



Theses and Dissertations

2020-11-16

Formulating a Crowd State Prediction Problem for Application to Crowd Control

Brooks A. Butler
Brigham Young University

Follow this and additional works at: <https://scholarsarchive.byu.edu/etd>



Part of the [Physical Sciences and Mathematics Commons](#)

BYU ScholarsArchive Citation

Butler, Brooks A., "Formulating a Crowd State Prediction Problem for Application to Crowd Control" (2020). *Theses and Dissertations*. 8718.
<https://scholarsarchive.byu.edu/etd/8718>

This Thesis is brought to you for free and open access by BYU ScholarsArchive. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of BYU ScholarsArchive. For more information, please contact scholarsarchive@byu.edu, ellen_amatangelo@byu.edu.

Formulating a Crowd State Prediction Problem for
Application to Crowd Control

Brooks A. Butler

A thesis submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of
Master of Science

Sean Warnick, Chair
Mark K. Transtrum
Christopher J. Archibald

Department of Computer Science
Brigham Young University

Copyright © 2020 Brooks A. Butler

All Rights Reserved

ABSTRACT

Formulating a Crowd State Prediction Problem for Application to Crowd Control

Brooks A. Butler
Department of Computer Science, BYU
Master of Science

This project considers a new application of crowd control, namely, keeping the public safe during large scale demonstrations. This problem is difficult for a variety of reasons, including limited access to informative sensing and effective actuation mechanisms, as well as limited understanding of crowd psychology and dynamics. This project takes a first step towards solving this problem by formulating a crowd state prediction problem in consideration of recent work involving crowd behavior identification, crowd movement modeling, and crowd psychology modeling. We build a non-linear crowd behavior model incorporating components of personality modeling, human emotion modeling, group opinion dynamics, and group movement modeling. This model is then linearized and used to build a state observer whose effectiveness is then tested on system outputs from both non-linear and linearized models. We show that knowledge of the crowd emotion equilibrium is necessary for zero-error convergence; however, other parameters, such as individual personality parameters of crowd agents, may be approximated and zero-error convergence still achieved given the crowd equilibrium point and sign of agent opinions. We conclude that using this model class to predict live crowd emotion may be impractical due to the need for knowledge of individual agent personality parameters to simulate the crowd equilibrium. Directions for future work are discussed.

Keywords: Networks, estimation, contagion, emotion, modeling

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Related Work	4
1.2.1	Crowd Psychology	4
1.2.2	Control Models	5
1.2.3	Crowded Scene Analysis	5
1.3	Contributions	6
2	Background	8
2.1	Individual Personality	8
2.2	Emotion	10
2.3	Opinion	11
2.4	Movement	12
3	Crowd Behavior Model	14
3.1	Preliminaries	14
3.1.1	OCEAN parameters	14
3.1.2	PAD model	15
3.1.3	Opinion	16
3.1.4	Position	16
3.1.5	Distance penalty	16
3.2	Emotion dynamics	16

3.3	Movement dynamics	19
3.4	Opinion dynamics	20
3.5	Complete model	21
3.5.1	Emotion	21
3.5.2	Movement	22
3.5.3	Opinion	22
3.6	Crowd Simulation Results	23
4	Linearized Crowd Model	27
4.1	Preliminaries	27
4.1.1	Jacobian Matrix	27
4.2	Model Linearization	28
4.2.1	Emotion Dynamics Jacobian Matrix	29
4.2.2	Emotion Dynamics Equilibrium	31
4.2.3	Linearized Model	32
5	State Estimation	34
5.1	Building a State Observer	34
5.2	Equilibrium Point Shifting of Non-linear Model	36
5.3	Necessity of Equilibrium Knowledge	38
5.4	State Estimation Simulated Results	40
5.4.1	Simulation Methods	40
5.4.2	Comparing Linearized and Non-linear System Results	41
5.4.3	Comparing Non-linear System Predictions of Close vs. Far Agents	45
5.4.4	State Estimation with Unknown Personalities	47
5.4.5	Larger Crowds	47
5.4.6	Edge Case: Emotion Dynamics Limit Cycle	49

6	Conclusions and Future Work	53
6.1	Conclusions	53
6.2	Future Work	54
6.2.1	Crowd Modeling	54
6.2.2	State Estimation	55
6.2.3	Data Collection	55
	References	57
	Appendices	65
A	Python Code	66
A.1	CrowdEmotionFunctions.py	66
A.2	model_tests.py	79

Chapter 1

Introduction

This thesis presents efforts to model realistic crowd behavior and solve the state estimation problem for predicting crowd state given crowd system measurements. In this section, we address the specific motivation for this work, as well as previous work in relation to solving this and similar problems. Finally, we give a summary of our findings and the structural layout of work performed for this thesis.

1.1 Motivation

With the recent protests in the wake of the George Floyd incident [2], the public has watched in dismay as either crowds have turned violent or the people's right to assemble has been squelched [77]. Finding the happy medium, where peaceful public outcry is enabled, is not an easy task. On the one hand, public protests can easily induce mob violence, as we've observed so very recently, and police have a responsibility to keep both people and property safe from the angry hoard. On the other hand, the excuse of protecting public safety can be used as an excuse to suppress public assembly and open, free speech altogether, as we have also seen very recently in both Hong Kong [1] and Washington DC [37]. Developing technologies that enable safe, peaceful public assembly, instead of weaponizing an armed militia to subdue them, is the objective of our work.

To achieve this objective, we look to the field of Control Theory, specifically to problems involving crowd control. Crowd control is an emerging application of network control theory that considers contexts in which crowds can be effectively managed. Crowds are

typically modeled as networks, and managing them involves controlling the values of certain variables within or throughout the network, often with limited and distributed sensing and actuation mechanisms.

Examples of network control problems that focus on human networks include the control of epidemics [58], [60], shaping opinion dynamics in social networks [63], [82], data-driven marketing [52], and the spread of rumors [27] or (fake) news [81], [59], [3], [6], [61] within a population. All of these applications involve driving the values of certain quantities in networks of interacting people.

Nevertheless, some of these network control problems involve large numbers of people that share physical proximity. This, then, becomes the primary feature we use to distinguish the subclass of *crowd control problems*. These problems demand special attention because, in contrast to the examples listed above, they often not only demand distinct models of aggregate human behavior, but also involve aggregate sensing or actuation mechanisms that cannot independently act on individual nodes in the network.

One important class of such crowd control problems that has received significant interest is the evacuation problem; see for example [7], [66], [76], [78], [80]. These problems are characterized by the crowd being in a confined area while under stressful circumstances, and the objective is to quickly relocate the crowd. Models that drive this work not only need to describe how the crowd's physical properties, such as location and density, change under various circumstances, but they also begin to characterize psychological properties of the crowd as an entity (and not, for example, as a network or aggregate of individuals). Information entropy, for example, is used in recent work to characterize "panic" as a property of crowds in evacuation scenarios [85].

Although we hope to build from evacuation models in formulating the *demonstration-safety problem*, there are a number of important differences. First, while evacuations consider a single crowd, we envision a context where two distinct crowds are co-located and possibly intermixed. This is the case, for example, when a public demonstration induces the

spontaneous formation of another group, politically or ideologically opposed to the first, as observed, for example, in postgame altercations at sporting events [30], [74] and at the 2017 Unite the Right rally in Charlottesville, VA [44].

Another important difference is that evacuation models typically consider passive controls, sometimes built into the architecture and design of meeting places, while demonstration-safety problems may consider active controls, including the placement of peacekeepers within the crowd or the use of crowd-dispersion mechanisms. Although current practice tends to use such mechanisms to suppress and disrupt demonstrations—effectively preventing the realization of the entire mode of expression—our ultimate goal is to understand whether such mechanisms might effectively be employed in a more strategic and surgical manner to support demonstration activity by effectively ensuring the safety of all members of the crowd.

Finally, sensing mechanisms available in evacuation scenarios, such as video, may not always be readily available in demonstration-safety situations. This is because evacuation, say of a sporting arena or nightclub, generally occurs at locations that can be pre-instrumented, both with cameras and with the necessary lighting to make video useful. Demonstrations, on the other hand, often occur spontaneously in arbitrary, outdoor locations—and frequently at night. In such circumstances, obtaining real-time measurements of the crowd state that can be used to effectively direct the placement of peacekeepers, and the strategic local use of crowd dispersion mechanisms, may be challenging—or even a complete roadblock for demonstration-safety. This project leverages previous work by the author comparing the informativity audio versus video sensor data which suggests that audio is comparable to video for identifying crowd violence. This work will be the first step in formulating a research program for *predicting* crowd emotional state by considering psychological model creation and behavior identification and prediction for real crowds.

1.2 Related Work

The study of crowds and crowd behavior can be found throughout several disciplines of scientific research. The depth of this research, however, is still limited to an introductory level in most fields. We sort current literature on crowd behavior and analysis into the following categories: crowd psychology, control models, and crowded scene analysis.

1.2.1 Crowd Psychology

Theories on the psychology of crowds have been postulated in academic literature since as early as the late 19th century [38]. However, in contemporary literature, little to no concrete conclusions have been drawn or thoroughly tested on the subject. Some even refer to crowd psychology as the “elephant man” of the social sciences, referring to its being something of a social enigma [29]. Popular theories for modeling crowd behavior, as discussed by [29], include contagion, emergence, and convergence theories. Contagion theory models the spread of emotion through a crowd similarly to the spread of a disease through a population, with the likelihood of an agent being “infected” by another crowd member’s emotion being measured by some form of susceptibility factor. Emergence theories contend that crowd behaviors emerge from a particular grouping of crowd members that otherwise would not be present in the behavior of individuals (ie. the whole is not simply composed of the sum of the parts) while convergence theory argues that crowds are the result of groups of like-minded individuals converging to the same ideas, goals, and behaviors.

Contemporary research involving crowd psychology range from engineering design for crowd evacuation [71], to sentiment extraction and identification from text analysis of crowds on social media [50], [25], [40], to the proper policing of crowds at and around European football games [30], [74]. Additionally, crowd psychology may be used as a contextual backdrop to inform ideas on general policy for public order policing [67], which is the main interest of this research. Perhaps most relevantly, recent work on directly parameterizing crowd emotions for simulation has been performed by [12], [13], [41], [84]. This work uses

a contagion model to simulate the spread of emotion and utilizes many ideas developed by literature published for video game development.

1.2.2 Control Models

A significant interest has been shown in the proper modeling of crowd movement during high stress scenarios. One scenario is the optimal evacuation of a crowd during an emergency. For details on the various proposed models for crowd evacuation see [7], [66], [76], [78], [80]. In many of these models crowds are compared to fluids, governed by some form of fluid mechanics, and they obey the law of conservation of mass. There is also a proposed model for panic quantification—in association with crowd evacuation—based on principles of information entropy [85]. Additionally, other movement based models have been proposed that attempt to simulate crowd behavior based on an entropic path-integral model [34]. However, none of the discussed models attempt to directly characterize crowd emotion nor make predictions on crowd psychological state, which is the main goal of this project.

1.2.3 Crowded Scene Analysis

The rise of computer vision in machine learning has encouraged research across many fields of study involving the automatic classification of images. When computer vision is applied to video, i.e. multiple consecutive images in time, to detect specific behaviors of objects or persons, it becomes the field of *action identification* [31]. In particular, the classification and analysis of surveillance video has become increasingly popular as video surveillance technology becomes more widespread and commonly used. Additionally, the growth of crowd phenomena captured in these videos has led to development of crowded scene analysis and abnormal or anomalous crowd behavior detection [39].

Proposed methods for crowd behavior detection include various forms of optical flow [10], [11], [48], [57], [75], pixel behavior entropy calculation [68], and foreground-background separation [83]. The identification of crowd violence can be classified as an extension of

crowded scene analysis and thus far has only involved applications of computer vision techniques. Examples of methods used to identify crowd violence through video are [15], [26], [32], [36], [49], [51], [70], [73].

1.3 Contributions

In all of these cases, we see that although crowds have been studied in various ways and for various purposes, the estimation of emotional state of a crowd remains unexplored. Our work in this area builds on previous work to detect violence in crowds but seeks to predict violence by estimating the emotional state leading up to outbursts. Our specific contributions include:

1. Bringing together relevant models from a rich variety of disciplines,
2. Combining distinct models from the literature on crowd movement, emotional evolution, and opinion dynamics to create a novel crowd model,
3. Formulating the state estimation problem for this model,
4. Linearizing the model around its equilibrium point to build an algorithm that solves the state estimation problem,
5. Demonstration that the solution built from the linearization works on the actual non-linear system.

The remainder of this thesis will be structured as follows. In Chapter 2 we cover more thoroughly the literature communities from which we build our crowd behavior model, then in Chapter 3 we construct our model and note where it differs from the literature. Because our model differs from the literature largely through the addition of non-linear components, we then discuss methods for linearizing our crowd behavior model in Chapter 4. Using this linearized model, we then build a state estimator and evaluate its effectiveness at state prediction on both the linearized and non-linear crowd models given their system outputs.

Finally, we discuss in Chapter 6 our conclusions, among which are that this model requires unreasonable knowledge of individual crowd agent parameters, and propose directions for future work.

Chapter 2

Background

As this work is built from pre-existing models in the literature, we review the respective contributions made by each research community regarding each state variable in our model. Background for modeling methodologies is given in four parts: individual personality modeling, human emotion modeling, group opinion dynamics, and crowd movement modeling.

2.1 Individual Personality

Personality modeling is central to modeling how humans behave, interact, and react to certain situations and stimuli. As a crowd is comprised of a group of individuals with potentially different behavioral tendencies, it is necessary that we choose a model that represents the differences in individuals that you may find in a real crowd. There have been many models for human personality as posed by researchers in the field of human psychology. One subclass of commonly used personality models are type based models, where people are classified into qualitatively different types based on certain behaviours. Some examples of type based personality models include the Myer-Briggs type indicator model [53], type A/B personality classification [20], and John L. Holland's RIASEC vocational model [54]. The discrete nature of these personality models, however, make them difficult to use quantitatively in numerical modeling. Thus, a more continuous personality model is preferred for this research.

For a continuous model of personality, few personality models have been as widely used and accepted as the Five Factor Model (FFM), or OCEAN model, of personality [45].

This model proposes that an individual's personality can be described by a five-dimensional vector of real numbers, where each vector element describes a distinct personality factor for that person. The five personality factors are described as follows, with descriptions for low vs. high values for each factor:

- Openness: consistent/cautious vs. inventive/curious,
- Conscientiousness: easy-going/careless vs. efficient/organized,
- Extroversion: solitary/reserved vs. outgoing/energetic,
- Amiability: challenging/detached vs. friendly/compassionate,
- Neuroticism: secure/confident vs. sensitive/nervous

Conventionally, each personality value is bounded on the interval of $[-1, 1]$ to give a notion of high versus low amounts of a given personality factor, with neutral values being close to zero.

One of the more useful characteristics of this model is its ability to treat individual personalities as real valued vectors, which lends itself to useful operations attributed to vector spaces, such as a mathematical measure of similarity through the dot product for example. Consequently, the OCEAN model has been used to predict customer responses to targeted advertising [43], and is even speculated to be one of the models used by Cambridge Analytica to predict and manipulate responses to targeted political advertisements during the US 2016 presidential elections [9]. Additionally, OCEAN parameters have been used to model individual and crowd behavior and expression for simulation in artificial environments such as video games [12], [13]. This model is also used to simulate other more specific scenarios including political rallies and hazardous situations [41], [84].

Another useful feature of the OCEAN model is its ability to map an individual's personality to other personality features such as empathy [35] and resting emotional state [46], indicative of one's inherent or "natural" personality. These features are both useful for modeling crowd behavior and will be discussed in greater depth in Chapter 3.

2.2 Emotion

According to research in psychology, there are three major approaches to affect modelling of human emotion [23]: categorical, dimensional and appraisal-based approaches. The categorical approach claims that there exist a small number of emotions that are basic, hard-wired in our brain and recognised universally [14]. However, one shortcoming of this approach is that it does not always capture the complexity of human emotion responses which can be subtle in their differences. The dimensional approach of emotion modeling posits that any human emotion can be defined as a point in some emotion space, where each axis of this coordinate space represents some nearly orthogonal component of emotion. In the appraisal-based approach emotions are generated through continuous, recursive subjective evaluation of both our own internal state and the state of the outside world (relevant concerns/needs) [69]. However, due to the subjective nature of this approach it can be difficult to concretely define emotions and emotional responses across all individuals. Because our model requires consistency of emotional classification across all agents and a continuous space over which emotions can evolve, we choose a dimensional approach to emotion modeling. For a more rigorous survey of emotion modeling see [24].

This work uses the pleasure, arousal, dominance (PAD) model of emotion [47] to characterize the spectrum of human emotion for crowd agents. This model supposes three orthogonal real-valued axes for describing human emotional state, where each axis represents the level of pleasure, arousal, and dominance respectively that a given agent is feeling. The pleasure axis describes the either positive or negative feelings experienced by an agent and is tied to an agent's evaluation of their current situation, with positive values showing a positive evaluation and vice versa. The arousal axis represents a level of mental activity ascribed to a given agent, where negative values may denote states such as sleepy, non-energetic, bored, relaxed, etc., and positive values attributed to high alertness, stimulated, etc. Finally, the dominance axis represents an agent's feelings of control where positive values indicate feelings of influence, autonomy, control, etc., and negative values indicate feelings of submission, lack

Octant	P	A	D	Octant	P	A	D
Relaxed	+	-	+	Anxious	-	+	-
Dependent	+	+	-	Disdainful	-	-	+
Exuberant	+	+	+	Bored	-	-	-
Docile	+	-	-	Hostile	-	+	+

Table 2.1: Labels for each of the PAD space octants as given by [22]

of control, etc. For an overview of the PAD model for human emotion see [5].

This model for emotion is useful for its ability to characterize a continuous space for emotion where a great number of simple and complex emotions can be described. In [22] correlations are given between PAD values and emotions described by the OCC (Ortony, Clore, Collins) model for human appraisal [56], as well as a general definition for emotions found in each PAD octant as described in Table 2.1.

Another important aspect for modeling emotion dynamics is emotional contagion. Much like how the spread of disease can be modeled by describing the change in a susceptible, infected, and recovered (SIR) agents [58], [60], we can also model the spread of emotion through a crowd of connected agents using emotional contagion [21], where an agents susceptibility and ability to infect are described by their personality parameters. This method of describing crowd emotion dynamics using emotional contagion is used by [12], [41], [84] to model how emotions expressed by surrounding agents in a crowd affect individual agents. This will be discussed in greater detail in Chapter 3.

2.3 Opinion

The field of opinion dynamics (OD) seeks to describe how opinions change over time in a group of connected agents with various levels of influence over each other. The most basic model for OD, called the French-DeGroot (FG) model [16], assigns stochastic weights for each agent in relation to every other agent, including themselves, that represent the level of influence each agents holds over every other agent in the simulation. We use the

Friedkin-Johnson (FJ) model of opinion dynamics [19], [18], [17], being very similar to the FG model except for one addition, which describes how an agent's opinion changes given a certain stubbornness value. This determines how likely an agent is to change their opinion in relation to their original opinion at the start of the simulation. For a more in-depth overview of the field of opinion dynamics in social networks, including a stability analysis for the FJ model, the see [64], [65].

