2020-05-15

Parameter Estimation of Cancer Cell Dynamics

Avery M. Ingegneri
Worcester Polytechnic Institute

Erika H. Sanborn
Worcester Polytechnic Institute

Jarred O. Measmer
Worcester Polytechnic Institute

John Paul Williams
Worcester Polytechnic Institute

Maxwell J. Pellerin
Worcester Polytechnic Institute

See next page for additional authors

Author

Avery M. Ingegneri, Erika H. Sanborn, Jarred O. Measmer, John Paul Williams, Maxwell J. Pellerin, Robert B. Duff, , , , , , , and

Project Number: AA1 0001

Parameter Estimation of Cancer Cell Dynamics

A Major Qualifying Project Report

Submitted to The Faculty

of

Worcester Polytechnic Institute

In partial fulfillment of the requirements for the

Degree of Bachelor of Science

by

Lynne Moore

April, 2020

Approved:

Professor Andrea Arnold

Professor Sarah Olson

Abstract

The goal of this project is to model cancer cell dynamics and their sensitivity to different cancer treatments, as well as use a variety of parameter estimation techniques to provide experimental insights. Cancer cells are highly heterogeneous in terms of their ability to proliferate and evade cell death. Our research utilizes a local lab's data of interactions between cancer cell populations and three different drug treatments. Biological data is inherently noisy due to internal mechanisms and measurement inaccuracy. This paper introduces several variations of a model developed by our team to capture the dynamics of this data system, as well as the various filtering methods utilized for parameter estimation. All variations of our model include a natural growth rate, natural or cell induced death rate, and a drug-induced death rate. These parameters are immeasurable and potentially vary with time. We utilize the statistical (Bayesian) based, Ensemble Kalman Filter for parameter estimation as it accommodates noise in data by producing state and parameter distribution estimations. Long term, estimation of model parameters may lead to insight with regards to the mechanisms behind the drugs causing cell death as well as a better understanding of data time points that are the most relevant to capture the dynamics.

Executive Summary

In a healthy biological environment, cells undergo a self-regulatory division cycle. When a cell enters the cell cycle, checkpoints need to be passed for a cell to pass on healthy genes to the next generation of cells. If the cell does not pass any of the three main checkpoints within the cycle, the cycle repeats until only healthy genes are passed on. Cancer cells, however, have a cycle mutation which overrides these checkpoints, resulting in mutated, unhealthy genes being passed on. This does not imply that cancer cell division is faster than healthy cell division. Instead, cancer cells do not re-enter the cycle if a mutation in genes is detected at a checkpoint. In contrast, a healthy division process would see a mutated cell continue through the cycle until the mutated genes are returned to a healthy status. This mutation also influences the cell cycle of the healthy cells also found within the population.

Cancer treatment research often targets the irregular cell cycle by working to regulate and restore the cycle to that of a healthy cell. Other treatments induce cell death, removing the possibility of cancer division to happen at the source. Both courses, or a combination of both, are used to restore normalcy to the greater population's cell division processes. Though there have many breakthroughs in the field, very little is still known about the dynamical interactions between healthy cells, cancer cells, and cancer treatments.

In this project, we present a coupled system of ordinary differential equations to model the dynamics of this ever evolving system and utilize parameter estimation techniques to provide researchers with experimental data collection insights. Our proposed model includes a combination of time-invariant, or constant, and dynamic time-varying parameters to capture the changing growth and death rates of cells. We use experimental data as well as knowledge about common biological models, such as exponential and logistic growth, to develop a model and generate synthetic data to run trials of the parameter estimation techniques. Specifically, we propose a model that includes parameters for natural growth rate of live cells, natural death rate, and drug induced death. We utilized the Ensemble Kalman Filtering algorithm as our main parameter estimation method, as it is easily adaptable to accommodate time-varying parameters and produces a natural measure of uncertainty for both state and parameter estimations. Our goal with this project is to highlight important population mechanisms that can be described with time-varying components, experimental time scales and frequency of data collection, and proper data collection methods for optimized estimation of model components.

When considering a model with strictly constant parameter values, the Ensemble Kalman Filter produces estimations of state and parameter values. The estimations of the varying growth and death rates give insight into the frequency and type of data collected. Subsets of our synthetic data at different equidistant and random time intervals were compared to show the importance of observations for both live and dead cancer cell counts. Our findings for necessary data collection and frequency of data collection were supported by adjusting the observation and system errors within the filter, as changing these two quantities shift the estimation of state and parameter values. For example, when limiting the live cell data observations to three over the timespan, any change in the system and observation errors resulted in the filter being able to accurately and confidently capture the state and parameter values. This same technique was used to support our final claim that data should be collected on both live and dead cell counts to obtain reasonable estimations of all necessary constant parameter values in our proposed model.

Strictly autonomous dynamic systems are rarely found in the real world, so adjustments with time-varying natural growth parameters were also explored in our research. By modifying the Ensemble

Kalman Filter to account for parameter tracking, the parameter estimation techniques were adjusted to allow for potential time-varying components. The same tests were run on models containing three potential gradually decreasing natural growth parameters. When considering these dynamic systems, the estimation technique requires more frequent data collection for live cell counts, while also maintaining the need for dead cell observations to capture all state and parameter values in the model. Though the dynamics are slow in our proposed model, the setup of the proposed simulations should provide the framework for further adjustments to the cancer cell dynamics model.

Acknowledgements

I would like to acknowledge my advisors **Professor Andrea Arnold** and **Professor Sarah Olson**, for all of their help and guidance throughout this past year. Thank you Professor Arnold for guiding me through the development of my EnKF algorithm and supporting my understanding of all the moving parts of parameter estimation. Thank you Professor Olson, for providing understandable biological explanations and help throughout the model development. Thank you both for supporting my research and report over this past year.

We would all like to acknowledge **Dr. Mike Lee** and associates, for providing our team with our project motivating data. Thank you for meeting with our team and allowing us to use your experimental data within our research.

Finally, we would like to acknowledge the NSF grant (NSF/DMS-1819203) that funds Professor Arnold's research.

Contents

List of Figures

Chapter 1

Cancer Cell Dynamics Modeling

1.1 Cell Cycle Dynamics

1.1.1 Healthy versus Cancer Cell Cycle

When modeling cancer cell dynamics, the interactions between cells, internal mechanisms that affect external interaction, and many other factors need to be considered. Experts have spent years trying to learn about this complicated ecosystem, and while the fundamental structures are well studied for healthy populations, populations of unhealthy cells are still being studied and questioned to this day [25].

All cells undergo a regulatory cell cycle to maintain a proper balance of healthy cells within the population. A regulatory cell cycle follows a simple path with several check points along the way. Before a cell can enter mitosis, the process of cell division, a signal must be sent that shows the cell will pass on healthy genes to the next cells. If the healthy signal is not sent, the cell will not undergo the regulatory process again until a healthy mitosis signal is met. When this cell cycle is disrupted, the health of the entire population is in danger. Cancer cells have a mutation that disrupts the cell cycle eliminating the self-regulation found in healthy cells. This implies that cancer cells will pass on unhealthy genes, which does not necessarily mean that cancer cells divide faster than normal cells [27]. Figure 1.1 depicts the growth of cells within the regulated and unregulated cell division cycles.

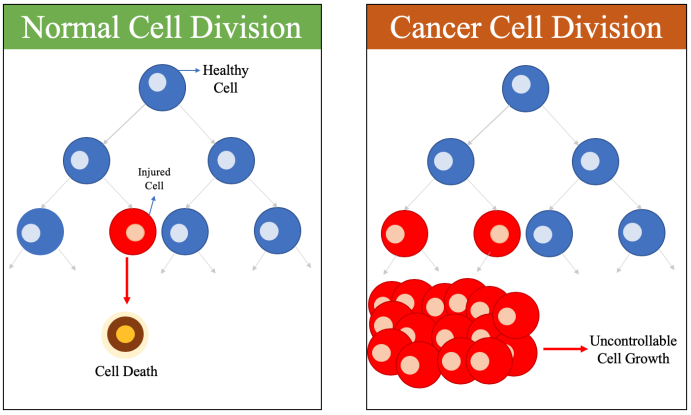


Figure 1.1: This is a schematic representing the basic difference between the regulated and unregulated cell division cycles that are common with healthy and cancer cells respectively. Image is reproduced based on schematic found in [23].

As depicted in Figure 1.1, a normal cell will only divide when it has the ability to pass on only healthy genes to its daughter cells. If the cell is unable to pass on healthy genes, the cell will either enter the cell cycle again or induce death cell death as shown in Figure 1.1, depending on the number of iterations the cell has undergone. Cancer cells do not have this regulatory process, which means they divide without proper regulation [6].

1.1.2 Cancer Cell Cycle with Cancer Treatment Introduction

The idea that cancer cells cannot regulate the cell cycle is a main motivating factor for potential cancer treatments. It is not enough, however, to just look at the irregularity in cancer cell division. When considering a model to describe the dynamics of cancer cells, many experts emphasize the need to consider cancer drug interactions within the cell cycle [27]. There are two main ways that cancer treatments regulate and control the spread of cancerous cells within a population; they either target the cell division process by preventing cell division until healthy genes can be passed on, or they initiate the sequence of cell death [9]. Experts use these two types of cancer treatments, or combinations of both, to optimize the amount of cancer cell death, while maintaining the livelihood of the surrounding healthy cells [22].

Regulating cell division is the more complicated course of treatment, but it is considered to be safer when prescribed to humans. The goal of this method is to revert the abnormal cell division time of cancer cells to the average healthy cell division time period of 18-24 hours [8]. The period of 18-24 hours is also used as a reference point for treatments that induce cell death. Cancer cells introduce mutations to the greater cell population, thus creating a hostile environment for healthy cell division. To combat these mutations, cancer treatments restore normalcy to the population by utilizing healthy cell division processes as benchmarks for cell division and cell death [12].

There are hundreds of active and FDA approved cancer drugs in circulation today, all of which target cell cycle checkpoints and/or induce cell death if the cell cannot regain its self-regulatory cycle [7]. The three drugs considered through our sample data include Camptothecin, RSL3, and SGI-1027. These three drugs are introduced at different times in the cell cycle, and exhibit both pathways of cancer cell regulation. As an example, the drug RSL3 initiates ferroptosis, a specific type of cell death, where as SGO-1027 sparks a tumor suppressant gene that will slow the division process [8]. The variations of treatments within each of these drugs are key components within any model meant to describe the dynamics of a cancerous population. Our team developed our proposed model with the basic biological principals presented in this chapter, as well as allowing model flexibility for the addition of cancer treatments to the population.

1.2 Common Mathematical Models for Cell Growth

Along with the biological constraints highlighted in the previous section, we must familiarize ourselves with the common mathematical models used within the cancer research field. It is imperative to note that any model meant to confidently fit cell data must describe the entire population, not just an individual cell within the population [5]. The internal dynamics of a cell are a necessity, but a model must also describe the interactions between cells and treatments.

The goal of this section is to outline several universally accepted models for cell growth and introduce our proposed model. Our proposed model incorporates fundamental pieces from the outlined models.

When introducing our proposed model, we support our thought process by examining the growth patterns of dead cell experimental trial data and comparing them to the common growth models. We further develop our model by introducing potential variations of terms and parameters explored throughout our research. Specifically, we show how time-varying growth parameters can accommodate variations in data based on different cancer treatments.

1.2.1 Ordinary Differential Equation Models

Most biological systems are not easily described by simple models due to the complicated dynamics of the system. Mathematical models are developed for many different purposes and often focus on a subset of factors within biological systems. There are a variety of differential equations commonly used to model aspects of cell populations. Some focus on the growth of the system, while others consider the interactions of population members and their environment. No matter what a model focuses on, or assumes, the main goal of any model is the same: to reproduce and explain curves that are produced through experimentation [6].

Data collected through lab experiments can help motivate the basis of a mathematical model. Experimental data is usually sparse due to capacities of machines and other factors. When a model is developed to capture the dynamics of this data, synthetic data sets can be generated for further computer simulations. This allows researchers to test various amounts, types, and frequencies of data collection to be tested on without the time-consuming and expensive lab experiments. From here, models can be used to provide deeper insight into important population mechanisms such as potential time-varying components, experimental design components such as time scales and moments of importance, and proper data collection for optimized estimation of events.

With cancer cell populations, it is imperative to model the growth and death rates of the tumor cells. Ordinary differential equations (ODEs) are commonly used models due to their natural ability to describe the rate of change in cell populations, both live and dead, with respect to time. There are multiple different ODE models that are widely accepted within the modeling community for cancer cell growth, all of which assume a well-mixed population. The two most common are the exponential and logistic growth models, which are elaborated upon in the following sections.

The two common ODEs presented in this section describe the basics of cell growth, but need further development to explain all the dynamics within a cancerous cell population. Further dynamics can possibly be captured through the LED Model, a lag model commonly used within the cancer research field. This section will focus on explaining the basics of the common exponential and logistic models, as well as introducing the LED model and our initial proposed model that incorporates different aspects from the common models.

Exponential Growth

The most common and universally accepted model for cell growth is the exponential growth model. Exponential growth is known to happen within every cell population within a short time frame [2]. This is why the model is often utilized as the first line of fitness for more in-depth, concise models.

Exponential growth allows for unlimited growth, which is very rarely found in nature due to resource limitations [8]. Biologists believe, however, it can provide a baseline for most experiments that utilize a single-species population [5]. The basic form of the exponential growth equation,

$$\frac{dP}{dt} = rP \tag{1.1}$$

demonstrates the relationship between $P = P(t)$, the total number of cells in a population at time t, and r, the population growth rate in terms of inverse time. In the current form, Eq. (1.1) only depicts how the population will grow over time. Integrating Eq. (1.1) by separation of variables method yields the solution

$$P(t) = P_0 e^{rt} \tag{1.2}$$

providing the total number of cells in the population at time t. In Eq. (1.2), P_0 is the number of cells in the initial population and r is still the growth rate, both of which are typically assigned constant values. The infinite growth allowed in exponential models can be seen in Figure 1.2 below. This figure also demonstrates the growth rate r is the driving force behind the steepness of the curve.

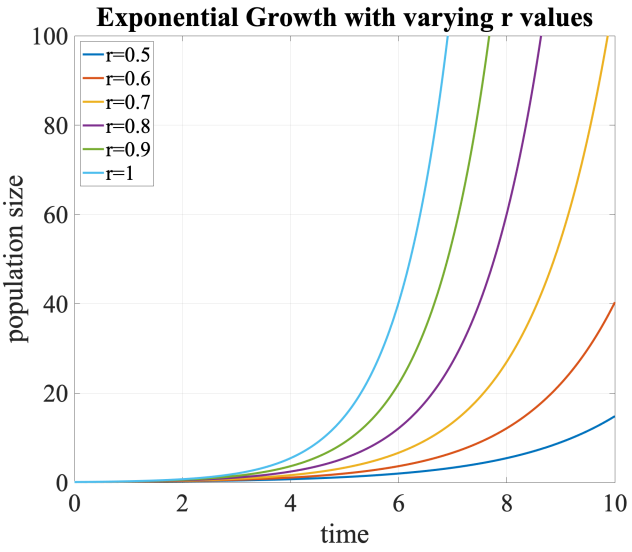


Figure 1.2: Exponential growth model described in Eq. (1.1). Each line depicts the solution to Eq. (1.1) over the time span, $0 \leq t \leq 10$, with $P_0 = 0.1$, and values of the growth rate, $r \in [0.5, 1]$. The growth rate r changes the steepness of the curve; for example, the larger the growth rate, the steeper and faster the curve grows.

The exponential growth model provides a basic framework for population growth, but should be developed further as most natural phenomena do not follow it perfectly. This is commonly due to limitations in resources that stop infinite growth. The general ideas laid out in the exponential growth model are motivation for the logistic growth model.

Logistic growth

Much like the exponential growth model, the logistic growth model is used to describe the change in population size over time. The logistic growth model is best described as an extension of the exponential growth model. By including a second term, the logistic growth is better suited to capture natural phenomena. The extension is introduced by adding a second term to the exponential growth model to generate the logistic functional form,

$$\frac{dP}{dt} = rP - \frac{rP^2}{K} \tag{1.3}$$

which describes the growth of the population while also accounting for potential limitations of the system. There are two main dynamical changes within this limitation term. First, is the incorporation of the environmental carrying capacity, denoted as K, which has units of population. This term is the maximum number of members a population can sustain under environmental limitations and is the source of the flip in concavity of the logistic curve seen in Figure 1.3. The carrying capacity takes into consideration the effects of limited resources that exponential growth is unable to capture. The second addition accounts for the pairwise interactions between the cells within the population, P^2.

The first half of the equation, rP, is the main body of the exponential growth model (Eq. (1.1)). The latter component incorporates the gradual reduction of population size due to the carrying capacity and fighting for resources. Complex interactions cells have with each other and their surrounding environments, such as fighting for resources or space [5], are accounted for in the extension. Logistic growth models are also considered more reliable than exponential growth because they utilize the carrying capacity as a population's cell maximum capacity. While exponential growth models allow for infinite growth, the logistic model guarantees the population size will stay contained below the carrying capacity. Any change in initial population size or growth rates does not change the carrying capacity of the system. If $P_0 < K$ the population will grow based on the growth rate, but will never pass this carrying capacity. If $P_0 > K$, no matter the growth rate, the population will decrease until the capacity is met. Figure 1.3 below is a visualization of this phenomena.

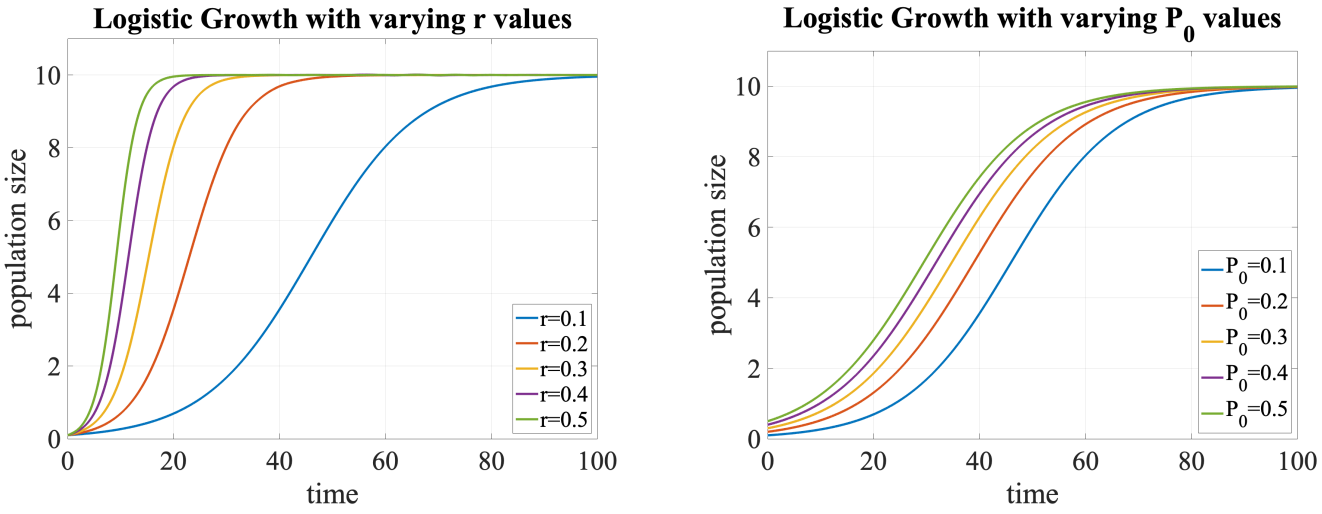


Figure 1.3: *Left*: Solution to the Logistic growth model model described in Eq. (1.3). Each line depicts the solution to Eq. (1.3) over the time span, $0 \leq t \leq 100$, with initial population size $P_0 = 0.1$, carrying capacity $K = 10$, and varying growth rate, $r \in [0.1, 0.5]$. As the growth rate, r, increases, the solution reaches the carrying capacity K faster. *Right*: Solution to the Logistic growth model model described in Eq. (1.3). Each line depicts the solution to Eq. (1.3) over the time span, $0 \leq t \leq 100$, with growth rate $r = 0.1$, carrying capacity $K = 10$, and initial population size $P_0 \in [0.1, 0.5]$. As the initial population, P_0, increases, the curve is shifted closer to the carrying capacity and requires less time to reach the carrying capacity K.

While logistic growth provides a more natural framework for population growth, it does not explicitly account for population death [8]. Logistic growth only describes the slowing of population growth due

to limitations in resources found in nature. Most experiments remove the fight for resources with the intentions of gaining insight into how the cells interact with one another when the population can thrive. In our proposed model, we do no incorporate carrying capacity directly. We do, however, include a natural slowing of growth, which could be due to resource limitation, through a time-varying natural growth parameter.

1.2.2 Lag Exponential Death (LED) Model

While the ODE systems previously mentioned are often used to describe the growth of the live population, it is important to consider models that also show the interactions between live and dead cells. Many researchers in the cancer field utilize the Lag Exponential Death (LED) model [3, 16, 13]. For the purposes of this paper, we will not go into detail about the dynamics of the LED model. Instead we will just introduce its potential as a model that shows the connections between the live and dead cells in the system.

