \times Act_{Al}$, (iii) $q^0 \subseteq q^0_{Al} \times q^0_{Bo}$, (iv) ! is the message transmitted by leader, (v) ? is the message received by follower, (vi) W is the vector of phrase complexity values and E is the vector of phrase energies that are known by the leader and follower agents, and (vii) \rightarrow is the transition relation that obeys the following rule.

It is assumed that there exists synchronous message passing between two transition systems over a memoryless channel c that is used to send (c!) and receive a message (c?) such that the transition relation satisfies,

$$\frac{q_{Bo} \xrightarrow{c(Act_{Bo})!} \hat{q}_{Bo} \wedge q_{Al} \xrightarrow{c(Act_{Bo})?} \hat{q}_{Al}}{Act_{Al}} \xrightarrow{q_{Al}} (2.39)$$

The numerator of the formula (2.39) illustrates leader's transition based on the decided action Act_{Bo} from state q_{Bo} to \hat{q}_{Bo} . This is transmitted as a signal through the channel $c(Act_{Bo})!$ and the follower has the simultaneous transition from q_{Al} to \hat{q}_{Al} based on the received signal $c(Act_{Bo})$?. The denominator represents the transition of the overall system (2.38) from the state $\langle q_{Bo}, q_{Al} \rangle$ to $\langle \hat{q}_{Bo}, \hat{q}_{Al} \rangle$. Using the model defined in Equation (2.38), we introduce the state transition diagram representation for BSP (Fig.2.31).

In BSP all of the moves start and end with same pose. Thus, in Fig. 2.31, q_{Bo}^0 and q_{Al}^0 represent the states as defined in Equation (2.35) which correspond to the poses of Alice and Bob, respectively. Blue arrows represent the state transitions based on the physical motions (described by the *(.,.) operator) required to perform each move A, B, C, D. The notation $c(Act_{Bo})!$ represents the communication of the Bob's decided transition $q_{Bo}^0 \xrightarrow[Act_{Bo}]{c(Act_{Bo})!} q_{Bo}^0$ to the follower. Alice has the simultaneous



Figure 2.31: Finite state machine representation of beginner salsa performance (BSP)

transition $q_{Al}^0 \xrightarrow{c(Act_{Bo})?}{Act_{Al}} q_{Al}^0$ with respect to the received signal $c(Act_{Bo})$? from Bob. For instance, if the agents perform move D, the leader has a transition $q_{Bo}^0 \xrightarrow{c(*(0,0)_D)!}{*(0,0)_D} q_{Bo}^0$ that is signaled to Alice through the channel c such that Alice has the corresponding transition $q_{Al}^0 \xrightarrow{c(*(0,0)_D)?}{*(0,0)_D} q_{Al}^0$. Hence, the state of the overall system changes such that $< q_{Bo}^0, q_{Al}^0 > \xrightarrow{*(0,0)_D} < q_{Bo}^0, q_{Al}^0 >$, which will be denoted as a move D.

Unless stated otherwise, we will assume that c is a noiseless and memoryless channel so the follower is capable of decoding every signal transmitted by the leader. Problems such as channel bandwidth and noise during the signal transmission will be discussed in the following section. Transition graph for the ILS is much more complex due to the topological constraints introduced in Section 2.3.2. In Fig. 2.32, states of the leader and follower including the transitions (blue arrows) are depicted. Moreover, the topological link diagram representations are illustrated. For instance, in Fig. 2.17 one of the link diagram representations is shown on the bottom left of the figure. This corresponds to the initial pose of the leader and follower. In Fig. 2.32 the same topological link occurs when the system (2.38) is in the state $\langle q_{Bo}^1, q_{Al}^1 \rangle$.

Using the transition model representations given for BSP and ILS, we conclude with the following result about generated dance sequences by using the transition



Figure 2.32: Finite state machine representation of Intermediate Level Salsa (ILS)

model given in Equation (2.38).

Proposition 2.4.4 The allowable sequences generated by using the transition model (2.38) for the set of moves described in BSP and ILS can be represented by the concatenation of the brackets as in the following,

$$[\langle q_{Bo}, q_{Al} \rangle_{ji}, \langle q_{Bo}, q_{Al} \rangle_{jf}], [\langle q_{Bo}, q_{Al} \rangle_{ji}, \langle q_{Bo}, q_{Al} \rangle_{jf}], \dots$$
(2.40)

where *i* and *f* stand for the initial and final state (pose), respectively, and $j \in \mathcal{M}_{BSP}$ for BSP or $j \in \mathcal{M}_{BLS}$ for ILS.
Discussion: This result is an extended version of the bracket representation given in Proposition 2.3.7. Here, the bracket representation is extended to involve the distinct states to represent the poses of both the leader and the follower. Each bracket [.,.] represents a move performed by the agents such that the concatenation of brackets is used to represent dance move sequences.

The goal of constructing such framework is to create an automaton that generates letter sequences and investigate the notion of 'optimality' in dance. The transition system representation will be used to understand the underlying structure of the leader-follower interactions in salsa by capturing the discrete states of the leader and follower and the transitions based on the communicated signals. In Section 2.2.2, in order to define 'optimality' in a dance performance, two complexity metrics are defined: the energy consumption and the phrase complexity. Integration of these complexity metrics to the model (2.38) will be discussed in what follows.

2.4.2 Optimized Transitions for Salsa Choreography

We focus on the problem of understanding how to structure the sequence generation of the model (2.38) by introducing W and E as vectors of energy and average phrase complexity constraints in order to generate 'optimal' dance sequences. We are also interested in finding proper communication protocols for the agents to successfully achieve dance phrases. In Section 2.2.1, it is discussed the that artistic appeal of a dance sequence perceived by human judges is highly correlated to the average phrase (4-letter phrase) complexity. Hence, we incorporate this metric in our state transition model by a vector W. As an example, assume that the agents Alice and Bob are required to generate 20-letter long sequences. We partition a 20-letter sequence into 4-letter phrases and assign a complexity value for each phrase as in Fig. 2.33.

For this particular case, we define the complexity vector W introduced in the



Figure 2.33: Example deconstruction of 20 letter sequence

model (2.38), $W = (w_1, w_2, w_3, w_4, w_5)$ where $w_i \in \{0, 0.811, 1, 1.5, 2\}$ and i = 1, 2, 3, 4, 5. The possible values of w_i correspond to the complexity values calculated by the formula (2.1), e.g. phrase (AAAA) has phrase complexity w = 0 but (BACD) has phrase complexity w = 2. We assume that Alice and Bob have access to a look-up table that has the phrase complexity values assigned for each 4-letter phrase and the leader makes decisions for the complexity values of the phrases generated in a sequence. Below, we illustrate an example sequence generated by Alice and Bob with the moves from BSP.

Seq: (ABDC)(BCAC)(CCBB)(DADD)(BBBB)

Seq is constrained by a complexity vector W = (2, 1.5, 1, 0.811, 0) so that Bob chooses phrases to satisfy W that guarantees decreasing phrase complexities through the sequence generation.

Another, complexity metric proposed in Section 2.2.2 is the energy expenditure. Thus, an energy vector $(E = (e_1, e_2, e_3, ...))$ is introduced to the model (2.38) in order to constrain the energy consumed by the agents in performing each phrase.

Case 1: Unlimited Energy

In this case it is assumed that the agents can expand an unlimited phrase energy during sequence generation or equivalently all of the entries of vector E are infinity. Hence, the only constraint on the automated sequence generation is the phrase complexity vector W. Since Alice has information about only the phrase complexity W in the look up table, Bob is required to communicate the phrase he chooses for a given phrase complexity to Alice. Thus, we are interested in finding the communication protocols between the agents for the execution of the moves to be performed in a phrase *i* which is constrained by w_i . For simplicity, we assume that the channel used by the leader to send (*c*!) and follower to receive (*c*?) a signal is noiseless (the case with noise will be discussed in Section 2.5. Here, we use similar notation introduced in study [37] to quantify the amount of information exchange in distributed control systems. For an action ($a \in Act_{Bo}$) chosen by Bob, we assume that the codeword sent by Bob to Alice is represented by, $\zeta_B = K_{Bo}(a)$, where $K_{Bo}(.)$ is the coding function that maps the possible choices onto bits. If U_{Bo} represents the range of K_{Bo} , then the amount of bits needed for Bob to communicate his decision is calculated by $\lceil \log_2(|U_{Bo}|) \rceil$ with the units in bits. For instance, if Bob is required to communicate an action *a* from the set of four moves in BSP or eleven moves from ILS, he needs to exchange $\lceil \log_2(4) \rceil = 2$ or $\lceil \log_2(11) \rceil = 4$ bits, respectively. A protocol Ω for communicating the choices in a phrase is defined in the following.

Definition 2.4.5 A communication protocol Ω for a phrase *i* which is constrained by w_i generated by the system (2.38) is given by, $\Omega\{K_{Bo}^1, K_{Bo}^2, K_{Bo}^3, K_{Bo}^4\}$, where the superscripts are ordered from 1-4 since a phrase has 4 letters.

Using the definitions presented above we introduce the notion of communication complexity for our transition system.

Definition 2.4.6 For a 4-letter phrase *i*, the communication complexity of Ω is calculated by

$$D(\Omega) = \sum_{k=1}^{4} \lceil \log_2(|U_{Bo}|^{(k)} \rceil.$$
 (2.41)

Next result will present the communication complexity bounds with respect to the level of the dance to be generated by the model (2.38).

Proposition 2.4.7 The bounds on the communication complexity $D(\Omega)$ of a phrase *i* constrained by $w_i \in \{0, 0.811, 1, 1.5, 2\}$ with the set of moves defined in BSP are found as,

$$2 \le D(\Omega) \le 8. \tag{2.42}$$

Moreover, the communication complexity bounds in ILS are found as,

$$4 \le D(\Omega) \le 16,\tag{2.43}$$

where the units are in bits.

Proof: For simplicity, we begin with proving the bounds in BSP in that similar arguments will hold for ILS. Since, the phrases with w = 0 are repetitions of a single letter and w is known by the agents, communication complexity becomes 2 bits which is simply signaling only the first letter from the set $\{A, B, C, D\}$. For instance, for signaling the phrase (AAAA), the leader signals the first letter which requires $log_2(4) = 2$ bits of information exchange and rest of the sequence is deterministic to both Alice and Bob with the assumption that they both know w. Using a similar argument, it is easy to conclude that the communication complexity value reaches maximum 8 bits which occurs for multiple phrases. This corresponds to the value $3.\lceil \log_2(4)\rceil + \lceil \log_2(3)\rceil = 8$ which is simply signaling three letters of a phrase with four possibilities of the moves $\{A, B, C, D\}$ and the last letter with three possibilities with respect to the w_i constraint (For instance, when w = 1.5 three distinct letters appear in a phrase such that signaling the first three letters of the phrase (ABC-) with four choices corresponds to exchanging $3.\log_2(4) = 6$ bits and the last letter (-) will be either A, B or C which requires $\lceil \log_2(3) \rceil = 2$ bits).

A Similar approach is used to calculate the communication complexity bounds on ILS. We conclude that when w = 0, for signaling one letter from the 11 letters in ILS, Bob needs to communicate $\lceil \log_2(11) \rceil = 4$ bits of information. However, the upper bound corresponds to signaling three letters of a 4-letter phrase with 11 possibilities of moves and last letter with 10 possibilities such that $3 \cdot \lceil \log_2(11) \rceil + \lceil \log_2(10) \rceil = 16$ bits are required to exchange.

Case 2: Limited Energy

In this case, we are interested in generating automated sequences that are optimal in terms of the phrase complexity and phrase energy metrics. By using the vectors Wand E in the model (2.38), one may construct two distinct optimization problems. In the first problem the agents are required to perform sequences with the highest phrase energy e_i in the look-up table for a given phrase complexity w_i . Here, we focus on the second problem of which the agents' energy consumption \mathbb{E}_i in a phrase is assumed to be restricted by the energy vector E for a given phrase complexity vector W. In this problem the agents seek to perform the phrase with highest energy consumption without exceeding their energy budget. We present the problem formulation below. *Problem Statement:* For the purpose of generation of automated dance sequences the agents will seek to solve the following optimization problem for a given phrase i,

$$\max \quad \mathbb{E}_i \tag{2.44}$$

Subject to
$$\mathbb{E}_i \le e_i$$
 (2.45)

For a given
$$w_i$$
 (2.46)

where *i* is a four letter phrase, \mathbb{E}_i is the energy consumed by the agents to generate phrase *i* and e_i , w_i are the energy and phrase complexity constrains. For a given w_i there exist phrases that minimizes \mathbb{E}_{min} and maximizes \mathbb{E}_{max} energy consumption such that $\mathbb{E}_{min} \leq \mathbb{E}_{max}$. The energy packet e_i for a phrase *i* is assumed to be distributed such that $\mathbb{E}_{min} \leq e_i \leq \mathbb{E}_{max}$, which guarantees a solution to the problem given in Equations (2.44)-(2.46). The uniqueness of the solution will be discussed by using the following result. **Proposition 2.4.8** Each 4-letter phrase *i* constructed by the moves in \mathcal{M}_{BLS} and a constraint w_i , has a unique energy consumption \mathbb{E}_i .

Proof: The proof is by brute force calculation of the phrase energies using the methods introduced in Section 2.4.2. The energy consumption for each letter is computed and used to calculate the energy required for performing each distinct phrase. Fig. 2.34, illustrates the plot of the phrase complexity versus the phrase energy values.