Typically, opinion dynamics models are applied to networks such as social media environments or other internet based communications. However, in our case with live crowds, we conjecture that an agent's opinion is in many ways a crucial factor in how they will react to the emoting of other agents around them. As such, we will need to determine some measure of influence each crowd member would hold over every other crowd member, which will be discussed further in Chapter 3.

2.4 Movement

One of the unique challenges of modeling live crowd behavior is determining how a crowd member will move in a given scenario. In the controls community, there is a strong body of literature for modeling crowd movement during high stress scenarios. As discussed, one scenario is the optimal evacuation of a crowd during an emergency. For details on the various proposed models for crowd evacuation see [7], [66], [76], [78], [80]. In many of these models crowds are compared to fluids, governed by some form of fluid mechanics, and obey the law of conservation of mass. Other movement based models include panic quantification—in association with crowd evacuation—based on principles of information entropy [85] and simulating crowd behavior using an entropic path-integrals [34].

Another strong group of literature for crowd based movement is found in the flocking and active matter research communities, whose focus is often more general in attempting to model many types of crowd phenomena such as flocks of birds [4], [62], [28], schools of fish [33], groups of drones [79], active matter [42], etc. These models are useful for considering

coordinated and semi-coordinated crowd efforts, which also allows for crowd formations and particular groupings to occur. For a review of flocking algorithms and theory see [55].

In this research, we choose to model the movement of individual agents, as each agent's movement and behavior should be governed by the inner states of their emotion, opinion, and personality respectively. An initial simplified model of movement is implemented to describe how these various states affect individual agent movement. However, a more complex movement model may be implemented in future work that allows for more realistic crowd groupings and crowd formations.

Chapter 3

Crowd Behavior Model

In this chapter we define a new model dynamics for crowd emotion, movement, and opinion respectively. Before discussing these models, however, we cover some preliminary variable definitions. Additionally, we highlight where our model draws from the literature and where we deviate to better describe our crowd behavior dynamics.

3.1 Preliminaries

To describe crowd behavior, we assume an agent based model, i.e. each crowd member is described individually given a set of initial conditions and the collective behavior of the individual crowd members describe our crowd dynamics. Furthermore, each agent is assigned a fixed personality, defined by the OCEAN model [45] as discussed in Section 2.1, which affects how certain agents influence and are influenced by other agents in the crowd. We assume fixed personalities for agents due to the shorter time scale over which our simulations are run.

3.1.1 OCEAN parameters

Define the set of OCEAN parameters $p \in \mathbb{R}^5$ for an individual as the vector

$$p = [\psi^O, \psi^C, \psi^E, \psi^A, \psi^N]^T \quad (3.1)$$

where $\psi^O, \psi^C, \psi^E, \psi^A, \psi^N$ represent the level of openness, conscientiousness, extroversion, amiability, and neuroticism respectively for a given agent. As is common practice, each

parameter is bounded in the range of $[-1, 1]$. To describe the susceptibility of an agent to being influenced by the emotions of other crowd members, we require some measure of empathy, or the ability to understand and share the feeling of another, for each agent. In psychological literature, empathy was found to be positively correlated with all five factors of personality in the OCEAN model, with correlation values being drawn between the basic empathy scale (BES) and personality factors. Thus, we also define the empathy $\epsilon \in \mathbb{R}$ for an agent as a linear combination of its OCEAN parameters as found in the literature [35]

$$\epsilon = 0.354\psi^O + 0.177\psi^C + 0.135\psi^E + 0.312\psi^A + 0.021\psi^N. \quad (3.2)$$

Due to our bounding of OCEAN parameters to in interval $[-1, 1]$, agent empathy ϵ will also be bounded on that same interval.

3.1.2 PAD model

We define the emotion for a given agent as the vector $e \in \mathbb{R}^3$ where $e = [e_P, e_A, e_D]^T$ and e_P, e_A, e_D represent an agent's current pleasure, arousal, and dominance respectively. We bound the emotion values for each agent such that $\|e\| \leq 1$, where $\|\cdot\|$ is the 2-norm. Furthermore, the literature also defines a mapping from an agents personality score to their resting emotional state [46], ie. the natural emotional state an agent tends to when otherwise unaffected. Let the mapping from an agent's OCEAN parameters to its resting emotion vector e^* be defined as

$$e^* = Bp = \begin{bmatrix} e_P^* \\ e_A^* \\ e_D^* \end{bmatrix} \quad (3.3)$$

where $B \in \mathbb{R}^{3 \times 5}$ is a set of constants defined in [46] based on a mapping of OCEAN personality scores to resting emotional state and e_P^*, e_A^*, e_D^* are the resting values of pleasure, arousal, and dominance respectively for a given agent.

3.1.3 Opinion

Define $o \in \mathbb{R}^n$ as the vector of n opinions/viewpoints of an agent relating to specific issues, where each element represents separate issue. Positive values denote a supporting viewpoint and negative values an opposing viewpoint. All opinion values are bounded between $[-1, 1]$.

3.1.4 Position

We define the position of an agent as $x \in \mathbb{R}^2$. For this study, we choose a 2-dimensional movement space as many crowd situations can be modeled in as agents moving in the same plane. We can imagine exceptional circumstances where it may be necessary to model crowd movement in $x \in \mathbb{R}^3$, such as in a mall environment with multiple tiers of movement, but for simplicity we only consider $x \in \mathbb{R}^2$.

3.1.5 Distance penalty

To model how greater distances reduce the relative affects of agents on each other, we define a distance penalty from agent i to agent j as

$$d_{ij} = \exp(-\eta \|x_i - x_j\|) \quad (3.4)$$

where η is a positive real constant controlling the penalty rate.

3.2 Emotion dynamics

In this section we define the emotion dynamics for each crowd member i . To begin, we define the discrete time emotion dynamics for crowd member i as

$$e_i[k + 1] = e_i[k] + \Delta e_i[k] \quad (3.5)$$

where $\Delta e_i[k]$ represents the change in emotion for agent i at time k of the simulation. We

define $\Delta e_i[k]$ in two parts

$$\Delta e_i[k] = A_i(e_i[k] - e_i^*) + \lambda(s_i, e_{j \neq i}) \quad (3.6)$$

where $A_i(e_i[k] - e_i^*)$ is a linear time invariant (LTI) model that ensures agent's emotion converges to its defined resting emotion state[46] e^* if otherwise unaffected. The linear dynamics for emotional convergence are defined by A_i

$$A_i = \begin{bmatrix} -\alpha n_i & 0 & 0 \\ 0 & -\beta n_i & 0 \\ 0 & 0 & -\gamma n_i \end{bmatrix} \quad (3.7)$$

where α, β, γ are real positive constants controlling the rates of convergence for pleasure, arousal, and dominance respectively. Because negative values for the parameter neuroticism, ψ^N , are associated with emotional stability, we define n_i for a crowd member as

$$n_i = \frac{-\psi_i^N + 1}{2} \quad (3.8)$$

which scales n_i to to the interval $[0, 1]$.

The second term $\lambda(s_i, e_{j \neq i})$ represents the effect of emotional contagion from other crowd members. For the effect of contagion on agent i with respect to every other agent j we define an initial model for emotional contagion as

$$\lambda(s_i, e_{j \neq i}) = \frac{1}{N-1} \sum_{j \neq i}^N d_{ij} e_j q_j s_i \quad (3.9)$$

where N is the number of agents in the crowd, d_{ij} is the distance penalty between two agents, e_j is the current emotional state of crowd member j , $q_j = \frac{\psi_j^E + 1}{2}$ is the extroversion of agent j scaled to the interval $[0, 1]$, and

$$s_i = \frac{\epsilon_i + 1}{2} \quad (3.10)$$

is an agent's susceptibility to being influenced by other's viewpoints and emotions. This initial model, similar to the models posed by [21], [12], [41], and [84], is inspired by the SIR model of disease spread through a population where an agent's extroversion or expressiveness q_j measures their capability to infect other agents with their emotion, and the susceptibility of agent i is defined by their empathy as measured by their OCEAN parameters.

However, this model fails to account for how other agent's opinions, and their subsequent expression of said opinions, affects the change in emotion for another given agent. For example, according to the above model if agent j is approving of a certain sentiment or idea expressed at a political rally, they will express a positive emotion and thereby begin to infect other agents with their same emotion. But, consider the case where agent i is in complete opposition to the opinion of the expressive agent j . In this case, the sentiment of expressing approval and positive emotion would evoke a negative reaction in agent i according to their opinion. Thus, the above model, in some regard, requires that crowd members opinion be somewhat homogeneous in order to simulate realistic behavior.

To account for more complex behavior including agent opinion, we add a sign change to the model. By taking the dot product of two agent's opinion vectors we can measure how similar each agent is in opinion to another agent, with positive values indicating agreement and negative values disagreement. Thus, the sign of the dot product of two agent opinion vectors can inform the model how a particular agent may react to another's expressed emotion. However, simply flipping the sign of the emotion may also yield unrealistic crowd behavior, such as violence only spreading among agents who agree with each other. This introduces the idea that certain emotions are positive contagious, i.e. they will always infect other agents with that same emotion regardless of opinion, while other emotions may be positive or negative contagious, i.e. agents may be infected with the opposite emotion the depending of the similarity of agent opinion. To decide which emotions are positive contagious and which emotions are positive or negative contagious, we refer to the emotion label for each octant in the PAD space as shown in Table 2.1. For our model, we choose that

emotions where the sign of the pleasure and arousal are the same (e.g., the hostile, bored, exuberant, and docile octants) always add in the contagion model, whereas emotions with differing signs in arousal and dominance (e.g., the relaxed, anxious, dependant, disdainful octants) are subject to sign change dependant on opinion. This sign change is reflected in the improved emotional contagion equation

$$\lambda(s_i, e_{j \neq i}) = \frac{1}{N-1} \sum_{j \neq i}^N d_{ij} e_j q_j s_i h_{ij} \quad (3.11)$$

where $h_{ij} \in \{-1, 1\}$ determines whether agent j contributes positively or negatively to the emotion of agent i defined by

$$h_{ij} = (1 - \text{ReLU}[-\text{sgn}(e_{A_j} e_{D_j}) - \text{sgn}(o_i^T o_j)]) . \quad (3.12)$$

3.3 Movement dynamics

We define the movement dynamics for an agent i as

$$x_i[k+1] = x_i[k] + \Delta x_i \quad (3.13)$$

we define the velocity Δx_i of an agent in two parts, the first being a diffusion component which encourages agents to spread out in the movement space while a second emotional component drives agents to move either towards or away from each other depending on relative emotion and opinion

$$\Delta x_i = \overrightarrow{\text{Diffusion}_i} + \overrightarrow{\text{Emotion}_i}. \quad (3.14)$$

The component of an agent's movement relating to the natural tendency for people to spread

out in a given space is defined as

$$\overrightarrow{Diffusion}_i = \frac{\rho}{N-1} \sum_{j \neq i}^N d_{ij}(x_i - x_j) \quad (3.15)$$

where ρ is a positive real constant controlling the diffusion rate. For the sake of simplicity, we assume a bounded area (by peacekeepers, physical boundaries, etc.). A flocking model of group movement may be applied for future work.

The movement component relating to an agent's emotion is defined as

$$\overrightarrow{Emotion}_i = \frac{\chi}{N-1} \sum_{j \neq i}^N d_{ij}(x_j - x_i) \|e_j\| q_j h_{ij} \quad (3.16)$$

where χ is a positive real constant controlling the emotion attraction rate, $q_j = \frac{\psi_j^E + 1}{2}$ is the extroversion of agent j , which is a measure of the level at which they express their emotions scaled from $[0, 1]$. We apply the same additive rules that are implemented for the the emotional contagion term in Equation 5.26

3.4 Opinion dynamics

We use the Friedkin-Johnsen (FJ) model of opinion dynamics,

$$o_i[k+1] = \nu \omega_i \sum_{j=1}^N w_{ij}[k] o_j[k] + (1 - \nu \omega_i) u \quad (3.17)$$

where $\omega_i = \frac{\psi_i^O + 1}{2}$ is an agent's openness to being influenced by other's viewpoints, $\nu \in \mathbb{R} \leq 1$ is a situational parameter, independent of agents, that determines how hardened an agent's viewpoint will be depending on a given scenario, $u = o_i(0)$ is the agent's initial viewpoint at the beginning of the simulation, and w_{ij} is the weight that agent i places on the opinion of agent j . Let $w_{ii} = 1 - \omega_i$ represent the weight agent places on their own opinion (ie. the stubbornness of the agent). For the effect that other agents may have on opinion, let us first

define an influence score for agent j on agent i as

$$I_{ij} = a \frac{p_i^T p_j}{\|p_i^T p_j\|} + b \frac{e_i^T e_j}{\|e_i^T e_j\|} + c \frac{o_i^T o_j}{\|o_i^T o_j\|} \quad (3.18)$$

where a, b, c are positive real constants that determine how much importance agents place on personality, emotion, and opinion similarities respectively in determining opinion influence.

Now, let us define the weights for agents where $i \neq j$ as

$$w_{ij} = \frac{\omega_i}{W_{j \neq i}} (d_{ij} q_j I_{ij} + |W_{min}|) \quad (3.19)$$

$$W_{j \neq i} = \sum_{j \neq i}^N (d_{ij} q_j I_{ij} + |W_{min}|) \quad (3.20)$$

$$W_{min} = \min_{j \neq i} d_{ij} q_j I_{ij} \quad (3.21)$$

This ensures that $\sum_{j=1}^N w_{ij} = 1$ as is required by the FJ model.

3.5 Complete model

In summary, our crowd model is described by the following state dynamics for each individual agent i with respect to every other agent j , where $i, j = 1, 2, \dots, N$, with N being the number of agents in the crowd.

3.5.1 Emotion

$$e_i[k+1] = e_i[k] + \Delta e_i[k] \quad (3.22)$$

$$\Delta e_i[k] = A_i(e_i[k] - e_i^*) + \lambda(s_i, e_{j \neq i}) \quad (3.23)$$

$$\lambda(s_i, e_{j \neq i}) = \frac{1}{N-1} \sum_{j \neq i}^N d_{ij} e_j q_j s_i h_{ij} \quad (3.24)$$

$$h_{ij} = (1 - ReLU[-sgn(e_{A_j} e_{D_j}) - sgn(o_i^T o_j)]) . \quad (3.25)$$

3.5.2 Movement

$$x_i[k+1] = x_i[k] + \Delta x_i[k] \quad (3.26)$$

$$\Delta x_i[k] = \overrightarrow{Diffusion}_i + \overrightarrow{Emotion}_i \quad (3.27)$$

$$\overrightarrow{Diffusion}_i = \frac{\rho}{N-1} \sum_{j \neq i}^N d_{ij} (x_i - x_j) \quad (3.28)$$

$$\overrightarrow{Emotion}_i = \frac{\chi}{N-1} \sum_{j \neq i}^N d_{ij} (x_j - x_i) \|e_j\| q_j h_{ij} \quad (3.29)$$

3.5.3 Opinion

$$o_i[k+1] = \nu \omega_i \sum_{j=1}^N w_{ij}[k] o_j[k] + (1 - \nu \omega_i) u \quad (3.30)$$

$$w_{ij} = \frac{\omega_i}{W_{j \neq i}} (d_{ij} q_j I_{ij} + |W_{min}|) \quad (3.31)$$

$$I_{ij} = a \frac{p_i^T p_j}{\|p_i^T p_j\|} + b \frac{e_i^T e_j}{\|e_i^T e_j\|} + c \frac{o_i^T o_j}{\|o_i^T o_j\|} \quad (3.32)$$

$$W_{j \neq i} = \sum_{j \neq i}^N (d_{ij} q_j I_{ij} + |W_{min}|) \quad (3.33)$$

$$W_{min} = \min_{j \neq i} d_{ij} q_j I_{ij} \quad (3.34)$$

3.6 Crowd Simulation Results

To ensure that our model exhibits believable crowd behaviors, we simulate our crowd model as described in Section 3.5. In figures 3.1 and 3.2 we show two scenarios with the crowd's location, opinion, and emotion states plotted at ten-step intervals. The purpose of these simulations is to show the ability for our model to exhibit both violent and non-violent behavior depending on certain parameter settings and initial conditions. In both scenarios, each agent is assigned only one opinion, where $o_i \in \mathbb{R}$. We denote each agent's opinion by the shape of their marker, where a negative and positive valued opinion is represented by a square or a circle respectively. We show the magnitude of an agent's opinion by the relative size of their marker. Each agent's location is plotted in \mathbb{R}^2 and their location at a given time step is shown by their assigned marker. Finally, to give a general idea of the agent's emotional state, we assign each agent a color according to which PAD octant the emotion falls in, as described in Table 2.1, with the appropriate colors shown in Table 3.1.

In Figure 3.1 we show a simulation where 25 agents are given a random personality p , location x , opinion o , and initial emotion e . In this scenario, we see that agents of positive valued opinions (circles) begin to gather physically and share emotional contagion of disdain amongst each other, while those with negative opinions (squares) are eventually repelled by the gathering of the positive opinion agents, causing them to disperse.

In Figure 3.2 we show a similar simulation, with random personalities p and locations x ; however, we choose that every third agent in the simulation is initialized with both an extremely hostile emotion, where $e = [P, A, D] = [-1, 1, 1]$, and an extreme opinion where

Octant	P	A	D	Octant	P	A	D
Relaxed	+	-	+	Anxious	-	+	-
Dependent	+	+	-	Disdainful	-	-	+
Exuberant	+	+	+	Bored	-	-	-
Docile	+	-	-	Hostile	-	+	+

Table 3.1: Labels for each of the PAD space octants as given by [22], with added color to text to denote the general emotion state of agents shown in Figures 3.1 and 3.2.

$o \in \{-1, 1\}$. The remaining agents given random initial emotions and opinions. We see that in this case we see the hostile emotion “infect” each crowd agent regardless of opinion, resulting in a strong gathering of hostile emotion between agents of differing opinions which can be interpreted as violence. Thus, showing our model can exhibit both violent and non-violent behavior in a believable manner.