In cancer cell dynamic models, the growth of the dead cell population, C_{dead}, is just as important as the live cell population, C_{live}. As mentioned earlier, cancer treatments impact the evolution of C_{live} over time by either resetting cell division or inducing cell death. Along with the change of C_{live} over time, we must also calculate the evolution of C_{dead} and the relationship between the two. A simple way researchers quantify this relationship is through the LED model and lethal fraction scoring (LF). The LED model uses a the lethal fraction scoring ratio,

$$LF(t) = 1 - \frac{C_{live}(t)}{C_{live}(t) + C_{dead}(t)} \tag{1.4}$$

which requires data, or at minimum, assumptions on both live and dead cells at any point, t [3]. Equation (1.4) is used to calculate the ratio of live to dead cells at a specific time, but does not generate the change in this ratio over time. Instead lethal fraction scores provide the basis for the LED. Specifically, the lethal fraction scores are parameterized within the LED model

$$LF(t) = LF_0 + (LF_P - LF_0)\left(1 - e^{-D_R(t-D_0)}\right) \tag{1.5}$$

where t is time, and D_0 is the point in which the lethal fraction begins to grow exponentially. Beyond this, LF_0 represents the average value of the lethal fraction before D_0, D_R is the maximum cell death rate, and the LF_P is the plateau of lethal fraction score [3].

Lethal fraction scores explain the proportion of the population that is in the live versus dead cell population. The difference between the two subsets of the population provide researchers with a scoring mechanism for treatment effectiveness. A larger lethal fraction implies that more of the entire population is within the dead cell population. When considering treatment for cancer, the more cancer cells that are dead, the better the drug is working.

Though our team does not explicitly use the LED model, we do believe that using the lethal fraction scoring mechanism for future research into the parameter estimation techniques ability to describe the effectiveness of the particular experimental drug will eventually be necessary.

1.3 Initial Model Development

A mathematical model used to describe the interactions between cancer cells and cancer treatments should utilize components of the common biological growth models presented in this chapter. The model also needs to capture the movement of any available observed data. In our case, our team was provided data collected through a set of experiments conducted at University of Massachusetts (UMass) Medical School in Worcester, MA. The data set consisted of cumulative dead cell counts over a 48 hour timespan and initial and final values of live cell counts for three different cancer treatments at varying concentrations. Each treatment concentration was used in the experiment twice to show potential consistent behaviors. Further details regarding experimental structure and data collection methods can be found in Chapter 3.1. In this section, however, we explain how the basic growth patterns exemplified by the data influenced our development of our proposed model.

Though we assumed this experimental data would mimic the growth patterns of the commonly used models, we could not assume this to be true without visualizing the data. Figure 1.4 depicts examples of the raw cancer treatment data plotted to visualize the general growth of the dead cells.

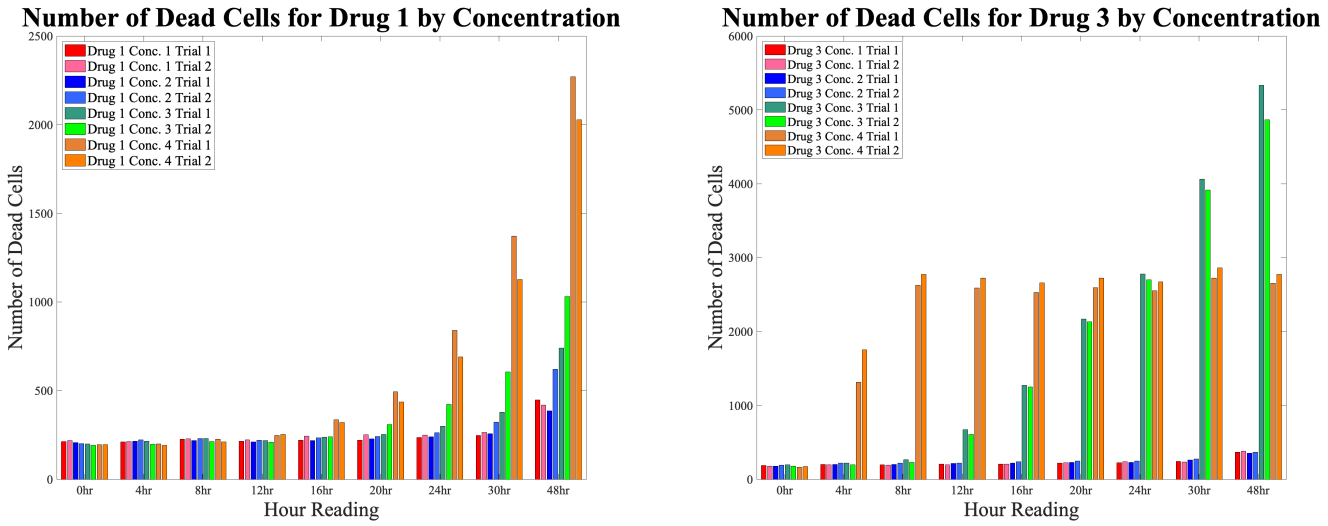


Figure 1.4: *Left*: Plot of experimental data provided for dead cell counts at the specified times of data collection. Each paired color represents Trials 1 and 2 for each particular concentration of the same drug. This particular drug demonstrates a potential exponential growth pattern for dead cells over the 48 hour trial run. *Right*: Plot of experimental data provided for dead cell counts at the specified times of data collection. Each paired color represents Trials 1 and 2 for each particular concentration of the same drug. Drug 3 demonstrates the anomalies that can be found over the variety of cancer treatments. This particular drug induces a different type of cell death resulting in immediate death for all cells within the population.

Plotting the dead cell count data allowed our team to assume that when introduced to certain treatments, the dead cell population would grow exponentially, while it would not with others. With this said, there are many examples of cancer treatments where this exponential growth pattern might not appear; Drug 3, shown in the right image of Figure 1.4, shows potential anomalies each cancer treatment could impose. This indicates any model we develop to capture the dynamics of this data must have flexible parameters that can be adjusted based on the given treatment.

As previously mentioned, our goal for this project is to develop a model that could capture the different dynamics of the interactions between cancer cells and different treatments. The troubling issue with cancer treatments is that each tumor cell can interact differently with treatments. Though the actual growth patterns of dead cells differs for each treatment, a general model can be developed to explain the dynamics of the system with varying parameter and time scales. Our team developed the final proposed model, depicted in Equations (1.8) and (1.9), by looking at the varying growth patterns and using our knowledge of the common ODE models to express the dynamics.

Before reaching the end result, our team broke down the model into two coupled equations describing the change in live and dead cells over time. We first looked at the basic dynamics within the dead cell population, as our data only provides information on this part of the total group. When looking at the potential exponential natural growth of the dead cells, our team developed this equation

$$\frac{dC_{dead}}{dt} = \beta C_{dead} \tag{1.6}$$

where β is the cell-induced death rate, and $C_{dead} = C_{dead}(t)$ is the total number of dead cells in the population at time t. As mentioned earlier in this section, our team knew we could not stop here when modeling the provided data, as the exponential model is meant to be an ideal situation. The experiments conducted at UMass introduced the initial cancer population to cancer treatments that could potentially induce cell death. Eq. (1.6) does not account for drug-induced cell death. To combat this, our team introduced the additional term to generate

$$\frac{dC_{dead}}{dt} = \beta C_{dead} + \gamma C_{live} \tag{1.7}$$

where γ is the rate at which the drug-induced cell death occurs and $C_{live} = C_{live}(t)$ is the total number of live cells. This new term allows for cells that were previously a part of the live cell population to join the dead cell population through a prescribed death rate. The total number of cells in the system depends on both the live and dead cell populations. Eq. (1.7) depicts the potential dynamics of the dead cell population, but to better understand the population as a whole, we must include a coupled ODE that describes the live cell dynamics. Our team used basic biological principles to develop our proposed model,

$$\begin{aligned} \frac{dC_{live}}{dt} &= \alpha C_{live} - \beta C_{dead} - \gamma C_{live} && (1.8) \\ \frac{dC_{dead}}{dt} &= \beta C_{dead} + \gamma C_{live} && (1.9) \end{aligned}$$

which depicts a coupled ODE system with shared initial conditions $C_{live}(0)$ and $C_{dead}(0)$. The parameter α is a constant that corresponds to the growth rate of cells within the population. The coefficient β is the cell-induced death rate, which coincides with the the fact that once a cell dies it releases toxins that alert surrounding cells of ensuing death and may cause those cells to move from the live cell population to the dead cell population. Finally, the coefficient γ is the rate of death due to the specific drugs being implemented into the population.

For the duration of this paper, our proposed cancer cell model will be depicted by Equations (1.8) and (1.9) unless otherwise stated. Our initial considerations assume constant parameters, α, β and

γ, but we later modify the model to account for time-varying parameters. Specifically, we consider a variety of time-varying growth parameters $\alpha(t)$ as a potential way to incorporate the drug-induced slowing of the live cell population due to lack of resources and eventual elimination of live cells by the dead cell population reaching its determined maximum.

Chapter 2

Parameter Estimation Techniques and Example Inverse Problem

In mathematical modeling, there are two types of problems, the *direct problem* and the *inverse problem*. Direct problems describe the process of finding the solution to a given model provided a set of assumed initial factors [14]. In contrast, inverse problems are generally defined as the process of recovering information about the unknown factors that generated an observed solution [17]. Direct problems are usually well-posed, meaning they have a known unique solution that is not greatly impacted by noise in the data. The purpose of these problems is to predict the data generated by the well-posed model [14, 19]. Direct problems assume the factors of the model are known and want to answer the question: what will the result be with these factors? Inverse problems, unlike their counterpart, have little to no information about the initial conditions, especially the parameter values in the model [14]. Inverse problems use observed data and want to answer the question: what factors produced this result?

This is often considered the more interesting and flexible approach to a mathematical modeling problem. More often than not, we want to learn more about the immeasurable factors that produced this data. While it would be much easier to describe a system if we could assume every component, this is not always plausible. The majority of real world systems, especially biological ones, have too many characteristics to measure all at once. Thus, the very nature of biological systems requires the inverse problem approach to mathematical modeling. To combat this problem, researchers have utilized parameter estimation techniques that use observable system components to infer about the immeasurable state and parameter values within a system.

State variables, usually, describe the physical components of a system. In a biological system like ours, the state variables are the number of live and dead cells in the population. In a physical system, like our example system described later in this chapter, the state variables are position or velocity. These components of a mathematical model are the values that researchers are more likely to observe or measure. In contrast, parameter values are most often immeasurable, meaning they need to be estimated through various techniques. Parameters can come in many different forms, and based on their internal dynamics, they require different estimation techniques.

In this chapter, we set out to further describe the different types of parameters and how they impact the system over time. We will introduce two commonly used estimation techniques, the Nelder-Mead Algorithm and Kalman Filtering, as well as compare their abilities to estimate a simple oscillatory mass-spring system.

2.1 Parameters in Autonomous and Non-autonomous Dynamic Systems

Before we can delve deeper into the different types of parameters, it is necessary to understand the systems these parameters help define. A system can be defined as a grouping of entities that can be used to describe the whole of a natural event [11]. The different types of systems depend on any changes that occur to a physical property over a given amount of time.

Very few natural properties can be described as truly constant, which is why the majority of systems are defined as *non-autonomous dynamic systems*. More specifically, a non-autonomous dynamic system most often consists of a system of ODEs that describe the evolution over time of the state variables [11]. Where an *autonomous* system consists of parameters of interest that do not change over time. For example, an autonomous system has a right hand side (RHS) of the system not explicitly dependent on time t (i.e. $dP/dt = rP$). A non-autonomous system would have a RHS explicilty dependent on t. Truly autonomous dynamic systems do not occur in the natural world. Most non-autonomous dynamic systems, however, do have time-invariant, or constant, components. These time-invariant components often appear in the form of constant parameters.

Constant parameters are the simplest form of a parameter, and are often the easiest to implement when developing a model as their impact on the dynamic system can be well monitored. Some aspects of a system can easily be described with a constant parameter, but more often than not, they can not fully capture all aspects of a dynamic system. As stated above, truly autonomous systems do not commonly occur in the natural world. Constant parameters are very helpful as they can simplify certain aspects of a model that are not of interest or are well understood in the field. For example, most biological systems assume a constant exponential growth rate for a population. This assumption allows researchers a baseline to extend the other less well-known aspects of a model, while still maintaining the biological integrity of their model.

Non-autonomous systems are an extension of autonomous systems. To extend these, time-varying parameters are added to accommodate stronger dynamics within a system. Depending on the needs or dynamics of the system, the functional form of these parameters may vary. For the purposes of this paper, our team explored three different time-varying parameter forms for our research. The first is a sigmoidal function,

$$f(t) = \frac{1}{1 + e^{\tau t - s}} \tag{2.1}$$

which decreases over time as shown in the left-most image in Figure 2.1. Time-varying parameters, $f(t)$, often introduce new hyperparameters to the system. In the case of Equation (2.1), $f(t)$ is the time-varying parameter for the system as a whole, but new hyperparameters, τ and s, impact the shape and slope of the new form. For a logistic function like Equation (2.1), $\tau \geq 0$ shifts the rate of decline from the maximum to the minimum point, while $s \in \mathbb{R}$ shifts the blue line in the left most image in Figure 2.1 left or right.

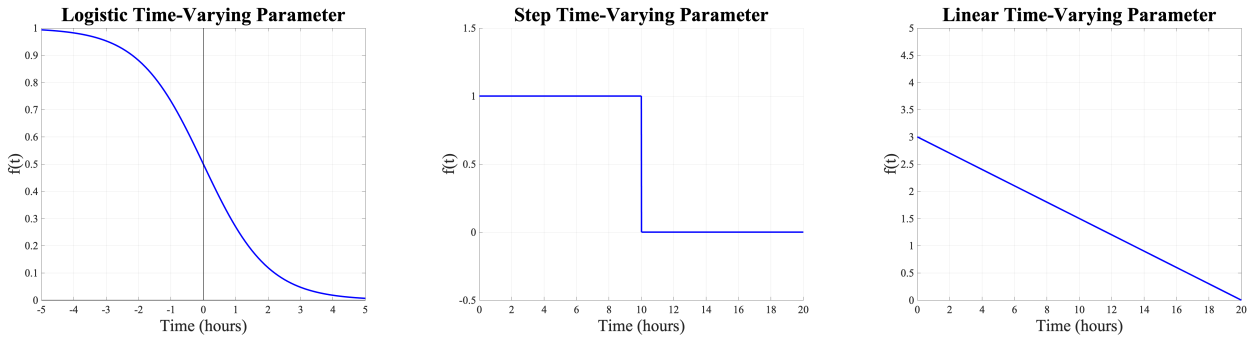


Figure 2.1: *Left*: Graph of the time-varying sigmoidal growth parameter, blue line, as depicted in Equation (2.1) with $\tau = 1$ and $s = 0$ from time $-5 \leq t \leq 5$. *Middle*: Graph of a time-varying step parameter, Eq. (2.2), with $f_1 = 0$, $f_2 = 1$ and $t_1 = 10$ from time $0 \leq t \leq 20$. *Right*: Graph of a linear time-varying parameter, as depicted in Equation (2.3) with $m = 0.15$ and $b = 3$ from time $0 \leq t \leq 20$.

The other two images shown in Figure 2.1 are a step-function, middle, seeing a decrease in the parameter value at a certain point t_1,

$$f(t) = \begin{cases} f_0 & t \leq t_1 \\ f_1 & t > t_1 \end{cases} \tag{2.2}$$

and a linear function, right,

$$f(t) = b - mt \tag{2.3}$$

where b is the initial value of $f(t = 0)$ and m is the slope of the line.

There are a variety of time-varying parameters that greatly impact the dynamics of a system. The choice of the functional form of a parameter, may it be constant or time-varying, is dependent on the knowledge of the system. More often than not, the exact form of a potentially time-varying parameter will be unknown, but the general movement of it will be known. For example, for our developed model (Eq. (1.8) and (1.9)), we must understand the nature of cancer treatments. Cancer treatments usually have an onset-time, a point in time where the treatment starts to induce death. This means that to properly use a time-varying drug-induced death rate, γ, we must use a time-varying parameter that changes at time t_1 like the step function (Eq. (2.2)). This exact form might not be exactly known, but the general path is necessary when making these assumptions.

It should also be noted that choosing parameters greatly impacts the potential changes in the state variables over time as well as the type of estimation technique chosen to capture these dynamics. Parameter values, in general, are the most volatile component of a system, as they have the greatest impact on the state variables. The choice of constant, time-varying, or a combination of both will change the dynamics of the system and the amount of observable data needed to provide a proper estimation of the system.

2.2 Parameter Estimation Techniques

The choice of parameters within a model greatly impacts the estimation technique implemented on the system. No estimation technique is inherently better or worse than another, but should be

evaluated for the given problem before being utilized. In other words, every estimation technique has its strengths and weaknesses which should then be compared to the results of another estimation. In this section, we will introduce the two methods used for estimating the parameters of our proposed model, the Nelder-Mead Simplex Algorithm and the Kalman Filtering methods.

2.2.1 Direct Optimization versus Statistical Bayesian Approaches

The two techniques used in our study are examples of two very different approaches to parameter estimation, Direct Optimization and Bayesian. These two approaches differ in a variety of ways, but share the common goal of using measurable state data and model assumptions to find a set of parameters that capture the dynamics of a system. They differ in how the noise level of the data is incorporated, what initial assumptions are required, and criteria for estimations to properly capture the dynamics of the observed data.

Data collection for biological systems is noisy by nature. Both intrinsic and extrinsic noise needs to be accounted for when modeling a biological system [18, 20]. Intrinsic noise within a cancer cell accounts for all the internal workings of the cell that are not being explicitly modeled by the system. Extrinsic noise accounts for inaccurate data collection methods. An example is a machine's inability to capture full cell fluorescence. A parameter estimation approach must be able to quantify this noise, as it can explain identifiability issues for the system as a whole.

The direct optimization approach to parameter estimation produces a point estimation of parameters. In other words, direct approaches take in a reasonable initial guess for model parameters, and searches exhaustively within this range for an optimal set of parameters that best fits the data [18]. The key here is that the initial guesses for the parameter values need to be within a reasonable range of the final optimal parameter values. This reasonable range depends on the assumptions of the model being used to capture the system dynamics. This extensive search does not take into account the inherent noisiness of the data; which, as mentioned previously, is a necessity when modeling biological cell dynamics. Instead, direct approaches focus on optimizing a cost function. In most cases, the goal is to reduce the error between the observed data and the model solution with the estimated parameter values. This approach does not sequentially update the parameter values to minimize this error. Instead, the entirety of the observed data is compared directly to the solution of the model with the estimated parameter values. The Nelder-Mead Simplex method is an example of a direct optimization approach to parameter estimation.

Like their direct optimization counterpart, Bayesian based approaches also utilize observed data and an assumed form of a model to estimate the unknown parameters. However, Bayesian approaches estimate parameters based on prior knowledge of the system, may it be the initial guesses of the parameter values or the estimations at the previous time point. They also produce an estimation distribution, usually defined as a posterior distribution [20], instead of the point estimation found in direct optimization. These state and/or parameter value distributions accommodate the inherent noise of data by providing a general range of possible values for the given time step. While most direct approaches require more care when choosing the initial parameter guesses because they may not converge to the optimal value, the initial parameter guesses in the Bayesian framework do not need to be known or well understood. Instead, Bayesian approaches use a random sample for the initial guess based on prior knowledge or distributions.

Kalman filtering methods are well known and commonly used Bayesian based parameter estimation methods. We explicitly use the Ensemble Kalman Filtering method in our study. We delve further into the algorithm mechanics in this section.

2.2.2 Nelder-Mead Simplex Algorithm and fminsearch

As mentioned above, there are a variety of different methods to estimate parameters. A simple method used to estimate constant, or static, parameters is the MATLAB® function **fminsearch**. **fminsearch** implements the Nelder-Mead simplex algorithm, a direct optimization approach to parameter estimation. The goal of this technique is to optimize a cost function by minimization [10]. In the context of finding a best fit line for a set of data, the cost function is described as the residual between the expected and the real values at any given time.

This method iteratively generates solutions to a cost function, with the goal being to minimize said cost function, until an optimal solution is found or the maximum number of iterations is met. At each iteration, a region is generated around a potential optimal solution. Once it is generated, the vertices are either *reflected, expanded, contracted, shrunk* or any combination of these scalar processes. These processes are often represented by the following: a reflect, b expand, c contract and d shrink; with the only requirement to satisfy $a > 0$, $b > 1$, $0 < c < 1$ and $0 < d < 1$ [10]. The built in MATLAB® implementation of the Nelder-Mead algorithm, **fminsearch**, uses the standard choice of parameter values,

$$[a \;\; b \;\; c \;\; d] = [1 \;\; 2 \;\; 1/2 \;\; 1/2] \tag{2.4}$$

unless otherwise specified by an alternate input. It is with these restrictions in mind that the Nelder-Mead algorithm iterates through its process of shifting the parameters towards an optimal value. Each iteration of the parameter estimation goes through a process where the objective function values are weighted, compared and shifted based on the shifts stated above. Figure 2.2 is a visual representation of Nelder-Mead algorithm iterations and further details about the shifts is clearly stated in [10].