Figure 2.34: The energy and phrase complexity values of the all of the possible phrases in agents' look up table. The blue circles represent the calculated phrase complexity and energy values for each 4-letter phrase in BSP. The red circle represents the phrase with highest energy (CCCC) and green circle is with the highest complexity (ABCD)

The blue circles in Fig. 2.34 represent the total of 35 distinct 4-letter phrases that can be performed for phrase complexity values 0, 0.811, 1, 1.5 and 2 which correspond to performing one, two, three and four distinct letters in a 4-letter phrase, respectively. For instance, for phrase complexity $w_i = 2$, the agents perform a permutation of the letters A, B, C and D. Another case is when $w_i = 0$ such that the agents can perform one of the phrases from the set $\{AAAA, BBBB, CCCC, DDDD\}$. Using similar arguments, unique energy consumption values of the phrases that are constrained by w_i are computed and shown in the Fig. 2.34.

Remark: The solution of the optimization problem (2.44)-(2.46) is unique in terms of the number of distinct letters in a phrase. For instance, for $w_i = 2$ and $e_i = 3000$, the solution set includes the permutations of the letters A, B, C, D. However for $w_i = 0$ and $e_i = 3000$ the only possible solution is the phrase (DDDD).

For the application of the proposed framework to an actual salsa performance, we conducted a new set of experiments by using a set up similar to the study presented in Section 2.1. The four possible moves A, B, C and D in BSP were introduced to experienced real salsa dancers who were asked to generate sequences by using these moves. Moreover, we asked our dancers to perform sequences that they think would be artistically 'high energy', 'low Energy', 'surprising' and 'unsurprising' sequence. Each performance is 20 letters long and recorded as video. Four generated sequences by our experienced dance pair are shown below.

Seq1:(BDCB)(DBCB)(DDBB)(CCDB)(DDBB)Seq2:(BBBB)(BBBA)(ACAA)(AAAD)(AAAB)Seq3:(AAAA)(ABAA)(BABA)(DACA)(DABC)Seq4:(ABDC)(DABC)(BBAB)(AABA)(BBBB)

The sequences are ordered such that 'high energy', 'low energy', 'surprising' and 'unsurprising' correspond to Seq1, Seq2, Seq3, Seq4, respectively. We would like to generate similar sequences to those of Seq 1-4 by the model given in Equation (2.38) and by solving the optimization problem introduced in (2.44)-(2.46), with complexity W and energy vector E. We calculated the energy and the complexity values of the phrases in each sequence generated by the real dancers and illustrated in Fig. 2.35.

		_	Total	Av. Ph.
	W	E	Energy	Complexity
High energy	(1.5 1.5 1 1.5 1)	(2006 2064 2559 2260 2181)	12093	1.3
Low energy	(0 0.81 0.81 0.81 0.81)	(2462 2462 2163 2840 2163)	11072	0.649
Surprising	(0 0.81 1 1.5 2)	(2240 2181 2123 2579 2521)	11646	1.0623
Unsurprising	(2 2 0.81 0.81 0)	(2521 2521 2064 2182 2006)	11295	1.1245

Figure 2.35: The table illustrating the energy and phrase complexity values for four sequences generated by the real dancers

From the table it can be concluded that the sequences 3 and 4 are constructed such that the phrase complexities are increasing and decreasing, respectively, during the evolution of the sequence with $W_{Seq3} = \{0, 0.811, 1, 1.5, 2\}$ and $W_{Seq4} = \{2, 2, 0.811, 0.811, 0\}$. Also, it is observed that Seq1 has the maximum and Seq2 has the minimum total energy. Thus, we conclude that by using W and E vectors calculated for the sequences Seq 1-4, as constraints in the model (2.38), our agents can generate similar dance sequences to those created by the real dancers with artistic objectives such as generating a 'surprising' or a 'high energy' dance sequence.

Up to this point, the signal transmission from the leader dancer to the follower dancer is assumed to be noiseless. However, in a real dance performance mistakes may occur due to corrupted communication signals transmitted by the leader or to inexperience of the follower. Next section will investigate the leader-follower interactions when there exists noise in the communication channel.

2.5 Measuring the Performance of a Leader in Collective Behavior

In this section, the results of an experiment will be discussed in order to understand the proficiency level in dance. In this experiment, various skill level leaders were asked to perform unstructured dance sequences with the same follower. Both the leaders and the follower were introduced to Beginner Level Salsa. They were given a library of moves and the gestures which were used by the leader to signal the upcoming move to the follower during the dance. Moreover, the dancers were asked to continue dancing even if they make a mistake in the execution of any of the moves in BSP. The video recordings of the dance sequences started after the subjects confirmed that they feel comfortable executing the moves and the signals. The sequences performed by the dancers were recorded with a video camera and are transcribed in Appendix C. The notation m is used to distinguish the mistakes that occurred during a performance. A mistake is defined as a motion of leader or follower that can not be classified by a knowledgeable observer as a member of the set of the moves $\{A, B, C, D\}$. The order of the experiment followed from Seq5 to Seq7 in which Beginner 1 (Bg1), Expert (Ex), Beginner 2 (Bg2) level leaders performed, respectively with the same follower. An *Expert* is defined as a leader who has more than three years of experience in salsa. Since the follower was also new to salsa, the order of the leaders in recordings is chosen as Bg1-Ex-Bg2, respectively to eliminate the effect of experience gained by the follower during the experiment.

In order to discuss the results of this experiment, we modify the communication channel (c) defined in the transition model (2.38). The communication between finite state machines that represent the dancers is assumed to be achieved by a multi-input erasure channel (MEC) with a finite channel capacity.

In Fig. 2.36, the erasure channel scheme used for the transition model is shown.



Figure 2.36: Multi-input Erasure Channel designed for the agents in model (2.38)

Erasure channels are widely studied in the information theory literature [43] to model communication systems in which messages occasionally are lost. The general definition of a channel capacity is given by, $C = \sup_{p_X(x)} I(X;Y)$, where I(X;Y) is the mutual information between the input X and output Y, and the supremum is taken over all possible probability distributions of x, $p_X(x)$ [43]. For further analysis we list the assumptions for the model (2.38).

Assumptions: 1. We assume that the leader communicates 2 bits for channel use in order to signal his decision (Act_{Bo}) of a move from the set $\{A, B, C, D\}$. In actual human dance performances, communications are encoded as arm and body gestures, but for the sake of our abstract model, the communications are carried out by exchanging packets of bits 2. The dancers were not restricted to follow any prescribed complexity W or energy E constraint. 3. It was assumed that the mistakes would be equally likely when performing each move in the sequences Seq5-Seq7 (Appendix C), i.e. the error probability is independent of the move to be executed.

For the interpretation of the data presented in Appendix C, we calculate the channel capacity for a multi-input erasure channel with four inputs that are the moves $\{A, B, C, D\}$ and five outputs $\{A, B, C, D, e\}$ where e represents the erasure.

Proposition 2.5.1 The channel capacity C for a multi-input erasure channel with four inputs and five outputs and with a probability of erasure α is computed as

$$C = 2(1 - \alpha). \tag{2.47}$$

Proof: We begin the proof with expanding the capacity (C) formula such that

$$C = \max_{p_X(x)} (H(Y) - H(Y|X)).$$
(2.48)

By assumption 2, the probability of erasure is α is equally likely for each input A, B, C, D so that

$$C = \max_{p_X(x)} (H(Y) - H(\alpha)).$$
(2.49)

Using a proof similar to that given in [43] for the binary case, we call an event E = 0when Y = e(erasure), and E = 1 otherwise, such that

$$H(Y) = H(Y, E) = H(E) + H(Y|E).$$
(2.50)

Here, $H(E) = H(\alpha)$ since the probability of an erasure is assumed to be α for each input. The conditional entropy H(Y|E) is expanded such that

$$H(Y|E) = p(E=0)(H(Y|E=0) + p(E=1)H(Y|E=1).$$
(2.51)

The term H(Y|E = 0) = 0 in Equation (2.51) in that there exist no uncertainty in Y when E = 0. So the Equation (2.50) becomes

$$H(Y, E) = H(\alpha) + (1 - \alpha)H(Y|E = 1).$$
(2.52)

By plugging Equation (2.51) into (2.49), we get the capacity formula such that,

$$C = \max_{p_X(x)} (H(\alpha) + (1 - \alpha)(H(Y|E = 1) - H(\alpha)),$$
(2.53)

$$C = \max_{p_X(x)} ((1 - \alpha)(H(Y|E = 1))).$$
(2.54)

Let us assume that the probability distribution for each letter is given by π_A , π_B , π_C , π_D . Then, for the case when E = 1, the probability distribution that maximizes the

entropy of Y is the uniform distribution such that $\pi_A = \pi_B = \pi_C = \pi_D = 1/4$. Thus, the channel capacity becomes $C = 2(1 - \alpha)$.

From the empirical data given in Appendix C, we calculate erasure probabilities $\bar{\alpha}$ for three distinct cases. By counting the number of mistakes m and dividing it to the total number of letters, we found that $\bar{\alpha}_{Bg1} = 0.159$, $\bar{\alpha}_{Bg2} = 0.147$, $\bar{\alpha}_{Ex} = 0.056$, respectively. Channel capacities for three cases are calculated by the formula (2.47) and the relationship between the capacities are given by, $C_{Bg1} = 1.68 \leq C_{Bg2} = 1.70 \leq C_{Ex} = 1.88$, for three leaders. The expert dancer creates a higher capacity communication link with the follower.

Remark: In the channel capacity calculations, $\bar{\alpha}$ is used to distinguish an empirical estimate with the actual probability of erasure α . Also, in assumption 2 we assume that the errors occur equally likely in the execution of each move. In Section 2.3.2, the moves with the follower's rotation are introduced to define Intermediate Level Salsa. Hence, the error probability can be expected to be higher in more complex moves and the investigation of the channel capacities for such cases is the subject of the future work.

2.6 A Robotic judge to Evaluate Group Motion

We seek to answer the question: How can a third party robot evaluate 'success' in human or animal group behavior? In team sports, such as basketball, American football or soccer, evaluation would be simply based on the execution of a team strategy that results in a score on offense or prevents the opponent team from scoring on defense. In animal group behavior, success can be defined as the protection of the group members from predators or finding a food source. However, if one considers performance art, such as dance, the definition of success might not be that clear, since the overall goal in art forms is not as explicit compared with those in sports or animal behavior. Nevertheless, we observe that, in dance competitions, the judges' scores consistently agree with each other, which means that they may use similar performance metrics in judging the performances. This brings up the question of whether there exists a formal way of evaluating a dance performance. In what follows, this question is examined by using the transition model representation of salsa dance introduced in Section 2.4 and by incorporating performance metrics to evaluate salsa performances. We explain how this framework will allow a robotic judge to potentially replace human judges and in doing so to become the most fair judge ever.

2.6.1 A Modified Transition Model for Dance Move Recognition

In order to describe the dancers' movement as well as track their motions, we use the humanoid robot introduced in Section 2.4.1 (Fig. 2.30) to represent different parts of a dancer's body as illustrated in Fig. 2.37. It is assumed that there exist *critical points* (red circles in Fig. 2.37) that represent tracked points on the humanoid robot representation. The number and locations of the critical points are chosen in order to be able to distinguish the moves performed in BLS and ILS. We recall the notation introduced in Section 2.4.1 such that a, b, l represent arm, the body and leg of a dancer and R and L represent right and left arm/leg, respectively. For instance, x_{Ra} represents x-coordinate of the critical point tracked on the dancer's right (R) arm (a). We define a state q to represent a pose p in salsa as a 15-tuple such that

$$q = (x_b, y_b, z_b, x_{Ra}, y_{Ra}, z_{Ra}, x_{La}, y_{La}, z_{La}, x_{Rl}, y_{Rl}, z_{Rl}, x_{Ll}, y_{Ll}, z_{Ll}).$$
(2.55)

Each component of the vector q is the corresponding x, y or z coordinate of each critical point. The notations q_{Al} and q_{Bo} represent the discrete states of the leader and the follower in a pose, respectively.

We also recall the state transition system for a dance pair presented in Equation



Figure 2.37: A Humanoid robot that represents an actual dancer. The red circles represent five critical points that are named as the Body, Left/Right arm and Left/Right Leg. The number of critical points are chosen to distinguish the moves in Beginner Level Salsa and Intermediate Level Salsa.

(2.1),

$$G = (Q, Act, \to, q^0, !, ?), \tag{2.56}$$

where $Q = Q_{Bo} \times Q_{Al}$, $Act \subseteq Act_{Bo} \times Act_{Al}$, $q^0 \subseteq q^0_{Al} \times q^0_{Bo}$, $!(Act_{Bo})$ is the signal transmitted by leader, $?(Act_{Bo})$ is the signal received by follower, \rightarrow is the transition. It is assumed that there exists a synchronous message passing between these two transition systems such that follower can estimate the upcoming move without an error. The details of the communication protocols between the dance pair is given in Section 2.4.2. In this section, we are going to use model (2.56) for the dance move recognition executed by a dancer pair from the recordings of the locations of the critical points.

In Section 2.4.1, the transition models are defined for BLS (Fig. 2.31) and ILS

(Fig. 2.32) and the allowable sequences are generated as follows:

$$[\langle q_{Bo}, q_{Al} \rangle_{ji}, \langle q_{Bo}, q_{Al} \rangle_{jf}], [\langle q_{Bo}, q_{Al} \rangle_{ji}, \langle q_{Bo}, q_{Al} \rangle_{jf}], \dots$$
(2.57)

where *i* and *f* stand for the initial and final state (pose), respectively, and $j \in \mathcal{M}_{BLS}$ for BLS or $j \in \mathcal{M}_{ILS}$ for ILS and each bracket represents a move.