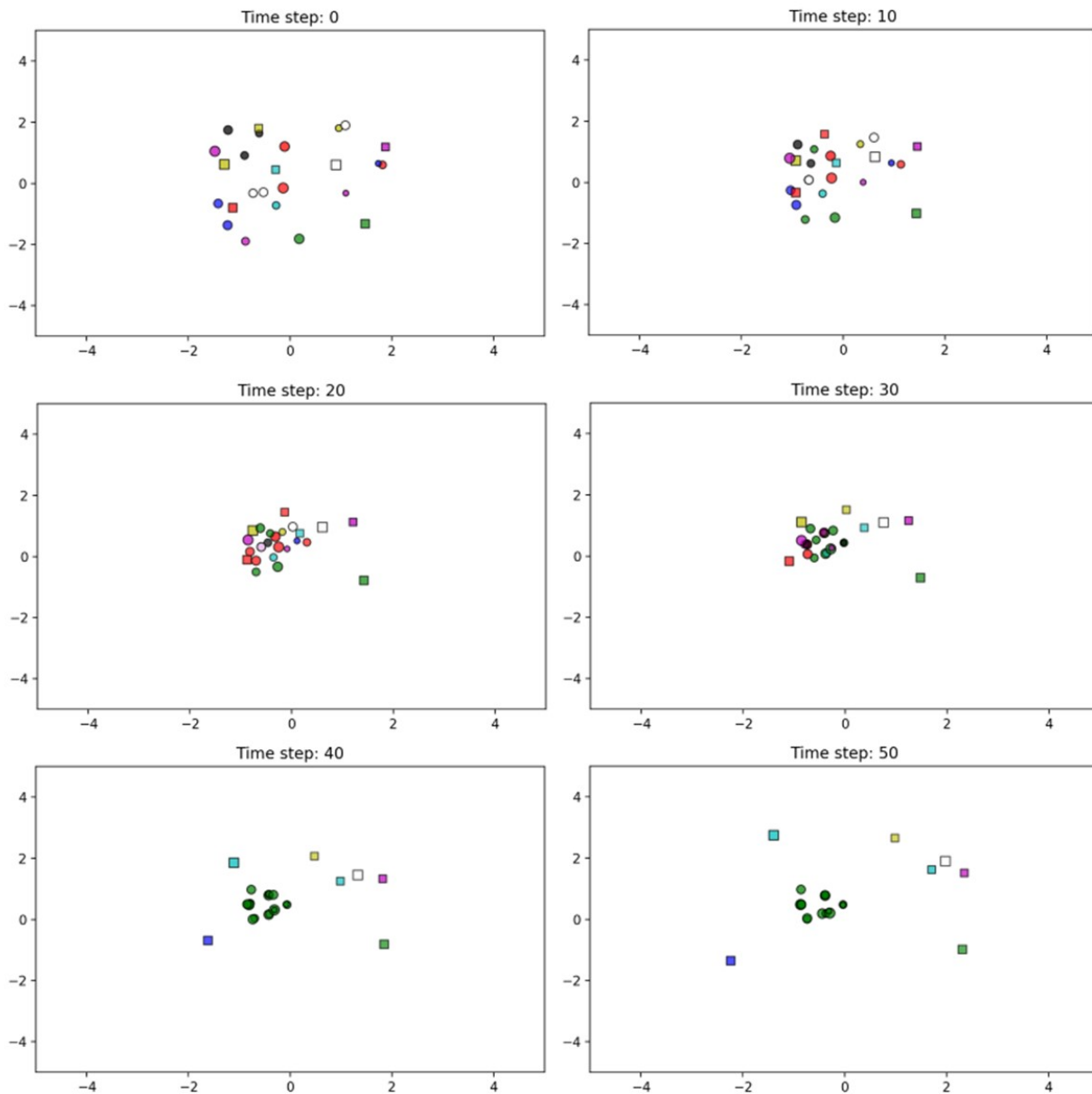


Figure 3.1: A simulation of 25 agents initialized with all random parameters and initial states. Opinion sign and magnitude is represented by shape and size respectively, position is plotted in \mathbb{R}^2 , and the emotion value represented by color as described in Table 3.1.

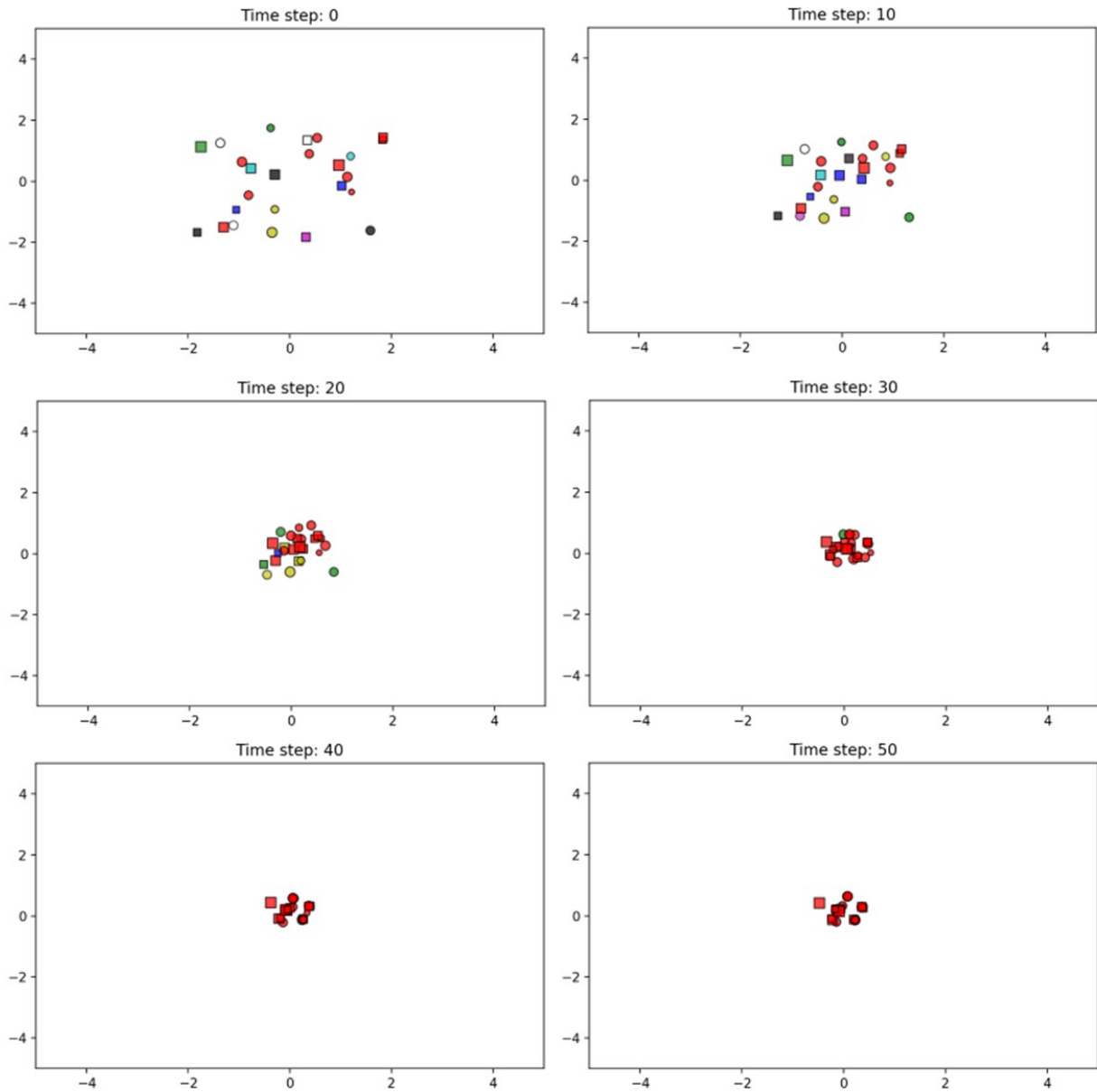


Figure 3.2: A simulation of 25 agents initialized with all random parameters and initial states, except for every third agent which is initialized with a strongly hostile emotion and strong opinion $o \in \{-1, 1\}$. Opinion sign and magnitude is represented by shape and size respectively, position is plotted in \mathbb{R}^2 , and the emotion value represented by color as described in Table 3.1.

Chapter 4

Linearized Crowd Model

As the model described in Chapter 3 is clearly non-linear, particularly in the characterization of emotion dynamics, this chapter will describe a linearized version of said model. We will then use this linearized model to solve the state estimations problem as posed in Chapter 5.

4.1 Preliminaries

In this section we review some preliminaries necessary for model linearization.

4.1.1 Jacobian Matrix

The Jacobian matrix is a key component in the linearization of non-linear systems and is defined as the matrix of first order partial derivatives of the system

$$f(x) = [f_1(x), f_2(x), \dots, f_n(x)] \quad (4.1)$$

as follows:

$$J(x) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1}(x) & \frac{\partial f_1}{\partial x_2}(x) & \dots & \frac{\partial f_1}{\partial x_n}(x) \\ \frac{\partial f_2}{\partial x_1}(x) & \frac{\partial f_2}{\partial x_2}(x) & \dots & \frac{\partial f_2}{\partial x_n}(x) \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1}(x) & \frac{\partial f_n}{\partial x_2}(x) & \dots & \frac{\partial f_n}{\partial x_n}(x) \end{bmatrix} \quad (4.2)$$

4.2 Model Linearization

While comprehensive analysis of non-linear systems is typically difficult, we can draw some understanding of the local behavior of our non-linear model by linearizing it around a valid equilibrium point. For this exercise, we assume that the position of each agent at any given time step k is known, and that the opinion o_i for each agent i is fixed for the duration of the simulation, making the opinion a parameter in our emotion dynamics equations rather than an evolving state. Thus, we focus our attention on the emotion dynamics of the crowd as defined in Chapter 3

$$e_i[k + 1] = e_i[k] + \Delta e_i[k] \quad (4.3)$$

$$\Delta e_i[k] = A_i(e_i[k] - e_i^*) + \lambda(s_i, e_{j \neq i}) \quad (4.4)$$

$$\lambda(s_i, e_{j \neq i}) = \frac{1}{N-1} \sum_{j \neq i}^N d_{ij} e_j q_j s_i (1 - ReLU[-sgn(e_{A_j} e_{D_j}) - sgn(o_i^T o_j)]) \quad (4.5)$$

However, one difficulty arises when taking the partial derivative of Equation 4.5 due to the non-differentiable functions of $ReLU(x)$ and $sgn(x)$, which is necessary for constructing the Jacobian matrix. To circumvent this, we can approximate both functions with their differentiable counterparts where

$$ReLU(x) \approx \frac{1}{\tau} \ln(1 + \exp(\tau x)), \tau \gg 1 \quad (4.6)$$

$$sgn(x) \approx \frac{2}{1 + \exp(-\tau x)} - 1, \tau \gg 1 \quad (4.7)$$

which allows us to replace the term $1 - ReLU[-sgn(e_{jA} e_{jD}) - sgn(o_i^T o_j)]$ in Equation 4.5

with

$$h_{ij} = 1 - \frac{1}{\tau} \ln \left[1 + \exp \left(\tau \left[\left(\frac{2}{1 + \exp(-\tau e_{A_j} e_{D_j})} - 1 \right) - \left(\frac{2}{1 + \exp(-\tau o_i^T o_j)} - 1 \right) \right] \right) \right] \quad (4.8)$$

In the following subsections we discuss the construction of the Jacobian for the system described in Equation 4.3 using the above substitutions, methods for finding the equilibrium of said system, and the final construction of the linearized crowd emotion dynamics model.

4.2.1 Emotion Dynamics Jacobian Matrix

To construct the Jacobian matrix for 4.3 we must first construct the complete list of nonlinear equations $f(e)$. Since Equation 4.3 actually describes the evolution of three emotional states for each agent, namely their pleasure, arousal, and dominance states, the function $f(e)$ will be a function mapping $\mathbb{R}^{3n} \rightarrow \mathbb{R}^{3n}$ where

$$f(e_i) = \begin{bmatrix} \dot{e}_{P_i} \\ \dot{e}_{A_i} \\ \dot{e}_{D_i} \end{bmatrix} = \begin{bmatrix} e_{P_i}[k] + \Delta e_{P_i}[k] \\ e_{A_i}[k] + \Delta e_{A_i}[k] \\ e_{D_i}[k] + \Delta e_{D_i}[k] \end{bmatrix} = \begin{bmatrix} e_{P_i}[k] - \alpha n_i (e_{P_i}[k] - e_{P_i}^*) + \lambda_P (s_i, e_{P_{j \neq i}}) \\ e_{A_i}[k] - \beta n_i (e_{A_i}[k] - e_{A_i}^*) + \lambda_A (s_i, e_{A_{j \neq i}}) \\ e_{D_i}[k] - \gamma n_i (e_{D_i}[k] - e_{D_i}^*) + \lambda_D (s_i, e_{D_{j \neq i}}) \end{bmatrix} \quad (4.9)$$

and $\lambda_P, \lambda_A, \lambda_D$ represent the contagion of pleasure, arousal, and dominance respectively from

every other agent j in the crowd.

$$\begin{aligned}
\lambda_P(s_i, e_{P_{j \neq i}}) &= \frac{1}{N-1} \sum_{j \neq i}^N d_{ij} e_{P_j} q_j s_i h_{ij} \\
\lambda_A(s_i, e_{A_{j \neq i}}) &= \frac{1}{N-1} \sum_{j \neq i}^N d_{ij} e_{A_j} q_j s_i h_{ij} \\
\lambda_D(s_i, e_{D_{j \neq i}}) &= \frac{1}{N-1} \sum_{j \neq i}^N d_{ij} e_{D_j} q_j s_i h_{ij}.
\end{aligned} \tag{4.10}$$

Thus, we can describe the Jacobian for the system as

$$J(e) = \begin{bmatrix} \frac{\partial \dot{e}_{P_i}}{\partial e_{P_j}} & \frac{\partial \dot{e}_{P_i}}{\partial e_{A_j}} & \frac{\partial \dot{e}_{P_i}}{\partial e_{D_j}} \\ \frac{\partial \dot{e}_{A_i}}{\partial e_{P_j}} & \frac{\partial \dot{e}_{A_i}}{\partial e_{A_j}} & \frac{\partial \dot{e}_{A_i}}{\partial e_{D_j}} \\ \frac{\partial \dot{e}_{D_i}}{\partial e_{P_j}} & \frac{\partial \dot{e}_{D_i}}{\partial e_{A_j}} & \frac{\partial \dot{e}_{D_i}}{\partial e_{D_j}} \end{bmatrix} \tag{4.11}$$

where

$$\frac{\partial \dot{e}_{P_i}}{\partial e_{P_i}} = 1 - \alpha n_i, \quad \frac{\partial \dot{e}_{A_i}}{\partial e_{A_i}} = 1 - \beta n_i, \quad \frac{\partial \dot{e}_{D_i}}{\partial e_{D_i}} = 1 - \gamma n_i \tag{4.12}$$

$$\frac{\partial \dot{e}_{P_i}}{\partial e_{A_i}} = \frac{\partial \dot{e}_{P_i}}{\partial e_{D_i}} = \frac{\partial \dot{e}_{A_i}}{\partial e_{P_i}} = \frac{\partial \dot{e}_{A_i}}{\partial e_{D_i}} = \frac{\partial \dot{e}_{D_i}}{\partial e_{P_i}} = \frac{\partial \dot{e}_{D_i}}{\partial e_{A_i}} = 0 \tag{4.13}$$

$$\frac{\partial \dot{e}_{A_i}}{\partial e_{P_j}} = \frac{\partial \dot{e}_{D_i}}{\partial e_{P_j}} = 0. \tag{4.14}$$

To simplify further partial derivative expressions we define

$$\phi_j = - \left(\frac{2}{1 + \exp(-\tau e_{A_j} e_{D_j})} - 1 \right) - \left(\frac{2}{1 + \exp(-\tau o_i^T o_j)} - 1 \right) \tag{4.15}$$

and

$$\theta_j = \frac{2\tau \exp(\tau(\phi_j - e_{A_j} e_{D_j}))}{(1 + \exp(\tau\phi_j)) (1 + \exp(-\tau e_{A_j} e_{D_j}))^2} \quad (4.16)$$

making

$$\frac{\partial \dot{e}_{P_i}}{\partial e_{P_j}} = \frac{d_{ij} q_j s_i}{N-1} \left(1 - \frac{1}{\tau} \ln(1 + \exp(\tau\phi_j)) \right), \quad j \neq i \quad (4.17)$$

$$\frac{\partial \dot{e}_{A_i}}{\partial e_{A_j}} = \frac{d_{ij} q_j s_i}{N-1} \left(1 + e_{A_j} e_{D_j} \theta_j - \frac{1}{\tau} \ln(1 + \exp(\tau\phi_j)) \right), \quad j \neq i \quad (4.18)$$

$$\frac{\partial \dot{e}_{D_i}}{\partial e_{D_j}} = \frac{\partial \dot{e}_{A_i}}{\partial e_{A_j}} \quad (4.19)$$

$$\frac{\partial \dot{e}_{P_i}}{\partial e_{A_j}} = \frac{d_{ij} q_j s_i}{N-1} e_{D_j} e_{P_j} \theta_j, \quad j \neq i \quad (4.20)$$

$$\frac{\partial \dot{e}_{P_i}}{\partial e_{D_j}} = \frac{d_{ij} q_j s_i}{N-1} e_{A_j} e_{P_j} \theta_j, \quad j \neq i \quad (4.21)$$

$$\frac{\partial \dot{e}_{A_i}}{\partial e_{D_j}} = \frac{d_{ij} q_j s_i}{N-1} e_{A_j}^2 \theta_j, \quad \frac{\partial \dot{e}_{D_i}}{\partial e_{A_j}} = \frac{d_{ij} q_j s_i}{N-1} e_{D_j}^2 \theta_j, \quad j \neq i. \quad (4.22)$$

Using this Jacobian we can now linearize our crowd emotion model around a valid equilibrium point.

4.2.2 Emotion Dynamics Equilibrium

In order to solve for the equilibrium point of 4.3, we set $e_i[k+1] = e_i[k]$ yielding $\Delta e_i[k] = 0$.

Thus

$$A_i(e_i[k] - e_i^*) + \lambda(s_i, e_{j \neq i}) = 0 \quad (4.23)$$

However, because each agent i has three emotional states, where the contagion of all three states depends on the arousal and dominance states of every other agent j , we have that solving for the root of the system

$$F(e) = \begin{bmatrix} -\alpha n_1(e_{P_1} - e_{P_1}^*) + \lambda_P(s_1, e_{P_{j \neq 1}}) \\ \vdots \\ -\alpha n_N(e_{P_N} - e_{P_N}^*) + \lambda_P(s_N, e_{P_{j \neq N}}) \\ -\beta n_1(e_{A_1} - e_{A_1}^*) + \lambda_A(s_1, e_{A_{j \neq 1}}) \\ \vdots \\ -\beta n_N(e_{A_N} - e_{A_N}^*) + \lambda_A(s_N, e_{A_{j \neq N}}) \\ -\gamma n_1(e_{D_1} - e_{D_1}^*) + \lambda_D(s_1, e_{D_{j \neq 1}}) \\ \vdots \\ -\gamma n_N(e_{D_N} - e_{D_N}^*) + \lambda_D(s_N, e_{D_{j \neq N}}) \end{bmatrix} = 0 \quad (4.24)$$

is non-trivial, even in the case of a two person crowd, due to the terms $\lambda_P, \lambda_A, \lambda_A$ which all contain the state variables e_{A_j}, e_{D_j} inside of our $ReLU(x)$ and $sgn(x)$ approximations as defined in Equations 4.6 and 4.7 respectively. We can, however, run our non-linear system until it reaches equilibrium and use this simulated equilibrium point to linearize our non-linear system.

4.2.3 Linearized Model

Now that we have an expression for the Jacobian of our system and method for numerically approximating the equilibrium point, we can construct a linearized model. Let e_{eq} be the equilibrium point of our non-linear crowd emotion dynamics where

$$e_{eq} = [e_{P_1}, \dots, e_{P_N}, e_{A_1}, \dots, e_{A_N}, e_{D_1}, \dots, e_{D_N}]^T \quad (4.25)$$

then, our linear model becomes

$$\tilde{e}[k + 1] = Q\tilde{e}[k] \quad (4.26)$$

where

$$Q = J(e_{eq}) \quad (4.27)$$

and

$$\tilde{e}[k] = e[k] - e_{eq}. \quad (4.28)$$

It should be noted that our linearized model will have a shifted coordinate system where the new equilibrium is now centered at the origin. Using this model we can now build a state observer, which will be discussed in Chapter 5.