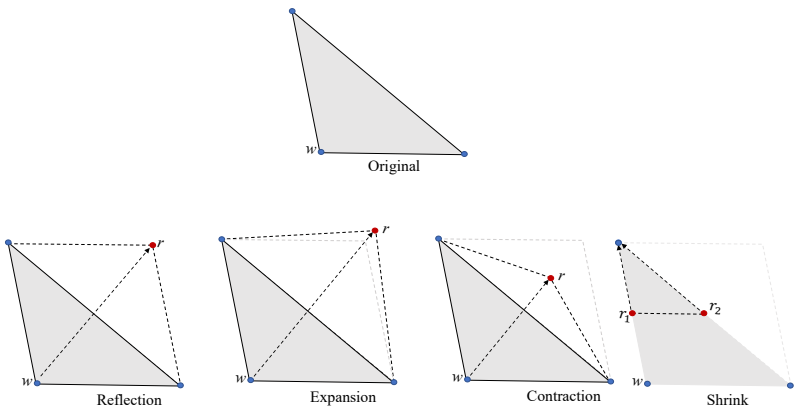


Figure 2.2: 2-dimensional visualization of Nelder-Mead Algorithm Iteration possibilities. After each iteration, the same four shifts, reflection, expansion, contraction and shrink, can be utilized to find the optimal solution for the point w.

Within Figure 2.2, the point w is the point being shifted through the Nelder-Mead process. During reflection, expansion, and contraction, it is only the point w that is being shifted to its new value r. If the process indicates the need for a shrink, there are two new points, r_1 and r_2 that can be considered for the next iteration of shifts.

Though the Nelder-Mead algorithm is one of the most used direct search methods, it is not guaranteed that the algorithm will converge to a critical point [10]. Unlike most linear optimizing algorithms, Nelder-Mead can converge to a solution that is not the optimal solution. The optimal solution is found at the global minimum, which is a critical or intersection point of some or all constraint equations. This happens more often within larger dimensional problems, but has also been seen in lower dimensions. The algorithm can get stuck at local minimums and remains unconstrained unless constraints are provided. In most biological cases, the signs of the parameters being estimated are usually known. The lack of constraints and guaranteed optimization can cause problems with certain dynamical models. This is why the Nelder-Mead algorithm should only be implemented for small dimensioned problems.

2.2.3 Kalman Filtering

Parameter estimation techniques vary based on their abilities to correctly encompass the dynamics of the system and each parameter over time. The majority of techniques rely heavily on the estimation of only static parameters, as they are dynamically constant and remain constant over the time intervals. Though static parameters are appealing for studies, they are very rarely used in natural circumstances. Thus, it is helpful to utilize parameter estimation techniques that will adjust for dynamic, non-static parameters.

There are a variety of techniques for dynamic, time-varying parameter estimation. The most commonly used are statistical (Bayesian) based approaches, such as the sequential Monte Carlo methods or Ensemble Kalman filters [1]. The technique used for our work is the augmented Ensemble Kalman filtering method (EnKF), because of its ability to accommodate a variety of differential equation models.

The EnKF algorithm consists of two steps; the prediction step and the analysis, or observation update, step. During the prediction step, the value of the prior distribution and an evolution equation are utilized to predict the values of the states and parameters in the posterior distribution at the next time step. The observation step utilizes an observed value to correct the prediction made in the previous step [1]. The steps of algorithm implementation are below.

To begin, the current density at time j is represented as the ensemble,

$$S_j = \{(x_j^1, \theta_j^1), (x_j^2, \theta_j^2), ..., (x_j^N, \theta_j^N)\} \tag{2.5}$$

where the ensemble is comprised of N combinations of state x_j and parameters θ_j at each time t_j. As an example, for the cancer cell model in Eq. (1.8) - (1.9), our ensemble has N vectors for our states, C_{live} and C_{dead}, and parameters, α, β and γ. This ensemble is representative of the underlying probability distribution,

$$\pi(x, \theta|y) \tag{2.6}$$

at time t_j. From here, the prediction step uses the state evolution equation, in our case, it is the solution to our proposed differential equation model (1.8)-(1.9), to form the prediction ensemble

$$x_{j+1}^n = F(x_j^n, \theta_j^n) + v_{j+1}^n, \;\; n = 1, 2, ..., N \tag{2.7}$$

where v_{j+1}^n is the error in the model prediction. This error can be changed to alter the filter's confidence in its prediction. The prediction step will see changes in the states, but the parameter values are not updated during this step. In other words, we set $\theta_{j+1}^n = \theta_j^n$. Then an augmented state and parameter vector,

$$z_{j+1}^n = \begin{bmatrix} x_{j+1}^n \\ \theta_{j+1}^n \end{bmatrix}, \;\; n = 1, 2, ...N \;\; , \tag{2.8}$$

is created. This vector includes a set of N, d dimensional vectors, where d is the number of state and parameter variables in each ensemble. From the augmented state and parameter vector, the prediction ensemble mean is calculated by

$$\bar{z}_{j+1} = \frac{1}{N} \sum_{n=1}^{N} z_{j+1}^n, \;\; \bar{z}_{j+1} \in \mathbb{R}^{d \times d} \tag{2.9}$$

and the prediction covariance matrix is calculated by

$$\Gamma_{j+1} = \frac{1}{N-1} \sum_{n=1}^{N} (z_{j+1}^n - \bar{z}_{j+1})(z_{j+1}^n - \bar{z}_{j+1})^T, \Gamma_{j+1} \in \mathbb{R}^d \tag{2.10}$$

which results in a square, $d \times d$, covariance matrix.

From here the analysis step starts. An artificial observation ensemble is formed around a true observed value y_{j+1},

$$y_{j+1}^n = y_{j+1} + w_{j+1}^n, \;\; n = 1, 2, ..., N \tag{2.11}$$

where w_{j+1}^n is the observation error drawn from a Gaussian distribution with mean 0 and covariance D_{j+1}. Then the observation model prediction ensemble is generated

$$\hat{y}_{j+1}^n = g(x_{j+1}^n, \theta_{j+1}^n), \;\; n = 1, 2, ..., N \tag{2.12}$$

where g is the observation function that allows the observed values to be compared to the prediction ensemble members in the next few steps. The final step involves calculating the augmented posterior ensemble

$$z_{j+1}^n = z_{j+1}^n + K_{j+1}(y_{j+1}^n - \hat{y}_{j+1}^n), \;\; n = 1, 2, ..., N \tag{2.13}$$

where K_{j+1} is the Kalman gain which is found by

$$K_{j+1} = \mathcal{S}_{j+1}^{z\hat{y}} \left(\mathcal{S}_{j+1}^{\hat{y}\hat{y}} + D_{j+1} \right)^{-1} \;\; . \tag{2.14}$$

Within this equation, $\mathcal{S}_{j+1}^{z\hat{y}}$ and $\mathcal{S}_{j+1}^{\hat{y}\hat{y}}$ are the covariances between the augmented prediction and the observation predictions, and the observation prediction respectively. These are calculated in the same manner as Equation (2.10). Also, the Kalman Gain introduces the observation noise covariance, D_{j+1}, which is the matrix that accounts for any adjustments made by the filter to accommodate noise in the data [1].

The EnKF is a particular algorithm within the Kalman Filtering family that allows for estimation of both state and parameter values. The algorithm presented in this section is used when all parameters are defined as constant or static parameters. Further changes can be made to the EnKF presented by adding a random walk model through the defined walk step ξ,

$$\theta_{j+1}^n = \theta_j^n + \xi_j^n, \;\; \xi_j^n \sim \mathcal{N}(0, \sigma_\xi^2) \tag{2.15}$$

during the prediction step of the filter to accommodate time-varying parameters. This addition will not change the estimation of parameters that are constant; if a parameter is meant to be a constant, the random walk model will not alter the EnKF's estimation of this constant parameter. The main caveat with this addition is the sensitivity of the algorithm to the dynamics of the system. Studies have shown that the time-varying parameter needs to change at a much slower rate than the dynamics of the entire system for the parameter dynamics to be captured by the filter [26]. It should also be noted that the choice of the drift variance σ_ξ can alter the confidence and accuracy of the algorithm's estimation of both state and parameter values [26].

2.3 Parameter Estimation on Example Problem

Parameter estimation techniques, such as those outlined in this chapter, provide researchers with a way to verify and support experimental design. The posterior distributions produced through Bayesian based approaches, such as the Kalman Filtering methods, can be very useful when designing future experiments [20]. Direct optimization approaches, such as Nelder-Mead, can be optimal for systems with well known behaviors. Both of these approaches can be useful when trying to gain information about the behaviors of a system. Every system, however, is unique and no one parameter estimation technique should be used for all systems. Some systems require more flexibility in their estimation, thus an estimated distribution would be a more desired final product.

To better illustrate this sentiment, our team implemented both estimation techniques described in this chapter on a simple mass-spring model

$$mp''(t) + bp'(t) + kp(t) = 0 \tag{2.16}$$

where $p(t)$ is the position of mass at time t, $m > 0$ is mass, $b \geq 1$ is a damping coefficient, and $k \geq 0$ is the spring constant. This model can be written as a first-order linear system of differential equations

$$\frac{dp}{dt} = v \tag{2.17}$$

$$\frac{dv}{dt} = -\frac{k}{m}p - \frac{b}{m}v \tag{2.18}$$

where $v(t) = p'(t)$ is the velocity of the mass at time t.

This model was chosen as our experimental toy inverse problem because it has well established behavior with given parameter values. With the parameter values, $[m\ \ k\ \ b] = [10\ \ 5\ \ 3]$, the solution for both state variables will be a damped oscillation that should approach zero over time. We also used this example as a test problem to validate our EnKF algorithms before applying them to our cancer cell dynamics model shown in Eq. (1.8) and (1.9).

2.3.1 Synthetic Data Generation

Before we could implement the two estimation methods on the mass-spring model (2.16), we first needed to generate a set of synthetic data for the experiments. This set of synthetic data was generated by assuming our initial state values, $p(0) = 1$ and $v(0) = 0$, under the constant parameters $[m, k, b] = [10, 5, 3]$. Then we added Gaussian noise with mean zero and standard deviation, 10% of the standard deviation in the data points. It should be noted that we assumed that we only had measurable data on the position, state variable p, of the mass over time. This data was used throughout the experimentation described in the following sections.

2.3.2 Nelder-Mead

We first explored the results of the Nelder-Mead Simplex Algorithm on the mass-spring model. We implemented MATLAB® 's **fmincon** to solve for the values of the three parameters, m, k, and b, such that the residual

$$e = y - \hat{y} \tag{2.19}$$

between the assumed form of the model (Eq. (2.17) and (2.18)) with the given parameter values, $\hat{y}$, and the synthetic data, y, was minimized. Specifically, $\hat{y}$ is the true solution for the position equation and our sunthetic data, y, is only the observed position at time t. As mentioned in Section 2.2.2, **fminsearch** is the commonly used function for Nelder-Mead. This function does not allow for restriction on output values, so to combat this our team utilized the Nelder-Mead algorithm with **fmincon**, which is equivalent to **fminsearch** but with respect to specified constraints. In our case, we wanted to restrict the parameter values that could optimize the cost function (2.19) to be strictly positive. Figure 2.3 below shows the results of the Nelder-Mead technique on the toy model.

Though the actual values of the parameters, $[m, k, b] = [10, 5, 3]$, are close to the minimized cost function parameter values, $[m, k, b] = [10.009, 4.7927, 3.6811]$, we can see that the Nelder-Mead Algorithm is still not fully capturing the dynamics in the system. This is especially prevalent when comparing the peaks and troughs of the oscillations of the true and estimated velocity state variable (right of Figure 2.3). The true path of the velocity sees much more dramatic peaks and troughs than the Nelder-Mead estimation. This is due to the algorithm trying to produce one set of parameters that minimizes the collective residual of all data points rather than looking at each data point at every time point as an individual residual meant to be minimized.

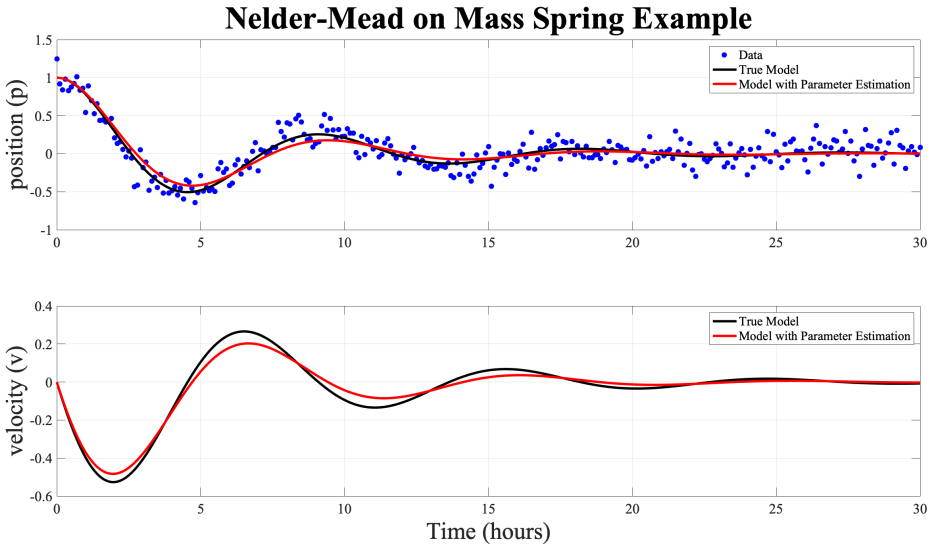


Figure 2.3: Result of both state variables, position and velocity, estimation, under **fmincon**. Red Line indicates the mass-spring model (Eq. (2.16)) with the final estimation of parameters $[m\ \ k\ \ b] = [10.009\ \ 4.7927\ \ 3.6811]$.Black line depicts the mass-spring model with the true parameter values $[m\ \ k\ \ b] = [10\ \ 5\ \ 3]$.

Figure 2.3 also leaves out any information on the parameter values. If we did not know the true parameter values, we would be unable to determine if this estimation is close to the actual values. In this instance, we do know the true parameter values, so we can say that the Nelder-Mead algorithm can estimate the state and parameter variables to an extent. For this particular problem, however, this estimation is not enough. The extremely dynamic nature of the system indicates a need to understand the parameters and their impact on the system as whole. The need for more information supports the idea that the Nelder-Mead Simplex Algorithm should just be a starting point of estimation and further techniques should be implemented on this model.

2.3.3 Kalman Filtering Results

To combat the issues of the Nelder-Mead Algorithm, our team turned to the family of Kalman Filtering Algorithms. Specifically, our team implemented both the Classic and Ensemble Kalman filter on this system. In this section, we will show the difference between the Classic and Ensemble Kalman filtering methods as well as explain the benefits of using these techniques on this toy model, as well as our proposed model.

Classic Kalman Filtering

The Classic Kalman Filter is a special case of the Bayesian parameter estimation approach. The EnKF, as described in Section 2.2.3, is a more flexible extension of the Classic Kalman Filter. The Classic Kalman Filter assumes an initial Gaussian distribution for the state values of a system and utilizes linear stochastic equations to describe the states variables. There is no updating of the parameter values; instead the Classic sequentially estimates the state of the system and analytically computes the covariance of these state values at every time step t [18]. The EnKF, as previously mentioned, does not analytically compute the covariance but instead estimates the covariance matrix based on the mean of the ensemble members.

For the purposes of this paper, we did not utilize the Classic Kalman filter on our proposed model. We did, however, wish to highlight the difference between the Classic and EnKF on a well understood system like the damped oscillation system presented in Equations (2.17) and (2.18).To highlight the difference between the Classic and EnKF, we will look at the the linear stochastic equations needed for the classic filter. These can be written in terms of a coefficient matrix A and a state vector x, given as

$$\frac{dx}{dt} = Ax = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{b}{m} \end{bmatrix} \begin{bmatrix} p \\ v \end{bmatrix} \quad , \tag{2.20}$$

which is then discretized with the state evolution equation

$$x_{j+1} = (I + hA)x_j \quad . \tag{2.21}$$

This state evolution equation allows for the analytical computation of the state vector mean and covariance evolution over time. Beyond this, the Classic Kalman Filter does not have the capability to estimate state and parameter values simultaneously. Instead, it uses the initial assumed parameter values of m, k and b, along with the observed data at the previous time points t_j, to estimate the models state variables, p, v at t_{j+1} [1]. Figure 2.4 shows the results of the Classic Kalman Filter Estimation on the same data used for Nelder-Mead simulation.

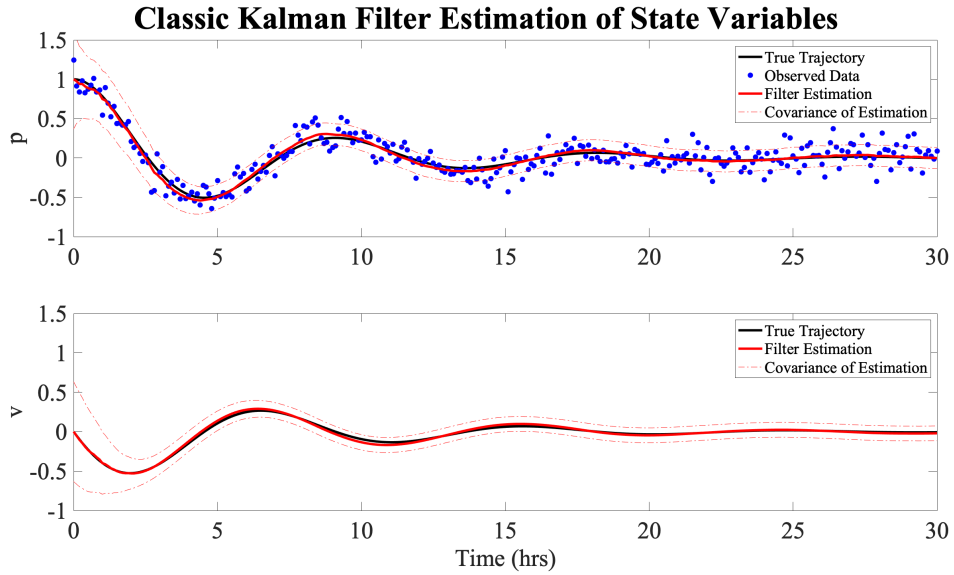


Figure 2.4: Results of the Classic Kalman Filter on the mass-spring toy model shown in Eq. (2.20). Results are based on the notion that there is no known data for the velocity state variable, v. Intake of observed data on the varying position, p, over time, along with the fixed initial parameter values, $[m\ k\ b] = [10\ 5\ 3]$. The system error covariance matrix, C = 0.001, and observation error, D = 0.0167. Note the difference between the Nelder-Mead and Classic graphs is the addition of the covariance estimations (dashed red line) around the mean estimate. *Top*: Prediction (solid red line) of the true solution (solid black line) of position state variable. The prediction is the mean of the underlying Gaussian distribution. *Bottom*: Prediction (solid red line) of the true value (solid black line) of velocity state variable.

Though it is important to have accurate information on the state variables, the very nature of most inverse problems demands inferences about all parts of the model. In our toy model as well as our cancer cell dynamics model we want to produce the Ensemble Kalman Filter as it will estimate state and parameter values, as well as estimating for nonlinear systems.

Ensemble Kalman Filter on Toy Model

As noted above, the Classic Kalman Filter is unable to estimate the state and parameter values simultanenously. In most instances, being able to do both estimations on linear and nonlinear systems is necessary. These instances can call for a variety of different parameter estimation methods, but our team decided on the Ensemble Kalman Filtering (EnKF) method.

See Section 2.2.3 for further details on the EnKF. This section will discuss the results of implementing EnKF on the mass-spring synthetically generated data described in Section 2.3.1, which has been used for both the Nelder-Mead and Classical Kalman examples. Along with the synthetic data we also need to generate initial ensemble members for all of our state and parameter values. Each state and parameter ensemble consists of $N = 100$ members. The initial p ensemble members are drawn normally from $\mathcal{N}(1, 0.1^2)$. The same process is conducted for the initial v ensemble members. The initial parameter ensemble members are all drawn uniformly from $U(9.5, 13.5)$, $U(2.5, 4.5)$, and $U(0, 4)$, for m, k and b respectively. It should also be noted that **ode45** is utilized when generating a prediction at $t = t + 1$. Figure 2.5 shows the state variable estimations generated by the EnKF algorithm, which should be compared to Figure 2.4 above.