We use this bracket representation to distinguish the moves (letters) performed by the dancers. For such a purpose, the initial and final states of the leader and follower dancers need to be observed. Moreover, observation of the transition relations are needed in order to avoid ambiguities caused by the moves that start and end with the same pose. The recognized sequence will be evaluated by a robotic judge, which computes a score based on the performance metrics.

2.6.2 Measuring the Performance of the Robotic Judge

We are particularly interested in building a robotic judge that observes and evaluates the artistic success of a dance performance. The idea is similar to the judges that appear in the Olympic games or dance competitions. The judges in these contests have criteria that measure artistic reflection and also the complexity of the execution. It would be difficult for a robotic judge to evaluate the warmth of a dancer's smile but instead it can evaluate artistic appeal of a performance by using the energy and complexity metrics. The overall scheme of a robotic judge evaluating performance art is shown in Fig.2.38.

Robotic judge has two components: an observation component and an evaluation component. We use the abstract model given in (2.38) to represent the leader and follower dancer as humanoid robots with tracked critical points. The goal of the observation component is then to estimate a sequence of states and the transition (Q_o, \rightarrow_o) that best fit the observed sequence of tracked critical points and the model.



Figure 2.38: A robotic judge scheme to evaluate salsa. The robotic judge involves two components: An *observation component* and an *evaluation component*

Here, Q_o is the set of observed states including the initial and final states (poses) of the leader and the follower in a move and \rightarrow_o is the observed transition between the initial and final state such that $\rightarrow_o: (q_i, a) \mapsto q_f$ where $q_i, q_f \in Q$ and $a \in Act_o$ represents the actions that are executed by the agents between the initial pose and final pose.

The purpose of the observation component is to deconstruct a salsa performance into a letter (move) sequence with the bracket representation proposed in (2.57). This is achieved by detecting the x-y-z coordinates of the critical points defined as in vector q in (2.55) so that the sensory information can be mapped to a pose in the dance by a function s such that $q_o = s(x_b, y_b, z_b, \dots x_{Ll}, y_{Ll}, z_{Ll})$ for Alice and Bob. This is similar to the idea of template matching which is widely studied in computer vision [45]. Simply, the tracked points' coordinates are compared with the values that are contained in a library of poses with an allowed deviation δ . After detecting initial and final pose, the algorithm resets and starts to track the new move. In the previous sections, we have shown that there may be multiple moves with equivalent initial and final poses. To avoid ambiguities in dance move detection, we also include the observation of the transition \rightarrow_o . This transition is captured by tracking the velocities v_x , v_y and v_z for each critical point.

The evaluation component first decomposes the observed sequence into 4-letter phrases. It then calculates observed phrase complexity W_o and observed phrase energy E_o as described in Section 2.5. It finally computes the score of the observed sequence based on the following *Score* function,

$$Score = a.E_{total} + b.W_{ave} + c. \tag{2.58}$$

This function is a linear combination of E_{total} and W_{ave} where E_{total} is the sum of the energy values of the phrases and W_{ave} is the average phrase complexity that is calculated by dividing the total phrase complexity value by the total number of phrases that appear in a sequence.

The Score function is constructed as a linear function in that it fits the data collected from a previous experiment which is reported in Section 2.2. Thus, in this study we use the previous data as a training set to estimate the coefficients a, b and c in Score function (Fig. 2.39). The Score function learned from the data has the form

$$Score = -17.94 + 16.E_{total} + 0.833.W_{ave}.$$
(2.59)

In order to validate our robotic judge, we asked our experienced salsa dancers to perform four new sequences (each of them having 20 letters) by using the moves in BLS. All of the sequences are performed by the same two dancers in order to exclude the effect of artistic reflection of a dancer's personal demeanor. The sequences



Figure 2.39: The least squares regression plane with coefficients estimated from the previous data (Section 2.2). The x-coordinate is the average phrase complexity of a dance sequence (bits), the y-coordinate is the total energy consumed by the dancers (hectometers) and the z-coordinate is the associated score assigned by the human judges.

constructed by the dancers are given below.

$$V1 : (BDCB)(DBCB)(DDBB)(CCDB)(DDBB)$$
$$V2 : (BBBB)(BBBA)(ACAA)(AAAD)(AAAB)$$
$$V3 : (AAAA)(ABAA)(BABA)(DACA)(DABC)$$
$$V4 : (ABDC)(DABC)(BBAB)(AABA)(BBBB)$$

The video recordings of the dance sequences are shown to human subjects (subjects without dance experience) with a random order. Average scores in 1-to-10 scale that are collected from 15 judges are, $Score_{V1} = 9.1$, $Score_{V2} = 3.09$, $Score_{V3} = 3.7$ and $Score_{V4} = 5.79$.

The same video recordings are fed into the robotic judge. For the observation

component, we use a Microsoft Kinect sensor in order to track the x-y-z coordinates of the critical points that are defined in the vector q in (2.55). Microsoft's open source C++ algorithm is modified for the purpose of our experiment such that two distinct libraries are contained in a movement library and transition library. The q vectors that represent the poses of the leader and the follower are integrated to the Kinect algorithm as a library so that the algorithm seeks to match the tracked coordinates of the critical points to one of the possible poses from the library with the maximum deviation δ . Moreover, a library of move transitions is incorporated into the C++ code which includes the deviations of the coordinates with respect to time for distinct physical moves described by the *(.,.) operator. The initial pose shown in the upperleft corner of the Fig. 2.17 is incorporated into the algorithm as a trigger to start tracking critical points of the dancers (Fig. 2.40).



Figure 2.40: A snapshot of the User Interface of the robotic judge that uses Kinect C++ algorithm to track the critical points on the stick figure representations of the dancers. Recognized dance move is shown on the right bottom corner to the user.

The timer starts and stops with the recognition of initial pose and final pose respectively. In Fig. 2.40, a snapshot of the algorithm is shown including the stick figure representation of a dancer and the detected letter which is illustrated in the right bottom corner. Finally, the algorithm computes the E_{total} and W_{ave} values for a recognized sequence which are then supplied to the Score function given in Equation (2.59).

The dancers performed the sequences given as V1, V2, V3 and V4 in the view of the Microsoft Kinect sensor. Four sequences are deconstructed by the robotic judge and average phrase complexity and energy values are computed for each sequence which are then fed to the Score function. The score values are calculated as 5.29, 2.88, 4.28, 3.69 for the sequences V1, V2, V3 and V4 respectively. The correlation coefficient between the subjects' scores and the scores assigned by the robotic judge is calculated as R = 0.81. The strong correlation implies that our judge performs well enough in matching human's perceived artistic appeal of a dance performance.

Chapter 3

Collective Group Motion in Bat Cave Emergence

In this chapter, we discuss collective behavior of bats through the analysis of their three dimensional trajectories recorded during a cave emergence. The trajectories are computed with stereoscopic methods using data from synchronous thermal videos that were recorded with high temporal and spatial resolution from three viewpoints. The analysis targets to understand the flight behavior of bats when they react to other group members as well as their navigation strategies in feature rich environments. The first part of the chapter discusses the statistical analysis of 405 bat trajectories. We examine the possibility that bat trajectories are governed by optical flow sensing that interpolates periodic distance measurements from echolocation. Using an idealized geometry of bat eyes, we introduce the concept of *time-to-transit*. Earlier studies [46, 47, 48] suggest that this quantity is computed by the animals' visual cortex. We propose several optical flow based control laws in order to generate synthetic bat-like trajectories in a simulated environment. Examination of the generated trajectories suggests that bat motions are governed by reaction to the key features in the environment as well as by spacial memory.

In the second part of the chapter, leader-follower interactions are examined in order to understand how a leader bat's flight behavior differs from a follower bat in emergence. A new concept *virtual loom* is introduced and incorporated into the proposed model to generate motions that fuse reactions to the environmental features, spacial memory and the other bats in close proximity during flight. Parts of the analysis discussed in what follows are presented in [30] and [31].

3.1 The Flight Behavior of *Myotis velifer* Bats

The bat flight behavior analysis is based on data recovered from a large collection of 3dimensional video records of *Myotis velifer*, emerging from a cave on the Bamberger Ranch Preserve in Johnson City, Texas. These are cave-roosting bats that live in southern North America and Central America and are a large bodied species of the *Myotis genus*, weighing about 14 grams and having a wingspan of 30 cm [49, 50].



Figure 3.1: Myotis velifer Bat, Credit: photo by Roger W. Barbour

Bats perceive features within their environment using complex combinations of sensory organs. Many species, including the M. velfer studied in the present document, are able to perceive distances to objects by means of echolocation. It is likely, however, that vision also plays a role in guiding these animals as they fly. Similar to other Yangochiroptera (microbats), M. velifer have relatively small eyes that are principally directed sidewards from opposite sides of the head (Fig. 3.1). Their retinas are rod-dominated, making them well suited to their nocturnal niche. In addition,

their retinas contain dense horizontal connections but few vertical connections, which suggests that they are specialized to detect motions and contours under nocturnal illumination at the expense of high feature discrimination acuity [51]. However, contrary to traditional belief that bats possess a simplistic visual system, recent evidence suggests that several bat species, including M. velifer, have functional S opsin genes [52]. It is unclear whether these genes are expressed in M. velifer retinae, but the presence of functional genes suggests that M. velifer may be able to see UV light and thus possess mesopic vision that is effective at dusk and dawn and on brightly moonlit nights [53]. The optic nerves of M. velifer's left and right eyes remain separate. Each optic nerve crosses over completely to the contralateral side of the brain [51]. This anatomical characteristic suggests that, at least at the lower level, information from each eye is processed separately by the brain.

In what follows, we present the details of the experimental procedure for recording the trajectories of M. *velifer* bats in their natural habitat.

3.1.1 Experimental Procedure for Recording Bat Trajectories

Raw bat flight data were collected shortly after sunset on 30 May, 2011. The bat colony resides in an artificial cave located approximately 50 meters from the point of observation. Upon exiting the roost, individuals immediately begin to disperse over the landscape by following the margin of a forest fragment toward an open flight corridor over a paved ranch road. We collected thermal infrared video of bats with three thermal cameras (FLIR ThermoVision SC8000, FLIR Systems, Inc.) placed along the flight corridor. Our camera system operated at 131.5 Hz with 1024 \times 1024 resolution and used 25 mm lenses. We chose this location because there was an abundance of natural obstacles in the flight corridor and because it was sufficiently far from the roost that bats presumably had accreted into flight groups but were not sufficiently far from the roost to have split from each other to forage separately.

Cameras were placed linearly and perpendicular to the flight direction of the bats. Camera viewing angles were selected so as to optimize reconstruction accuracy at points of direct interaction between bats and a natural obstacle (a hanging vine), and to maximize flight track duration.

On average, each bat was recorded for approximate 300 frames. This was accomplished by localizing the vine (see Fig. 3.2) at a central focal point in each of the three camera views. The 3D geometry of the scene was calibrated by waving an object of known dimension through the shared view volume of the three cameras, in this case a 1.56m PVC wand, and direct linear transformation (DLT) coefficients were calculated from pairs of wand points (for more information refer to [54]). A technician gathered 2D coordinates of each bat in each of the three views using custom annotation software developed by our research group. Flight trajectories were then reconstructed in 3D as described in [55]. For hand annotated positions, some human-generated noise was introduced to the flight trajectories. This uncertainty was smoothed as described in the following section.

3.2 Analysis of the 3D *Myotis velifer* Trajectories

Using three dimensional video recordings of *Myotis velifer* bats, numerical reconstructions of 405 different trajectories were created (Fig. 3.2). Errors inevitably appear in these reconstructions due to uncertainties arising from bats flying outside the 3D calibration region, from occasional occlusions, and from misidentifying homologous points on the bats body in the three views, especially when its size in at least one view is small. Smoothing and filtering were carried out along the lines discussed in [56] but in this case using cubic spline smoothing $\hat{\mu}$ with a smoothing factor $\lambda = 0.85$. (See [57]).



Figure 3.2: Trajectories of 405 bats along with the positions of the three cameras, vine and the pole. Black triangles represent the key features in the environment such as the trees and the big tree branches

$$\min_{\hat{\mu}} \{\lambda \sum_{i=1}^{n} (Y_i - \hat{\mu}(x_i))^2 + (1 - \lambda) \int_{x_1}^{x_n} \hat{\mu}''(x)^2 dx \}.$$
(3.1)

Over the range of the parameter, $0 \le \lambda \le 1$, $\lambda = 0$ corresponds to a linear least squares fit to the data and $\lambda = 1$ corresponds a cubic spline interpolation passing through every data point. The parameter λ is chosen such that the smoothing is good enough for noise cancellation without loosing too much information. Our goal is to understand the behavior of the bats in the average sense. Hence, in order to investigate features such as mean trajectory and variance along the trajectories, the smoothed trajectories are parameterized by arc length. The data is filtered such that the trajectories that appear in the field of view less than 1 second are excluded which result i total 254 bat trajectories. The filtered trajectories are shown in Fig. 3.3.



Figure 3.3: Smoothed bat trajectories illustrated by the color coding to represent their distinct reactions to the vine and the pole

Finally, over the 254 trajectory segments that were retained for study, it was noted that the bats descended at a fairly steady rate so as to follow the descending slope of a hill. These trajectories appeared to be largely confined to a plane, with only small deviations above or below. Hence our initial attempt to understand how the bats' sensory perceptions were guiding their movement has been focused on models of motion control in this plane. The cameras that recorded the flights were located just outside and to the left of the rectangular region displayed in the Fig. $3\cdot 3$. The bats entered the field of view of the three video cameras from the left. At the top of the figure, the small triangles correspond to trees in a wooded area, and at the center of the figure there are two significant features labeled vine and pole. The vine is a natural feature that runs from the ground up to a fairly high tree branch so that the bats must fly either to the left or the right of it. The pole was placed as a vertical marker to calibrate the cameras, and its height was such that bats could either fly over or around it.