Chapter 5

State Estimation

In this chapter we propose a solution to the state estimation problem for the model formulated in Chapter 3 using its linearized form as constructed in Chapter 4. We begin by reviewing the fundamentals of state estimation and methods for building a linear state observer. We then build a state observer from the linearized crowd emotion model and evaluate its effectiveness at state prediction on both our linearized and non-linear model.

5.1 Building a State Observer

In control theory, a system is considered observable if its internal states can be inferred from knowledge of the system's outputs. Let our system take the form

$$\begin{aligned}\varepsilon[k + 1] &= Q\varepsilon[k] \\ y[k] &= C\varepsilon[k]\end{aligned}\tag{5.1}$$

where $\varepsilon \in \mathbb{R}^{3N}$ is our emotion state vector where N is the number of agents in the crowd, $Q \in \mathbb{R}^{3N \times 3N}$ is the linear state evolution matrix derived from the Jacobian of our non-linear model as defined in Chapter 4, $y \in \mathbb{R}^{3M}$ is our system output vector where M is the number of crowd sensors, and $C \in \mathbb{R}^{3M \times 3N}$ is the state output matrix. We define the state output

matrix C for our model as follows

$$C = \begin{bmatrix} c_{11} & \dots & c_{N1} & 0 & \dots & 0 & 0 & \dots & 0 \\ 0 & \dots & 0 & c_{11} & \dots & c_{N1} & 0 & \dots & 0 \\ 0 & \dots & 0 & 0 & \dots & 0 & c_{11} & \dots & c_{N1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ c_{1M} & \dots & c_{NM} & 0 & \dots & 0 & 0 & \dots & 0 \\ 0 & \dots & 0 & c_{1M} & \dots & c_{NM} & 0 & \dots & 0 \\ 0 & \dots & 0 & 0 & \dots & 0 & c_{1M} & \dots & c_{NM} \end{bmatrix} \quad (5.2)$$

where $c_{ij} = \frac{1}{r_{ij}}$ and $r_{ij} \in \mathbb{R} > 0$ is the distance of crowd member i to sensor j . We use this model for sensor measurement as it is analogous to a model for measuring sound pressure levels using acoustic sensors. We choose an acoustic sensor model as recent work [8] shows that general crowd mood may be reasonably identified using acoustic measurements. This state output matrix also assumes that the crowd emotion states are ordered as

$$\varepsilon = [\varepsilon_{P_1}, \dots, \varepsilon_{P_N}, \varepsilon_{A_1}, \dots, \varepsilon_{A_N}, \varepsilon_{D_1}, \dots, \varepsilon_{D_N}]^T \quad (5.3)$$

making the measurement of crowd, pleasure, arousal, and dominance independent of each other where

$$y = [\bar{\varepsilon}_{P_1}, \bar{\varepsilon}_{A_1}, \bar{\varepsilon}_{D_1}, \dots, \bar{\varepsilon}_{P_M}, \bar{\varepsilon}_{A_M}, \bar{\varepsilon}_{D_M}]^T \quad (5.4)$$

and $\bar{\varepsilon}_{P_j}, \bar{\varepsilon}_{A_j}, \bar{\varepsilon}_{D_j}$ are the weighted sums of of pleasure, arousal, and dominance respectively for the general crowd as measured from sensor j .

For state estimation, we use a Luenberger observer model defined by

$$\begin{aligned} \hat{\varepsilon}[k+1] &= Q\hat{\varepsilon}[k] + L(y[k] - \hat{y}[k]) \\ \hat{y}[k] &= C\hat{\varepsilon}[k] \end{aligned} \quad (5.5)$$

where $\hat{\varepsilon}$ is our state estimate, \hat{y} is our measurement estimate, and $L \in \mathbb{R}^{3N \times 3M}$ is chosen such that the error $\bar{e}[k] = \varepsilon[k] - \hat{\varepsilon}[k]$ converges to zero as $k \rightarrow \infty$. By solving for $\bar{e}[k+1]$ it is easy to show that

$$\bar{e}[k+1] = \varepsilon[k+1] - \hat{\varepsilon}[k+1] = (Q - LC)\bar{e}[k] \quad (5.6)$$

proving that the system will asymptotically converge to zero error so long as the spectral radius of $Q - LC$ is less than one. We use the steady state Kalman gains matrix as calculated from the Kalman filter [72] as our choice for L in the state observer model.

As mentioned, we use our linearized crowd emotion dynamics model as defined in Chapter 4 for Q . However, since testing this state observer only requires a set of measurements y from the real system, we can build a state observer using the linearized crowd emotion model, then test its effectiveness at predicting the states given output from the non-linear crowd emotion model. To account for the shifted equilibrium of the linearized system, the outputs of our non-linear crowd model are shifted to reflect an origin centered equilibrium point. This, however, mandates the necessity for knowing the true equilibrium point of the system in order to estimate its state. We discuss the implications of such a necessity in the following section.

5.2 Equilibrium Point Shifting of Non-linear Model

As remarked in Chapter 4, our linearized crowd model will have a shifted coordinate system from our non-linear crowd model, which may have a non-zero equilibrium point. Since our state observer is built from our linearized model, we must account for this shift accordingly when making state estimations for our non-linear system. Let our non-linear model be

defined as

$$\begin{aligned}\varepsilon[k+1] &= f(\varepsilon[k]) \\ y[k] &= C\varepsilon[k]\end{aligned}\tag{5.7}$$

where $\varepsilon[k]$ is the value of the crowd emotion states at time step k , $f(\varepsilon)$ is our non-linear model, C is the same known state output matrix as defined in Section 5.1, and y is the system output. Furthermore, let the equilibrium point of this system be non-zero and defined as

$$\varepsilon^* = f(\varepsilon^*)\tag{5.8}$$

Now, let

$$\begin{aligned}\varepsilon[k] &= \varepsilon^* + \delta\varepsilon[k] \\ y[k] &= C(\varepsilon^* + \delta\varepsilon[k])\end{aligned}\tag{5.9}$$

where $\delta\varepsilon$ is some shifted ε such that it reaches equilibrium at the origin. Then

$$\varepsilon[k+1] = \varepsilon^* + \delta\varepsilon[k+1] = f(\varepsilon^* + \delta\varepsilon[k+1])\tag{5.10}$$

making our shifted system notated as

$$\begin{aligned}\delta\varepsilon[k+1] &= f(\varepsilon^* + \delta\varepsilon[k+1]) - \varepsilon^* \\ \delta y[k] &= C\delta\varepsilon[k]\end{aligned}\tag{5.11}$$

Thus, to correct for the non-zero shift in our non-linear system outputs we shift δy accordingly

$$\delta y[k] = C\delta\varepsilon[k] = y[k] - C\varepsilon^*\tag{5.12}$$

making the measurements from our non-linear system compatible with our state observer. To plot the results and compare our state estimate with values of the non-linear system states, we simply add the equilibrium shift back to the state predictions, making our state prediction error for the non-linear system

$$\bar{e}[k] = [k] - (\hat{\varepsilon}[k] + \varepsilon^*). \quad (5.13)$$

5.3 Necessity of Equilibrium Knowledge

We propose the following theorem in relation to the necessity of equilibrium knowledge for our non-linear system.

Theorem 1. *Let $f(\varepsilon) : \mathbb{R}^n \rightarrow \mathbb{R}^n$ describe the non-linear state evolution of $\varepsilon \in \mathbb{R}^n$. Furthermore, let ε^* describe a non-zero equilibrium point for $f(\varepsilon)$, given some initial state ε_0 , such that $f(\varepsilon^*) = \varepsilon^*$. If ε^* is unknown, or incorrectly assumed, then state estimation by Luenberger observer will converge to a constant non-zero error.*

Proof. For the system $f(\varepsilon) : \mathbb{R}^n \rightarrow \mathbb{R}^n$, let $\tilde{\varepsilon}^*$ be an incorrect estimate of the equilibrium, given some initial condition ε_0 , such that

$$\varepsilon[k] - \tilde{\varepsilon}^* = b \neq 0; k \rightarrow \infty. \quad (5.14)$$

Furthermore, let our non-linear system output be shifted accordingly to allow for linear state estimation

$$y[k] = C(\varepsilon[k] - \tilde{\varepsilon}^*) \quad (5.15)$$

where $y \in \mathbb{R}^m$ is the output of the system and the state output matrix $C \in \mathbb{R}^{m \times n}$ is known,

making the error in our state estimations defined as

$$\bar{e}[k] = \varepsilon[k] - \tilde{\varepsilon}^* - \hat{\varepsilon}[k] \quad (5.16)$$

where $\hat{\varepsilon}[k]$ is our state estimate at time step k . Let our discrete time Luenberger observer then be defined as

$$\begin{aligned} \hat{\varepsilon}[k+1] &= Q\hat{\varepsilon}[k] + L(y[k] - \hat{y}[k]) \\ \hat{y}[k] &= C\hat{\varepsilon}[k]. \end{aligned} \quad (5.17)$$

where $Q \in \mathbb{R}^{n \times n}$ represents the correct linearization of the system $f(\varepsilon)$ and $L \in \mathbb{R}^{n \times m}$ is chosen such that $Q - LC$ is asymptotically stable at the origin. By substituting for y and \hat{y} in the state estimation equation we have

$$\hat{\varepsilon}[k+1] = Q\hat{\varepsilon}[k] + LC(\varepsilon[k] - \tilde{\varepsilon}^* - \hat{\varepsilon}[k]) \quad (5.18)$$

and applying Equation 5.14 as the system $f(\varepsilon)$ reaches equilibrium yields

$$\hat{\varepsilon}[k+1] = Q\hat{\varepsilon}[k] + LC(b - \hat{\varepsilon}[k]). \quad (5.19)$$

To solve for the steady state dynamics of this system we set $\hat{\varepsilon}[k+1] = \hat{\varepsilon}[k]$ and solve for $\hat{\varepsilon}[k]$ which yields

$$\hat{\varepsilon}^* = (I - Q + LC)^{-1}LCb \quad (5.20)$$

where $\hat{\varepsilon}^*$ is the steady state value of our state observer. Consequently, the error of our state predictions will be driven to

$$\bar{e} = \varepsilon - \tilde{\varepsilon}^* - \hat{\varepsilon}^* = b - (I - Q + LC)^{-1}LCb \quad (5.21)$$

$$\bar{e} = (I - (I - Q + LC)^{-1}LC) b \quad (5.22)$$

as $k \rightarrow \infty$. Thus, our system will converge to a non-zero error so long as knowledge of the equilibrium is incorrect. \square

Because knowledge of the true equilibrium is required for zero error convergence, we must consider the practical application and limitations of this model class. In particular, as our model requires knowledge of the crowd personality parameters to simulate the true equilibrium point, we recognise the difficulty in obtaining such information in a live crowd scenario.

However, given these practical limitations, the purpose of this research is to explore the conditions under which crowd state estimation may be achievable. Thus, having shown that equilibrium knowledge is necessary for zero error convergence, we perform a set of simulated experiments using our model to simulate crowds and observe the error convergence of our Luenberger observer given a set of measurements y from the real system.

5.4 State Estimation Simulated Results

This section shows the results from building and testing a state observer on simulated data for a multi-agent crowd. In our results, we show state estimation for random configurations of multi-agent crowds where personalities, locations, and opinions are randomly chosen. We evaluate our state observer performance on simulated crowd data from our non-linear model and examine when high and low performance may be expected.

5.4.1 Simulation Methods

In this section we show the numerical results of testing our state observer on simulated data. To start, we choose a set of random parameter values for each agent in the crowd. We then approximate the equilibrium point for each random multi-agent system by simulating the

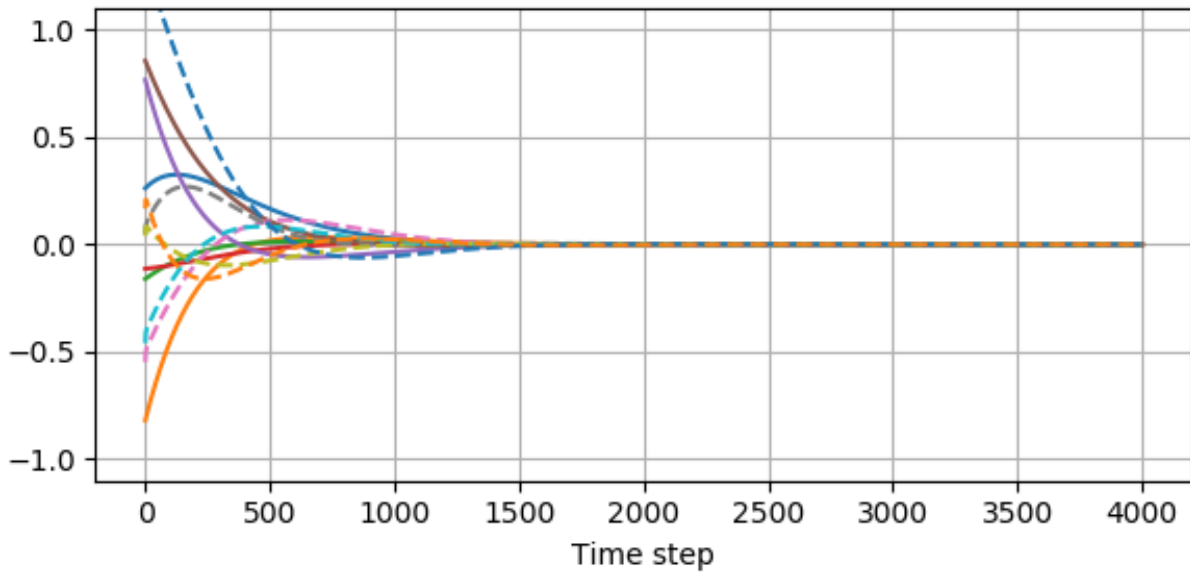
non-linear model until it reaches equilibrium $x^* = x[k]$ where

$$\|x[k-1] - x[k]\| < \epsilon \quad (5.23)$$

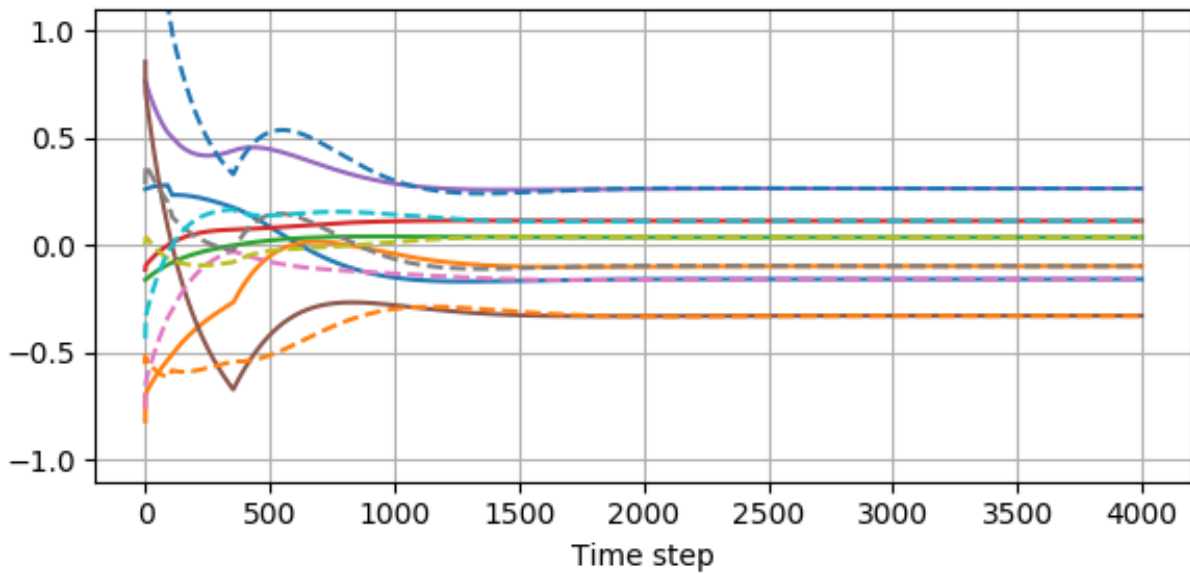
and ϵ is some tolerance of change. We then use this equilibrium point to linearize our non-linear system as described in Chapter 4. After linearization, the steady state Kalman gains K are calculated for each system using a Kalman filter. Finally, we substitute K for L in our state observer model as defined in Section 5.1 and test the convergence of our state estimator to the true state values over time starting with a random guess \hat{x}_0 for the states and given y for both linearized and non-linear systems. We discuss our findings below.

5.4.2 Comparing Linearized and Non-linear System Results

Figure 5.1 shows the convergence of our state observer to the true state values over time for our linearized system and its non-linear counterpart. Parameter values for both systems are identical as is the initial guess for the states \hat{x}_0 . As expected, our state observer performs well on the linearized system, which is characteristic of its performance on all linearized systems, and the state error approaches zero as $k \rightarrow \infty$. We also see that the same state observer performs well on the non-linear system and steadily converges to the true state of the system, as shown in 5.1(b). This, however, is not characteristic of its performance on all non-linear systems, as shown in Figure 5.3, where some parameters lead to slow or irregular convergence. We even see in 5.3(b) that some parameter settings may lead to high outlier error. We hypothesize that this error is caused by the contribution of the non-linear component (ie. the emotional contagion) being larger than the linear contribution of the emotion model and test this hypothesis in the following section. Finally, we test the robustness of our state observer to Gaussian noise, as shown in Figure 5.2. We add noise to the system measurements y and still maintain error convergence even at levels of 10% noise,

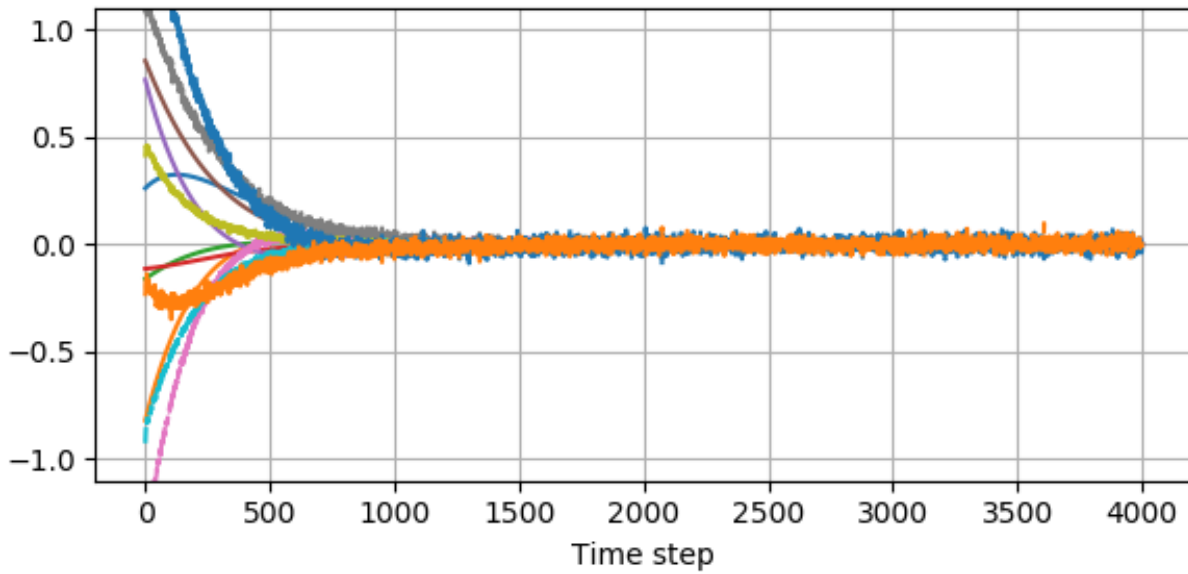


(a)

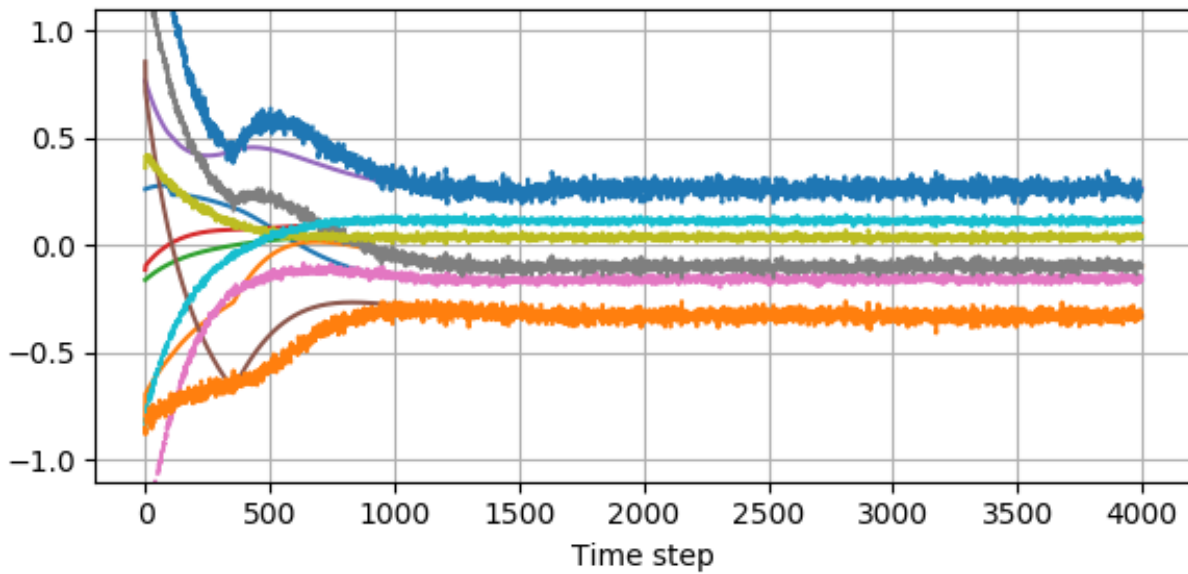


(b)

Figure 5.1: An example of emotion state predictions (dotted lines) versus actual state values (solid lines) for a linearized emotion model (a) and its non-linear version (b) over time using the same state observer for a two-agent crowd system.