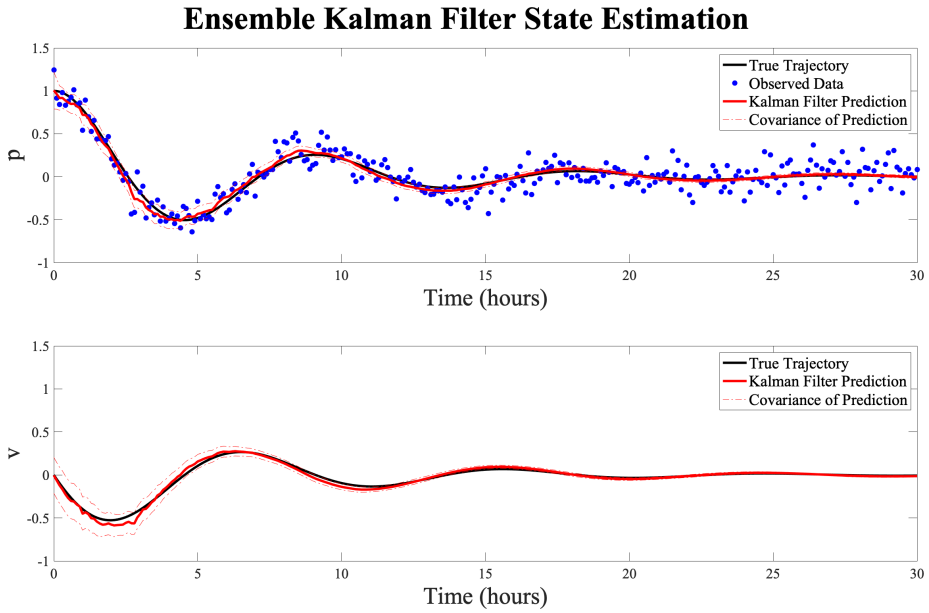


Figure 2.5: Results of the Ensemble Kalman Filter on the mass-spring toy model shown in Eq. (2.20). Results are based on the intake of observed data on the varying position, p, over time, along with the a uniform distribution of initial parameter values, $[m\ k\ b] = [10\ 5\ 3]$. The system error coefficient, C = 0.001, and observation error, D = 0.0167. *Top*: Prediction (solid red line) of the true value (solid black line) of position state variable. *Top*: Prediction (solid red line) of the true value (solid black line) of velocity state variable.

As previously stated, one of the major improvements on the Classic Kalman Filter in the EnKF is the addition of parameter distribution estimations. The estimation of the state variables, p and v are meant to act in the same manner for both algorithms. With this in mind, the state vector covariances of the EnKF are much tighter than those of the Classic. This shows that the EnKF produces a much more confident prediction.

After seeing that the EnKF is producing a confident estimation of the state variables, we can now look at the estimation of the parameters. These estimations are also accompanied by the measure of uncertainty, or the covariance at each time step. Figure 2.6 below shows the parameter estimation that accompanied the state variable estimation shown in Figure 2.5.

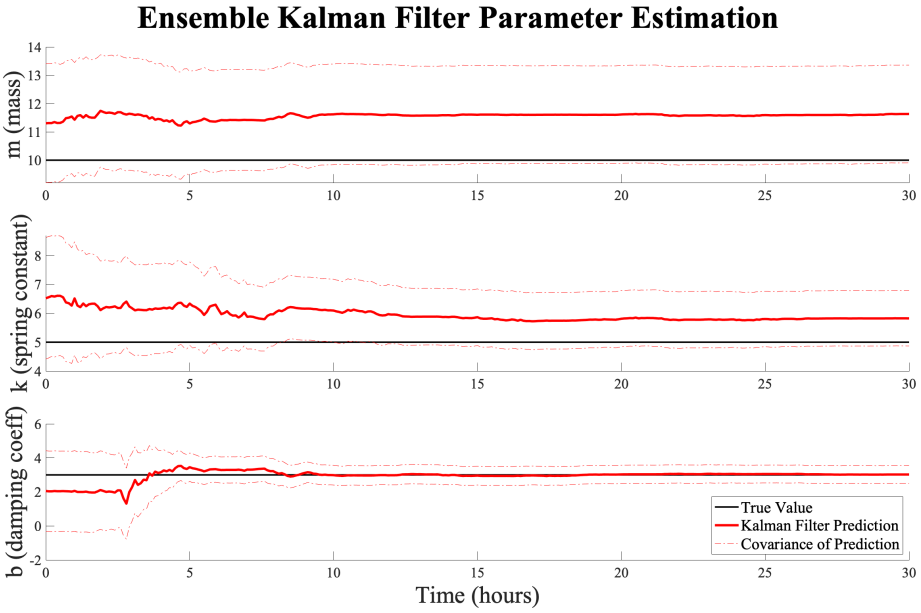


Figure 2.6: Estimation of parameter values is produced along with state estimations shown in Figure 2.5. The true parameter values are as follows, $[m\ k\ b] = [10\ 5\ 3]$. Final mean of the posterior distribution is as follows: $[m\ k\ b] = [11.6323\ 5.8232\ 3.0250]$. The prediction produced by the EnKF for each parameter is represented by the solid red line. The true value of the parameter is represented by the solid black line. *Top*: Estimation of parameter m, mass. *Middle*: Estimation of parameter k, spring constant. *Bottom*: Estimation of parameter b, damping coefficient.

Parameter estimation utilizing EnKF can provide insight into the dynamics of a model. EnKF is a reliable estimation technique, but it cannot always capture all parameters in a model. There are many different factors that come into play when a parameter cannot be properly captured, but the most common is there are deeper dynamics within the parameter, such as a time-varying component. In the case of the mass-spring model, the parameter m, the mass, can not be confidently captured by EnKF, indicating identifiability issues. The spring constant, k, can be captured with more confidence depending on the observation and system errors, while the damping coefficient, b, is accurately captured. All of these behaviors differ based on the dynamics of the system, but for the sake of this paper, we will not delve deeper into the possible further dynamics within our toy model.

Chapter 3

Experimental Data Collection and Synthetic Data Generation

3.1 Experimental Data

The data collected through experimentation was provided to our team by Dr. Mike Lee at UMass Medical School in Worcester, MA [16]. This data describes the interaction between different cancer drugs and a cancerous cell population. An example of the provided data is shown below in Figure 3.1.

Drug w/	Concentration	Sytox 0 hr	Sytox 4 hr	Sytox 8 hr	Sytox 12 hr	Sytox 16 hr	Sytox 20 hr	Sytox 24 hr	Sytox 30 hr	Sytox 48 hr
Drug 1:	0.0158	213	211	226	215	221	220	236	247	447
Drug 1:	0.05	220	229	230	231	236	244	255	256	414
Drug 1:	0.158	206	215	219	210	219	228	240	256	385
Drug 1:	0.5	198	211	216	220	253	432	763	1252	2198
Drug 1:	1.58	199	215	230	219	237	254	299	378	740
Drug 1:	5	195	193	208	241	333	634	1082	1528	2447
Drug 1:	15.8	196	199	227	248	336	493	840	1372	2270
Drug 1:	0.0158	218	213	229	222	243	252	250	264	419
Drug 1:	0.05	219	221	227	220	225	225	236	269	504
Drug 1:	0.158	202	223	230	221	233	242	262	323	620
Drug 1:	0.5	196	202	214	206	222	241	285	359	772
Drug 1:	1.58	191	197	212	208	239	308	423	605	1030
Drug 1:	5	198	207	214	225	300	435	727	972	1746
Drug 1:	15.8	195	191	211	254	321	435	690	1127	2028

Figure 3.1: Subset of experimental data provided to our team by UMass Medical School researchers. Data provided to our team includes three different cancer drugs at different concentrations. Two trials were conducted for each drug concentration. Columns labeled 'SYTOX hr' show the cumulative number of dead cells within the population at the specified hour. In other words, 'SYTOX 8 hr' is the cumulative number of dead cells at hour 8. For further visual of provided data, see Figure 1.4.

This data is broken down into four main facets: drug type and concentration; SYTOX counts at each hour interval; dead cell (at final SYTOX reading); and T_0 total cells. These four pieces were the main focus of the initial development of our model. To better understand the model developed, shown in Equations (1.8) and (1.9) we must first understand the data being used to measure the credibility of the model.

As was stated in Section 1.1.1, the type and concentration of cancer drug drastically impacts the dynamics within a cell population. Cancer inhibitors impact the cell proliferation cycle in different

ways, forcing a model to be dependent on which drug is active within the population. Dr. Mike Lee and associates repeated their testing on hundreds of different drugs with multiple trials for each drug and concentration pair. Our team was only provided with three common drugs with similar growth patterns. This largest, and most important, piece of the data is the SYTOX counts at the specified hour intervals. These points indicate the growth pattern of the dead cell population, C_{dead}.

SYTOX is a family of "nucleic acid staining dyes" [4] used to mark dead-cells within single-cell experiments. These dyes emit large quantities of fluorescence during the DNA binding process, while also maintaining a steady "signal-to-noise ratio", and providing colors across the color spectrum [4]. Though the specifics behind the dye's interaction within each cell are unnecessary for our model, it is still necessary to understand the mechanics behind data collection utilizing SYTOX dyes as an indicator of cell death.

Before the cancer treatments could be introduced to the population, the plate sat overnight for 24 hours to induce regular cell death. This meant the total number of cells at $t = 0$ consisted of approximately 200 dead cells. From here, each drug was injected into an individual plate with approximately the same number of initial cells as the other plates. The interactions between the cells and the cancer treatment were then monitored using these SYTOX dyes for 48 hours. The SYTOX hour measurements seen in the raw data in Figure 3.1 are the intervals at which the amount of fluorescence was evaluated; these totals were then recorded for further exploration. These SYTOX readings were taken every 4 hours, for $0 \leq t \leq 24$, then inconsistently for $24 < t \leq 48$. From here, the final count of the dead cells, d_{final}, within the population was recorded as the number of dead cells present at $t = 48$.

Once d_{final} was recorded, the experiment conductors assumed exponential growth, of the dead cells, as well as the total number of cells in the population. This assumption was used to calculate the initial number of cells within the population. This information was gathered, and a collective average was found to be $T_0 \approx 2209.97$. From here, the initial number of dead cells , $C_{dead}(0) \approx 200$ was subtracted from T_0 to get $C_{live}(0) \approx 2000$.

3.1.1 Finding Reasonable Initial Parameter Values

Before we could run our parameter estimation techniques on our proposed model, we needed to understand the potential ranges of our parameters, α, β and γ, on our observed data. For the purposes of this paper, we generated our synthetic data based on the ranges for only one of the three cancer treatments' dead cell data. These ranges are treatment dependent, as each treatment induces a different dead cell growth pattern as seen in Figure 1.4.

The potential ranges for each parameter were found by fixing values for two parameters, using the initial values of $C_{live}(0) = 2000$, $C_{dead}(0) = 200$, then graphing the solutions to Eq. (2.17) and (2.18) with a vector of values for the third parameter. These solutions were then compared to our observed data. This was conducted for each parameter. Figure 3.2 below is an example of the graph produced for each drug dosage.

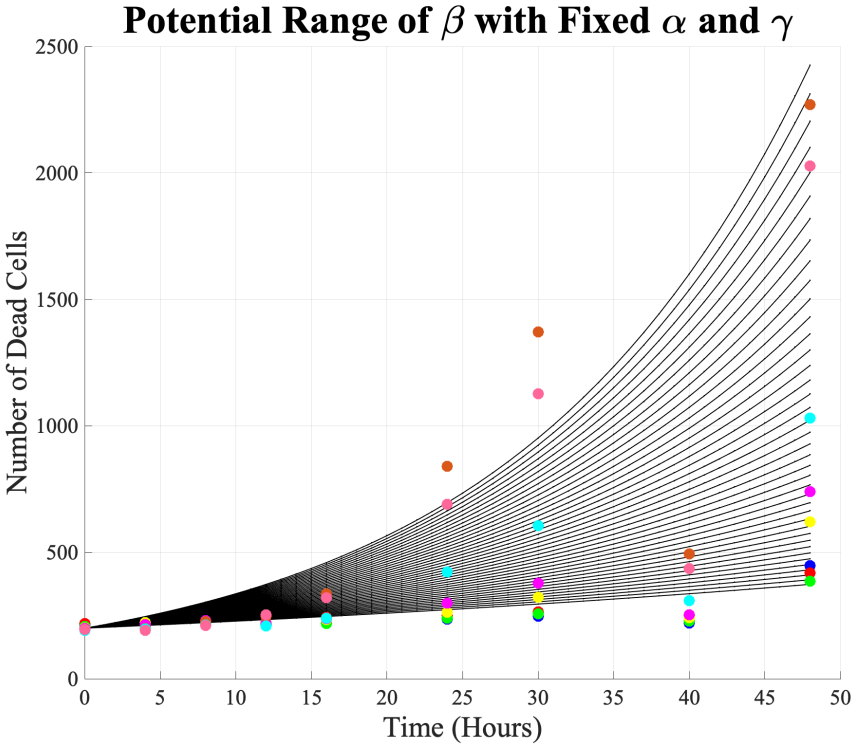


Figure 3.2: Example outcomes for range of cell-induced death parameter β. Solutions are the black lines with initial conditions $C_{live}(0) = 2000$, $C_{dead}(0) = 200$, $\alpha = 0.01$, and $\gamma = 0.001$. The tested range for $\beta \in [0.0130.052]$. Dots are the cumulative number of dead cells for Drug 1. All data shown in Figure 3.1 is represented by points on this image. Each color represents a concentration and trial for Drug 1.

This initial simulation indicated a positive cell-induced death rate, β, between 0.013 and 0.052 could fit this specific drug and dosage. These initial tests were meant to provide us with a gauge on reasonable death rate values for our future iterations of the model. This also is the starting point for generating synthetic data that mimics the raw data.

Though the ranges found above are helpful, they do not show the impact of dynamically changing all three parameters, α, β and γ, simultaneously on the fit of the model to the observed data. To solidify our findings from the experimental range shown in Figure 3.2, we also implemented the built-in MATLAB® function **fmincon**. As shown in Section 2.3, the Nelder-Mead algorithm can produce an accurate depiction of a set of data. These particular ranges were used as a reasonable range of parameter values that would fit our data.

We implemented the constrained **fmincon** to minimize the residual between the model and our provided data points and restrict our parameter values to $\alpha, \beta, \gamma \geq 0$. Figure 3.3 is the outcome of using the model presented in (1.8) and (1.9) on the first of our three cancer treatments.

At first glance, we can see that the model prediction using **fmincon** estimates parameter values that correspond to decreased accuracy in state variables at latter time points. The sparsity of our data or the inability of the model prediction with the **fmincon** optimized parameter values to fully represent the dynamics of our model could be factors in this potential discrepancy.

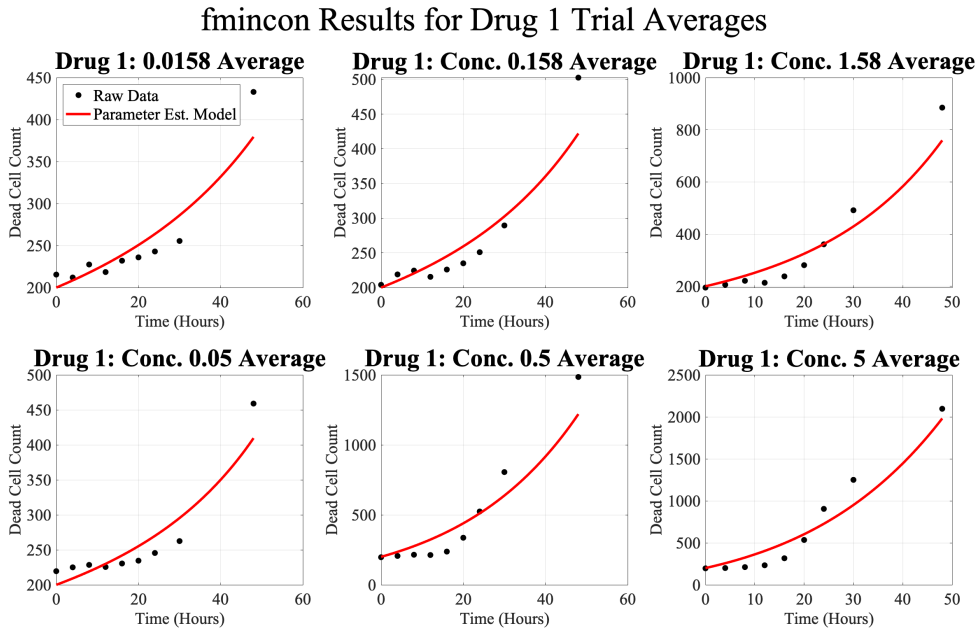


Figure 3.3: Graphs of the fmincon results for Drug 1. The optimal set of parameters for each drug is used to solve the model (Eq. (1.8) and (1.9)). All models have the same initial conditions, $C_{live} = 2000, C_{dead} = 200, \alpha = 0.01, \beta = 0.01$ and $\gamma = 0.01$. The initial guesses of the parameter values were found through the experimentation depicted in Figure 3.2. Each graph represents the average between the two trials of Drug 1 at each concentration. The solution to the model is then graphed in red, compared to the true data points, black dots.

To back up this thought, we need to test the capabilities of **fmincon** with our model. We did this by generating synthetic data with known input parameter values, running this generated data through **fmincon**, then comparing the optimized estimation to our known input. An example of this process is shown in Figure 3.4 below.

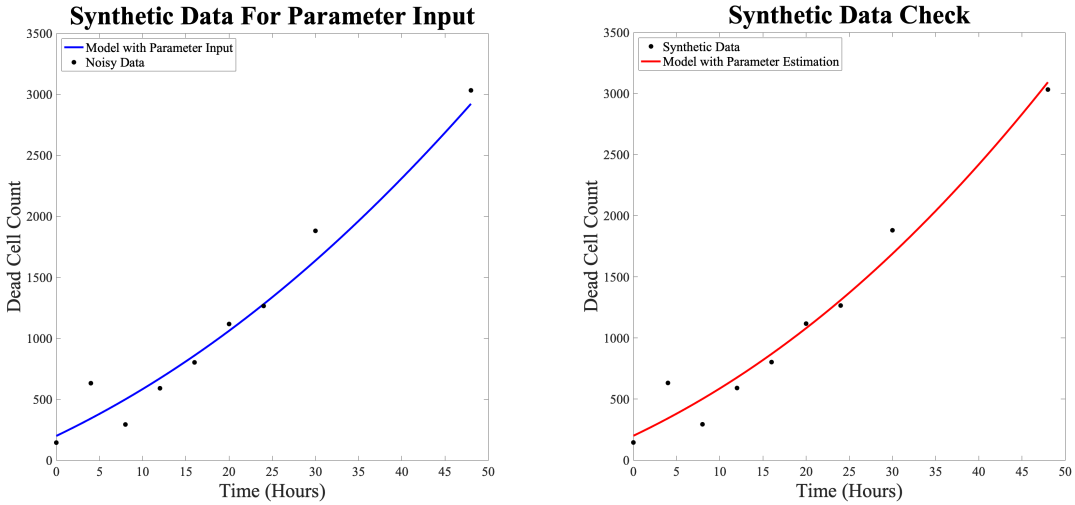


Figure 3.4: *Left*: Synthetic data generated using Equations (1.8) and (1.9) with parameter Values:$[\alpha \ \ \beta \ \ \gamma] = [0.023 \ \ 0.02 \ \ 0.015]$. *Right*: Parameter estimation utilizing synthetic data to check output values: $[\alpha \ \ \beta \ \ \gamma] = [0.0209 \ \ 0.0248 \ \ 0.0144]$.

This test, along with looking at further parameter input values, supports our hesitancy with using **fmincon** for our data and model. The input parameter values are relatively close to our **fmincon** estimation, but this can not be said for other data sets that we were provided. Specifically, running **fmincon** on the provided SGI-1027 data did not generate a curve that fit the shape of our data. While the majority of our trial data was easily fit by our data, the dynamics within the SGI-1027 data indicated the need for either another parameter estimation mechanism or a shift in our initial model by adding more dynamics, such as time-varying parameters or interaction terms.

3.1.2 Synthetic Data Generation for Kalman Filter Estimations

Constant Growth Parameter

The data provided to our team was collected through a UMass Medical School experiment that looked solely at the number of dead cells within a controlled environment. This data was collected at varying time points over a 48 hour interval. We generated synthetic data by setting our initial state variable values, $C_{live}(0) = 2000$ and $C_{dead}(0) = 200$, equal to the provided experimental data at time $t = 0$. We also initialized the parameter values, $\alpha = 0.01, \beta = 0.01$ and $\gamma = 0.001$ based on our final results of the average parameter estimations using **fmincon**. Observation noise was added to C_{live} and C_{dead} separately in proportion to their respective standard deviations. Figure 3.5 shows the set of synthetic data generated using our model with constant parameters.