The data is initially classified into six groups with respect to the bats' reaction to the obstacles (Pole and Vine). Along -x direction the letters L and R are used to denote whether the bats fly from the left of the obstacles or the right respectively. Also, it is known that the vine is naturally attached to the ground on one end and a tree branch on the other so bats have to fly either left or the right of it. The pole is artificially placed in a vertical position such that bats can either fly over or around it. Hence, the letters U and O is used to denote the classes which fly under or over the pole respectively. By using the definitions above, six groups of bats are named as, Group 1: LLU (39 bats) (Red), Group 2: LRU (3 bats) (Black), Group 3: RRU (14 bats) (Blue), Group 4: RLU (27 bats) (Green), Group 5: LO (73 bats) (Yellow), Group 6: RO (98 bats) (Pink). For instance, Group 1 represents the bats flying left of the vine, left of the pole and under the pole which is symbolized as "LLU".

In Fig. 3.4 mean trajectories of the six sub classes of bats are shown with the associated color coding. The upper half of the graph shows groups 1,2 and 3 from left to right and lower half are the mean trajectories of the groups 4,5 and 6. The triangles in the plots represent the key points marked during the experiments including the trees in the area, vine and the pole. The mean paths are calculated using the step



Figure 3 \cdot **4**: Mean trajectories for six subgroups of bats

sizes of 0.1 meter intervals along the arclength and the mean points are connected to each other. The variances in the x, y, z directions of sample points at each arclength position are illustrated by drawing the variance ellipsoid by calculating the 3-by-3 covariance matrix, its eigenvalues and eigenvectors respectively. For the subgroups, along the mean path of the trajectories the evolution of the variance ellipsoid is plotted by using a MATLAB algorithm. In order to keep statistical significance, a threshold is set to the algorithm such that it runs the simulation for at least 20 trajectories which is illustrated in Fig. 3.5. The simulations are shown only for the four subgroups (LLU, RLU, LO and RO) since the number of trajectories in Group 2 and Group 3 is not sufficient to achieve a positive semi-definite covariance matrix. Hence, the results would be statistically insignificant for variance analysis.



Figure 3.5: The simulation of the variance ellipsoid for four subgroups of bats

Visual inspection of Fig. 3.5 suggested that the differences among the trajectories with respect to the pole were insignificant. Thus we divided the set of trajectories into two major classes: the first comprised of 115 bats who flew to the left of the vine and the second being the 139 bats who flew to the right. Having parameterized all trajectories by arc length, we adapted the viewpoint that we could recreate the flight path of a typical bat by computing mean trajectories within each class. The mean path of the 115 bats who flew to the left of the vine, and the mean path of the 139 bats who flew to the right of the vine (Fig. 3.10) will be used in the following section for the comparison with the generated synthetic trajectories.

3.2.1 Motion Control Using Optical Sensor Feedback

Sebesta and Baillieul [58] showed how the well-known optical parameter τ (time-tocontact) could be used to guide the motion of a moving optical sensor. It is worth redoing this analysis with a special emphasis on the eye geometry of the *M. velifer*. In [47] and [58], τ was described in terms of a geometric picture that is most appropriate for animals (like humans) that have a forward facing field of view. However, the eye geometry of the *M. velifer* is explained in Section 3.1 and it is illustrated that their eyes are shifted toward the sides of the head giving them a wider angle of view and, at the same time, giving them higher acuity in resolving objects that are off to one side or the other [51]. Because of the placement of their eyes and due to neural connectivity patterns of their photoreceptors, these bats have enhanced ability to orient themselves with respect to features in the lateral visual field. We conjecture that the bats may use optical flow sensing of features in the lateral field to guide their motions. In order to explore this, we introduce the concept of *time-to-transit* and discuss how time-to-transit is easily determined from the movement of feature images on an animal's retina or on the image plane of a camera. Consider the idealized planar



Figure 3.6: Optical flow of feature images for a sideward-looking imaging system

vehicle depicted in Fig. 3-6. The direction of motion is aligned with the vehicle body frame x-axis, and the feature O is observed with a pinhole camera system whose camera axis is aligned with the negative body frame y-axis. The image of the point feature O is at d_i , a negative quantity as it is depicted in the Fig. 3-6. We suppose the vehicle moves in the direction of its body frame x-axis at a constant speed v. If the motion is initiated at a point x_0 along the line of flight at time t = 0, it will cross a line that is perpendicular to the line of flight and passes through the feature point O at time $t = \frac{x_0}{v}$. This quantity is called the time-to-transit, and we denote it by τ . We note that as the figure is drawn, the image point corresponding to the feature in our idealized camera lies at d_i in the body frame x-axis, and the focal point lies at fon the body frame y-axis (f is the camera focal length). It is clear that the similarity of triangles implies the relationship

$$\frac{d}{x(t) - d_i(t)} = -\frac{f}{d_i(t)},$$
(3.2)

and from this it follows that,

$$\frac{d_i}{\dot{d}_i} = \frac{x_0}{v} - t. \tag{3.3}$$

This is zero when $t = x_0/v$ (the time at which the vehicle crosses the line perpendicular to its path and passing through O). At t = 0, we see that $d_i/\dot{d}_i = x_0/v = \tau$ is the time- to-transit. The general conclusion is that $d_i(t)$ is the location of an image feature, $\tau(t) = d_i/\dot{d}_i$ is the time remaining until the camera is directly abeam of the actual feature, provided that the speed and heading are held constant.

Referring to Fig. 3.6., we adapt a simple kinematic model of planar motion

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} v\cos(\theta) \\ v\sin(\theta) \\ u \end{pmatrix}, \qquad (3.4)$$

where v is the forward speed in the direction of the body-frame x-axis, and u is the

turning rate. The time-to-transit a feature located at (x_w, y_w) is found as,

$$\tau = \frac{\cos(\theta)(x_w - x) + \sin(\theta)(y_w - y)}{v}.$$
(3.5)

Figure 3.7: Maximizing time-to-transit when flying trough two key features

If we consider the case when there exist two distinct features O_1 and O_2 located at (x_1, y_1) and (x_2, y_2) respectively (Fig. 3.7.), the heading that maximizes the difference time-to-transits $\tau_2(\theta) - \tau_1(\theta)$ is aligned with the direction $(x_2 - x_1, y_2 - y_1)$ and at this heading we have $\tau'_2(\theta) - \tau'_1(\theta) = 0$.

Theorem 3.2.1 Consider point features O_1 , O_2 located respectively at (x_1, y_1) and (x_2, y_2) . Let $\tau_j(t)$ be the time-to-transit associated with the feature O_j , j = 1, 2. Suppose the initial orientation, θ_0 of the vehicle is such that $\tau_2 > \tau_1$ (which implies that $\cos(\theta)(x_2 - x_1) + \sin(\theta)(y_2 - y_1) > 0$). Further assume that the vehicle travels at constant speed v = 1, then for any k the steering control law,

$$u = u(t) = k[\tau'_{2}(\theta(t)) - \tau'_{1}(\theta(t))], \qquad (3.6)$$

where $\tau' = \delta \tau / \delta \theta$ will asymptotically align the vehicle in parallel with the line segment from O_1 to O_2 . *Proof:* Let (v_x, v_y) designate the planar direction from feature O_1 to O_2 :

$$\begin{pmatrix} v_x \\ v_y \end{pmatrix} = \begin{pmatrix} x_2 - x_1 \\ y_2 - y_1 \end{pmatrix} \frac{1}{\sqrt{(x_2 - x_1)^2 - (y_2 - y_1)^2}}.$$
(3.7)

The function $V(\theta) = 1 - \cos \theta v_x - \sin \theta v_y$ will serve as a Lyapunov function with Lie derivative

$$\mathcal{L}_u(\theta(t)) = \frac{\partial V}{\partial \theta} \dot{\theta}, \qquad (3.8)$$

$$= -k\rho(-\sin\theta v_x + \cos\theta v_y)^2, \qquad (3.9)$$

where $\rho = (x_2 - x_1)^2 - (y_2 - y_1)^2$. The set $V(\theta) \leq 1$ is compact and invariant under the motion, and by LaSalle's theorem the motion evolves asymptotically toward the set

$$\{\theta: V(\theta) \le 1\} \cap \{\theta: \mathcal{L}_u V(\theta) = 0\}.$$

The unique value of θ , $0 \le \theta \le 2\pi$, lying in this set is such that $(\cos \theta, \sin \theta) = (v_x, v_y)$. This specifies that the asymptotic direction of the motion is aligned with the line from O_1 to O_2 as stated in the theorem.

Remark 1.2.1: The theorem is conservative in the sense that the hypotheses are intended to restrict the initial conditions to configurations in which the moving vehicle has a non-zero component of its motion in the direction of the line from O_1 to O_2 . If the vehicle had access to its orientation θ in space and to the world-frame coordinates of the features $(x_1, y_1), (x_2, y_2)$, then the control law (3.6) could be written $u(k) = k[-\sin \theta(x_2 - x_1) + \cos \theta(y_2 - y_1)]$. Assuming the system has access this global configuration information, the control law will steer the vehicle to alignment with the feature from every initial configuration except those in which the initial θ_0 has the vehicle aligned with the direction $((x_1 - x_2), (y_1 - y_2))$ (i.e. aligned in exactly the opposite direction from the goal alignment of the theorem). The existence of this singular direction is a familiar characteristic of kinematic control laws for vehicle models of the form (3.4) [73].

Remark 1.2.2: We note that the value of τ that is associated with a visible feature as specified in (3.5) is a purely geometric quantity. It will be negative in the case that the feature lies behind the vehicle on its current line of flight. For side looking eyes, it could still be visible, and would be perceived as being both negative and becoming increasingly negative as the motion continues. The important point is that the control law (3.6) does not assume that either feature lies ahead of the vehicle on its current heading.

Remark 1.2.3: The theorem states what is achieved by keeping the difference $\tau_2 - \tau_1$ at its maximum value. How flying animals might detect this maximum is not understood, but we speculate that small magnitude saccadic eye movements might be used as an energy efficient means of detecting what corrections to the vehicle's heading are needed to keep $\tau_2 - \tau_1$ at its maximum value.

Remark 1.2.4: (Model validity) Field observations together with the unsmoothed reconstructed flight data make clear that the relatively smooth trajectories that are produced by the model (3.4) do not capture the continual rapid, short distance lateral, pitching, and rolling motions that make the animal flight movements anything but smooth. It is our working assumption that these high-frequency deviations from smoothness can be thought of as noise that can be ignored in our initial attempt to synthesize vision-based bat-like trajectories.

To study the curvature of trajectories prescribed by (3.4), we normalize the problem (in Fig. 3.7) such that $(x_1, y_1) = (0, 0)$, $(x_2, y_2) = (1, 0)$ and v = 1. The steering control is then written $u(t) = -k \sin \theta(t)$. Note that this is just the curvature. It follows as a corollary of Theorem 3.2.1 and Remark 1.2.1 that for $\theta_0 \neq \pi$, the absolute value |u(t)| is a monotonically decreasing function that asymptotically approaches zero. The rate of decrease is determined by the gain parameter k, and for any threshold $\alpha > 0$, we can explicitly compute the time $T_{\alpha}(k)$ at which |u(t)| becomes less than α . This is made precise in the following way.

Theorem 3.2.2 Consider the system 3.4 with constant speed v = 1 and the steering law,

$$u = -ksin(\theta(t)). \tag{3.10}$$

Let $k > \alpha$ and $0 < \alpha < k |sin(\theta_0)|$. Then u(t) is a monotonically decreasing that takes on value α at time $t = T_{\alpha}(k)$ where

$$T_{\alpha}(k) = \frac{1}{k} \left[\log(\frac{\theta_0}{2}) - \log(\tan(\frac{1}{2}\sin^{-1}(\frac{\alpha}{k}))) \right].$$
(3.11)

Proof: First we note that the steering equation

$$\dot{\theta} = -k\sin\theta, \quad \theta(0) = \theta_0,$$
(3.12)

can be integrated explicitly in closed form to give

$$\theta(t) = 2\tan^{-1}\left(\tan(\frac{\theta_0}{2})e^{-kt}\right). \tag{3.13}$$

The signed curvature of the trajectory generated by 3.4 is given by

$$\kappa(k,t) = u(t) = -k \sin\left[2 \tan^{-1}\left(\tan(\frac{\theta_0}{2})e^{-kt}\right)\right].$$
(3.14)

For each k, as noted above $\kappa(k, t)$ is a monotonically decreasing function of t so that for each α with $0 \leq \alpha \leq k |\sin(\theta_0)|$, we have a unique solution t > 0 to the equation $\alpha = \kappa(k, t)$. Elementary but slightly tedious algebra yields this solution as

$$t_{\alpha} = \mathbb{T}_{\alpha}(k) = \frac{1}{k} \left[\log \left(\frac{\theta_0}{2} \right) - \log \left(\tan \left(\frac{1}{2} \sin^{-1} \left(\frac{\alpha}{k} \right) \right) \right) \right], \quad (3.15)$$

proving the theorem.