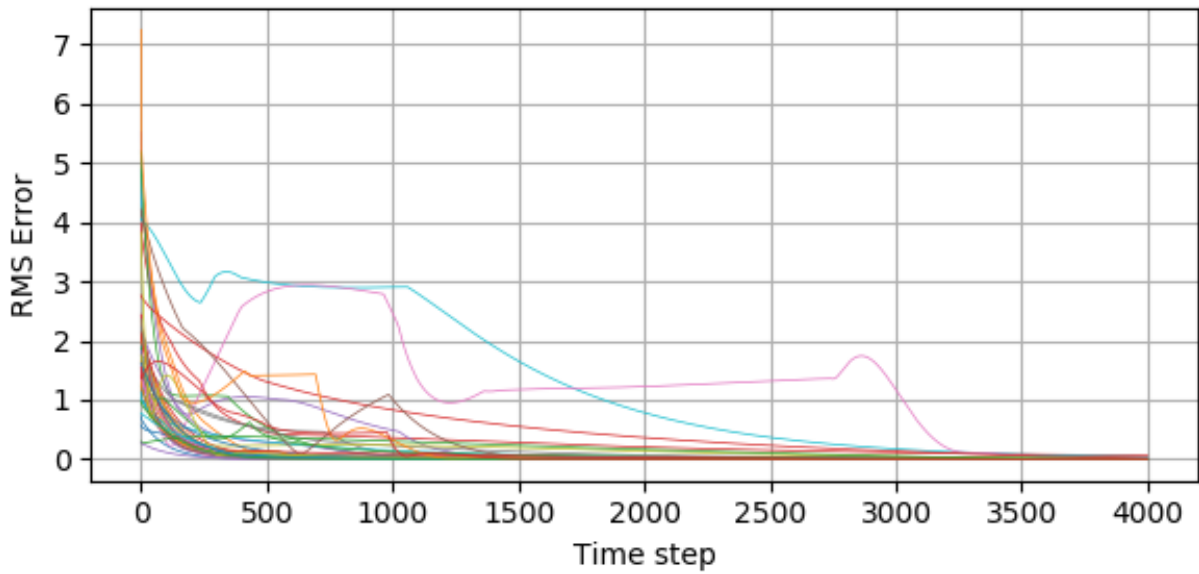


(a)

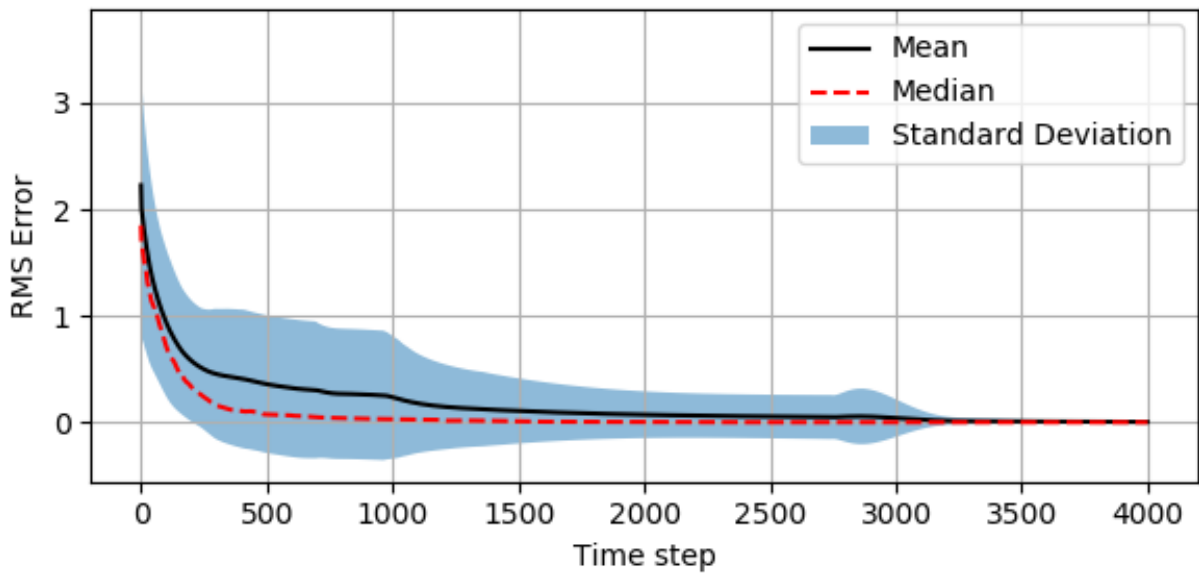


(b)

Figure 5.2: State observer predictions for the same systems shown in Figure 5.1 with 10% Gaussian noise added to system measurements. Our linear system predictions are shown in (a) and non-linear system predictions shown in (b).



(a)



(b)

Figure 5.3: The root-mean-square error of the actual versus predicted states for 50 randomly generated systems, with individual RMS error shown in (a) and general statistics shown in (b).

although the error level is increased as expected.

5.4.3 Comparing Non-linear System Predictions of Close vs. Far Agents

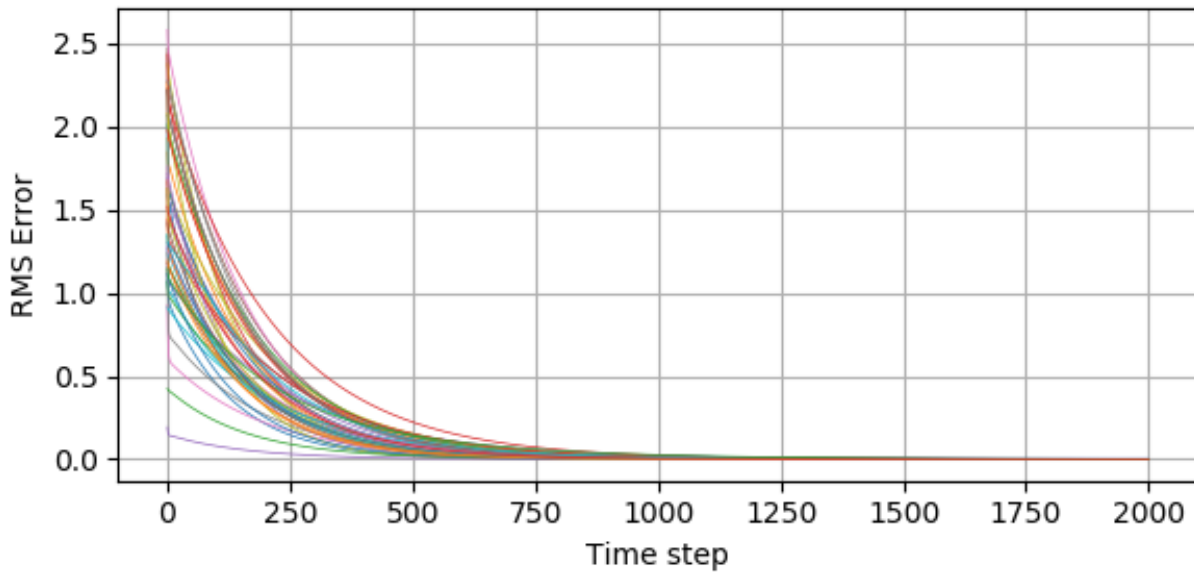
In this section we test our hypothesis that the contribution of the non-linear component of our crowd emotion model is the cause for the increase in our state prediction error depending on system parameters. We consider the non-linear emotion dynamics model as described in Chapter 3

$$e_i[k + 1] = e_i[k] + \Delta e_i[k] \quad (5.24)$$

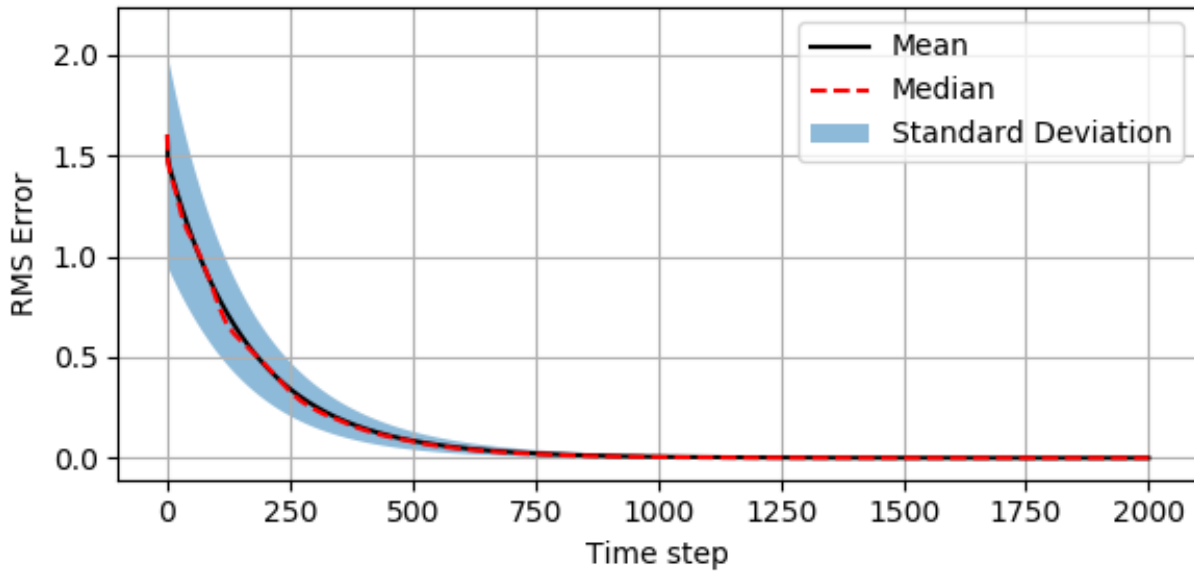
$$\Delta e_i[k] = A_i(e_i[k] - e_i^*) + \lambda(s_i, e_{j \neq i}) \quad (5.25)$$

$$\lambda(s_i, e_{j \neq i}) = \frac{1}{N-1} \sum_{j \neq i}^N d_{ij} e_j q_j s_i (1 - ReLU [-sgn(e_{jA} e_{jD}) - sgn(o_i^T o_j)]) \quad (5.26)$$

where $\Delta e_i[k]$ is comprised of two components, a linear component $A_i(e_i[k] - e_i^*)$ and a non-linear component $\lambda(s_i, e_{j \neq i})$. To test the effect that the non-linear component has on our state observer predictions, we took the same set of random initial parameters chosen for comparing predictions on linear and non-linear systems as shown in Figures 5.1-5.2 and manually set the distance of the agents from each other to be double the original distance. By increasing the distance between agents we increase the distance penalty d_{ij} on emotional contagion as calculated in Equation 5.26, allowing for the linear term to dominate. As a result, Figure 5.4 shows a more consistent convergence rate for the same two-agent systems with agents further apart from each other, confirming our hypothesis.



(a)



(b)

Figure 5.4: The root-mean-square error of the state predictions versus the actual states of the same systems from Figure 5.3 except all agent distances to each other are doubled, causing the linear component of the emotion dynamics to dominate. (a) shows the individual RMS error while (b) shows general statistics of the state prediction error.

5.4.4 State Estimation with Unknown Personalities

Despite the necessity of equilibrium knowledge for zero-error convergence, we may still consider a scenario where the true equilibrium of the crowd is known but the exact parameter values for all crowd members is unknown. In this case, instead of using the true personality parameters for crowd agents we assume incorrectly that all agent's personality scores are homogeneous at an average OCEAN personality parameter value of zero, resulting in all scaled personality parameters used in the model to be set at 0.5. We also assume that the sign of each agent's opinion is known, ie. what "team" each agent is on. We then linearize the model around the true equilibrium point of the system and examine the state estimation convergence for our approximated model.

In Figure 5.5 we show an example of the emotion states over time for a two-agent crowd and the corresponding state estimates given the linearized system Q where all scaled personality parameters are assumed to be homogeneous. Additionally, we show in Figure 5.6 the root-mean-square (RMS) error of the true state values and state estimates over time for 1,000 random simulations with the same assumption of average personality parameters given to construct Q . We can see that even when incorrectly guessing the personality parameters for the crowd, but given the true equilibrium point and sign of each agent's opinion score, we can achieve zero error convergence in our state estimates. It should also be noted that the error in the state estimation example shown in Figure 5.5 converges significantly before the system reaches equilibrium, which is characteristic for all two-agent systems tested. Repeating this experiment for a 20-agent crowd system (60 emotion states) yields similar results in error convergence.

5.4.5 Larger Crowds

We show the effectiveness of our state observer in predicting the emotional state of a larger crowd of 20 agents in Figures 5.7 and 5.8. We see that both the linearized system in Figure

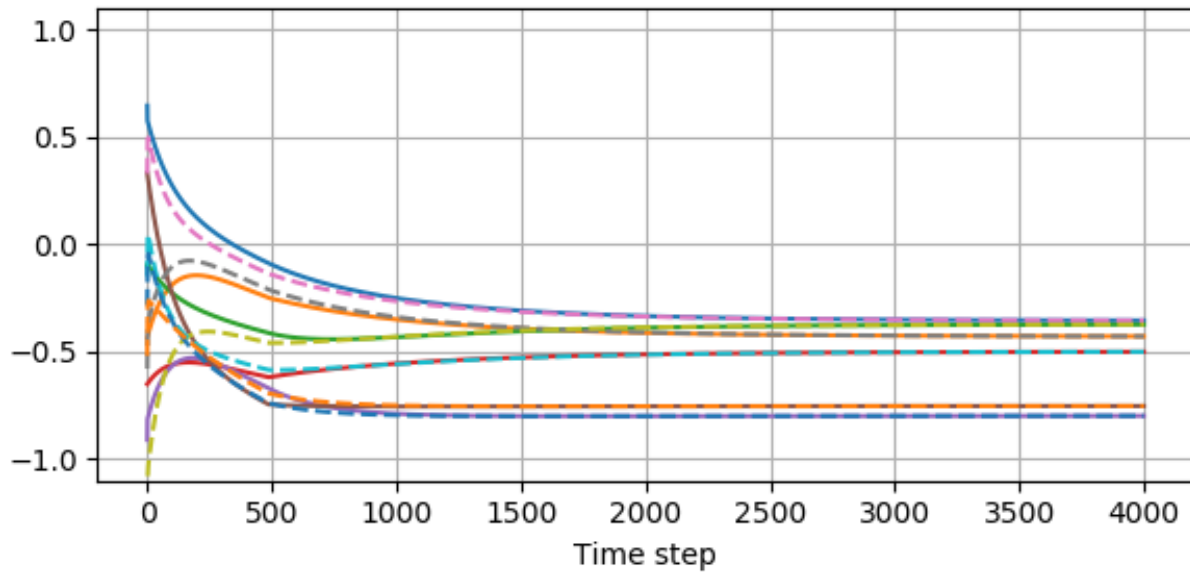


Figure 5.5: An example of emotion state predictions (dotted lines) versus actual state values (solid lines) for the nonlinear crowd emotion dynamics system for a two-agent crowd where the linearized model given to the state estimator assumes incorrectly that all personality parameter values are average.

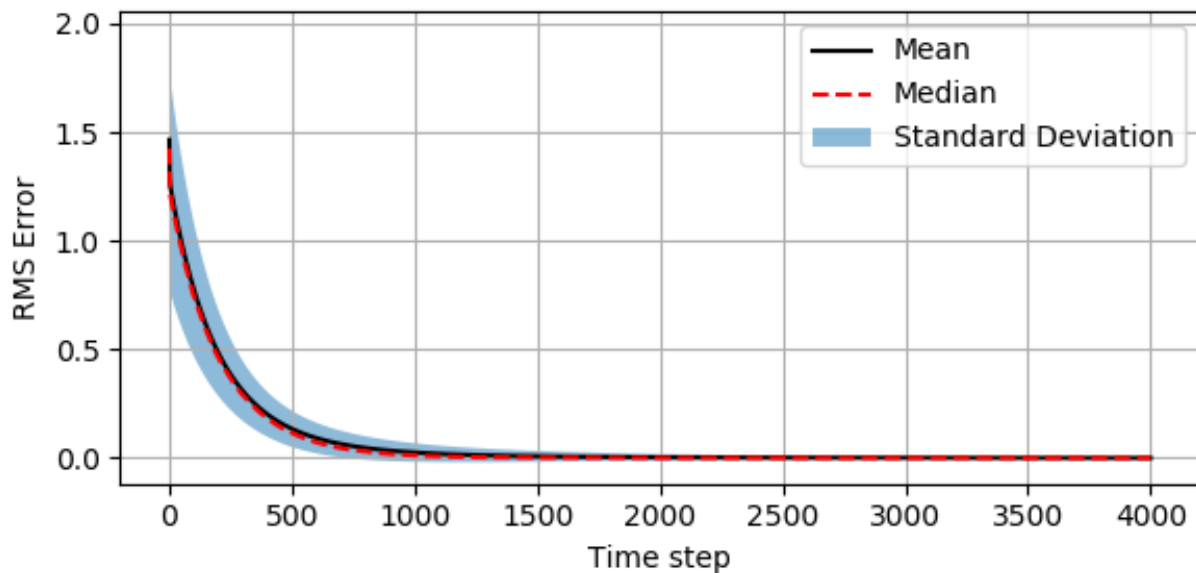


Figure 5.6: The general statistics of the root-mean-square error for the actual versus predicted states for 1,000 randomly generated two-agent systems, where the linearized model given to the state estimator assumes incorrectly that all personality parameter values are average.