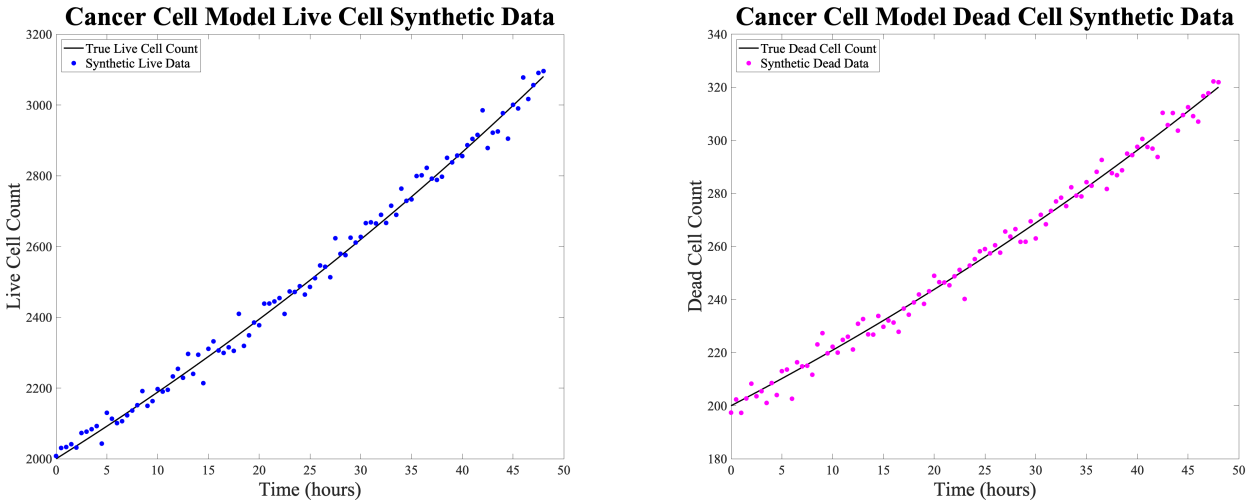


Figure 3.5: *Left*: Synthetic data generated for C_{live} using Equations (1.8) and (1.9) with initial values $C_{live}(0) = 2000$ and $C_{dead}(0) = 200$, and parameter values:$[\alpha\ \beta\ \gamma] = [0.01\ 0.01\ 0.001]$. Noise added to data in proportion to the standard deviation of the true trajectory of C_{live}. *Right*: Synthetic data generated for C_{dead} using same conditions as synthetic data for C_{live}. Noise added to data in proportion to the standard deviation of the true trajectory of C_{dead}.

Along with our synthetic observation data, we generate an initial ensemble of state and parameter values. Our ensemble consists of $N = 100$ ensemble members for each state and parameter variable. The initial C_{live} ensemble members are equivalent to $C_{live}(0) = 2000$ plus 10% of Gaussian noise drawn normally from $\mathcal{N}(0, 1)$. The same process is conducted for the initial C_{dead} ensemble members. The initial parameter ensemble members are all drawn uniformly from $U(0.015 - 0.02, 0.015 + 0.02)$, $U(0.015 - 0.02, 0.015 + 0.02)$, and $U(0.005 - 0.02, 0.005 + 0.02)$ respectively. It should also be noted that **ode45** is utilized when generating a prediction at $t = t + 1$.

Time-Varying Growth Parameter Forms

Though a constant growth parameter is a reasonable assumption for the provided data, the possibilities of further dynamics based on onset times of drugs may induce a time-varying alpha, $\alpha(t)$. The dynamics that could be captured within this time-varying growth term include a delay in drug onset time which is usually present in systems such as this one [21]. Though there are a variety of possible time varying growth parameters, our team explored three forms of $\alpha(t)$,

$$\alpha(t) = \alpha_0 \frac{1}{1 + e^{\tau t - s}} \tag{3.1}$$

$$\alpha(t) = \alpha_0 - mt \tag{3.2}$$

$$\alpha(t) = \begin{cases} \alpha_0 & t \leq t_1 \\ \alpha_1 & t > t_1 \end{cases} \tag{3.3}$$

as previously mentioned in Section 2.1 (Figure 2.1). These three functional forms of $\alpha(t)$ were chosen as they all integrate the natural slowing of growth within the population due to limitations of resources or cells being placed into the dead cell population. Looking at each $\alpha(t)$ we introduce new hyperparameters that have a reasonable range of possible values. For Equation (3.1), two constant parameters, τ and s are introduced; where τ impacts the speed at which $\alpha(t)$ changes over time, and s is a shift term that introduces the time, t, at which $\alpha(t)$ starts to change. For Equation (3.2), the slope, m, of the line is considered as a constant parameter. Finally, in Equation (3.3) we see the addition of α_0, α_1 which are the two values of a step function that changes at time t_1. Though all of these functional forms vary in their shape, they all share the common constant,

$$\alpha_0 = \frac{ln(2)}{18} \tag{3.4}$$

which was chosen based on an average cell division time of 18 hours in the absence of drug-induced cell death [8].

Equations (3.1), (3.2), and (3.3) are all based on the natural assumption that the growth of live cells will slow over time. To accommodate these variations of a time-varying growth parameter, our team implemented each of the three forms into our proposed model,

$$\frac{dC_{live}}{dt} = \alpha(t)C_{live} - \beta C_{dead} - \gamma C_{live} \tag{3.5}$$

$$\frac{dC_{dead}}{dt} = \beta C_{dead} + \gamma C_{live} \tag{3.6}$$

and generated new sets of synthetic data for each of the functional forms. The procedure described for the constant parameter model was utilized to generate synthetic data for each time-varying parameter.

Specifically, our team uses the following values for all hyperparameters introduced for each $\alpha(t)$:

Sub-Parameter	Value
τ	0.14
s	5
m	$\alpha_0 - 0.003$
α_1	0.02

The dynamics of the system vary based on which form on $\alpha(t)$ is chosen. This implies that the real values of the new parameters introduced within the different forms were calculated to mimic the data provided by the UMass Medical School experiments.

Chapter 4

Numerical Results

In this section, we demonstrate the effectiveness of the Ensemble Kalman Filter (EnKF), on our cancer cell model as well as provide information on how constant versus time-varying growth parameters affect the filter's ability to capture possible dynamics of the system. As a reminder, our proposed cancer cell model is given by the following coupled system of differential equations:

$$\frac{dC_{live}}{dt} = \alpha C_{live} - \beta C_{dead} - \gamma C_{live} \tag{4.1}$$

$$\frac{dC_{dead}}{dt} = \beta C_{dead} + \gamma C_{live} \tag{4.2}$$

with three constant parameters, α, β and γ. The α parameter in (4.1) will be modified to accommodate a time-varying growth parameter, $\alpha(t)$, as shown in Equations (3.5) and (3.6).

The performance of the EnKF on a system with constant parameters is much different than that of a system with time-varying parameters. It should be noted that when considering a constant parameter, the EnKF is producing a sequentially updated prediction of that constant parameter value. When the EnKF is estimating a time-varying parameter, however, the algorithm attempts to capture the movement or path of this parameter over time. This implies the EnKF predictions made for these two parameter types should be considered differently.

Presentation and Supporting Factors for Results

Before we can present our results, our team will first introduce the tools used throughout this section to support our claims. These tools include using combinations of the system error matrix $\mathbf{C}$, and the observation error matrix $\mathbf{D}$ within the filter. As used in Equation (2.7) through the addition of the v_{j+1}^n to generate the prediction ensemble, the system error matrix $\mathbf{C}$, is a diagonal matrix with hand-picked values meant to increase or decrease the filters trust in its own prediction. The observation error matrix $\mathbf{D}$, is a diagonal matrix with values equal to a proportion of the observation noise for C_{live} and C_{dead}, as shown in Eq. (2.14). Using different combinations of the system and observation noise changes the ability of the filter to estimate the provided data. Finding combinations of these errors that produce reasonable results depends on the dynamics of the model and is often the most difficult component of working with the EnKF. There are more combinations of $\mathbf{C}$ and $\mathbf{D}$ values than can be tested in a timely and inexpensive fashion. Thus, instead researchers rely on showing a certain number of combinations can result in the same general behavior of the EnKF for the given system.

In our case, we highlight four possible combinations of **C** and **D** matrices, and form conclusions based on if these combinations result in the filter being able to fully capture the model dynamics or not. For the rest of this chapter we will denote the system and observation errors as follows,

$$\mathrm{C} = \left[c_1;\ c_2\right], \mathrm{D} = \left[d_1;\ d_2\right] \tag{4.3}$$

which are the diagonal values for each matrix. Specifically, we say that c_1 and d_1 are the system and observation error respectively for state variable C_{live}. The system and observation errors used to shift C_{dead} are represented by c_2 and d_2.

Another major tool we use to show our results are the EnKF state and parameter estimation graphs. The synthetic data generated through process described in Section 3.1.2, has a set of known parameter values that produce a true trajectory of the state and parameter values under our proposed model (1.8)– (1.9) with no simulated noise. This true trajectory is depicted on the state and parameter estimation graphs used throughout this section as a solid black line. These true trajectories are used an internal checking mechanism for the EnKF predictions, depicted as a solid red line on the graphs. These predictions are the mean of the posterior distribution for all state and parameter values. True trajectories show our team if the EnKF is producing predictions that follow the known values of the model that produced our synthetic data. If the EnKF produces results that do not follow the true trajectory of the model and parameter values, we will need to either adjust our model as it may not capture the dynamics of the data, or adjust the internal mechanisms of the EnKF to accommodate the noise in the data.

Finally, the state and parameter estimation graphs will show the confidence measure of the prediction over time, which is depicted as the mean plus or minus twice the standard deviation of the posterior distribution. We hope to see this confidence measure tighten up over time by converging to the known, true values of the state and parameters. If, however, the confidence values become wider around the mean, it is implied that the given set of observed points may not provide sufficient information for an accurate and confident estimation. We use histograms of the predicted distributions to show whether or not the prediction is converging to the true state and parameter values.

4.1 Parameter Estimations and their Confidences

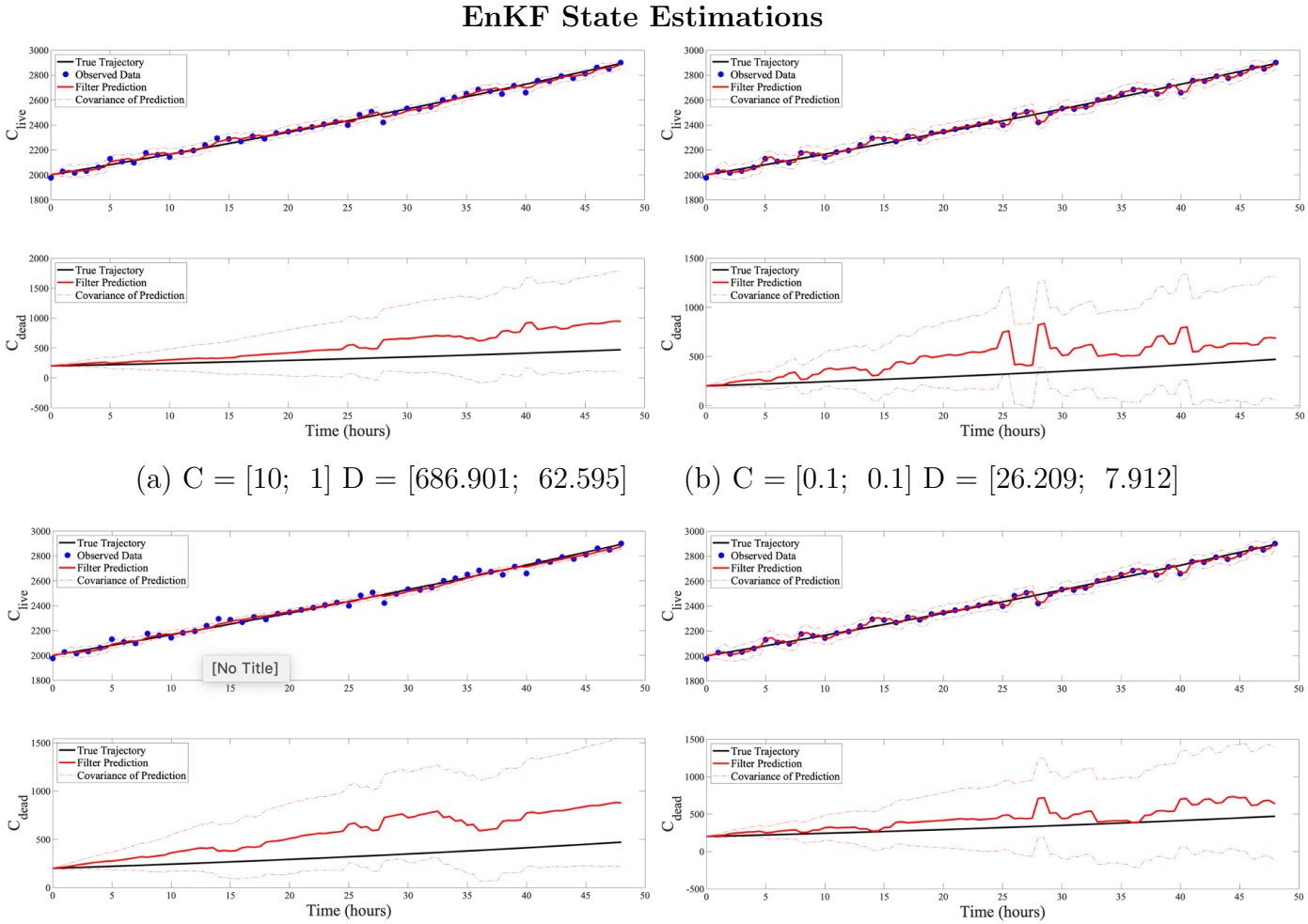
The first question our team wanted to address when using the EnKF algorithm on our model focuses on what parameters can be confidently estimated when certain data observations are collected. As previously mentioned, the majority of studies done in the field will collect data on live cell counts, while UMass Medical School researchers provided our team with data for dead cell counts over time, and an initial and final value for live cell counts. By focusing data collection on the growth patterns of the dead cell population, these researchers hope to show the relationship between the live and dead cell counts through a new lens [13]. There is currently no literature suggesting one data collection method provides researchers better insight into the underlying dynamics between cancer cells and treatments than another.

Our team utilized the EnKF algorithm as a way to test the various data collection methods used through the literature. In this section, we present our findings regarding sufficient collection of state variable data that allow for confident estimations of the unknown parameter values within our proposed cancer cell model.

4.1.1 Estimations with Live Cell Counts

Our first inquiry was into whether the live cell only data collection strategy would allow our model to accurately estimate state and parameter values. This strategy is widely used throughout the field, but does it provide the foundation for a confident estimation?

Figure 4.1 shows the EnKF estimates of the state variables when given live cell count data at every hour. The estimation of the live cell count is confident throughout the entirety of the time series, but the confidence in the dead cell estimation decreases over time. The cone shaped range of covariance, the red dotted lines on Figure 4.1, indicate a lack of confidence in the estimation. This pattern is apparent for a variety of combinations of system and observation errors in the same format as shown in Equation (4.3).



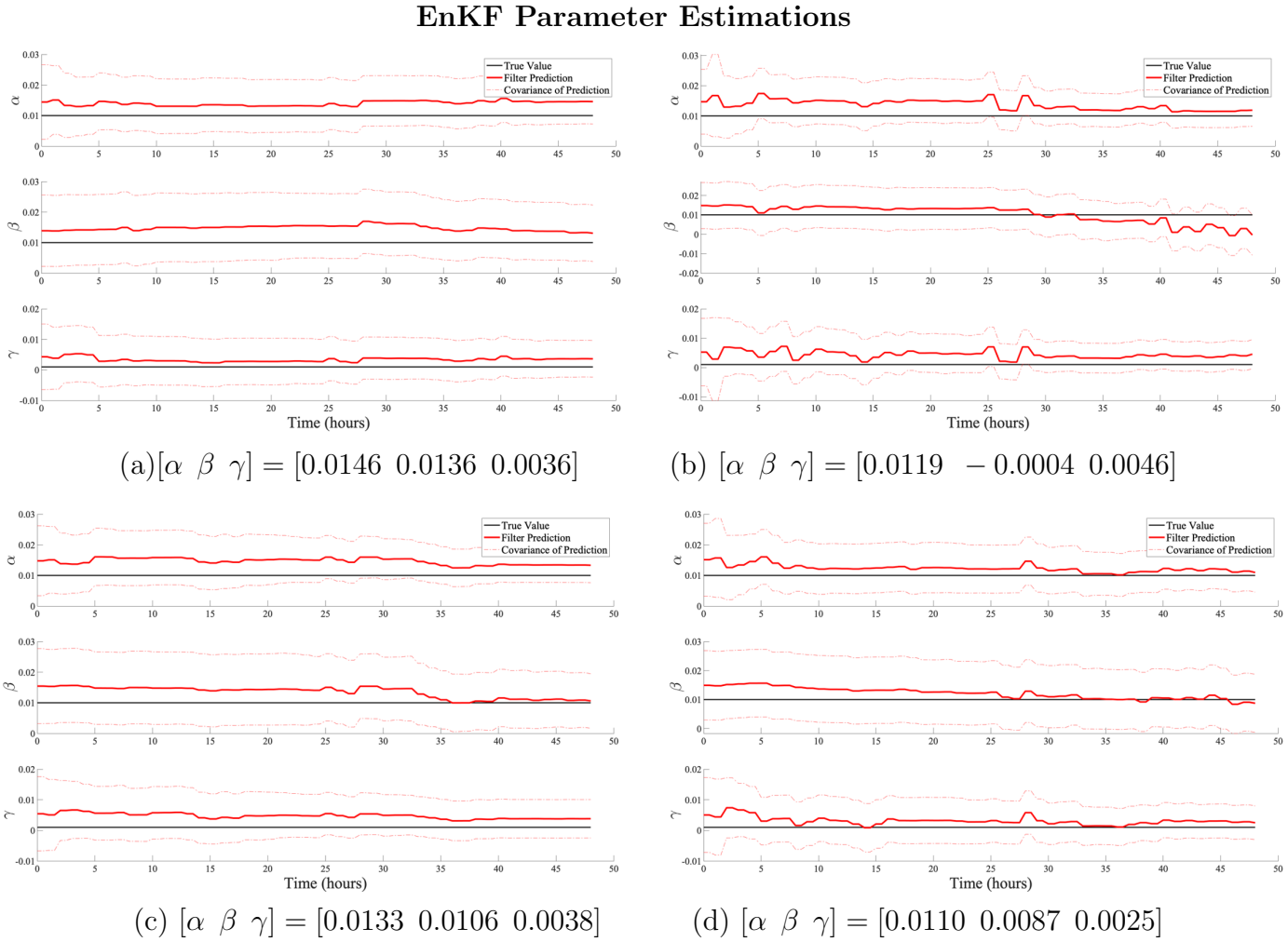
(a) C = [10; 1] D = [686.901; 62.595] (b) C = [0.1; 0.1] D = [26.209; 7.912]

(c) C = [0.1; 0.1] D = [686.901; 62.595] (d) C = [10; 1] D = [26.209; 7.912]

Figure 4.1: Graphs of EnKF estimation of both state variables, C_{live} and C_{dead}, when observations are collected every hour for live cell count, and no observations are collected for dead cell count. Each graph shows the results for a different combination of **C** and **D** values, shown underneath each image. Accompanying parameter estimations can be seen in Figure 4.2 with the same letter beneath the graph.

Along with not being able to confidently estimate the number of dead cells over time, Figure 4.2 indicates none of the parameters, α, β, or γ can be confidently captured. Though the covariances do not have the cone shape seen in Figure 4.1, the covariances do not converge onto the solid red prediction

line. This pattern is seen with the combinations of system, C, and observation, D, errors for each estimation. Figure 4.2 shows the various parameter predictions, solid red line, as compared to the true value, solid black line, for each combination of the system and observation errors shown in Figure 4.1 respectively.