The theorem is useful in understanding the time required for the control law 3.6 to align the flight path with a given straight line direction. Another potentially useful control law that can be based on sensed optical flow is one that keeps the time-to-transit fixed at zero. Consider three features forming the vertices of a triangle as depicted in Fig. 3.8. The vehicle (3.4) can be steered around the three features



Figure 3.8: Optical flow sensing can be used to guide the idealized vehicle to maintain a constant distance (dashed line) from three non-colinear obstacles A, B, and C.

by using the following control protocol. First, assuming the vehicle starts at or below the dashed line segment through feature A, apply control law (3.6) (more precisely, the control $u(t) = k[\tau'_B(\theta(t)) - \tau'_A(\theta(t))]$ so that it aligns itself with the segment AB. As soon as the condition $\tau_B = 0$ is met (i.e. when the vehicle transits the dashed line through B that is perpendicular to its line of travel), the vehicle switches to a control to keep $\tau_B \equiv 0$. This will cause the vehicle to execute a circular arc that maintains a constant distance from feature B. This control continues to be applied until the difference $\tau_C(t) - \tau_B(t)$ attains a maximum value, at which time the vehicle switches to the control law $u(t) = k[\tau'_C(\theta(t)) - \tau'_B(\theta(t))]$. It will continue in
this way in a direction parallel to the segment BC at the same distance to the side of the segment. In terms of the notation, the path segmentation is prescribed by $u_d[A, B:t] \rightarrow u_c[B:t] \rightarrow u_d[B, C:t]$, which will be explained in what follows..

It is important to emphasize that we have not attempted to incorporate neurologically based models of how bats might sense that transit time differences are at a maximum value or how they might fly so as to keep a transit time constant. Optimum seeking control laws and control laws that steer vehicles so as to keep sensed quantities constant are well known [59], but the details of how such control can be carried out using vision and other sensing by the animals remains an open question. In what follows, we consider a number of vision-based control strategies that give rise to flight paths resembling the bat trajectories described in Section3.2.

These control laws will be used based on a certain strategy to generate synthetic trajectories for the model 3.4 for comparison with the bat trajectories illustrated in the preceding section. It is of interest to discover what can be learned from trying to reproduce animal-like trajectories with suitably tuned versions of the simple control laws and protocols that have been discussed in the preceding sections. At the outset, there are two related but fundamentally different questions. First, how closely can we come to producing a "typical" bat trajectory using an idealized flight vehicle with various vision-based control laws of the form we have described. Here the term "typical' trajectory" refers to the mean trajectories described in the previous section. The second question, which may be more difficult, is can we produce simple models and protocols that predict and replicate the variability among the bats in this simple setting. This is to say, can we find control laws and protocols such that all 254 trajectory reconstructions can be reproduced by simple variations of the model parameters. We consider whether an appropriately sequenced set of *single feature* and *paired feature* vision based control laws to produce animal-like trajectories. The

Time-to-Transit vision-based steering controls		
$u_c[\mathcal{O};t]$	single-feature control	keeps $\tau_{\mathcal{O}}$ constant
	—feature circling	follows circular arc at
		constant radius from
		feature \mathcal{O} ,
$u_d[\mathcal{O}_1,\mathcal{O}_2;t]$	paired-feature control	aligns with the line
	-distance maintenance	segment from \mathcal{O}_1 to \mathcal{O}_2 ;
		see Thm.1.2.1
$u_p[\mathcal{O}_1,\mathcal{O}_2;t]$	paired feature control	control law from [58]
	— steers between features	to steer vehicle on
		path between $\mathcal{O}_1, \mathcal{O}_2$.

Figure 3.9: The control laws used to produce animal like trajectories.

table presented in Fig. 3.9 introduces the notation for the control laws used to navigate through the environment. Note that control law $u_p[O_1, O_2; t]$ is integrated from the study [58] which is navigating between two features instead of aligning itself with the line between them. The control law $u_d[O_1, O_2; t]$ is for aligning the vehicle flight path to the desired straight line direction and $u_c[O; t]$ is the control law to keep the time-to-transit at zero. Using these control laws, we seek to synthesize trajectories under two different hypotheses 1. *Cue-directed Strategy*: It assumes that the Bats generate their control strategy purely based on the sensory feedback about the environmental features 2. *Integrated Strategy*: It assumes that the bats use landmarks based on their spatial memory to select features from the environment and generate control strategies based on this filtered features.

The four trajectories (See Fig. 3.10) are generated by the following sequences of controlled motion segments:

• Red curve with squares: $u_p[A, vine] \rightarrow u_d[A, B] \rightarrow u_d[B, C] \rightarrow u_c[pole] \rightarrow u_d[E, F] \rightarrow u_d[., .]$ for the remaining features.



Figure 3.10: Mean bat trajectories (solid curves), synthesized trajectories based on Hypothesis 1 (curves with circles) and synthesized trajectories based on Hypothesis 2 (curves connecting squares). Features used for navigation under Hypothesis 1 are marked by black circles. The subset of features used under Hypothesis 2 are marked by circles with green interior. Red and blue curves correspond to groups of bats passing the vine and the left and the right, respectively.

- Blue curve with squares: u_c[vine] → u_d[B, C] → u_c[pole] → u_d[E, F]
 → u_d[.,.] for the remaining features.
- Red curve with circles: $u_p[A, vine] \to u_d[A, B] \to u_d[B, C] \to u_c[C] \to u_d[C, D]$ $\to d[., .]$ for the remaining features.
- Blue curve with circles: u_c[vine] → u_d[B, C] → u_c[C] → u_d[C, D] → -u_d[.,.] for the remaining features.

As shown by the relatively high rise of the circle-interpolated curves in Fig. 3.10., if a cue-directed strategy is adapted by the bats, after passing point C, both of these two groups would keep following the edge of the woods. On the other hand, if an integrated strategy is adapted by the bats, after passing point C, the bats can use their spatial memory or the memorized sensory features of significant landmarks (such as the trees marked as E and F) for navigation and then use these features to generate their control. Using the integrated strategy, instead of continuing to follow the border of the woods, the bats would take a short cut. By comparing the synthesized trajectories based on these two hypotheses and the mean bat trajectories, it can be found that the trajectories based on the integrated view fit the observed mean trajectories better. Such an observation suggests that the bats indeed adapt an integrated strategy for their navigation. (A pure spatial memory strategy can be rejected by the observation that none of the bats collided with the pole, which was placed there by the researchers during the experiment and unlikely memorized by the bats). An advantage of the integrated strategy over the cue-directed strategy is its time and energy efficiency. By taking the short cut (optimized route) instead of strictly following the forest edge, the bats can save flight time.

The analysis so far investigates bats' decision making based on the environmental features as well as their spacial memory. However, observation of the data suggests that these bats typically fly as a group. Hence, in what follows, we will study the leader-follower behavior of bats which is similar to the analysis of a dance pair discussed in Chapter 2.

3.3 Leader Follower Interactions of a Paired Bat Flight

For a group of animals navigating through a cluttered environment, each individual must utilize sensory cues from both the environment and its neighbors in order to coordinate its motion with the neighbors and achieve effective navigation. A superb example of group navigation is bats emerging from their roost in groups shortly after sunset and flying through a wooded flight corridor to reach their forage ground. In Section 3.2, we have analyzed the data recovered from a large collection of video records of a group of *Myotis velifer* emerging from a cave on the Bamberger Ranch Preserve near Johnson City, Texas, focusing on their sensorimotor behavior with respect to environmental features. In this section, we continue to investigate the same data set by considering the interactions between pairs of bats with the aim of establishing a more unified view of bat navigation behavior.

Based on the species involved and the nature of the flight, paired-animal flight interactions have been mainly studied in the context of two situations: chasing and following. Chasing refers to the case in which a predator tries to catch a prey. The studies [60] and [61] show that bats and dragonflies use a motion camouflage flight strategy, which minimizes motion parallax cues that the prey can extract from its optical flow. Following is less aggressive than chasing and is generally conspecific. In [62] the authors show that a follower bat displays such a behavior to conceal itself from the leader bat in order to increase its prey-capture performance.



Figure 3.11: Flight path statistics for 39 M. velifer are depicted. The red curve is the mean trajectory, and the blue ellipses (centered on the mean trajectory) represent a dispersion of one standard deviation. Two obstacles, a vine and a pole, are denoted as circles. The triangles are visual features in a wooded area (mostly tree branches) and the dotted lines define the edges of the wooded area.

In Section 3.2.1, we discussed the concept of *time-to-transit* and used it as the basis for a collection of vision-based steering control laws. It is argued that time-totransit is a biologically meaningful parameter that could probably be calculated in an animal's visual cortex, and steering control laws based on time-to-transit relative to single and pairs of environmental features were proposed. It was observed that along those portions of the flight corridor where environmental clutter was relatively dense, each of the motion segments needed to be focused on closely spaced features and was of short duration. Along these portions of the motion, switching between control laws (and features) was frequent. Using our control primitives, we were able to develop a motion strategy that would closely approximate the mean flight path of the bats (the red curve in Fig. 3.11). The question remained as to why many animals deviated significantly from this mean path. In this section, we propose that large excursions toward the boundary of the woods could be the result of a trailing bat following a leader according to a certain leader-follower protocol. Using the concept of virtual loom, we formulate a new steering law that produces simulated flight paths consistent with those of pairs of bats observed in the field.

3.3.1 Virtual Loom

We begin the analysis with introducing the flight kinematics of the leader and follower by following the model introduced in [63]. The dynamics of the leader are given as:

$$\begin{cases} \dot{\boldsymbol{r}}_l = v_l \boldsymbol{x}_l \\ \dot{\boldsymbol{x}}_l = v_l \boldsymbol{y}_l u_l \\ \dot{\boldsymbol{y}}_l = -v_l \boldsymbol{x}_l u_l, \end{cases}$$
(3.16)

where v_l is the speed of the leader, r_l is the position of the leader, x_l is the unit tangent vector to the trajectory of the leader, y_l is the corresponding unit normal vector, and the plane curvature u_l is the steering control for the leader. Similarly, the dynamics of the follower are given as:

$$\begin{cases} \dot{\boldsymbol{r}}_{f} = v_{f}\boldsymbol{x}_{f} \\ \dot{\boldsymbol{x}}_{f} = v_{f}\boldsymbol{y}_{f}u_{f} \\ \dot{\boldsymbol{y}}_{f} = -v_{f}\boldsymbol{x}_{f}u_{f}. \end{cases}$$
(3.17)

We assume that the leader and the follower have the same speed. This assumption is consistent with field data.



Figure 3.12: Frenet frame representations of the leader and the follower together with the follower's side-looking eye. L and F are the center axis points of the eyes of the leader and the follower, respectively. |f| is the focal length distance from the lens to the focal plane (retina). L' is the image point corresponding to L. α is the angle between \boldsymbol{x}_l and \boldsymbol{x}_f .

Fig. 3.12 illustrates the geometry of an idealized leader-follower pair moving in a horizontal plane. As noted in Section 3.2 the bat motions in our data set are approximately planar. The directions of motion are aligned with the vehicle body frame x-axes, i.e., \boldsymbol{x}_l and \boldsymbol{x}_f . The leader is observed by the follower with a pinhole camera system whose camera axis is aligned with the follower's negative body frame y-axis, i.e., \boldsymbol{y}_f . The relative position of the leader in the frame of the follower is $r = r_l - r_f$. The projections of r onto the x_f and y_f directions are written as:

$$r_x := \boldsymbol{r} \cdot \boldsymbol{x}_f \text{ and } r_y := \boldsymbol{r} \cdot \boldsymbol{y}_f,$$

$$(3.18)$$

respectively.

In terms of these kinematic models and the follower's imaging system model that is depicted in Fig. 3.12, we recall the definitions of optical flow parameters from [58] and Section 3.2.1. Suppose the follower's initial position is $\mathbf{r}_f(0) = (r_1(0), r_2(0))$ at time t = 0 and it is flying in \boldsymbol{x}_f direction with a constant speed v_f . In such a case, if the leader is stationary, the follower will cross the line of transit at time $\tau = r_x/v_f$. Here line of transit is the line that is perpendicular to the line of flight and passes through the origin of the leader frame, and r_x is the distance between $\mathbf{r}_f(0)$ and this same line of transit. This quantity τ is the time-to-transit. At the initial time (t = 0), d is the distance in the follower's image plane (bat retina) between the leader's image (L' in Fig. 3.12) and the principal camera axis point F (Fig. 3.12), then $\tau = d/d$. If the leader is not stationary, the definition still makes sense and is related to the relative velocities of the leader-follower pair. Of course if the leader and the follower are traveling in the same direction at the same speed, the image distance d does not change over time (d = 0), which reflects the fact that τ must be infinite. Since we shall be interested largely in the case where the leader and the follower fly at essentially identical speeds, we find it more convenient to work with the reciprocal of τ , which is called the *loom*. Since we shall be dealing in particular with situations in which the follower never reaches the point of transit, we define the *virtual loom* as follows:

Definition 3.3.1 For a leader-follower pair (Eqs. (3.16) and (3.17)), the virtual loom $\lambda(t)$ at time t is

$$\lambda(t) = \frac{[1 - \boldsymbol{x}_f(t) \cdot \boldsymbol{x}_l(t)] \boldsymbol{v}_f}{\boldsymbol{r}(t) \cdot \boldsymbol{x}_f(t)}.$$
(3.19)

Notice that $\lambda(t)$ has a unit that is inverse of time. For brevity, we use λ to represent

 $\lambda(t).$

From Fig. 3.12, we have the following relationship:

$$d = \frac{f}{r_y - f} r_x,\tag{3.20}$$

so the follower bat can estimate r_x by sensing d.

In addition, we define an equilibrium state for a pair as follows.

Definition 3.3.2 A leader-follower pair (Eqs. (3.16) and (3.17)) is said to be in a state of λ equilibrium if λ is zero.