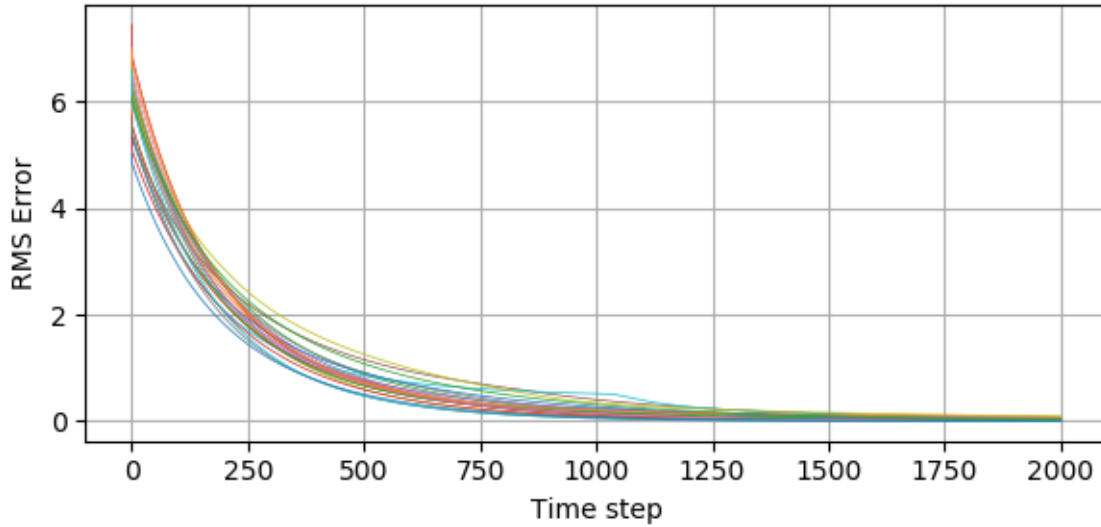
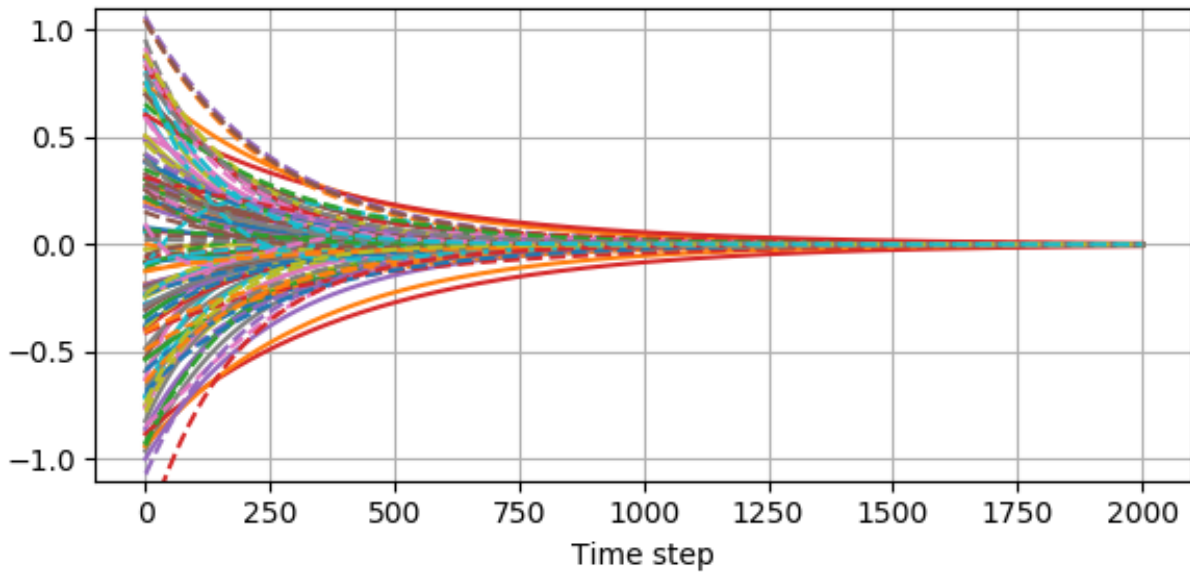


Figure 5.7: The root-mean-square error of the actual versus predicted states for 25 randomly generated 20-agent crowd systems.

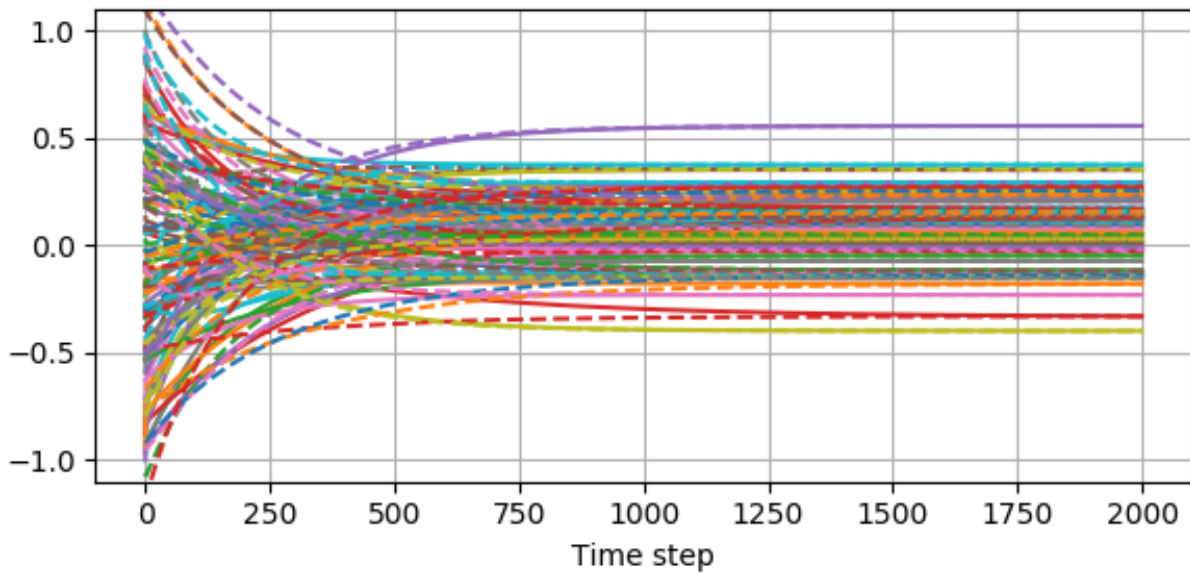
5.8(a) and the non-linear system in Figure 5.8(b) exhibit similar behavior to the two-agent crowd systems in Figure 5.1. It should be noted that as the crowd grows in size it becomes less likely for the error convergence to become irregular or show high outlier error as can happen with the two-agent system, provided that there is enough space between agents so not to amplify the non-linear terms of the emotion dynamics. This more steady convergence rate of random configurations of the 20-agent crowd is shown in Figure 5.7.

5.4.6 Edge Case: Emotion Dynamics Limit Cycle

While most crowd systems converge to a steady state equilibrium, there are some exceptional circumstances that can cause a two-agent crowd system to enter a limit cycle in its emotion dynamics. We show such an example in Figure 5.9 where the emotional states of the two-agent crowd cycle on a consistent period. This cycling can occur when two agents are close enough to each other to have the contagion term in their emotion dynamics dominate in addition to the two agents having differing opinions such that the dot product of their opinion vectors has a negative sign. Furthermore, we speculate that the initial conditions



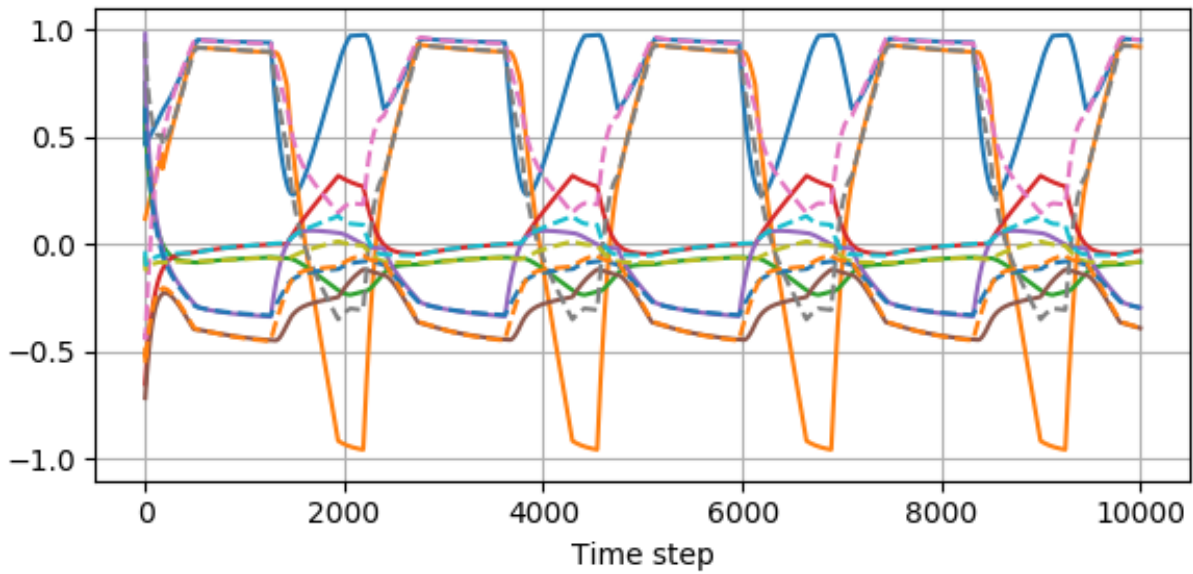
(a)



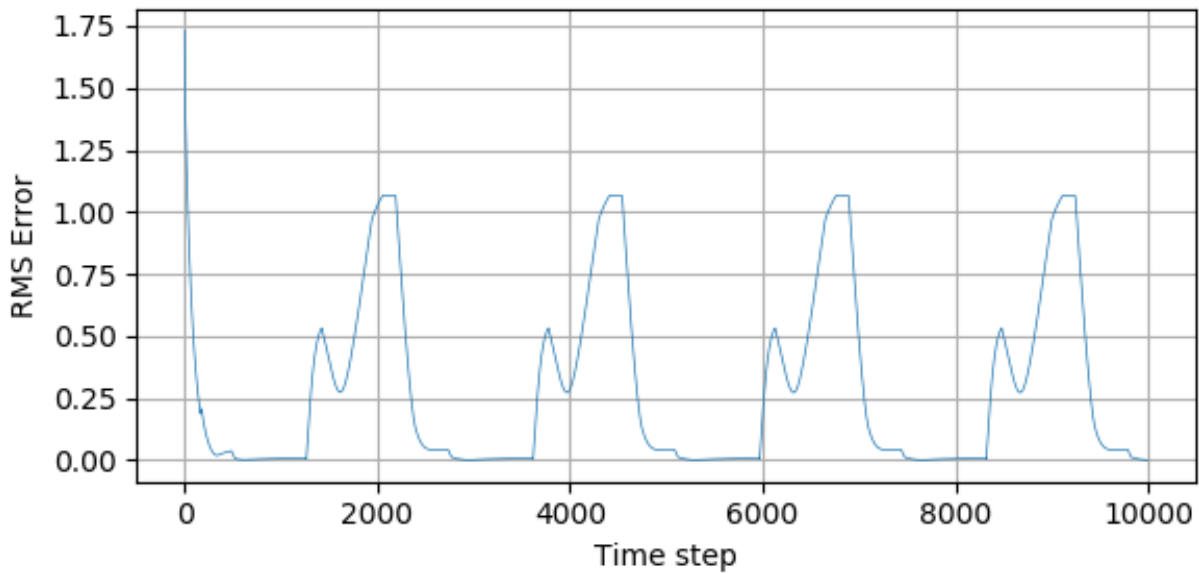
(b)

Figure 5.8: Emotion state predictions (dotted lines) versus actual state values (solid lines) for a linearized emotion model (a) and its non-linear version (b) over time using the same state observer for a 20-agent crowd system.

of their emotion state and the relationship of their personality parameters may need to start in a certain region with respect to their emotional equilibrium. It should also be noted that in our simulated non-linear model we bound the value of the emotion vector to have a magnitude of one, which is not accounted for in our model linearization, and may contribute to this emotion cycling. However, a complete study into the causes for these limit cycles is left for future work. It should be noted that in this research limit cycles were only observed in the two agent case and not in any larger crowd systems over numerous random simulations, perhaps suggesting that the conditions for limit cycles in this system may become increasingly strict as more agents are added to the system.



(a)



(b)

Figure 5.9: An example of a limit cycle in the emotion states of a two-agent crowd system. Predicted states (dotted lines) versus actual states (solid lines) are shown in (a) while the root-mean-square error between the predicted actual states is shown in (b).

Chapter 6

Conclusions and Future Work

In this thesis, we have reviewed the relevant literature pertaining to crowd behavior modeling, constructed a crowd behavior model to simulate the discrete time dynamics of emotion, movement, and opinion states of each agent, linearized that model around the simulated equilibrium point, and constructed a state observer from the linearized crowd emotion model. We discuss our conclusions and directions for future work in the following sections.

6.1 Conclusions

We have shown that knowing the value of the equilibrium point for the full nonlinear system is a necessary condition for convergence of this class of estimators, but otherwise not much information about the crowd is needed to obtain reasonable state estimates. In particular, zero-error convergence is possible even when the estimator erroneously uses nominal or average personality parameters in its model for each member of the crowd.

Of course, to simulate our system equilibrium, as done in this thesis, we require a knowledge of each agent's personality parameters and opinion vector, which we conclude is unreasonable knowledge to obtain in any realistic scenario. However, one can imagine taking crowd measurements for a sufficient amount of time for a live crowd to reach equilibrium and using this measured equilibrium in our model to drive state prediction error to zero, even with incorrect model parameterization. Unfortunately, this method would only hold for a stationary crowd, as a new equilibrium point would need to be simulated at each time step

as crowd positions may evolve over time.

These conclusions suggest a negative result for using the model class proposed in this thesis to practically predict crowd emotional state. However, because our initial motivation is to predict violence occurring in a crowd, we may still be able to find a practical solution to the state estimation of the crowd system by relaxing our detection and prediction goals. For example, if we try to instead predict when a crowd, or group of agents, may be entering an emotional *region* that characterizes violence rather than predict the exact crowd emotional state. It should also be considered that if a user of this method were to have no ethical restrictions in privacy or obtaining personal data, one may imagine a scenario where the personality parameters and opinion vectors may be inferred from personal data available through social media and other online mediums. These, and other ethical implications, should be considered in the future development of this research. We now discuss potential directions for future work building from the work performed in this thesis.

6.2 Future Work

We partition our discussion of future work into three sections: crowd modeling, state estimation, and crowd data collection.

6.2.1 Crowd Modeling

While the model constructed in Chapter 3 is built from the literature of research in relevant communities, there is more that can be done to improve model complexity and realism when testing state evolution on simulated crowds. For example, our movement model, while simple in execution, could apply a more complex flocking model that allows for agents to group together in complex ways, similar to how birds flocking or formations form in schools of fish. There is a rich body of literature, referenced in Section 2.4, that models complex movement of groups that was not implemented in this thesis. Because this thesis focuses primarily on the evolution of emotion in a crowd, there is more work to be done in carefully

modeling the interaction of movement, emotion, opinion between crowd agents and their respective dynamics.

6.2.2 State Estimation

We have shown that for many random configurations of crowd systems our posed model is observable (ie. the states can be inferred from the system outputs). However, a complete analysis on the general observability of our system has yet to be performed. An initial study into the observability of a simplified model of a two-agent crowd was conducted; however, expanding the analysis of the two-agent crowd to the N -agent crowd proved to be non-trivial and time was instead spent on showing model observability through experimental results. As such, the question of exactly when a crowd system becomes unobservable for N agents remains open.

Additionally, our state estimation results only show state evolution for a stationary crowd, where the position of each agent is known, and has not incorporated the movement component of our model. Future work may incorporated state estimation for a moving crowd, where the linearized system must be recalculated at each time step to account for changing positions. Furthermore, as addressed in our conclusion, we may attempt a more relaxed state estimation where we only attempt to detect emotional regions in the crowd rather than exact emotional state.

6.2.3 Data Collection

One of the greatest challenges in validating any crowd model is in obtaining high quality and high resolution data from real crowds. This thesis focuses on validation using simulated data, but future work may involve attempting validation on real crowd data which must be carefully collected from organic crowd events. This is no simple task and may be considered a field of research unto itself; however, to fully validate any crowd behavior model it will become necessary to have real crowd data to compare model predictions to actual crowd

behavior.

References

- [1] Hong kong: Timeline of extradition protests. *BBC*, Sep 2019.
- [2] George floyd: What happened in the final moments of his life. *BBC*, Jul 2020.
- [3] Hunt Allcott and Matthew Gentzkow. Social media and fake news in the 2016 election. *Journal of economic perspectives*, 31(2):211–36, 2017.
- [4] Pavlos Antoniou, Andreas Pitsillides, Tim Blackwell, Andries Engelbrecht, and Loizos Michael. Congestion control in wireless sensor networks based on bird flocking behavior. *Computer Networks*, 57(5):1167–1191, 2013.
- [5] Iris Bakker, Theo van der Voordt, Peter Vink, and Jan de Boon. Pleasure, arousal, dominance: Mehrabian and russell revisited. *Current Psychology*, 33(3):405–421, 2014.
- [6] Marco T Bastos and Dan Mercea. The brexit botnet and user-generated hyperpartisan news. *Social Science Computer Review*, 37(1):38–54, 2019.
- [7] Martin Burger, Marco Di Francesco, Peter A Markowich, and Marie-Therese Wolfram. On a mean field game optimal control approach modeling fast exit scenarios in human crowds. In *52nd IEEE Conference on Decision and Control*, pages 3128–3133. IEEE, 2013.
- [8] Brooks A Butler, Katrina Pedersen, Mylan R Cook, Spencer G Wadsworth, Eric Todd, Dallen Stark, Kent L Gee, Mark K Transtrum, and Sean Warnick. Classifying crowd behavior at collegiate basketball games using acoustic data. In *Proceedings of Meetings on Acoustics 176ASA*, volume 35, page 055006. Acoustical Society of America, 2018.
- [9] Carole Cadwalladr and Emma Graham-Harrison. Revealed: 50 million facebook profiles harvested for cambridge analytica in major data breach. *The guardian*, 17:22, 2018.
- [10] Sang-Hyun Cho and Hang-Bong Kang. Integrated multiple behavior models for abnormal crowd behavior detection. In *2012 IEEE Southwest Symposium on Image Analysis and Interpretation*, pages 113–116. IEEE, 2012.