(a)$[\alpha \ \ \beta \ \ \gamma] = [0.0146 \ \ 0.0136 \ \ 0.0036]$ (b) $[\alpha \ \ \beta \ \ \gamma] = [0.0119 \ \ -0.0004 \ \ 0.0046]$

(c) $[\alpha \ \ \beta \ \ \gamma] = [0.0133 \ \ 0.0106 \ \ 0.0038]$ (d) $[\alpha \ \ \beta \ \ \gamma] = [0.0110 \ \ 0.0087 \ \ 0.0025]$

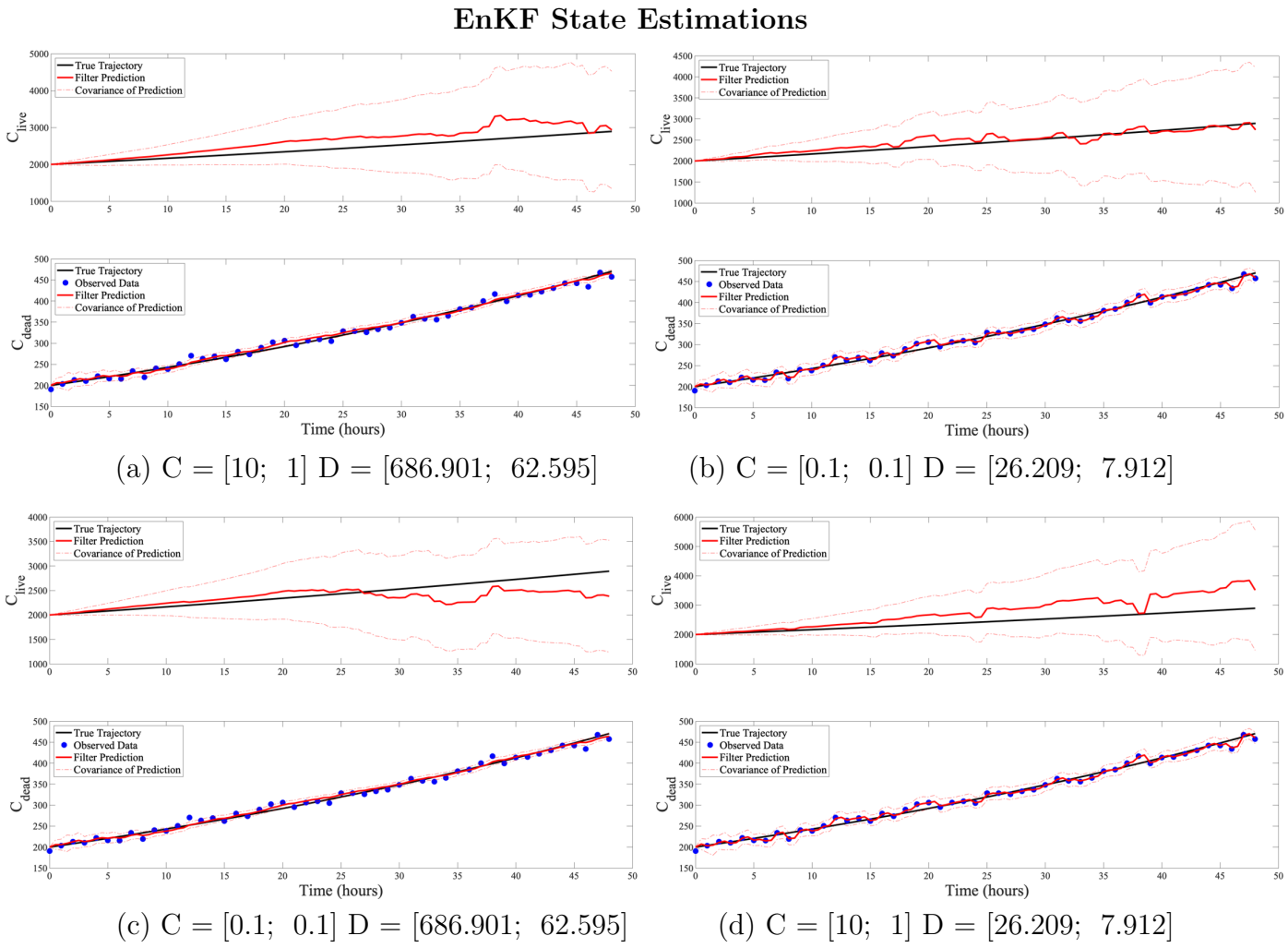
Figure 4.2: EnKF estimation of all parameter variables, α, β and γ, when observations are collected every hour for live cell count, while no observations are collected for dead cell count. Figures (a), (b), (c) and (d) correspond to the state variable estimations of Figures (a), (b), (c) and (d) in Figure 4.1 respectively. Caption under each graph indicates the mean of the final posterior ensemble, known as the EnKF estimation of the parameter values at time $t = 48$. All should be compared to the true values of the parameters, the solid black lines: $[\alpha \ \ \beta \ \ \gamma] = [0.01 \ \ 0.01 \ \ 0.001]$.

It should be noted that though the solid red prediction line in Figure 4.2 is following the true value line, the covariance shows more about the filters confidence in this prediction. The prediction itself is relatively close to the true value because the initial parameter ensemble is normally distributed close to the true value. This initial ensemble provides the filter with a starting point close to the true value, allowing it to stay within this window. If it did not stay within this window, then we would know that the EnKF is not effectively estimating any parameters for this model. The wide and non-converging confidence around the EnKF prediction suggests is not learning anything about the parameter values and is relying on the initial ensemble to maintain the current values of the parameters.

This information indicates that when implementing the EnKF algorithm on our proposed cancer model, we can not confidently estimate the state or parameter values when only provided data on live cell counts. This supports the claim made by the UMass researchers providing our team with data, that to understand the relationship between live cell growth and cell death when cancer cells are introduced to treatment methods, one must not only observe the live cell population [13].

4.1.2 Estimations with Dead Cell Counts

Now that we know live cell observations alone will not produce a confident estimation using EnKF, we next needed to test the algorithm with only dead cell observations. The same procedure implemented on live cell simulations was implemented for dead cell simulations. Figure 4.3 shows the results of running EnKF with the same combinations of system and observation errors.



(a) C = [10; 1] D = [686.901; 62.595]

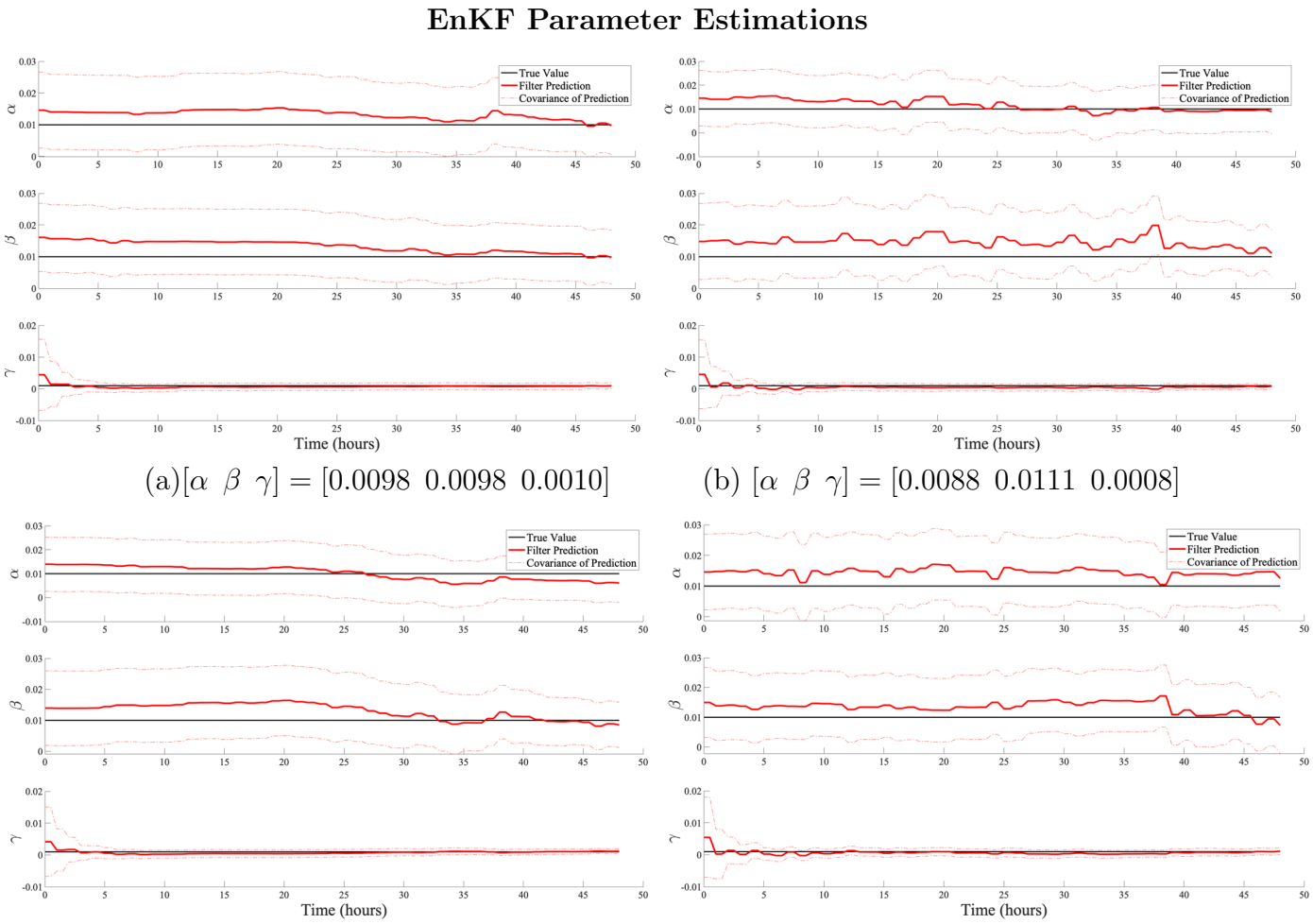
(b) C = [0.1; 0.1] D = [26.209; 7.912]

(c) C = [0.1; 0.1] D = [686.901; 62.595]

(d) C = [10; 1] D = [26.209; 7.912]

Figure 4.3: EnKF estimation of both state variables, C_{live} and C_{dead}, when observations are collected every hour for dead cell count, while no observations are collected for live cell count. Each graph shows the results for a different combination of **C** and **D** values, shown underneath each image. All exemplify the same pattern of EnKF algorithm not producing a confident or accurate estimation of C_{live}.

As suspected, when provided data on just dead cell counts the EnKF is unable to confidently capture the unobserved, live cell, state variable as time goes on. While the unobserved state variables act in a similar fashion for both live and dead observed, the parameter estimations have a different behavior. Figure 4.4 shows that although α and β are still not estimated confidently, γ can be estimated with increased confidence. Again, this behavior is consistent for a variety of different tested combinations of the observation and system errors in the algorithm.



(a)$[\alpha \ \ \beta \ \ \gamma] = [0.0098 \ \ 0.0098 \ \ 0.0010]$ (b) $[\alpha \ \ \beta \ \ \gamma] = [0.0088 \ \ 0.0111 \ \ 0.0008]$

(c) $[\alpha \ \ \beta \ \ \gamma] = [0.0061 \ \ 0.0085 \ \ 0.0012]$ (d) $[\alpha \ \ \beta \ \ \gamma] = [0.0126 \ \ 0.0073 \ \ 0.0011]$

Figure 4.4: EnKF estimation of all parameter variables, α, β and γ, when observations are collected every hour for dead cell count, while no observations are collected for dead cell count. Each graph corresponds to the state variable estimations shown in Figure 4.3 sharing the same letter below the image. All exemplify the same pattern of the EnKF algorithm being able to produce a confident estimation for γ, while the estimations for α and β are less confident.

The filter being able to accurately capture one parameter in our model under the given circumstances indicates that our model can be used to describe the dynamics of the system. It also suggests that further data points may be necessary to capture more parameter values. Our goal is to generate a model that can accurately capture the dynamics of the system, thus we we want produce confident estimates for as many parameters as possible. This implies that, though better than just live data, just dead cell data is not sufficient in estimating state or parameter values.

4.1.3 Estimations with Both Live and Dead Cell Counts

Now that we know having just dead or just live cell counts makes for an ineffective estimation of states or parameters, we need to consider cases when we have information on both cell counts. Figure 4.5 shows the estimations of the state variables, C_{live} and C_{dead}, over time. We see when data is collected for both live and dead cells at every hour from $0 \leq t \leq 48$, the algorithm produces state variables estimations with much more confidence. The algorithm has much more information to correct its estimation over time and is able to maintain a steady, tight covariance for both live and dead. The frequency of which these pieces of data is necessary to obtain this tightening covariance measure will be discussed in Section 4.2.

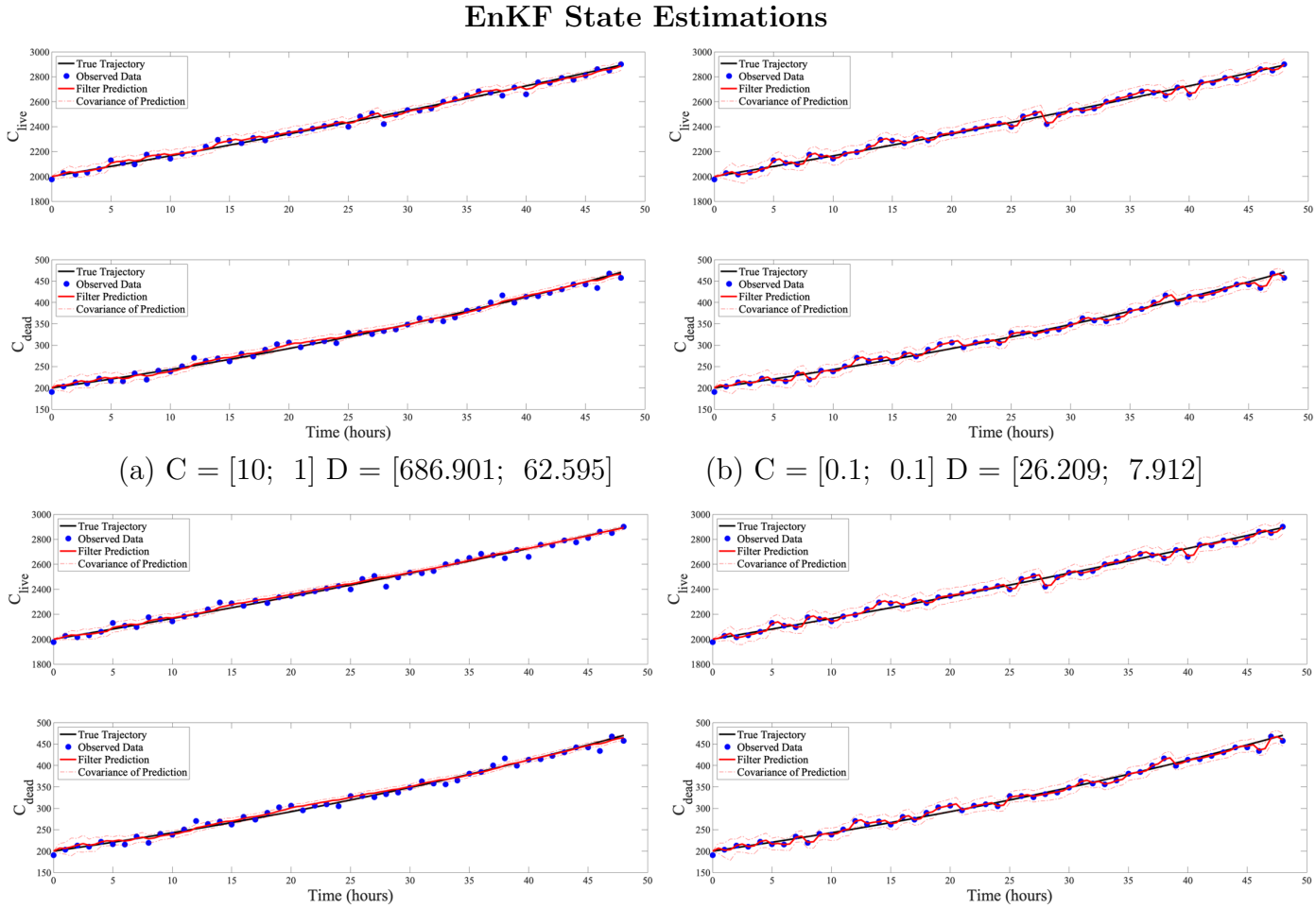
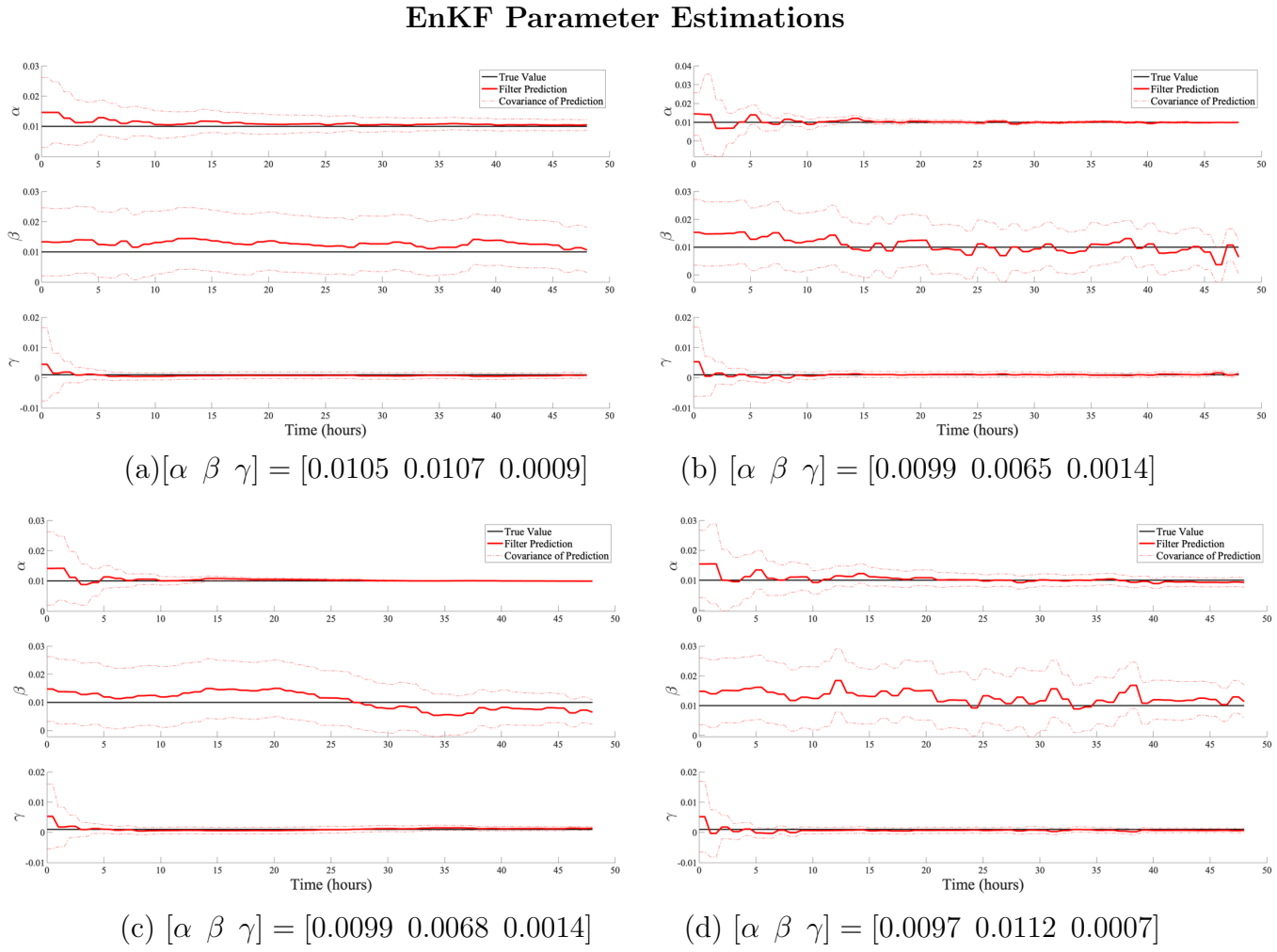


Figure 4.5: EnKF estimation of both state variables, C_{live} and C_{dead}, when observations are collected every hour for both live and dead cell count. Each graph shows the results for a different combination of **C** and **D** values, shown underneath each image. All exemplify the same pattern of EnKF algorithm producing accurate and confident estimations of both C_{live} and C_{dead}.

Figure 4.6 shows the parameter estimations that pair with the state estimations shown in Figure 4.5. We see the filter produces estimations for two of the three parameters, α and γ, that converge to their true values with covariance measures tightening around the prediction. This implies that not only is the filter capturing the true known value, but the filter itself is very confident that this estimation is true. Thus, when noisy data, without the parameter values being known, is used as an input for the

filter, the estimation of the parameters should fit the data.

With this in mind, we do not see the same behavior being exemplified by the β parameter. Though, with the given sets of system and observation noise, the filter is producing an estimation close to the true values, the covariances are not tightening around the prediction. Figure 4.6 below shows that when we add data for both live and dead, the filter is now able to confidently and accurately capture α and γ.



(a)$[\alpha \ \ \beta \ \ \gamma] = [0.0105 \ \ 0.0107 \ \ 0.0009]$

(b) $[\alpha \ \ \beta \ \ \gamma] = [0.0099 \ \ 0.0065 \ \ 0.0014]$

(c) $[\alpha \ \ \beta \ \ \gamma] = [0.0099 \ \ 0.0068 \ \ 0.0014]$

(d) $[\alpha \ \ \beta \ \ \gamma] = [0.0097 \ \ 0.0112 \ \ 0.0007]$

Figure 4.6: EnKF estimation of all parameter variables, α, β and γ, when observations are collected every hour for both live and dead cell counts. Each graph corresponds to the state variable estimations shown in Figure 4.5 sharing the same letter below the image. Caption under each graph indicates the mean of the final posterior ensemble, known as the EnKF estimation of the parameter values at time $t = 48$. All should be compared to the true values of the parameters, the solid black lines: $[\alpha \ \ \beta \ \ \gamma] = [0.01 \ \ 0.01 \ \ 0.001]$.All exemplify the same pattern of EnKF algorithm being able to confidently capture α and γ, while not being able to capture β.

When the EnKF is unable to capture a parameter this means one of two things; the parameter has more dynamical components that require the parameter to not be considered constant, or the parameter has little influence on the system as a whole. For a closer look at if the EnKF is producing an accurate and confident estimate of the β parameter, we must look at the posterior distributions compared to the prior for each parameter. When looking at the final posterior distributions of each parameter, we can

conclude the EnKF is not confidently capturing the value of the β parameter, resulting in identifiability issues.

To further support the notion that the EnKF can not capture the β parameter due to identifiability issues, we must compare the prior and final posterior distribution of all three parameters. This comparison will allow us to visualize if the filter is learning about the parameters or not. Though we believe all combinations of the system and observation errors produce the same behavior, we only present the prior and posterior distributions for the simulation depicted in Image (c) in Figures 4.5 and 4.6. These comparisons of the prior and posterior distributions for the three constant parameters α, β and γ can be seen in Figure 4.7 below.