Remark 1.3.1: If, as shown in Fig. 3.12, α is the angle between the headings of the two bats, then $\cos \alpha = \mathbf{x}_f \cdot \mathbf{x}_l$. Further, define *transiting* as the instant when the image of the leader on the follower's retina L' coincides with F, the focal point of the follower's retina, which corresponds to $r_x = \mathbf{r} \cdot \mathbf{x}_f = 0$. For two bats flying with the same constant speed $v_f = v_l = v$, a state of λ equilibrium means that the relative velocity of the two bats is zero and L' stays at the same position on the follower's retina. In this case, the follower bat can estimate α by sensing \dot{d} , the optical flow. A zero \dot{d} corresponds to a zero α . On the other hand, a non-zero \dot{d} implies that α is not zero and a transiting is going to happen if no adjustment is made by the follower. Finally, it is worth pointing out that, although in this dissertation we focus on visionbased control, bats can also use other sensory modalities, such as echolocation [64], to estimate time-to-transit τ or virtual loom λ .

Parallel (or near parallel) flight alignment (with $\alpha \approx 0$) has been observed in the mating activity of dragonflies [65], competitive prey capturing in bats [62] and tandem flight of swallows (unpublished results from the Hedrick Lab at UNC Chapel Hill). Benefits of such a flight pattern include aerodynamic efficiency (the follower can utilize the vortex of the leader's wingtip to save energy, known as 'vortex surfing'), stealth (the follower can conceal itself from the leader to increase its prey capturing probability) and echolocation efficiency (the follower bat can turn off its sonar or adopt a low duty cycle).

3.3.2 Statistical Analysis of the Bat Emergence

Previous studies of bat emergence times have been largely focused on how factors, such as sunset time, weather, and the existence of predators, affect the onset of the emergence and the mean emergence time [66], [67]. To our knowledge, there has been no study to model the fine details of emergence rates. However, there exists a rich set of literature on the modeling of human activity emergence, such as sending emails and initiating financial transactions [68].

We define the first time a bat appears in the video as its emergence time. By this, we get an ordered time sequence, $S := \{t_i, i = 1, ..., N\}$, where t_i is the emergence time of the *i*th bat and N = 254 is the total number of recorded bats. Notice that Fig. 3.11 only shows a fraction of the trajectories presented in Section 3.2. The Kolmogorov-Smirnov test is used to determine whether the sequence (or a subset of it) fits a Poisson model.

Fig. 3.13 shows the sampled rate parameter $\bar{\theta}(t)$ of the subset of emergence times that fall within the window [t, t + T]. It can be seen that $\bar{\theta}(t)$ is relatively constant before 200th second and its value is high; it falls rather sharply after 200th second; it becomes relatively constant again after 300th second. Our analysis has shown that the entire emergence time sequence S does not fit a Poisson model. However, the analysis also has shown that the truncated emergence time sequence $S_1 := \{t_i \in [0, 200]\}$ is able to pass the Kolmogorov-Smirnov test for a Poisson arrival process with a constant rate parameter $\bar{\theta}$ of 0.961(Another truncated sequence $S_2 := \{t_i \in [300, 450]\}$ was also tested. But it did not pass the test due to the the lack of enough data points for statistical significance). Further, it has been found that bats emerging within the



Figure 3.13: The sampled rate parameter $\bar{\theta}(t)$ of the subset $\{t_i \in [t, t+T]\}$ with $t \in [0, 450]$ and T = 120 seconds. The time axis corresponds to the whole duration of the recording period with 0 corresponding to the time the recording started.

duration [0, 200] account for 80 percent of the bats.

If we look at an interval of one second, a Poisson arrival process with a rate parameter 0.961 means that there is a 0.3825 probability that there is no bat within the interval, a 0.3676 probability that there is one bat within the interval, a 0.2499 probability (approximately 64 bats for the sample of 254) that there are two or more bats within the interval. Due to the high probability of having neighboring bats, in the next subsection, we will study whether the behavior of a leader bat affects the behavior of a follower bat and if it does, in what way.

3.4 Generating Synthetic Bat Trajectories

The 39 trajectories that are shown in Fig. 3.11 will be analyzed in this subsection that correspond to the group of bats passing the vine from the left and passing the



Figure 3.14: Example trajectories of a leader-follower pair. The black lines connect the paired bats' corresonding locations at different time slices.

pole from the left while flying lower than the upper end of the pole (Group 1,[LLU], as described in Section 3.2).

For the group of 39 trajectories shown in Fig. 3.11, we further select data segments of paired bats for analysis based on the following criteria: the paired bats need to be present simultaneously in the video for longer than 20 frames and the spatial separation between the paired bats must be shorter than 10 meters. A bat can perceive items within 10 meters with a good resolution via its eyes [69]. Given that the average speed of the observed bats is 10.17 m/s, this threshold corresponds to approximately one second difference between the two bats' emergence times). We say that the bat emerging earlier is the leader and the one emerging later is the follower. The trajectories of one such pair are shown in Fig. 3.14.

There is a correlation of R = 0.8894 (Pearson correlation coefficient) between the mean y coordinate of bats emerging within a fixed time window and the number of bats in the window. The result is shown in Fig. 3.15. As shown in Fig. 3.14, a higher y coordinate implies a smaller distance to the woods. Further, the larger the number



Figure 3.15: Mean y coordinates of bats emerging within a 40 second time window versus the number of bats in the window.

of bats emerging within a fixed time window, the shorter the average interval between the successive emergence of two bats, and the higher the probability of having a leader in front of a bat.

These correlations mean that a bat (a follower) behaves differently if there is another bat (a leader) in front of it. In order to further illustrate the behavior difference, we classify the 39 trajectories shown in Fig. 3.11 into four classes. They are

- C_1 : the bat is a single bat, which is neither a leader nor a follower (7 bats);
- C_2 : the bat is a single-role leader bat, which is a leader but not a follower (14 bats);
- C_3 : the bat is a dual-role bat, which is both a leader and a follower (4 bats);
- C_4 : the bat is a single-role follower bat, which is follower but not a leader (14 bats).



Figure 3.16: Flight path statistics of G_1 and G_2 are depicted. The red (green) curve is the mean trajectory of G_1 (G_2). The blue (black) ellipses (centered on the mean trajectory) represent a dispession of one standard deviation of G_1 (G_2).

We then combine the four classes into two groups: the leader group $G_1 = \{C_1, C_2\}$ and the follower group $G_2 = \{C_3, C_4\}$. The statistics of the two groups are shown in Fig. 3.16. It is quite obvious that the follower group curves more toward the wooded area than the leader group.

To conclude, as the number of bats emerging within an interval becomes larger or equivalently the initial distance (the emergence interval) between the leader-follower pair becomes smaller, the follower bat tends to stay closer to the wooded area and take a longer route than the leader bat. One possible interpretation of the observed effects is that the trailing bat is trying to align itself with the leader while staying a safe distance away from the obstacles, e.g. the pole. For the specific environment as shown in Fig. 3.16, a side effect of such a behavior is a larger excursion towards the woods for the follower bat.

3.4.1 Are Bats Pursuing?



Figure 3.17: Analysis results for the pair shown in Fig. 3.14: the baseline direction r/|r| (blue) and the angle between the baseline direction r/|r| and the follower's heading x_f (black). Both are represented as angles. For instance, the blue curve is computed by $\tan^{-1}(p_2/p_1)$ with p_1 and p_2 being the first and second component of r/|r|.

In this subsection, we analyze paired bats' behavior by checking the data against existing pursuit laws: classical pursuit, constant bearing, and motion camouflage [70]. In classical pursuit, the follower aligns its direction of motion \boldsymbol{x}_f with the baseline direction $\boldsymbol{r}/|\boldsymbol{r}|$, where the baseline \boldsymbol{r} is defined as $\boldsymbol{r}_l - \boldsymbol{r}_f$ in Section 3.3.1; in constant bearing, the follower keeps the angle between its heading \boldsymbol{x}_f and the baseline direction $\boldsymbol{r}/|\boldsymbol{r}|$ constant; in motion camouflage, the follower keeps the baseline direction $\boldsymbol{r}/|\boldsymbol{r}|$ constant.

Fig. 3.17 illustrates that none of these pursuit laws explains the behavior observed in Fig. 3.14. The baseline direction r/|r| (blue curve) does not stay constant, which violates motion camouflage pursuit; the angle between the baseline direction r/|r| and the follower's heading \boldsymbol{x}_f (black curve) is neither zero nor constant, which violates classical and constant bearing pursuits. The result implies that the follower bats are not pursuing the leader bats (by pursuing we mean that there is a moment when the follower intercepts the leader). The reasons may be as stated in Remark 1.3.1. Nevertheless an alternative interpretation is needed to explain the observed behavior. In the next section, we will propose a steering law and a navigation strategy the follower bat might use.

3.4.2 Steering Law For Following

In Section 3.2, we proposed an integrated strategy to explain the navigation behavior of *M. velifer* in a data set of 254 individuals (the same data set from which we are selecting the bat pairs studied here). We hypothesized that these bats used landmarks recalled from their spatial memory to select features from the environment and then generated control strategies based on these remembered features. Synthesized trajectories generated by using sequences of feature-based control primitives approximately fit the mean behavior of the bats. However, as noted in Section 3.3.2, bats following leaders behave differently form those that do not. The interaction between the bats is a factor that we have not considered in Section 3.2. In this section, we first propose a steering law that a follower bat may use to follow a leader. We then discuss a strategy that takes the leader-follower behavior into consideration and show that now the statistics, both the mean and the variance, of the synthesized trajectories fit with those of the data on bat pairs.

The planar steering law we study next is based on minimizing the virtual loom in a follower's perception of the leader's motion.

Theorem 3.4.1 Consider leader-follower pair (Eqs. (3.16) and (3.17)) with the following assumptions:

1. the control of the leader u_l is zero (the leader flies in a straight line);

2. r_x is positive (the leader is in front of the follower).

Then for k > 0, the follower with control

$$u_f = k\boldsymbol{x}_l \cdot \boldsymbol{y}_f = -k\sin\alpha, \tag{3.21}$$

will asymptotically align itself with the leader, i.e., $\alpha \to 0$ (and $\lambda \to 0$).

Proof: We take the unnormalized virtual loom as a Lyapunov function $V := 1 - \boldsymbol{x}_l \cdot \boldsymbol{x}_f$. This is 0 if $\boldsymbol{x}_l \cdot \boldsymbol{x}_f = 1$ ($\alpha = 0$) and positive otherwise. Its derivative along trajectories is

$$\dot{V} = -\dot{\boldsymbol{x}}_l \cdot \boldsymbol{x}_f - \boldsymbol{x}_l \cdot \dot{\boldsymbol{x}}_f = -u_f(\boldsymbol{x}_l \cdot \boldsymbol{y}_f) = -k(\boldsymbol{x}_l \cdot \boldsymbol{y}_f)^2.$$
(3.22)

which is zero when $\boldsymbol{x}_l \cdot \boldsymbol{y}_f = 0$ or equivalently $\boldsymbol{x}_l \cdot \boldsymbol{x}_f = 1$ ($\alpha = 0$).

Theorem 3.4.1 implies that if the leader is flying in a straight line, then the follower can utilize the virtual loom to achieve parallel flight with the leader. See Remark 1.3.1 for the explanation of how bats might estimate the virtual loom.

3.4.3 Simulation Results

Fig. 8 shows a pair of synthesized trajectories with the follower using control law (3.21) and leader with the steering control $u_l = 0$. It can be seen that with control law (3.21), the follower is approaching a parallel flight with the leader as described by Theorem 3.4.1. In the case that $u_l \neq 0$, the leader's trajectory is similar to the one depicted in Fig.3.14, and the synthesized follower's trajectory with control law (3.21) is qualitatively similar to the actual follower bat's trajectory as shown in Fig. 3.14. (Details will be elaborated in Section 3.4.4)

3.4.4 Is Pure Following Strategy Sufficient?

Fig. 3.19 shows the actual trajectories of a leader-follower pair and a synthesized follower trajectory (purple) by using control law (3.21) with the assumption that the



Figure 3.18: Synthesized trajectories with the follower using control law (3.21). Dots indicate starting locations. The black lines connect the pair's corresponding locations at different time slices.

follower only reacts to the leader without utilizing either its spatial memory or cues from the environment. The purple synthesized trajectory fits with the actual follower bat's trajectory (green) well for the segment that has x coordinates smaller than 9 meters. This implies that the follower bat synchronizes its motion with the leader inside the open space between the pole and the wooded area. However, after passing 9 meters, the discrepancy between the synthesized and actual trajectories becomes larger. The synthesized trajectory has the danger of colliding with the obstacles or losing track of the leader due to occlusion. Here, we need to consider a navigation strategy that integrates a rapid refocus of attention on the looming tree obstacles.

3.4.5 Integrated Strategy: Spatial Memory Fused with Reactions to Environment and Other Bats

The integrated strategy proposed in Section 3.2.1 is now extended so as to incorporate the leader-follower behavior. The navigation strategy is synthesized from three visionbased control primitives: a distance maintenance law $u_d[\mathcal{O}_1, \mathcal{O}_2]$, a circling control



Figure 3.19: Actual bat trajectories (red: leader bat, green: follower bat) and synthesized follower trajectories (purple: based on control law (3.21), blue: based on the integrated strategy).

law $u_c[\mathcal{O}_1]$ and a leader-follower control law $u_f[\mathcal{O}_1]$, where \mathcal{O}_1 and \mathcal{O}_2 are visually perceived features used in a particular control law and can be either static (for u_d and u_c) or moving (for u_f). The primitives u_d and u_c can be found in Section 3.2.1, while the primitive u_f is given in Eq. (3.21).