- [11] Shivali Choudhary, Nitish Ojha, and Vrijendra Singh. Real-time crowd behavior detection using sift feature extraction technique in video sequences. In *2017 International Conference on Intelligent Computing and Control Systems (ICICCS)*, pages 936–940. IEEE, 2017.
- [12] Funda Durupinar, Uğur Güdükbay, Aytok Aman, and Norman I Badler. Psychological parameters for crowd simulation: From audiences to mobs. *IEEE transactions on visualization and computer graphics*, 22(9):2145–2159, 2015.
- [13] Funda Durupinar, Nuria Pelechano, Jan Allbeck, Ugur Gudukbay, and Norman I Badler. How the ocean personality model affects the perception of crowds. *IEEE Computer Graphics and Applications*, 31(3):22–31, 2009.
- [14] Paul Ekman and Wallace V Friesen. *Unmasking the face: A guide to recognizing emotions from facial clues*. Ishk, 2003.
- [15] Hajer Fradi, Bertrand Luvison, and Quoc Cuong Pham. Crowd behavior analysis using local mid-level visual descriptors. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(3):589–602, 2016.
- [16] John RP French Jr. A formal theory of social power. *Psychological review*, 63(3):181, 1956.
- [17] NE Friedkin and Eugene C Johnsen. Social influence networks and opinion change—advances in group processes. 1999.
- [18] Noah E Friedkin. *A structural theory of social influence*, volume 13. Cambridge University Press, 2006.
- [19] Noah E Friedkin and Eugene C Johnsen. Social influence and opinions. *Journal of Mathematical Sociology*, 15(3-4):193–206, 1990.
- [20] Meyer Friedman and Ray H Rosenman. Association of specific overt behavior pattern with blood and cardiovascular findings: blood cholesterol level, blood clotting time, incidence of arcus senilis, and clinical coronary artery disease. *Journal of the American medical association*, 169(12):1286–1296, 1959.
- [21] Libi Fu, Weiguo Song, Wei Lv, and Siuming Lo. Simulation of emotional contagion using modified sir model: A cellular automaton approach. *Physica A: Statistical Mechanics and its Applications*, 405:380–391, 2014.

- [22] Patrick Gebhard. Alma: a layered model of affect. In *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, pages 29–36, 2005.
- [23] Didier Grandjean, David Sander, and Klaus R Scherer. Conscious emotional experience emerges as a function of multilevel, appraisal-driven response synchronization. *Consciousness and cognition*, 17(2):484–495, 2008.
- [24] Hatice Gunes, Björn Schuller, Maja Pantic, and Roddy Cowie. Emotion representation, analysis and synthesis in continuous space: A survey. In *Face and Gesture 2011*, pages 827–834. IEEE, 2011.
- [25] Martin Haselmayer and Marcelo Jenny. Sentiment analysis of political communication: combining a dictionary approach with crowdcoding. *Quality & quantity*, 51(6):2623–2646, 2017.
- [26] Tal Hassner, Yossi Itcher, and Orit Kliper-Gross. Violent flows: Real-time detection of violent crowd behavior. In *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–6. IEEE, 2012.
- [27] Zaobo He, Zhipeng Cai, and Xiaoming Wang. Modeling propagation dynamics and developing optimized countermeasures for rumor spreading in online social networks. In *2015 IEEE 35Th international conference on distributed computing systems*, pages 205–214. IEEE, 2015.
- [28] Frank Heppner and Ulf Grenander. A stochastic nonlinear model for coordinated bird flocks. *The ubiquity of chaos*, 233:238, 1990.
- [29] Michael A Hogg and Scott Tindale. *Blackwell handbook of social psychology: Group processes*. John Wiley & Sons, 2008.
- [30] James Hoggett and Clifford Stott. Crowd psychology, public order police training and the policing of football crowds. *Policing: An International Journal of Police Strategies & Management*, 2010.
- [31] Weiming Hu, Tieniu Tan, Liang Wang, and Steve Maybank. A survey on visual surveillance of object motion and behaviors. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 34(3):334–352, 2004.
- [32] Jian-Feng Huang and Shui-Li Chen. Detection of violent crowd behavior based on statistical characteristics of the optical flow. In *2014 11th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, pages 565–569. IEEE, 2014.

- [33] Andreas Huth and Christian Wissel. The simulation of the movement of fish schools. *Journal of theoretical biology*, 156(3):365–385, 1992.
- [34] Vladimir G Ivancevic, Darryn J Reid, and Eugene V Aidman. Crowd behavior dynamics: entropic path-integral model. *Nonlinear dynamics*, 59(1-2):351, 2010.
- [35] Darrick Jolliffe and David P Farrington. Development and validation of the basic empathy scale. *Journal of adolescence*, 29(4):589–611, 2006.
- [36] AS Keçeli and AYDIN Kaya. Violent activity detection with transfer learning method. *Electronics Letters*, 53(15):1047–1048, 2017.
- [37] Marissa J. Lang. D.c. protesters dig in for the long haul after nearly a month of demonstrations. *Washington Post*, Jun 2020.
- [38] Gustave Le Bon. *The crowd: A study of the popular mind*. T. Fisher Unwin, 1897.
- [39] Teng Li, Huan Chang, Meng Wang, Bingbing Ni, Richang Hong, and Shuicheng Yan. Crowded scene analysis: A survey. *IEEE transactions on circuits and systems for video technology*, 25(3):367–386, 2014.
- [40] Gale M Lucas, Jonathan Gratch, Nikolaos Malandrakis, Evan Szablowski, Eli Fessler, and Jeffrey Nichols. Goallll!: Using sentiment in the world cup to explore theories of emotion. *Image and Vision Computing*, 65:58–65, 2017.
- [41] Pei Lv, Zhujin Zhang, Chaochao Li, Yibo Guo, Bing Zhou, and Mingliang Xu. Crowd behavior evolution with emotional contagion in political rallies. *IEEE Transactions on Computational Social Systems*, 6(2):377–386, 2018.
- [42] M Cristina Marchetti, Jean-François Joanny, Sriram Ramaswamy, Tanniemola B Liverpool, Jacques Prost, Madan Rao, and R Aditi Simha. Hydrodynamics of soft active matter. *Reviews of Modern Physics*, 85(3):1143, 2013.
- [43] Sandra C Matz, Michal Kosinski, Gideon Nave, and David J Stillwell. Psychological targeting as an effective approach to digital mass persuasion. *Proceedings of the national academy of sciences*, 114(48):12714–12719, 2017.
- [44] Terry McAuliffe. The inside story of charlottesville—and how the violence could have been avoided. *Newsweek*, Jul 2019.

- [45] Robert R McCrae and Paul T Costa. Validation of the five-factor model of personality across instruments and observers. *Journal of personality and social psychology*, 52(1):81, 1987.
- [46] Albert Mehrabian. Analysis of the big-five personality factors in terms of the pad temperament model. *Australian journal of Psychology*, 48(2):86–92, 1996.
- [47] Albert Mehrabian. Pleasure-arousal-dominance: A general framework for describing and measuring individual differences in temperament. *Current Psychology*, 14(4):261–292, 1996.
- [48] Ramin Mehran, Alexis Oyama, and Mubarak Shah. Abnormal crowd behavior detection using social force model. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 935–942. IEEE, 2009.
- [49] Sadegh Mohammadi, Hamed Kiani, Alessandro Perina, and Vittorio Murino. Violence detection in crowded scenes using substantial derivative. In *2015 12th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–6. IEEE, 2015.
- [50] Arturo Montejo-Raez, Manuel Carlos Díaz-Galiano, Fernando Martinez-Santiago, and LA Ureña-López. Crowd explicit sentiment analysis. *Knowledge-Based Systems*, 69:134–139, 2014.
- [51] Hossein Mousavi, Sadegh Mohammadi, Alessandro Perina, Ryad Chellali, and Vittorio Murino. Analyzing tracklets for the detection of abnormal crowd behavior. In *2015 IEEE Winter Conference on Applications of Computer Vision*, pages 148–155. IEEE, 2015.
- [52] Maurice Mulvenna, Marian Norwood, and Alex Büchner. Data-driven marketing. *Electronic Markets*, 8(3):32–35, 1998.
- [53] Isabel Briggs Myers. *The myers-briggs type indicator: Manual (1962)*. 1962.
- [54] Margaret M Nauta. The development, evolution, and status of holland’s theory of vocational personalities: Reflections and future directions for counseling psychology. *Journal of counseling psychology*, 57(1):11, 2010.
- [55] Reza Olfati-Saber. Flocking for multi-agent dynamic systems: Algorithms and theory. *IEEE Transactions on automatic control*, 51(3):401–420, 2006.

- [56] Andrew Ortony, Gerald L Clore, and Allan Collins. *The cognitive structure of emotions*. Cambridge university press, 1990.
- [57] Geetha Palanisamy and TT Manikandan. Group behaviour profiling for detection of anomaly in crowd. In *2017 International Conference on Technical Advancements in Computers and Communications (ICTACC)*, pages 11–15. IEEE, 2017.
- [58] Philip E Paré, Carolyn L Beck, and Angelia Nedić. Epidemic processes over time-varying networks. *IEEE Transactions on Control of Network Systems*, 5(3):1322–1334, 2017.
- [59] Philip E Paré, Ji Liu, Carolyn L Beck, and Tamer Başar. A coupled bi-virus spread model in networked systems. In *2018 Annual American Control Conference (ACC)*, pages 4414–4419. IEEE, 2018.
- [60] Philip E Paré, Ji Liu, Carolyn L Beck, Angelia Nedić, and Tamer Başar. Multi-competitive viruses over static and time-varying networks. In *2017 American Control Conference (ACC)*, pages 1685–1690. IEEE, 2017.
- [61] Nathaniel Persily. The 2016 us election: Can democracy survive the internet? *Journal of democracy*, 28(2):63–76, 2017.
- [62] Lars Y Pomara, Robert J Cooper, and Lisa J Petit. Modeling the flocking propensity of passerine birds in two neotropical habitats. *Oecologia*, 153(1):121–133, 2007.
- [63] Anton V Proskurnikov, Alexey S Matveev, and Ming Cao. Opinion dynamics in social networks with hostile camps: Consensus vs. polarization. *IEEE Transactions on Automatic Control*, 61(6):1524–1536, 2015.
- [64] Anton V Proskurnikov and Roberto Tempo. A tutorial on modeling and analysis of dynamic social networks. part i. *Annual Reviews in Control*, 43:65–79, 2017.
- [65] Anton V Proskurnikov and Roberto Tempo. A tutorial on modeling and analysis of dynamic social networks. part ii. *Annual Reviews in Control*, 45:166–190, 2018.
- [66] Wei Qin, Baotong Cui, and Xuyang Lou. Feedback control design of crowd evacuation system based on the diffusion model. In *2017 29th Chinese Control And Decision Conference (CCDC)*, pages 2481–2486. IEEE, 2017.
- [67] Stephen Reicher, Clifford Stott, Patrick Cronin, and Otto Adang. An integrated approach to crowd psychology and public order policing. *Policing: an international journal of police strategies & management*, 2004.

- [68] Wei-Ya Ren, Guo-Hui Li, Jun Chen, and Hao-Zhe Liang. Abnormal crowd behavior detection using behavior entropy model. In *2012 International Conference on Wavelet Analysis and Pattern Recognition*, pages 212–221. IEEE, 2012.
- [69] Klaus R Scherer, Angela Schorr, and Tom Johnstone. *Appraisal processes in emotion: Theory, methods, research*. Oxford University Press, 2001.
- [70] Tobias Senst, Volker Eiselein, Alexander Kuhn, and Thomas Sikora. Crowd violence detection using global motion-compensated lagrangian features and scale-sensitive video-level representation. *IEEE transactions on information forensics and security*, 12(12):2945–2956, 2017.
- [71] Jonathan D Sime. Crowd psychology and engineering. *Safety science*, 21(1):1–14, 1995.
- [72] Dan Simon. Kalman filtering with state constraints: a survey of linear and nonlinear algorithms. *IET Control Theory & Applications*, 4(8):1303–1318, 2010.
- [73] Wei Song, Dongliang Zhang, Xiaobing Zhao, Jing Yu, Rui Zheng, and Antai Wang. A novel violent video detection scheme based on modified 3d convolutional neural networks. *IEEE Access*, 7:39172–39179, 2019.
- [74] Clifford Stott, Otto Adang, Andrew Livingstone, and Martina Schreiber. Tackling football hooliganism: A quantitative study of public order, policing and crowd psychology. *Psychology, public policy, and law*, 14(2):115, 2008.
- [75] Min Sun, Dongping Zhang, Leyi Qian, and Ye Shen. Crowd abnormal behavior detection based on label distribution learning. In *2015 8th International Conference on Intelligent Computation Technology and Automation (ICICTA)*, pages 345–348. IEEE, 2015.
- [76] Go Tanaka and Akira Kojima. A modeling of crowd behavior based on model predictive control. In *2006 SICE-ICASE International Joint Conference*, pages 5120–5124. IEEE, 2006.
- [77] Derrick B. Taylor. George floyd protests: A timeline. *New York Times*, Jul 2020.
- [78] Jiang Tianhe and Shangguan Ziheng. Optimal control on crowd evacuation with leader-follower model. In *2018 11th International Symposium on Computational Intelligence and Design (ISCID)*, volume 2, pages 236–239. IEEE, 2018.
- [79] Gábor Vásárhelyi, Csaba Virágh, Gergő Somorjai, Tamás Nepusz, Agoston E Eiben, and Tamás Vicsek. Optimized flocking of autonomous drones in confined environments. *Science Robotics*, 3(20), 2018.

- [80] Sabiha Amin Wadoo. Sliding mode control of crowd dynamics. *IEEE Transactions on Control Systems Technology*, 21(3):1008–1015, 2012.
- [81] Liang Wu and Huan Liu. Tracing fake-news footprints: Characterizing social media messages by how they propagate. In *Proceedings of the eleventh ACM international conference on Web Search and Data Mining*, pages 637–645, 2018.
- [82] Weiguo Xia, Ming Cao, and Karl Henrik Johansson. Structural balance and opinion separation in trust–mistrust social networks. *IEEE Transactions on Control of Network Systems*, 3(1):46–56, 2015.
- [83] Fanchao Xu, Yunbo Rao, and Qifei Wang. An unsupervised abnormal crowd behavior detection algorithm. In *2017 International Conference on Security, Pattern Analysis, and Cybernetics (SPAC)*, pages 219–223. IEEE, 2017.
- [84] Mingliang Xu, Xiaozheng Xie, Pei Lv, Jianwei Niu, Hua Wang, Chaochao Li, Ruijie Zhu, Zhigang Deng, and Bing Zhou. Crowd behavior simulation with emotional contagion in unexpected multihazard situations. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2019.
- [85] Rongyong Zhao, Qianshan Hu, Cuiling Li, Daheng Dong, Qiong Liu, Yunlong Ma, and Qin Zhang. Macroscopic crowd panic quantification model of crowd evacuation based on information entropy. In *2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, pages 388–392. IEEE, 2019.

Appendices

Appendix A

Python Code

A.1 CrowdEmotionFunctions.py

```
import numpy as np
import matplotlib.pyplot as plt

alpha, beta, gamma = .01, .01, .01
class CrowdMember:
    def __init__(self, personality, PAD_init, opinion_init, loc_init, agent_id):
        self.personality = personality
        self.empathy = (0.354*personality[0] +
                        0.177*personality[1] +
                        0.135*personality[2] +
                        0.312*personality[3] +
                        0.021*personality[4])

        self.s = (self.empathy+1)/2
        self.q = (personality[2,0]+1)/2
        self.omega = (personality[0,0]+1)/2
        self.n = (-personality[4,0]+1)/2

        self.PAD = PAD_init
        self.opinion = opinion_init
        self.u = opinion_init
        self.loc = loc_init

    B = np.array([[0,0, .21, .59, -.19],
                  [.15,0,0, .3, .57],
                  [.25, .17, .6, -.32,0]])
```



```

self.restingPAD = B@personality

self.alpha, self.beta, self.gamma = 0.01, 0.01, 0.01

self.restingDynamics = np.array([[ -alpha*self.n, 0, 0],
                                   [0, -beta*self.n, 0],
                                   [0, 0, -gamma*self.n]])

self.id = agent_id

class Crowd:
    def __init__(self, num_agents, personalities, PADs, opinions, locs):
        self.p = personalities
        self.PADs = PADs
        self.o = opinions
        self.locs = locs
        self.num_agents = num_agents

        self.agents = [CrowdMember(personalities[i],
                                    PADs[i],
                                    opinions[i],
                                    locs[i], i) for i in range(num_agents)]

    def update(self, PADs, o, locs):
        self.PADs = PADs
        self.o = o
        self.locs = locs
        self.agents = [CrowdMember(self.p[i],
                                    PADs[i],
                                    o[i],
                                    locs[i], i) for i in range(self.num_agents)]

def ReLU(x):
    return np.maximum(x,0)

```

```

def Sigmoid(x):
    return 1/(1+np.exp(-x))

def distance_penalty(agent_i, agent_j):
    eta = 1
    d_ij = np.exp(-eta*np.linalg.norm(agent_i.loc - agent_j.loc))
    return d_ij

def e_dot(agent_id, crowd):
    agent_i = crowd.agents[agent_id]
    A = agent_i.restingDynamics
    e = agent_i.PAD
    e0 = agent_i.restingPAD
    s_i = agent_i.s
    o_i = agent_i.opinion

    l = np.zeros((3,1))
    for agent_j in crowd.agents:
        if agent_j.id != agent_id:
            d_ij = distance_penalty(agent_i, agent_j)
            e_j = agent_j.PAD
            q_j = agent_j.q
            o_j = agent_j.opinion
            l += (d_ij*e_j*q_j*s_i*
                (1-ReLU(-np.sign(e_j[1]*e_j[2])-np.sign(o_i.T@o_j))))

    l = l/(crowd.num_agents-1)

    return A@(e-e0)+l

def x_dot(agent_id, crowd):
    rho = 1
    chi = 1
    agent_i = crowd.agents[agent_id]
    o_i = agent_i.opinion

```

```

diffusion, emotion = np.zeros(2), np.zeros(2)
for agent_j in crowd.agents:
    if agent_j.id != agent_id:
        d_ij = distance_penalty(agent_i, agent_j)
        e_j = agent_j.PAD
        e_mag = np.sqrt(e_j[0]**2+e_j[1]**2+e_j[2]**2)
        q_j = agent_j.q
        o_j = agent_j.opinion

        if(np.linalg.norm(agent_i.loc-agent_j.loc) > .5):
            move_e = 1
        else:
            move_e = 0

        if(np.linalg.norm(agent_i.loc-agent_j.loc) < .5):
            move_d = 1
        else:
            move_d = 0

        diffusion += move_d*d_ij*(agent_i.loc-agent_j.loc)
        emotion += move_e*(d_ij*e_mag*q_j*(agent_j.loc-agent_i.loc)*
            (1-ReLU(-np.sign(e_j[1]*e_j[2])
            -np.sign(o_i.T@o_j))))).flatten()

diffusion = np.tanh(rho*diffusion/(crowd.num_agents-1))
emotion = np.tanh(chi*emotion/(crowd.num_agents-1))

return diffusion+emotion

def norm(x,y):
    dot = x.T@y
    n = np.linalg.norm(x)*np.linalg.norm(y)
    return dot/n

def o_next(agent_id, crowd):
    nu = 0