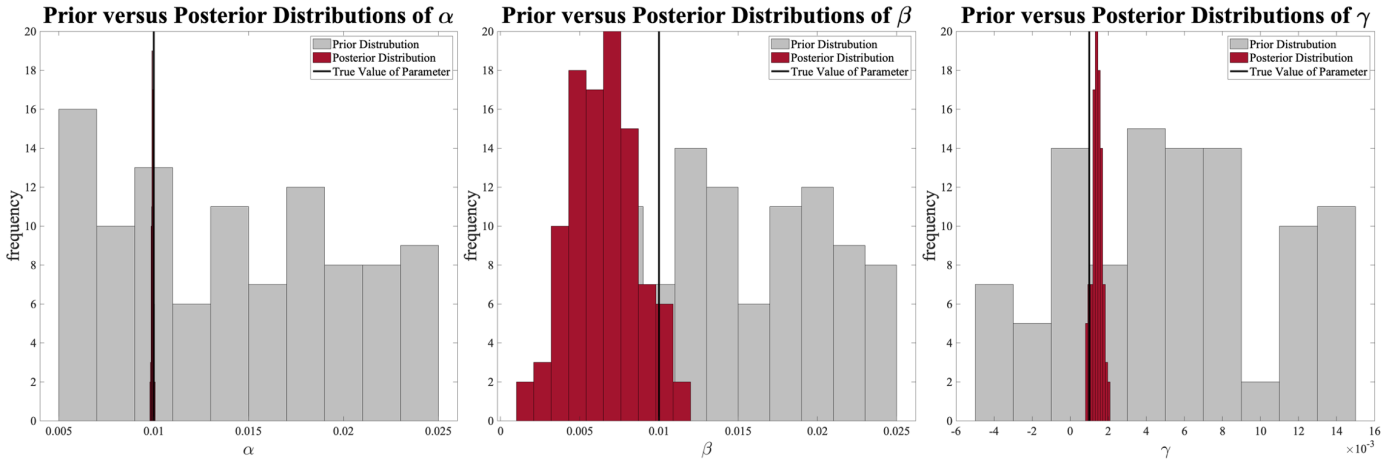


Figure 4.7: Histogram of the prior (initial, shown in grey) and posterior (final, shown in red) distributions of the ensemble for each parameter α, β and γ. For a confident parameter estimation, the posterior distribution should be very tight together. For an estimation to be deemed accurate, the bars of the histogram should have a mean at the true values of the parameter (black line). These graphs show that the posterior distribution of the β parameter (middle graph), indicates that final estimation is neither accurate, as the the peak of the distribution is not close to the true value, nor confident, as the distribution is wider than that of α or γ.

From Figures 4.7 and 4.6, we can say β is not fully identifiable under the given conditions. Though the EnKF does start to converge to the true value of β, the prediction still lacks confidence. Because of this, our team decided these numerical experiments indicate β is not fully identifiable and in remaining experiments we fix $\beta = 0$. The other option for β having more dynamics than a constant parameter can capture will not be explored in this paper, but our team does suggest running simulations with a potentially time-varying β parameter.

For the sake of the paper, from this point on, any reference to our proposed model will be referring the system with $\beta = 0$:

$$\frac{dC_{live}}{dt} = \alpha C_{live} - \gamma C_{live} \ , \tag{4.4}$$

$$\frac{dC_{dead}}{dt} = \gamma C_{live} \ . \tag{4.5}$$

4.1.4 Takeaways

Through these initial EnKF trials our team has identified two main takeaways. First, the natural death parameter, β, cannot be confidently estimated through the EnKF filter. Thus, our team chose to set $\beta = 0$ for future numerical experiments. Second, to confidently estimate state and parameter values, there needs to be observational data provided for both live and dead cell counts.

With the initial findings outlined throughout this section, our team now will take a deeper look into the frequency at which observational data for both live and dead cell counts should be collected to maintain the confidence and accuracy of the EnKF algorithm.

4.2 Time of Data Collection

In a perfect world, we would always have every piece of information needed to solve a problem. Though it would make most problems easier, there are rarely opportunities when all the information is available and more often than not, estimation based on what is known is necessary. Known observations and values are key components in model development and understanding. Any information that can be assumed or known about a model can provide an internal checking mechanism for predication accuracy.

Unlike the majority of the literature, the data provided for this project was focused on the total number of dead cells instead of live cells within the population over time. These observations were collected at non-equidistant time intervals over the 48 hour experimental timeframe. When looking at the EnKF prediction of our model, we looked specifically at how the frequency of data collection impacted the confidence of the prediction. As mentioned earlier, the EnKF algorithm has a built in uncertainty measure utilizing the covariances of the prediction ensembles. In our case, we compared the covariances of the EnKF prediction for varying live and dead cell data frequencies.

Figures 4.5 and 4.6 indicate collecting data for both live and dead at every hour interval provides a sufficient estimation of both state and parameter values. In an ideal world, hourly observations would be provided, but due to limitations in machine capacity, experimental data is collected at much fewer time points.

4.2.1 Same Frequency for Live and Dead Data Collection

When conducting frequency tests, our team needed to maintain the confidence of the EnKF on our model and reflect the limitations of machine capacity for data collection. To do so our team pulled subsets of a newly generated set of synthetic data, where $\beta = 0$, to mimic potential data collection time intervals. This new set of synthetic data uses the same process outlined in Section 3.1.2, with the exception of having $\beta = 0$.

When conducting frequency tests, it is important to note that the system and observation errors need to be fixed. For this section, our team selected the pairing of **C** and **D** values,

$$\mathrm{C} = \left[0.1;\ 0.1\right], \mathrm{D} = \left[1001;\ 12.4\right] \tag{4.6}$$

which reflect the pairing of errors that produced the smoothest estimation of state and parameter values for both live and dead cell data collection. This combination is shown in image (c) in both Figure 4.5 and 4.6, as C_{live} and C_{dead} having the same system error and observation error values, d_1 and d_2, equal to the observation noise of C_{live} and C_{dead} squared respectively.

Figure 4.8 shows the EnKF state and parameter predictions with fixed observation and system errors, as shown in Equation (4.6), and changing data collection frequency. Specifically, we show the results for data collected every two, four and eight hours.

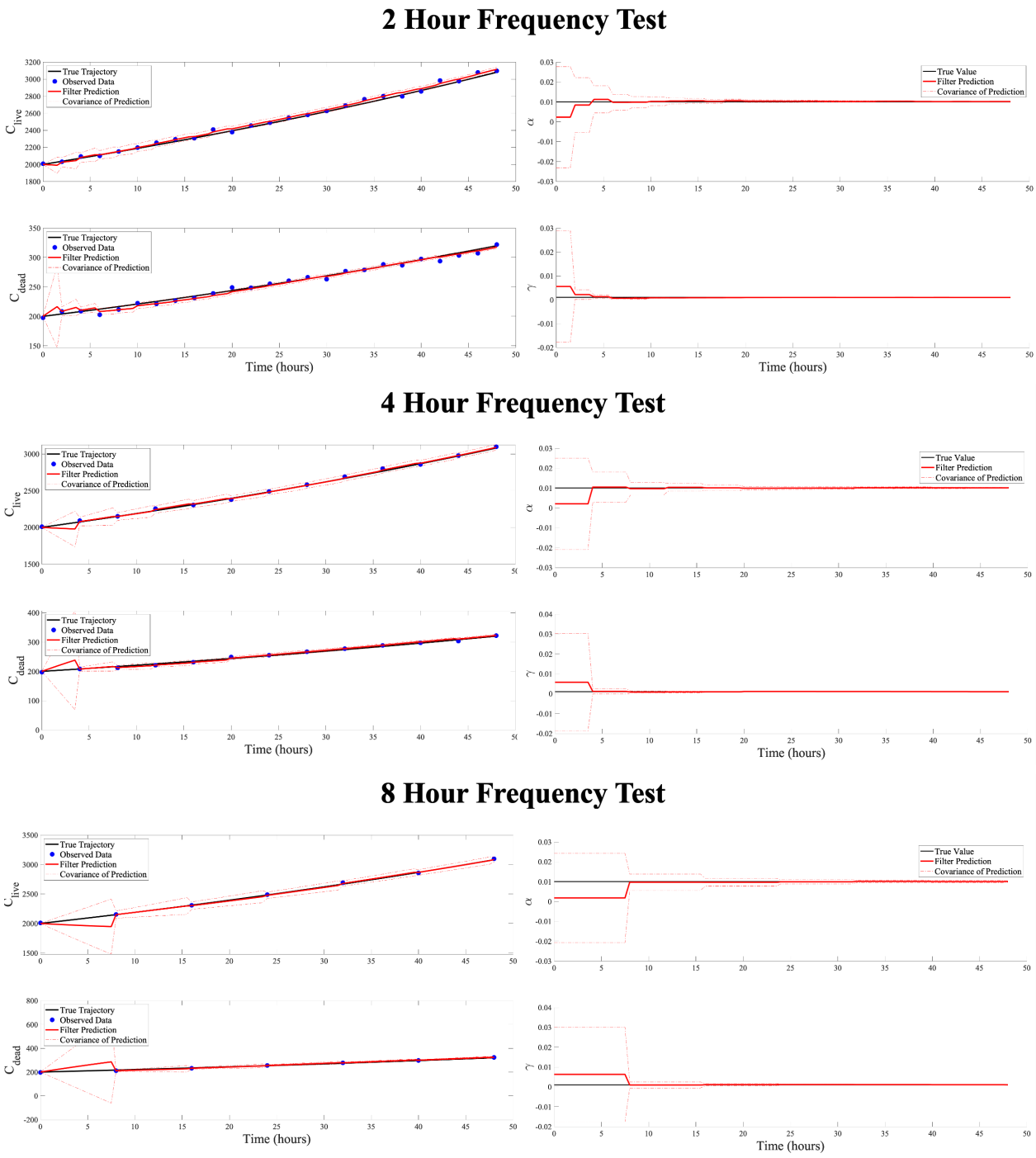


Figure 4.8: *Top Row*: EnKF estimation of state (top) and parameter variables (bottom) for the 2-hour interval of data collection. Final estimation of parameters: $[\alpha \ \gamma] = [0.0102 \ \ 0.001]$. *Middle Row*: EnKF estimation of state (top) and parameter variables (bottom) for the 4-hour interval of data collection. Final estimation of parameters: $[\alpha \ \gamma] = [0.0101 \ \ 0.001]$. *Bottom Row*: EnKF estimation of state (top) and parameter variables (bottom) for the 8-hour interval of data collection. Final estimation of parameters: $[\alpha \ \gamma] = [0.0100 \ \ 0.0011]$.

It should be noted that as the number of observations decreases, or the length of the time interval between data points increases, the initial covariance of the dead cell estimation increases. The EnKF algorithm takes in as much information at time t_j as possible to generate an estimation of all state and parameter values at time t_{j+1}. The interval $t_1 \leq t_j \leq t_2$ has the least amount of information for the filter to estimate on. This is why we see a larger covariance for both state variables. Though it is important to note this initial covariance, the goal of the filter is to have the covariances tighten around our true value over time.

As previously mentioned, the data provided to our team through the UMass experiments used a data collection frequency that is most similar to our simulation where data was collected every 4 hours. With this in mind, we believe the 4 hour equidistant time intervals for data collection best parallel conducted experimental procedures. Based on these results, we hypothesize that having data collected for both live and dead cell counts every 4 hours will maintain the confidence and accuracy of the EnKF estimation.

4.2.2 Differing Frequencies for Live and Dead Data Collection

As mentioned earlier, most machines used for these types of experiments do not have the capacity to collect observations for both live and dead cells every hour. Along with machine issues, most dyes used to fluoresce cells to allow for counting cannot provide accurate counts of both live and dead cells at the same time. Certain dyes, such as the SYTOX dye used in the UMass experiment, can only highlight the dead cells. Though this is an experimental limitation, our team has discovered the necessity for both live and dead cell count observations. With that said, we next looked at if there is a minimum amount of live cell observations that would still allow for a confident estimation of both state and parameter values. Utilizing the system shown in Equations (4.4) and (4.5) above, our team limited our subset of live cell counts with our assumed 4 hour interval for dead cell collection.

Though we experimented with smaller intervals, we would like to highlight two major cases for live cell collection, 24 hours and 48 hour intervals. These two intervals were a main focus throughout our simulations, as it mimicked the data collection method used in the UMass experiment. The goal of that experiment was to show more information can be gathered from observing dead cell counts and inferring live cell counts over time [13]. We wanted to see if this claim, only having live cell data at $t = 0$ and $t = 48$, is enough to maintain the confidence and accuracy of the EnKF estimation. Figure 4.9 shows the results of using the subset of live data at $t = [0,\ 48]$ paired with dead cell data collected at 4 hour intervals.

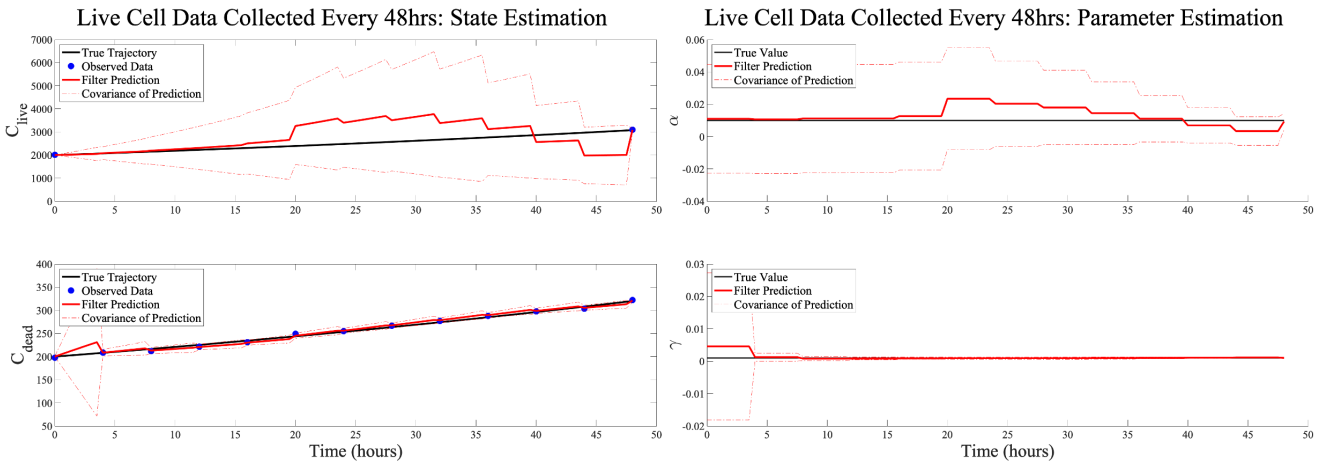


Figure 4.9: These graphs depict the state estimation (left) and the parameter estimation (right) when there is C_{dead} data collected every 4 hours and only C_{live} data collected at $t = 0$ and $t = 48$. The mean of the final parameter posteriors, at time $t = 48$, are as follows: $[\alpha \ \ \gamma] = [0.0093 \ \ 0.001]$. These results should be compared to the true values of these parameters: $[\alpha \ \ \gamma] = [0.01 \ \ 0.001]$. Result was obtained with the observation and system errors depicted in Equation (4.6).

The simulation conducted indicates an accurate estimation of the live cell state variable is most affected by limiting the number of observations. Specifically, we lose confidence in the C_{live} state variable estimation as we did when we had no live cell data. Figure 4.9 shows that even when we do have some live cell data, at time $t = 0$ and $t = 48$, the EnKF is unable accurately estimate C_{live} or α. We do, however, maintain a strong prediction of γ, which could potentially be due to the ODE for C_{dead}, Eq. (4.5), being strictly in terms of γ. This result tells our team that we need more data on live cells to maintain the confidence of the algorithm. Though we experimented with many intervals, our team found that it is necessary to add a third data point for live cell count half way through the experiment to allow the filter to correct its estimation of C_{live} and α. Figure 4.10 shows the result of simulation when collecting live cell data every 24 hours.

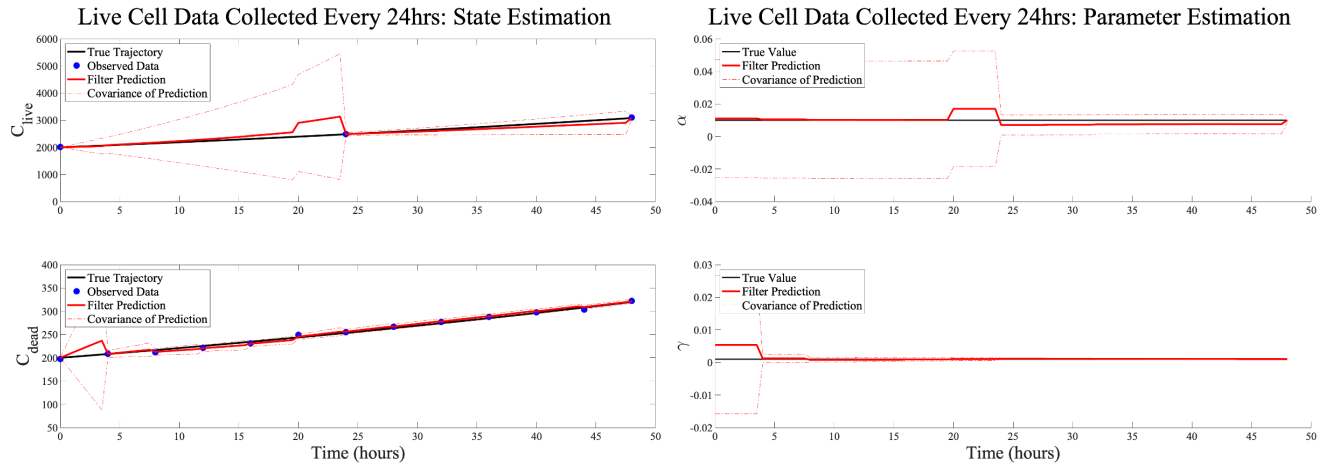


Figure 4.10: These graphs depict the state estimation (left) and the parameter estimation (right) when there is C_{dead} data collected every 4 hours and only C_{live} data collected every 24 hours. The mean of the final parameter posteriors, at time $t = 48$, are as follows: $[\alpha \ \ \gamma] = [0.0100 \ \ 0.001]$. These results should be compared to the true values of these parameters: $[\alpha \ \ \gamma] = [0.01 \ \ 0.001]$. Experiment was conducted with the observation and system errors depicted in Equation (4.6).

The third piece of data, at the halfway point of the time interval, allows for the filter to gain an important and corrective piece of information. This third data point corrects the estimation, which does start to diverge from the true trajectory when $0 \leq t \leq 24$, and narrows the confidence of the state variable estimation and fully allows the filter to capture both α and γ.

Though it is still ideal to have data collected for both live and dead cells at the same interval, there are experimental limitations that make this nearly impossible. Specifically, very few machines are capable of counting both live and dead cells without taking into account larger noise levels. We can further experiment with non-equidistant time intervals, such as the one implemented by the UMass researchers. With these intervals in mind, we still need to have the EnKF have a mean that converges to the true value and the covariance tighten around the prediction to conclude the filter is estimating the values correctly. Thus, our team has determined that a minimum of three live cell observations are necessary to allow the filter to regain confidence in its estimation. Figure 4.11 shows the simulation results of using the EnKF algorithm with this non-equidistant time-interval for data collection.

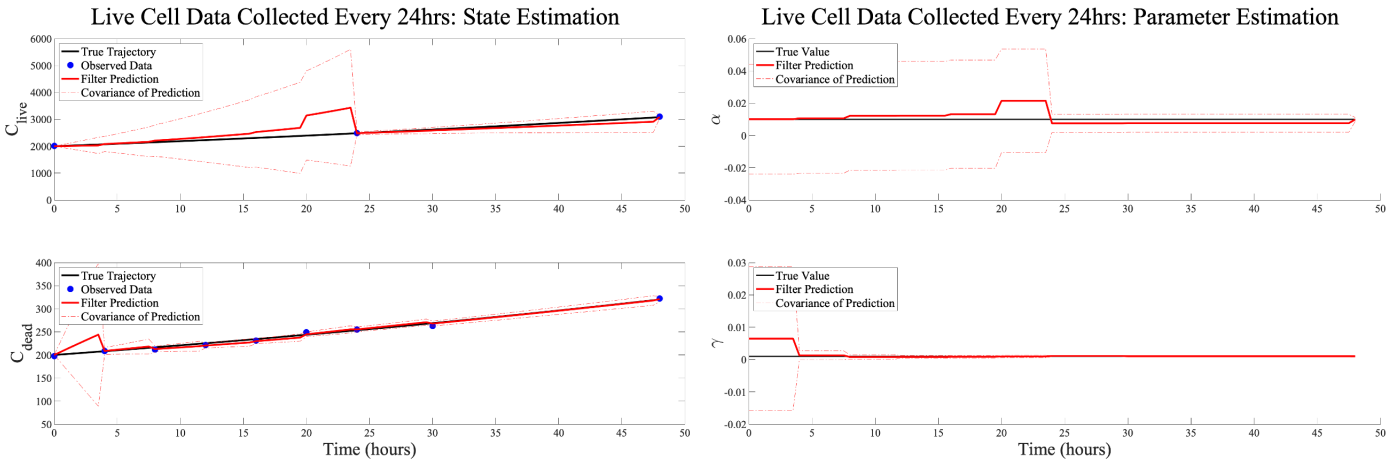


Figure 4.11: These graphs depict the state estimation (left) and the parameter estimation (right) when there is C_{dead} data collected through the method used by experimenters (every 4 hours for the first 24 hours, then sporadically for the final 24 hours). This data is paired with data collected for C_{live} every 24 hours. he mean of the final parameter posteriors, at time $t = 48$, are as follows: $[\alpha \ \ \gamma] = [0.0099 \ \ 0.001]$. These results should be compared to the true values of these parameters: $[\alpha \ \ \gamma] = [0.01 \ \ 0.001]$. Result was obtained with the observation and system errors depicted in Equation (4.6).