Fig. 3.20 shows the statistics of 100 synthesized trajectories based on the new integrated strategy. The vehicles are assumed to move according to Eqs. (3.16) and (3.17) with a constant speed. The vehicles appear in the field in accordance with a Poisson process. Their arrival locations and velocities are generated randomly by a Gaussian model with its mean and variance the same as those of the collected bat data. (We only simulate the bats' behavior after they pass feature 'a' as shown in Fig. 3.20.) The intersubjective distance between a pair of vehicles determines whether there exists a leader for the trailing vehicle to follow. If there exists a leader,



Figure 3.20: 100 synthesized trajectories based on the new integrated strategy. Labeled features are the ones that are assumed being memorized by the bats.

the follower vehicle relies on the leader and the control u_f for navigation. It switches to environment-cue-directed control u_d or u_c when it is on a collision course. On the other hand, if there does not exist a leader, the follower vehicle relies on its spatial memory and cues from the environment for navigation and the controls they can use are u_d and u_c . For Fig. 3.21, each trajectory is generated by a sequence of controlled motion segments as follows:

- If there does not exist a leader, the trajectory is generated by $u_c[\text{pole}] \rightarrow u_d[b,c] \rightarrow u_d[c,d] \rightarrow u_d[.,.]$ for the remaining features;
- If there exits a leader, the trajectory is generated by $u_f[\text{leader}] \to u_d[b,c] \to u_d[c,d] \to u_d[.,.]$ for the remaining features.

For Fig. 3.19, the follower trajectory (blue) is generated by the second strategy since it has a leader (red). We prescribe the switching between the primitives based on the nearest feature(s) in the follower's body x_f direction. For instance, the switching



Figure 3.21: The plot of the mean path and variance ellipse of 100 synthesized trajectories

from $u_f[\text{leader}]$ to $u_d[b, c]$ is triggered if feature b is closer to the follower than the leader in the x_f direction. Similarly, the switching from $u_d[b, c]$ to $u_d[c, d]$ is triggered if feature d is closer to the follower than feature b in the x_f direction.

A comparison between Fig. 3.21 and Fig. 3.11 shows that the synthesized trajectories accurately capture both the mean and the variance of the actual bat trajectories with the only difference being that the ellipses in Fig. 3.11 are slightly fatter. One possible explanation of the difference is that the sensors are assumed to be noiseless for the synthesized trajectories while this is not the case for actual bats. Similarity can also be observed between the actual follower's trajectory (green) and the synthesized trajectory based on the integrated strategy (blue) in Fig. 3.19. Such resemblances support our integrated strategy hypothesis.

By following another bat, in the context of navigation, a follower bat can save energy by adopting a low duty cycle echolocation or even turning off its sonar completely [71]. It can also be used by an inexperienced individual to follow an experienced one. In such a case, the leader (e.g. a female bat) is more familiar with the environment than the follower (e.g. a juvenile).

Chapter 4

Conclusions and Future Work

The contributions of this dissertation can be summarized as follows. In Chapter 2, we used salsa as a prototype to study dance pair interactions by means of gesture-based communication between the leader and the follower dancer. This was achieved by decomposing a dance move into distinct motion primitives and their corresponding motion signals. We defined a Beginner Level Salsa (BLS) as a dance performed by using the four fundamental salsa moves. We introduced a new motion description language by assigning a letter from an alphabet to each distinct eight beat salsa move. The constructed framework was used to search for "perceived artistic merit" of a dance performance by introducing two mathematical metrics that measure the energy expenditure of the dancers as well as the artistic expressivity of the dance phrases. We investigated a robotic dance pair to determine the energy consumed by the dancers during the execution of the moves. Moreover, we have proposed an entropy metric to capture the artistic expressivity in the performed four letter dance phrases. We conducted an experiment in which various dance performances were evaluated by judges and observed strong correlations between the judges' scores and the proposed mathematical metrics.

The framework was extended to Intermediate Level Salsa (ILS) by introducing seven new moves that involve the follower dancer's π or 2π degrees rotation in clockwise and counter clockwise direction. We constrained the dancers' movements by restricting them to keep hand contact throughout the performance. Under this constraint, we investigated the syntax of dance move transitions using the language of topological knot theory. Each dance move was decomposed into an initial and a final pose. The link diagram representations of the moves were extracted and the link invariants were computed. We concluded that the allowable dance move transitions decided by the leader are based on the syntactic requirement of matching the topology of the initial pose of a move with the topology of the final pose of the preceding move.

We introduced three new moves that allow the leader to break the hand contact in order to return to the most basic dance pose. The new alphabet was referred to as Extended Intermediate Level Salsa (EILS). We calculated a new link invariant, *Alexander Polynomial*, for each pose in EILS so that distinct physical dance motion primitives were represented by polynomial function manipulations.

We discussed the finite state machine representation (FSM) of a dance pair. Initial and final poses were defined as states that involve the roll, pitch and yaw angles of the eight links in stick figure representations of the dancers. We constructed a communication channel between the FSMs in order to represent the transmitted motion signals by the leader to the follower. In the case when there exists no noise, we showed the bounds of the cost for communicating the dance phrases. Moreover, we incorporated energy and entropy metrics into an optimization problem, which we refer to as a forward problem, in order to generate optimal dance performed by a robotic dance pair. We extended the analysis to the case when the communication channel is noisy. The communication channel was represented by a multi-input, multi-output erasure channel and we discussed how the model representing the pair with an expert level leader has a higher capacity channel in transmitting signals.

We described the solution of an inverse problem which involves the construction of a robotic judge to evaluate group execution. The robotic judge has an observation and an evaluation component. Observation was achieved by using a finite state machine representation of the dance pair. Dancers' body poses were tracked by using critical points in their stick figure representations. We showed that a dance performance can be deconstructed by the recognition of the initial and final dance poses and the motion signals. The evaluation component is based on a *Score* function that assigns a score to a deconstructed performance based on the energy and entropy metrics defined earlier. The performance of the robotic judge was validated by comparing its score values assigned on a new set of dance performances with the actual judges' evaluations.

One interesting future work direction is to extend the human collective behavior analysis to a group of dancers with multiple pairs. Rueda Salsa is a motivating example. By observing a rueda salsa performance, one may conclude that the dance involves two distinct group behaviors. In one case, each dance pair performs independently which can be perceived as local pairs generating move sequences by following the rules explained in Chapter 2. In the second case, the pairs separate to form a dance circle. The group behavior can be understood by adapting terminologies from graph theory. The circle formation is generated such that no two leaders or two followers are adjacent. This artistic constraint can be understood by a bipartite cyclic graph representation of a rueda circle. This reveals the interrelationship between the physical dance motion primitives and the change in the topology of the graph.

In Chapter 3 of this dissertation, we extended the analysis to the case when a group of bats emerge from a cave. We described the details of the experimental procedure to record *Myotis velifer* trajectories in their natural habitat. The recorded trajectories were smoothed and filtered by using cubic spline smoothing. The smoothed trajectories were used to classify the bats based on their decision making to avoid obstacles in their flight corridor. We introduced a new concept time-to-transit which is a quantity that can be computed by animals' visual cortex. We incorporated this concept to generate bat-like trajectories in a simulated environment that is identical to the bats' natural flight corridor. We concluded that the bats react to the key features in the environment and use spacial memory for navigation.

We also investigated the leader-follower interactions using the same data set. A leader bat was defined as the bat which flies in the range of the follower bats' vision or echolocation calls. We modified the concept time-to-transit to capture the configuration of a leader follower pair and introduced virtual loom. We generated a new set of synthetic trajectories driven by time-to-transit and virtual loom for individual or group flight. We concluded that the bats use beacons and spacial memory to navigate while reacting neighbors in close proximity.

The proposed control laws for imitating bats' navigation strategies during cave emergence were based on their optical flow sensing capabilities. However, it is a well known fact that bats predominantly use echolocation calls to perceive features during flight. Hence, we conducted a new set of experiments to study the learning behavior as well as the navigation strategies of bats. The experimental set up involved equipment to record bat trajectories as well as their echolocation calls during an emergence from the same cave in Johnson City, Texas. During the first day of the second experiment data was acquired as bats navigated through the same flight corridor as in the first set of experiment. The second day a new obstacle was introduced to observe behavioral changes by means of their call rates and trajectories. The experiment continued for six consecutive days to reveal their learning behavior. The preliminary results suggest that bats' call rates increased significantly when the new obstacle was introduced. Additionally, their reaction distance with respect to the obstacle was significantly closer. However, we observed a decrease in their call rate and an increase in their reaction distance on the consecutive days. The preliminary results suggest that they register the location of the pole to their spacial memory. The results of this experiment along with the leader follower analysis will be reported in future publications.

Bat cave emergence analysis has features in common with the analysis of dance since it involves bats reacting to the environmental features in addition to the internal group member interactions and leader-follower behavior. Lessons learned from both prototypes can be applied to real world applications including team athletics and military applications. The unifying element is constraining the agents to communicate by using the gestures and motions so that messages can not be detected decoded by the enemy. Appendices

Appendix A



Figure $\cdot 1$: The Foot Work of the Dancers and Performance of the Non-Holonomic Robots



Figure \cdot 2: The Foot Work of the Dancers and Performance of the Non-Holonomic Robots

Appendix B



Figure $\cdot 3$: Initial and final poses for eleven moves in Intermediate Level Salsa (ILS)



Figure $\cdot 4$: Initial and final poses for eleven moves in Intermediate Level Salsa (ILS)

Appendix C

Seq5-Beginner1:

ABmDAACmAmAmCABADAACAABAmCmACCmBA BCDCCDmABmABADACADmDCCACmBACAmBCD CABCDACBACmACABBCDmCBD Seq6-Expert: ABDACCBADDBCABDACCBBBDmCDBBDABBD CCBBCDDBCDABDBDBCCBBBDBBDBCBDBm CBBBAmBDDmBBCDDBBACmCAABD Seq7-Beginner7: AABmACABACDBmADCAAmBmCDABBmDAABC mAABCDmBABmDABAADCAABBDDCmABDBAB CBCmAADBBmDADBCBDBCmDBCm

Figure $\cdot 5$: Dance sequences performed by beginner and expert level dancers

References

- Swain, Daniel T., Iain D. Couzin, and Naomi Ehrich Leonard. "Real-time feedback-controlled robotic fish for behavioral experiments with fish schools." *Proceedings of the IEEE* 100.1 (2012): 150-163.
- [2] Zhang, He-Peng, Avraham Beer, E-L. Florin, and Harry L. Swinney, "Collective motion and density fluctuations in bacterial colonies." *Proceedings of the National Academy of Sciences* 107.31 (2010): 13626-13630.
- [3] Von Frisch, Karl, Seeley, Thomas D., Chadwick, Leigh E., The Dance Language and Orientation of Bees, Harvard University Press, 1993.
- [4] Schoellig A. P., Augugliaro F., and DAndrea R., "A platform for dance performances with multiple quadrocopters" in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)- Workshop* on Robots and Musical Expressions, 1-8, 2010.
- [5] Hsieh, M. Ani, Mallory, K., Forgoston E., and Schwartz, Ira B., "Distributed allocation of mobile sensing agents in geophysical flows" *American Control Conference (ACC)*, IEEE, 2014.
- [6] Schwager, Mac, Dean M. Anderson, Zack Butler, and Daniela Rus, "Robust classification of animal tracking data." *Computers and Electronics in Agriculture* 56.1 (2007): 46-59.
- [7] Stewart, A., Cao, M., Nedic, A., Tomlin, D., Leonard, N. E. "Towards humanrobot teams: Model-based analysis of human decision making in two-alternative choice tasks with social feedback." *Proceedings of the IEEE* 100.3 (2012): 751-775.
- [8] Novitzky, Michael, Charles Pippin, Thomas R. Collins, Tucker R. Balch, and Michael E. West, "Bio-inspired multi-robot communication through behavior recognition" *IEEE International Conference on Robotics and Biomimetics (RO-BIO)*,2012.
- [9] Laban, Rudolf (1975). Labans Principles of Dance and Movement Notation.
 2nd edition edited and annotated by Roderyk Lange. London: MacDonald and Evans. (First published 1956.)