```

```

a, b, c = 1, 1, 1

agent_i = crowd.agents[agent_id]
omega_i = agent_i.omega
u = agent_i.u
I = np.zeros(crowd.num_agents)
W_min = 10

for j, agent_j in enumerate(crowd.agents):
    if agent_j.id != agent_id:
        d_ij = distance_penalty(agent_i, agent_j)
        q_j = agent_j.q

        I[j] = (a*norm(agent_i.personality, agent_j.personality) +
                b*norm(agent_i.PAD, agent_j.PAD) +
                c*norm(agent_i.opinion, agent_j.opinion))

        if d_ij*q_j*I[j] < W_min:
            W_min = d_ij*q_j*I[j]

W = 0
for j, agent_j in enumerate(crowd.agents):
    if agent_j.id != agent_id:
        d_ij = distance_penalty(agent_i, agent_j)
        q_j = agent_j.q
        W += d_ij*q_j*I[j] + np.abs(W_min)

w = np.zeros(crowd.num_agents)
for j, agent_j in enumerate(crowd.agents):
    if agent_j.id != agent_id:
        d_ij = distance_penalty(agent_i, agent_j)
        q_j = agent_j.q
        w[j] = omega_i/W*(d_ij*q_j*I[j] + np.abs(W_min))
    else:
        w[j] = 1-omega_i

```

```

inf_others = 0
for i in range(crowd.num_agents):
    inf_others += w[i]*crowd.agents[i].opinion
inf_others = inf_others*nu*omega_i

inf_self = (1-nu*omega_i)*u

return inf_others + inf_self

def get_color(PAD):
    if((PAD[0] < 0) and (PAD[1] > 0) and (PAD[2] > 0)): # Hostile
        return 'r'
    elif((PAD[0] < 0) and (PAD[1] < 0) and (PAD[2] < 0)): # Bored
        return 'k'
    elif((PAD[0] < 0) and (PAD[1] < 0) and (PAD[2] > 0)): # Disstainful
        return 'g'
    elif((PAD[0] < 0) and (PAD[1] > 0) and (PAD[2] < 0)): # Anxious
        return 'b'
    elif((PAD[0] > 0) and (PAD[1] < 0) and (PAD[2] < 0)): # Docile
        return 'w'
    elif((PAD[0] > 0) and (PAD[1] > 0) and (PAD[2] > 0)): # Exuberant
        return 'y'
    elif((PAD[0] > 0) and (PAD[1] > 0) and (PAD[2] < 0)): # Dependant
        return 'c'
    elif((PAD[0] > 0) and (PAD[1] < 0) and (PAD[2] > 0)): # Relaxed
        return 'm'

def get_shape(o):
    if o[0] > 0:
        return 'o'
    else:
        return 's'

def simplified_model(crowd):
    N = crowd.num_agents
    A = np.zeros((N,N))

```

```

for i, agent_i in enumerate(crowd.agents):
    for j, agent_j in enumerate(crowd.agents):
        if i == j:
            A[i,j] = 1-agent_i.n
        else:
            d_ij = distance_penalty(agent_i, agent_j)
            q_j = agent_j.q
            s_i = agent_i.s
            A[i,j] = 1/(N-1)*d_ij*q_j*s_i
return A

def init_crowd(num_agents, area):
    p_init = []
    PADs_init = []
    o_init = []
    loc_init = []

    for i in range(num_agents):
        p_init.append(np.random.normal(loc=0.0,scale=.3,size=(5,1)))
        PADs_init.append(np.random.uniform(-1,1,(3,1)))
        p_init.append(np.random.normal(loc=0.0,scale=.3,size=(5,1)))
        o_init.append(np.random.uniform(-1,1,(1,1)))

        loc_init.append(np.random.uniform(-area,area,2))

    crowd = Crowd(num_agents, p_init, PADs_init, o_init, loc_init)

    return crowd

def distance(x,y):
    return np.sqrt(np.sum((x-y)**2))

def makeC(crowd, mic_loc, dim):
    C = np.zeros((dim,len(crowd.agents)*dim))
    if dim > 1:

```

```

    for agent in crowd.agents:
        r = distance(mic_loc,agent.loc)
        C[0,agent.id] = 1/r
        C[1,agent.id+len(crowd.agents)] = 1/r
        C[2,agent.id+len(crowd.agents)*2] = 1/r
    else:
        for agent in crowd.agents:
            r = distance(mic_loc,agent.loc)
            C[0,agent.id] = 1/r
    return C

def kalman_results(f, num_steps, dim, noise_scale, num_agents, plot=False):
    xs = []
    xps = []
    ys = []
    Ks = []
    zs = []
    for _ in range(num_steps):
        z = f.H@f.x + np.random.normal(loc=0,scale=noise_scale,size=(dim,1))
        f.predict()
        f.update(z)
        xs.append(f.x)
        ys.append(f.y)
        Ks.append(f.K)
        xps.append(f.x_prior)
        zs.append(z)

    savedK = np.copy(f.K)
    savedP = np.copy(f.P)

    if plot:
        if dim == 1:
            plt.figure()
            plt.title('State predecions')
            plt.xlabel('t')
            for i in range(num_agents):

```

```

        data = []
        for x in xs:
            data.append(x[i])
        plt.plot(data)
    plt.grid()

else:
    titles = ['P', 'A', 'D']
    plt.figure(figsize=(12,4))
    for i in range(dim):
        plt.subplot(1, dim, i+1)
        plt.title(titles[i])
        for j in range(num_agents):
            data = []
            for x in xs:
                data.append(x[j+i*num_agents])
            plt.plot(data)
        plt.grid()

plt.figure()
plt.plot(np.array(xs).squeeze(2))
plt.grid()
plt.savefig('results/state_predictions.png')

plt.figure()
plt.title('Residual')
plt.xlabel('t')
plt.plot(np.array(ys).squeeze(2))
plt.grid()
plt.savefig('results/residuals.png')

plt.figure()
plt.title('Kalman gains')
plt.xlabel('t')
for i in range(num_agents):

```



```

        data=[]
        for k in Ks:
            data.append(k[i])
        plt.plot(data)
    plt.grid()
    plt.savefig('results/kalman_gains.png')
    #
    plt.figure()
    plt.title('Measurements')
    plt.plot(np.array(zs).squeeze(2))
    plt.grid()
    plt.savefig('results/measurements.png')

plt.close()

return savedK, zs

def state_observer(A, C, L, x0, y_mes):
    x_hat = x0
    xs = []
    ys = []
    res = []

    for i, y in enumerate(y_mes):
        y_hat = C*x_hat
        x_h_next = A*x_hat + L*(y-y_hat)
        xs.append(x_h_next)
        ys.append(y_hat)
        res.append(y-y_hat)

        x_hat = x_h_next

    return xs, ys, res

def s_ReLU(x, tau):
    return np.log(1+np.exp(tau*x))/tau

```

```

def s_sgn(x,tau):
    # return np.tanh(tau*x)
    return 2/(1+np.exp(-tau*x))-1

def BuildJ(crowd, x, tau, equilibrium=True):
    tau_R = 10
    N = crowd.num_agents
    es = []
    for i in range(N):
        e = np.array([x[i], x[i+N], x[i+2*N]])
        es.append(e)

    J_P, J_A, J_D = np.zeros((N,N)),np.zeros((N,N)),np.zeros((N,N))
    J_P_A, J_P_D, J_A_D, J_D_A = (np.zeros((N,N)),
                                   np.zeros((N,N)),
                                   np.zeros((N,N)),
                                   np.zeros((N,N)))

    for i, agent_i in enumerate(crowd.agents):
        for j, agent_j in enumerate(crowd.agents):
            if agent_i.id == agent_j.id:
                if equilibrium:
                    J_P[i,i], J_A[i,i], J_D[i,i] = (-agent_i.alpha*agent_i.n,
                                                       -agent_i.beta*agent_i.n,
                                                       -agent_i.gamma*agent_i.n)
                    J_P_A[i,i], J_P_D[i,i], J_A_D[i,i], J_D_A[i,i] = (0.0,
                                                                        0.0,
                                                                        0.0,
                                                                        0.0)
                else:
                    J_P[i,i], J_A[i,i], J_D[i,i] = (1-agent_i.alpha*agent_i.n,
                                                       1-agent_i.beta*agent_i.n,
                                                       1-agent_i.gamma*agent_i.n)
                    J_P_A[i,i], J_P_D[i,i], J_A_D[i,i], J_D_A[i,i] = (0.0,
                                                                        0.0,

```

0.0,
0.0)

```
else:
    d_ij = distance_penalty(agent_i, agent_j)
    q_j = agent_j.q
    s_i = agent_i.s
    e_j = es[j]
    o_i, o_j = agent_i.opinion, agent_j.opinion

    phi_j = -s_sgn(e_j[1]*e_j[2], tau) - s_sgn(o_i.T@o_j, tau)
    theta_j = ((2*tau*np.exp(tau_R*phi_j - tau*e_j[1]*e_j[2])) /
                ((1+np.exp(tau_R*phi_j))*(1+np.exp(-e_j[1]*e_j[2]*tau))**2))
    quan = d_ij*q_j*s_i/(N-1)

    J_P[i,j] = quan*(1-s_ReLU(phi_j, tau))
    J_A[i,j] = quan*(1+e_j[1]*e_j[2]*theta_j - s_ReLU(phi_j, tau))
    J_D[i,j] = quan*(1+e_j[1]*e_j[2]*theta_j - s_ReLU(phi_j, tau))

    J_P_A[i,j], J_P_D[i,j] = (quan*e_j[0]*e_j[2]*theta_j,
                              quan*e_j[0]*e_j[1]*theta_j)
    J_A_D[i,j], J_D_A[i,j] = (quan*e_j[1]**2*theta_j,
                              quan*e_j[2]**2*theta_j)

z_block = np.zeros((N,N))

J = np.block([[J_P, J_P_A, J_P_D],
              [z_block, J_A, J_A_D],
              [z_block, J_D_A, J_D]])

return J

def BuildF_eq_pt(crowd, x, tau):
    N = crowd.num_agents
    es = []
    for i in range(N):
```

```

e = np.array([x[i], x[i+N], x[i+2*N]])
es.append(e)

F = np.zeros((3*N,1))
for i, agent_i in enumerate(crowd.agents):
    e_star = agent_i.restingPAD

    l = np.zeros((3,1))
    for j, agent_j in enumerate(crowd.agents):
        if agent_j.id != agent_i.id:
            d_ij = distance_penalty(agent_i, agent_j)
            e_j = agent_j.PAD
            q_j = agent_j.q
            s_i = agent_i.s
            o_i = agent_i.opinion
            o_j = agent_j.opinion

            l += (d_ij*e_j*q_j*s_i*
                  (1-ReLU(-np.sign(e_j[1]*e_j[2])-np.sign(o_i.T@o_j))))

    l = l/(N-1)

    F[i] = -(agent_i.alpha*agent_i.n)*(x[i]-e_star[0])+l[0]
    F[i+N] = -(agent_i.beta*agent_i.n)*(x[i+N]-e_star[1])+l[1]
    F[i+2*N] = -(agent_i.gamma*agent_i.n)*(x[i+2*N]-e_star[2])+l[2]

return F

```

```

def newtons_method_step(crowd, x, tau):
    J = BuildJ(crowd,x,tau,equilibrium=True)
    F = BuildF_eq_pt(crowd,x,tau)
    dx = -np.linalg.inv(J)@F

```

```

x_new = x + dx

```

```
return x_new
```

A.2 model_tests.py

```
import numpy as np
import matplotlib.pyplot as plt
import tqdm

from filterpy.kalman import KalmanFilter
from CrowdEmotionFunctions import *

num_tests = 50

seeds = [i for i in range(num_tests)]

mse_l = []
mse_nl = []

e_mes_l = []
e_mes_nl = []
min_count = 0
s_used = []
noise = False
noise_scale = 1e-2
test_dist = False
mic_loc = np.array([0,0])

num_agents = 20
area = num_agents
dim = 3

num_steps = 4000
tau = 100
num_steps_k = 500
for seed in tqdm.tqdm(seeds):
```

```

np.random.seed(seed)

crowd = init_crowd(num_agents, area)

if num_agents == 2 and test_dist == True:
    d = 1.7
    locs = [np.array([0,d]), np.array([0,-d])]
    crowd.update(crowd.PADs, crowd.o, locs)

x = np.random.uniform(low=-1,high=1,size=(dim*num_agents,1))
x0 = np.array(x, dtype='longdouble')

C = makeC(crowd, mic_loc, dim)

# Generate data for non-linear model test
e0 = []
for i in range(num_agents):
    e0.append(np.array([x0[i], x0[i+num_agents], x0[i+2*num_agents]]))
crowd.update(e0, crowd.o, crowd.locs)

t = []
x_test_nl = [x0]
y_test_nl = [C*x0]
for _ in range(num_steps):
    e_dots = [e_dot(agent.id, crowd) for agent in crowd.agents]

    e_curr = crowd.PADs
    e_new = ((e_curr[i]+e_dot)/np.max([1,np.linalg.norm(e_curr[i]+e_dot)]))
            for i, e_dot in enumerate(e_dots)])

    x_next = np.zeros((num_agents*dim,1))
    for i in range(num_agents):
        x_next[i] = e_new[i][0]
        x_next[i+num_agents] = e_new[i][1]
        x_next[i+2*num_agents] = e_new[i][2]
    x_test_nl.append(x_next)

```

```

e_star = x_next
y_test_nl.append(C@x_next)

crowd.update(e_new,crowd.o, crowd.locs)
t.append(Crowd(crowd.num_agents,crowd.p,crowd.PADs,crowd.o,crowd.locs))

for i, y in enumerate(y_test_nl):
    if noise:
        y_test_nl[i] = y - C@e_star
    else:
        y_test_nl[i] = (y - C@e_star +
            np.random.normal(loc=0,scale=noise_scale,size=(C@e_star).shape))

A = BuildJ(crowd, e_star, tau, equilibrium=False)

f = KalmanFilter (dim_x=num_agents*dim, dim_z=dim)
f.x, f.F, f.H = x0, A, C
noise_scale_k = 1e-3
f.R, f.Q = noise_scale_k, noise_scale_k
f.P *= noise_scale

K, zs = kalman_results(f,num_steps_k,dim, noise_scale, num_agents)

L = K

# Generate data for linearized model test
x = x0
x_test = [x]
y_test = [C@x]
for _ in range(num_steps):
    x_next = A@x
    x_test.append(x_next)
    if noise:
        y_test.append(C@x_next +
            np.random.normal(loc=0,scale=noise_scale,size=(C@x_next).shape))

```

```

else:
    y_test.append(C@x_next)
    x = x_next

x_hat = np.random.uniform(low=-1,high=1,size=(num_agents*dim,1))

xs, ys, res = state_observer(A,C,L,x_hat,y_test)
xs_nl, ys_nl, res_nl = state_observer(A,C,L,x_hat,y_test_nl)

diff = np.array(x_test).squeeze(2)-np.array(xs).squeeze(2)
mse_l0 = np.sqrt(np.sum(diff**2, axis=1))

diff = np.array(x_test_nl).squeeze(2)-(np.array(xs_nl).squeeze(2)+e_star.T)
mse_nl0 = np.sqrt(np.sum(diff**2, axis=1))

if True:
    s_used.append(seed)
    mse_l.append(mse_l0)
    mse_nl.append(mse_nl0)

    e_mes_l.append(res)
    e_mes_nl.append(res_nl)

plt.figure(figsize=(6,3))
plt.plot(np.array(x_test_nl).squeeze(2), label='actual')
plt.plot(np.array(xs_nl).squeeze(2)+e_star.T, '--', label='observer')
plt.ylim(-1.1, 1.1)
plt.grid()
plt.xlabel('Time step')
plt.tight_layout()
plt.savefig('so_states_nonlinear'+str(seed)+'.png')
plt.close()

plt.figure(figsize=(6,3))
plt.plot(np.array(x_test).squeeze(2), label='actual')
plt.plot(np.array(xs).squeeze(2), '--', label='observer')

```



```

plt.ylim(-1.1, 1.1)
plt.grid()
plt.xlabel('Time step')
plt.tight_layout()
plt.savefig('so_states_linear'+str(seed)+'.png')

plt.close()

e_l = np.array(e_mes_l).squeeze(3)**2
e_nl = np.array(e_mes_nl).squeeze(3)**2

e_l_mean = np.sqrt(np.mean(np.mean(e_l, axis=0),axis=1))
e_l_std = np.std(np.std(e_l, axis=0),axis=1)

e_nl_mean = np.sqrt(np.mean(np.mean(e_nl, axis=0), axis=1))
e_nl_std = np.std(np.std(e_nl, axis=0), axis=1)

t = [i for i in range(num_steps+1)]
figsize = (6,3)

mse_mean = np.mean(mse_nl, axis=0)
mse_median = np.median(mse_nl, axis=0)
mse_std = np.std(mse_nl, axis=0)
plt.figure(figsize=figsize)
plt.plot(t,mse_mean, 'k-', label='Mean')
plt.plot(t,mse_median, 'r--', label='Median')
plt.fill_between(t,mse_mean-mse_std, mse_mean+mse_std,
                 alpha=0.5, label='Standard Deviation')

plt.grid()
plt.legend()
plt.xlabel('Time step')
plt.ylabel('RMS Error')
plt.tight_layout()
plt.savefig('state_observer_error_test_nl.png')

```

```

plt.figure(figsize=figsize)
plt.plot(t,np.array(mse_nl).T, linewidth=.5 )
plt.grid()
plt.xlabel('Time step')
plt.ylabel('RMS Error')
plt.tight_layout()
plt.savefig('state_observer_error_test_nl_multiline.png')

```

```

mse_mean = np.mean(mse_l, axis=0)
mse_median = np.median(mse_l, axis=0)
mse_std = np.std(mse_l, axis=0)
plt.figure(figsize=figsize)
plt.plot(t,mse_mean, 'k-', label='Mean')
plt.plot(t,mse_median, 'r--', label='Median')
plt.fill_between(t,mse_mean-mse_std, mse_mean+mse_std,
                 alpha=0.5, label='Standard Deviation')

```

```

plt.grid()
plt.legend()
plt.xlabel('Time step')
plt.ylabel('RMS Error')
plt.tight_layout()
plt.savefig('state_observer_error_test_l.png')

```

```

plt.figure(figsize=figsize)
plt.plot(t,e_nl_mean,'k-', label='Mean')
plt.fill_between(t,e_nl_mean-e_nl_std, e_nl_mean+e_nl_std,
                 alpha=0.4, label='Standard Deviation')

```

```

plt.grid()
plt.legend()
plt.xlabel('Time step')
plt.ylabel('RMS Error')
plt.tight_layout()
plt.savefig('state_observer_error_test_nl_mes.png')

```

```
plt.figure(figsize=figsize)
plt.plot(t,e_l_mean,'k-', label='Mean')
plt.fill_between(t,e_l_mean-e_l_std, e_l_mean+e_l_std,
                 alpha=0.4, label='Standard Deviation')

plt.grid()
plt.legend()
plt.xlabel('Time step')
plt.ylabel('RMS Error')
plt.tight_layout()
plt.savefig('state_observer_error_test_l_mes.png')

plt.close()
```