- [10] Hutchinson-Guest, Ann. (1970). Labanotation: The System of Analyzing and Recording Movement, 4th revised edition (2005). New York: Theatre Arts Books. (First published 1954).
- [11] Hattori, Motofumi, Syun Nishizawa, Satoshi Tadokoro, Toshi Takamori, and Kazuhito Yamada, "The description of Bunraku puppet's motion in computer based on Labanotation" Proceedings of the 10th IEEE International Workshop on Robot and Human Interactive Communication, 2001.
- [12] LaViers, Amy, Egerstedt, Magnus, Chen, Yushan, and Belta, Calin, "Automatic generation of balletic motions" Proceedings of the 2011 IEEE/ACM Second International Conference on Cyber-Physical Systems. IEEE Computer Society, 2011.
- [13] LaViers, Amy, Chen, Yushan, Belta, Calin, and Egerstedt, Magnus, "Automatic sequencing of ballet poses" *IEEE Robotics & Automation Magazine*, 18.3 (2011): 87-95.
- [14] LaViers, Amy, and Egerstedt, Magnus, "Style based robotic motion." American Control Conference (ACC), IEEE, 2012.
- [15] Egerstedt, Magnus, Todd Murphey, and Jon Ludwig. "Motion programs for puppet choreography and control." *Hybrid Systems: Computation and Control*, Springer, Berlin Heidelberg, 2007. 190-202.
- [16] Leonard, Naomi E., George F. Young, Kelsey Hochgraf, Daniel T. Swain, Aaron Trippe, Willa Chen, Katherine Fitch, and Susan Marshall, *In the Dance Studio: An Art and Engineering Exploration of Human Flocking*, Controls and Art, Springer International Publishing, 2014. 27-49.
- [17] Leonard, Naomi Ehrich, George Young, Kelsey Hochgraf, Daniel Swain, Aaron Trippe, Willa Chen, and Susan Marshall, "In the dance studio: Analysis of human flocking" American Control Conference (ACC), IEEE, 2012.
- [18] Young, George, F., Scardovi, L., Cavagna, A., Giardina, I., and Leonard, N.
 E. "Starling flock networks manage uncertainty in consensus at low cost." *PLoS computational biology* 9.1 (2013): e1002894.
- [19] Hristov, Nickolay I., Margrit Betke, Diane EH Theriault, Angshuman Bagchi, and Thomas H. Kunz, "Seasonal variation in colony size of Brazilian free-tailed bats at Carlsbad Cavern based on thermal imaging" *Journal of Mammalogy* 91.1 (2010): 183-192.
- [20] Baird, Emily, Mandyam V. Srinivasan, Shaowu Zhang, Richard Lamont, and Ann Cowling, "Visual control of flight speed and height in the honeybee" From Animals to Animats 9. Springer, Berlin Heidelberg, 2006. 40-51.

- [21] Graham, Paul, and Thomas S. Collett. "View-based navigation in insects: how wood ants (Formica rufa L.) look at and are guided by extended landmarks." *Journal of experimental Biology* 205.16 (2002): 2499-2509.
- [22] Raghunathan, Dhananjay, and John Baillieul. "Relative motion of robots as a means for signaling." *Proceedings of the International Conference on Intelligent Automation and Robotics*, San Francisco, USA. 2008.
- [23] Pagello, Enrico, Antonio DAngelo, Federico Montesello, Francesco Garelli, and Carlo Ferrari, "Cooperative behaviors in multi-robot systems through implicit communication" *Robotics and Autonomous Systems* 29.1 (1999): 65-77.
- [24] Abras, Shadi, Sylvie Pesty, Stphane Ploix, and Mireille Jacomino, "An anticipation mechanism for power management in a smart home using multi-agent systems" 3rd IEEE International Conference on Information and Communication Technologies: From Theory to Applications (ICTTA), 2008.
- [25] Baillieul, John, Özcimder, Kayhan, "The Control Theory of Motion-Based Communication: Problems in Teaching Robots to Dance," *American Control Conference*, June 27-29, Montreal, Canada, pp. 4319 - 4326, 2012.
- [26] Baillieul, John, Ozcimder, Kayhan, Dancing Robots—The Control Theory of Communication Through Movement, Controls and Art, A. LaViers and M. Egerstedt (Eds.), Springer, 2014.
- [27] Ozcimder, Kayhan, "Communication Through Motion in Dance with Topological Constraints", American Control Conference (ACC), Portland, IEEE, Oregon, 2014.
- [28] Ozcimder, Kayhan, Kong, Zhaodan, Baillieul John, "Algorithmic Approaches to Artistic Movement", to appear in *IEEE Conference on Decision and Control*, CDC, 2014, LA, California.
- [29] Ozcimder Kayhan, Kong, Zhaodan, Wang Shuai, Baillieul John, "Salsa Robotic Judge: A Formal Exploration of Performance Art, submitted to Robotics and Automation Magazine, 2014.
- [30] Kong, Zhaodan, Ozcimder, Kayhan et al. "Optical flow sensing and the inverse perception problem for flying bats" *IEEE 52nd Annual Conference on Decision* and Control (CDC), 2013.
- [31] Kong, Zhaodan, Ozcimder, Kayhan et al. "Perception and Steering Control in Paired Bat Flight", *International Federation of Automatic Control*, 2014, South Africa.
- [32] LabanWriter software, http://dance.osu.edu/3 research gallery/laban writer.html.
- [33] Dutton D., The Art Instinct: Beauty, Pleasure, and Human Evolution, Bloomsbury Press, 288 pages, 2010.
- [34] Roads C. (ed.), The Computer Music Tutorial, MIT Press, 1996.
- [35] Baillieul J. and Wong W.-S., "The Standard Parts Problem and the Complexity of Control Communication," *Proceedings of the 48-th IEEE Conference* on Decision and Control, Shanghai, 2723-2728, December 16-18, 2009.
- [36] Wong W.S., "Control Communication Complexity of Distributed Control Systems," SIAM Journal of Control and Optimization, 48(3), 1722-1742, 2009.
- [37] Wong W.S. and Baillieul J., "Control Communication Complexity of Nonlinear Systems," *Communications in Information and Systems*, 9(1), 103-140, 2009. http://people.bu.edu/johnb/CIS-9-1-A5- wong.pdf.
- [38] Manikonda V. and Krishnaprasad P.S. and Hendler J, "A Motion Description Language and a Hybrid Architecture for Motion Planning with Non-holonomic Robots", In Proceedings of the IEEE International Conference on Robotics and Automation, 2021-2028, 1995.
- [39] Przytycki J. H., 2007. "History of Knot Theory", http://arxiv.org /pdf/ math/0703096v1.pdf.
- [40] Reidemeister K., Knotentheorie, Vol. 1, Ergebnisse der Mathematik und ihrer Grenzgebiete, Springer, Berlin, 1932.
- [41] Lickorish, W. B. Raymond, 1997. "An Introduction to Knot Theory", Graduate Texts in Mathematics, Springer-Verlag.
- [42] Waddell A. J. . "Topological invariants of knots and links", Transactions of the American Mathematical Society 30.2 (1928): 275-306.
- [43] Cover T. M. and Thomas J. A., *Elements of information theory*, 2nd, Ed. Hoboken, N.J.: Wiley-Interscience, 2006.
- [44] Murasugi Kunio, 1929. Knot Theory and Its Applications.
- [45] Aggarwal, J, K, Cai, Q, "Human Motion Analysis: A Review", Computer Vision and Image Understanding, Vol 73, No. 3, March, 428- 440, 1999.

- [46] Kaiser M.K. and Mowafy L., 1993. "Optical specification of time-to-passage: Observers sensitivity to global tau, *Journal of Experimental Psychology: Hu*man Perception and Performance, Vol 19(5), 1028-1040. doi: 10.1037/0096-1523.19.5.1028
- [47] Lee D.N. and Reddish P.E., 1981. "Plummeting gannets: a paradigm of ecological optics, *Nature*, 293:293-294.
- [48] Wang Y. and Frost B.J., 1992. "Time to collision is signaled by neurons in the nucleus rotundus of pigeons, *Nature* 356, 236 238 (19 March); doi:10.1038/356236a0
- [49] Farney J., Fleharty E.F. 1969. "Aspect ratio, wing span, and membrane areas of bats," *Journal of Mammalogy*, vol. 50, no. 2, 362-367.
- [50] Fitch J.A., Shump K.A., Jr., Shump A.U., 1981. "Myotis velifer," Mammalian Species, vol. 149. 1-5.
- [51] Jones M.P., Pierce Jr K.E., Ward D., 2007. "Avian Vision: A Review of Form and Function with Special Consideration to Birds of Prey, *Journal of Exotic Pet Medicine*, April 2007, 16.2 (2007), 69-87.
- [52] Muller B., Glosmann M., Peichl L., Knop G. C., Hagemann C., and Ammermuller J., 2009. "Bat eyes have ultravioletsensitive cone photoreceptors," *PloS one*, vol. 4, no. 7, p. e6390.
- [53] Wang D.T., Oakley J., Mower J., Shimmin L.C., Yim S., Honeycutt R.L., Tsao H., and Li W., 2004. "Molecular evolution of bat color vision genes, *Molecular Biology and Evolution*, vol. 21, no. 2, 295-302.
- [54] Abdel-Aziz, Y.J. and Karara, H.M. (1971)" Direct linear transformation from comparator coordinates in close range photogrammetry" In ASP Symposium on Close-Range Photogrammetry, 1-18.
- [55] Towne, G., Theriault, D.H., Wu, Z., Fuller, N., Kunz, T.H., and Betke, M. (2012) "Error analysis and design considerations for stereo vision systems used to analyze animal behavior", In Proceeding of IEEE Workshop on Visual Observation and Analysis of Animal and Insect Behavior (VAIB), in conjunction with the 21th International Conference on Pattern Recognition, volume 4. Tsukuba Science City, Japan.
- [56] Dey B. and Krishnaprasad P.S., 2012. "Trajectory Smoothing as a Linear Optimal Control Problem," in Proceedings of the 2012 Allerton Conference on Communication, Control, and Computing.

- [57] Wahba G., Wisconsin, 1983. "Bayesian Confidence Intervals for the Cross-Validated Smoothing Spline," *Journal of the Royal Statistical Society*, Series B (Methodological), Vol. 45, No. 1, 133-150.
- [58] Sebesta K. and Baillieul J., 2012. "Animal-inspired agile flight using optical flow sensing, in Proceedings of the 51st IEEE Conference on Decision and Control, Maui, December 10-13, 3727-3724, DOI: 10.1109/CDC.2012.6426163. Available from http://arxiv.org/abs/1203.2816
- [59] Baornov D. and Baillieul J., 2011. "A Motion Description Language for Robotic Reconnaissance of Unknown Fields," *European Journal of Control*, Sept.-Dec. 2011, Vol. 17:5-6, 512-525. DOI:10:3166/EJC.17.512-525.
- [60] Mizutani, A., Chahl, J.S., and Srinivasan, M.V. (2003) "Insect behaviour: Motion camouflage in dragon flies", *Nature*, 423(6940), 604-604.
- [61] Ghose, K., Horiuchi, T.K., Krishnaprasad, P.S., and Moss, C.F. (2006), "Echolocating bats use a nearly time optimal strategy to intercept prey", *PLoS biology*, 4(5), e108.
- [62] Chiu, C., Reddy, P.V., Xian, W., Krishnaprasad, P.S., and Moss, C.F. (2010) "Effects of competitive prey capture on flight behavior and sonar beam pattern in paired big brown bats, eptesicus fuscus", *The Journal of experimental biology*, 213(19), 3348-3356.
- [63] Justh, E.W. and Krishnaprasad, P.S. (2006). "Steering laws for motion camouflage" Proceedings of the Royal Society A: Mathematical, Physical and Engineering Science, 462(2076), 3629-3643.
- [64] Shaw, B.K., McGowan, R.S., and Turvey, M. (1991) "An acoustic variable specifying time-to-contact", *Ecological Psychology*, 3(3), 253-261.
- [65] Wagner, H. (1986)" Flight performance and visual control of flight of the free-flying house fly (musca domestica l.) ii. pursuit of targets", *Philosophical Transactions of the Royal Society of London, Biological Sciences*, 312(1158), 553-579.
- [66] Welbergen, J.A. (2006), "Timing of the evening emergence from day roosts of the grey-headed flying fox, pteropus poliocephalus: the effects of predation risk, foraging needs, and social context", *Behavioral Ecology and Sociobiology*, 60(3), 311-322.
- [67] Kunz, T.H. and Anthony, E.L. (1996), "Variation in the timing of nightly emergence behavior in the little brown bat, myotis lucifugus (chiroptera: Vespertilion- idae)" Contributions in Mammalogy: A Memorial Volume Honoring Dr. J. Knox Jones, Jr, 225-235.

- [68] Barabasi, A. (2005), "The origin of bursts and heavy tails in human dynamics", Nature, 435(7039), 207-211.
- [69] Wimsatt, W. (1970), Biology of Bats, Academic Press.
- [70] Wei, E., Justh, E.W., and Krishnaprasad, P.S. (2009), "Pursuit and an evolutionary game", Proceedings of the Royal Society A: Mathematical, Physical and Engineering Science, 465(2105), 1539-1559.
- [71] Chiu, C., Xian, W., and Moss, C.F. (2008). "Flying in silence: echolocating bats cease vocalizing to avoid sonar jamming", *Proceedings of the National Academy of Sciences*, 105(35), 13116-13121.
- [72] Lickorish, W. B. Raymond, 1997, An Introduction to Knot Theory, Graduate Texts in Mathematics, Springer-Verlag.
- [73] Baillieul, J., and A. Suri. "Information patterns and hedging Brockett's theorem in controlling vehicle formations." *Proceedings of the 42nd IEEE Conference on Decision and Control*, 2003 on. Vol. 1, 2003.

CURRICULUM VITAE

Hasan Kayhan Özcimder

119 Sutherland Road Apt K Brighton, MA 02135 Email: kayhan1983@hotmail.com

EDUCATION

Boston University, College of Engineering, Boston MA Master of Science (Mechanical Engineering), August 2011 Gazi University, Mechanical Engineering Department, Ankara/Turkey Bachelor of Science (Mechanical Engineering), June 2007

PROJECTS

Research Assistant (Fall 2010 - Fall 2014)

Conducting Research at the Intelligent Mechatronics Laboratory. The ongoing research aims to investigate the communication through motion in multi-agent systems. The research investigates collective behavior in human and animal groups.

COURSES

Adaptive Control, Robotics, Planning and Vision, Nonlinear Systems, Dynamical System Theory, Robust and Optimal Control, Introduction to Differential Geometry, Stochastic Processes.

COMPUTER SKILLS

MATLAB, Simulink, Mathematica environments, Windows XP/Vista/7, MS Office 2007, Latex, PHP, HTML, Blender

AWARDS

Turkish Ministry of Education STUDY ABROAD program Masters degree scholarship for the field of Mechatronics.