These time intervals were chosen after we established that to maintain a confident estimation of the state and parameter values using the EnKF we need at least 3 data points for live cell counts. This data collection frequency mimics the dead cell collection technique implemented by the UMass researchers will also provide sufficient estimations. Dead cell counts were collected at 4 hour intervals over the first 24 hours, then with less frequency over the final 24 hours.

As mentioned earlier, changing the observation and system error within the model impacted the confidence of the estimation as well as the trust in the data. Figure 4.11 shows that a confident estimation of the state and parameters is obtainable with dead cell data paired with at least three live cell measurements.

4.2.3 Takeaways

When considering the frequency of data provided to the EnKF algorithm it is necessary to incorporate real practices within synthetic data experimentation. In our case, the amount of live and dead cell information is dependent on the capabilities and capacity of the experimental data collection systems. To maintain a confident estimation of state and parameter values, our numerical experiments show that it is necessary to have at least three live cell count observations. Though more information is always helpful, limitations in collection methods may make it impossible to obtain the maximum amount of information. Live cell counts are a necessity to confidently capture the growth parameter α, but observing both live and dead cell counts at equal intervals is not always possible. Our team has determined that a minimum of three live cell observations are required to confidently estimate a constant α, γ and the state variables in this model.

4.3 Time-Varying Growth Parameter

All of the experiments presented thus far have implemented a constant growth parameter α. We have seen that our model with constant growth parameter can describe the dynamics seen in the experimental data provided to our team. Due to the limited amount of data for C_{live}, however, our team believes a model with a time-varying α parameter could accommodate a more dynamic system. As mentioned in Section 3.1.2, our team implemented three different time-varying parameters (Eq. (3.1), (3.2), and (3.3)) to test the EnKF algorithm's ability to estimate a system with a time-varying growth of live cells. It should also be noted that to allow for potentially longer data collection times, our team extended the time span for these experiments to $0 \leq t \leq 72$ hours. This extension was also added to allow parameters to evolve more slowly over time as well as allow the filter more time to adjust to the changes. This new time extension is within time frames of other experiments introduced in the literature [3].

4.3.1 Estimation of Time-Varying Growth

A time-varying parameter can be estimated using the EnKF algorithm in two ways: when the functional form of $\alpha(t)$ is assumed by the filter and unknown parameters of the function are estimated; and when the form is unknown by the filter. Either of these cases are straightforwardly implemented into the existing EnKF code by allowing for small adaptations to the existing algorithm estimating constant parameters.

When we assume the filter knows the functional form of the parameter, the filter knows to estimate the unknown hyperparameters within this known function. For example, if we use the first of the three functional forms, Equation (3.1), the prediction step of the EnKF will know that $\alpha(t)$ has a logistic form and will estimate the values of the two constant parameters, τ and s, within $\alpha(t)$.

When the functional form of $\alpha(t)$ is not assumed, which is common in real world problems, the EnKF is adapted to capture the movement of the time-varying parameter by incorporating a random walk model, ξ, in a similar fashion as [26]. Within the filter, there is no assumption about the functional form of α. This means, if the parameter is actually a constant, the random walk within the filter should estimate the parameter as constant.

These two adaptions of the EnKF generate different estimations of state and parameter values. The addition of a time-varying growth parameter, $\alpha(t)$, requires the experiments shown in Sections 4.1

and 4.2 to be conducted on new sets of synthetic data. The following numerical results will use the assumption that data is collected for both live and dead cell counts, at every hour. Further experiments were conducted to mimic those conducted in Section 4.2, but for the purposes of this paper, we focus on how how a time-varying growth parameter impacts the estimation of the other parameters and state values under ideal circumstances.

4.3.2 Results with Assumed Functional Form

Assuming the functional form within the prediction step permits the filter to estimate the hyper-parameters as constant values. Though this allows for the possibility of further known dynamics to be included in the model, the EnKF was developed with the intention of not knowing all of these dynamics. In most real world problems, the functional form of a time-varying parameter is unknown, and in this case, the EnKF should be adapted to use the random walk coefficient, ξ. With this is mind, our team believed for the purposes of this data, generating numerical results for both approaches to a time-varying parameter will allow for more understanding of the dynamics of the system. Any model describing the experimental data must be flexible as the parameters of the model depend on the drug used in the experiment. If the functional form of $\alpha(t)$ is assumed, the impact of the particular drug on the cancer cells can be studied at much finer detail. Thus, we ran several simulations using both approaches.

Based on our results of the EnKF on the system with constant α, our team only presents our findings when data is collected for both live and dead cell counts at every hour. Using the synthetic data generated to mimic our experimental data, our team first explored the ability of the EnKF to produce an accurate estimation of both the state variables, C_{live} and C_{dead}, and the new set of parameters under the assumption that our $\alpha(t)$ is of the logistic form shown in Eq. (3.1). Figure 4.12 shows a basic run of the EnKF, demonstrating the ability of the filter to estimate these system components.

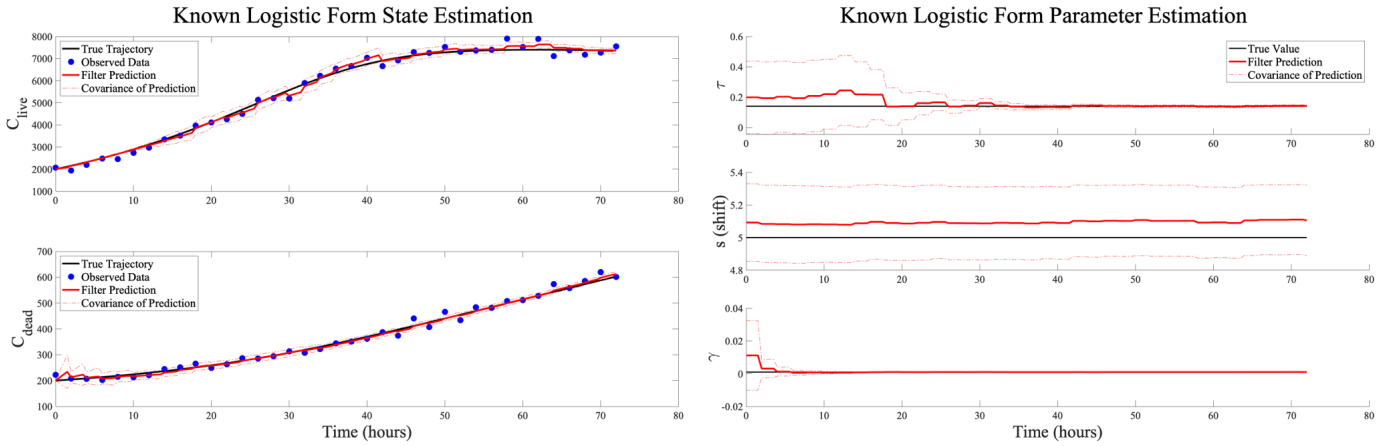


Figure 4.12: *Left*: State variable estimation for known logistic form of the time-varying $\alpha(t)$ growth parameter. Uses C = [0.1000; 0.01] and D= $[1.8837 \times 10^4;\ 0.0153 \times 10^4]$. *Right*: Parameter estimation for known logistic form of $\alpha(t)$. The true values of parameters, $[\tau\ \ s\ \ \gamma] = [0.14\ \ 5\ \ 0.001]$. Final estimation of parameter values: $[\tau\ \ s\ \ \gamma] = [0.1413\ \ 5.0986\ \ 0.0010]$.

From Figure 4.12, we are able to see that the filter produces a consistently strong estimation of the drug-induced death rate parameter, γ, even with a change in the form of α. Along with this, depending on the values of the observation and system errors, the EnKF will eventually capture the value of τ, but the parameter s is unable to be well identified by the filter.

The same process as above was then conducted on both the linear (Eq. (3.2)) and the piecewise function (Eq. (3.3)) forms. When using the linear $\alpha(t)$, the assumed form allows the EnKF to estimate the constant slope, m with a fixed y-intercept, α_0, which has the value shown in Equation (3.4). The piecewise $\alpha(t)$ estimates the values of α_0 and α_1 over the entire time span, only using one of the values during the prediction step depending on the current time t.

Figure 4.13 indicates that when the functional form of the $\alpha(t)$ is assumed, the new parameters introduced within $\alpha(t)$, m, α_0 and α_1, are all confidently captured by the EnKF. It should also be noted that for all three of our potential time-varying growth parameters, the filter's ability to well-capture γ is consistently maintained.

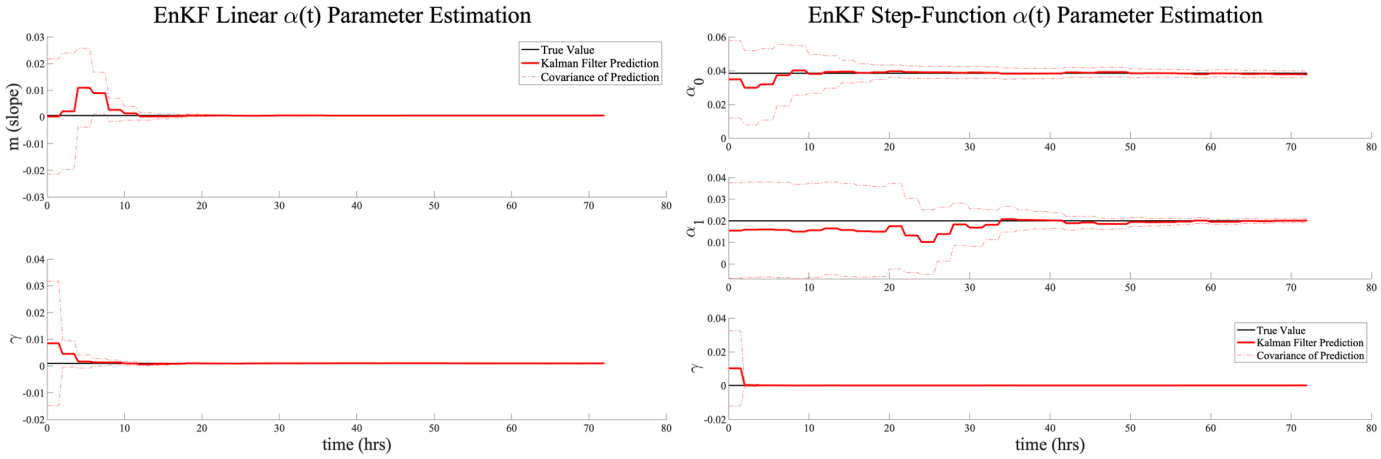


Figure 4.13: *Left*: Results of assumed linear growth parameter. Estimation treats the slope of $\alpha(t)$ and γ as constant parameters. The simulation conducted used four hour increments of the synthetic data generated using true parameter values of $[m \quad \gamma] = [2.4 * 10^{-4} \quad 0.001]$; final estimation of parameters $[m \quad \gamma] = [2 * 10^{-4} \quad 0.001]$. Also used assumed value for the observation error $D = [417.65; \ 240.07]$. *Right*: Results of assumed step-function growth parameter. Estimation treats the two values of $\alpha(t)$, α_0, α_1 and γ as constant parameters. The simulation conducted used synthetic data generated using true parameter values of $[\alpha_0 \ \alpha_1 \ \gamma] = [0.0385 \ 0.02 \ 0.001]$; final estimation of parameters $[\alpha_0 \ \alpha_1 \ \gamma] = [0.0378 \ 0.0201 \ 0.001]$. Simulation run used data collected at four hour increments while also assuming observation error $D = [724.11; \ 300.98]$. Both simulations used a system error, $C = [0.1; \ 0.1]$.

While the assumption of the functional form of a time-varying α parameter could allow for accurate estimations of the state and the majority of the new parameters, our team still needed to inquire about the filters capacity when the form is unknown. With the experimental data provided to our team in mind, the form of $\alpha(t)$ is most likely unknown. To mimic the real world problem, we must look at the EnKF with the parameter tracking adaptation.

4.3.3 Results with Unknown Functional Form

For more real world problems, most time-varying dynamics within a system are unknown and should be treated as such. To provide a more plausible tool for parameter estimation, we must assume the functional form is unknown to experimenters. Thus, we need to implement the EnKF with the random walk coefficient, ξ_n. There is a point in which the amount of random walk added to a parameter will disrupt the estimation even for the most dynamic of parameters. In our case, the random walk coefficient is slightly different for each time-varying form, but are all within one order of magnitude or the following:

$$\xi_n = \left[0.0001;\ 0.00001\right] \tag{4.7}$$

By adding this random walk coefficient, we allow the filter to alter its parameter estimation to capture any deviations from a constant parameter. With this in mind, it should also be noted that the random walk coefficient should not impact a truly constant parameter. In other words, the adaptation will estimate a parameter as a constant, if the parameter truly is a constant. In our case, we do add a small amount of randomness to γ, but the prediction should capture that γ is considered a constant parameter. Figure 4.14 below shows the results of a simulation run on the same set of synthetic data used for the assumed logistic functional form simulations.

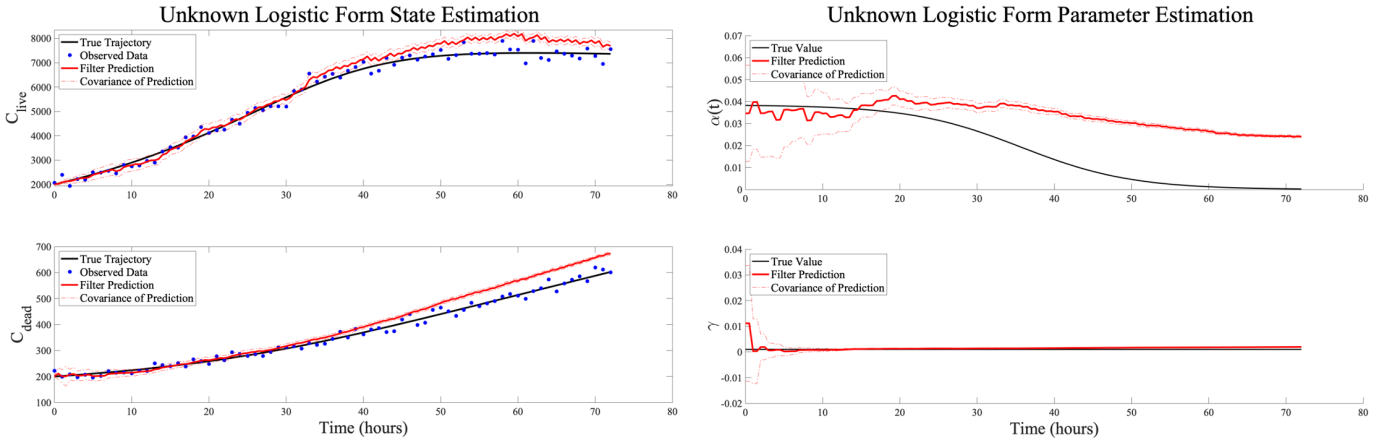


Figure 4.14: *Left*: State variable estimation for the unknown logistic form of the time-varying $\alpha(t)$ growth parameter. Uses C = $[0.1000;\ 0.01]$ and D= $[1.8837 \times 10^4;\ 0.0153 \times 10^4]$. *Right*: Parameter estimation for known logistic form of $\alpha(t)$. The true values of the constant parameter, $[\gamma] = [0.001]$. The mean of the final posterior: $[\gamma] = [0.0011]$.

From Figure 4.14, we see the EnKF algorithm can maintain an accurate estimation of γ, but the movement of the logistic $\alpha(t)$ is much harder to capture. The parameter tracking mechanism does trend towards approaching 0 for large t, but the full decent is not captured for any reasonable combination of **C** and **D** values. We only present the results of the combination where C = $[0.1000;\ 0.01]$ and D= $[1.8837 \times 10^4;\ 0.0153 \times 10^4]$.

This overestimation of both C_{live} and $\alpha(t)$ is due to the dynamics of the system changing slower than the dynamics of $\alpha(t)$. In other words, the movement in $\alpha(t)$ in this form is much faster than the movement of the system as a whole. The number of dead cells, C_{dead} is not greatly influenced by the dynamics of $\alpha(t)$ which is why the prediction of C_{dead} follows the true trajectory. C_{live}, on the other hand, is greatly impacted by a change in $\alpha(t)$ which explains the plateauing affect seen at hour 40 on the state estimation image in Figure 4.14.

The next big question, is whether or not this same overestimation pattern can be seen in our linear or step functional forms of $\alpha(t)$. Using our synthetically generated data for each $\alpha(t)$ system that was used in the assumed form simulations was also used for the parameter tracking estimation. Figure 4.15 shows the parameter estimations for $\alpha(t)$ and γ for both the linear and step functional forms.

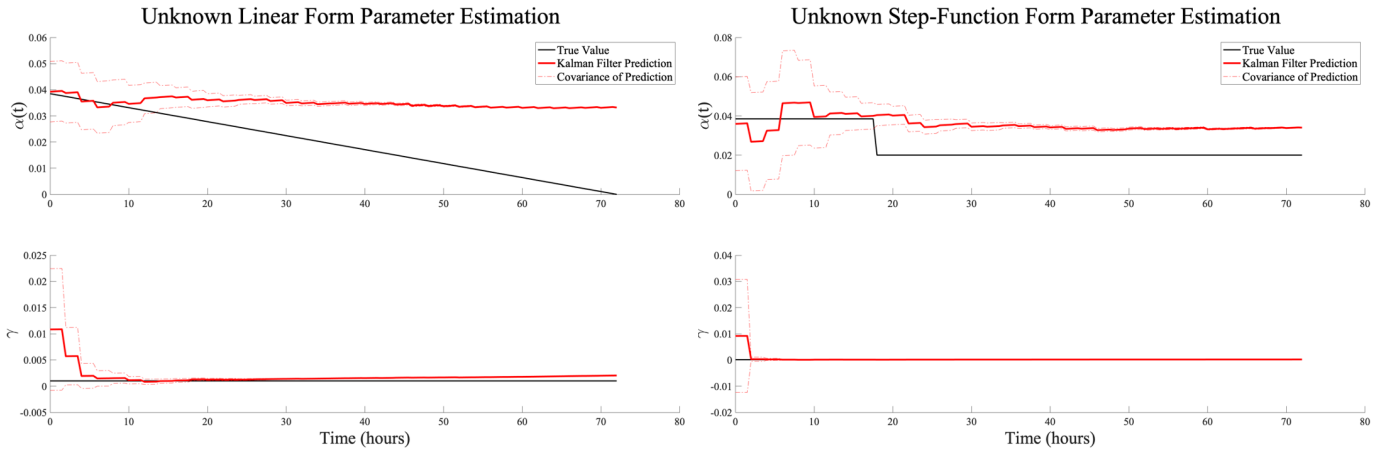


Figure 4.15: *Left*: Results of unknown linear growth parameter. The simulation conducted used synthetic data generated using the true form of $\alpha(t)$, depicted as the black line in the top image on the left. Simulation run used data collected at four hour increments while also assuming observation error $D = [3.214 \times 10^4;\ \ 0.013 \times 10^4;]$. *Right*: Results of assumed step-function growth parameter. The simulation conducted used synthetic data generated using the true form of $\alpha(t)$, depicted as the black line in the top image on the right. Simulation run used data collected at four hour increments while also assuming observation error $D = [2.691 \times 10^4;\ \ 0.020 \times 10^4]$. Both simulations used a system error of C = [0.1; 0.1].

Both the linear and step form of $\alpha(t)$ also demonstrate the behavior of overestimating the values of $\alpha(t)$ later on in the given time span. Specifically, the filter is predicting a smaller slope for the the linear $\alpha(t)$ than the true slope, and a smaller step for the step-function $\alpha(t)$. These results seem to suggest that the filter is unable to capture the true decline for any of the presented forms of $\alpha(t)$, resulting in an overestimation of the observed data points for C_{live} and C_{dead}. We are unable, however, to conclude this just from the above experiments. In other words, we need to explore the possibility that the produced parameter estimations produce a solution to our system of ODEs that fit our data.

To do this, our team first explored the impact of changing the value of the slope, the hyperparameter m, of the linear form of $\alpha(t)$ on the state variables C_{live} and C_{dead}. We fixed the true value of γ as the final estimated value provided by the filter, as well keeping the initial values of C_{live} and C_{dead} as 2000 and 200 respectively. Figure 4.16 depicts the variations of $\alpha(t)$ based on changing the slope and the impact of this change on C_{live}.

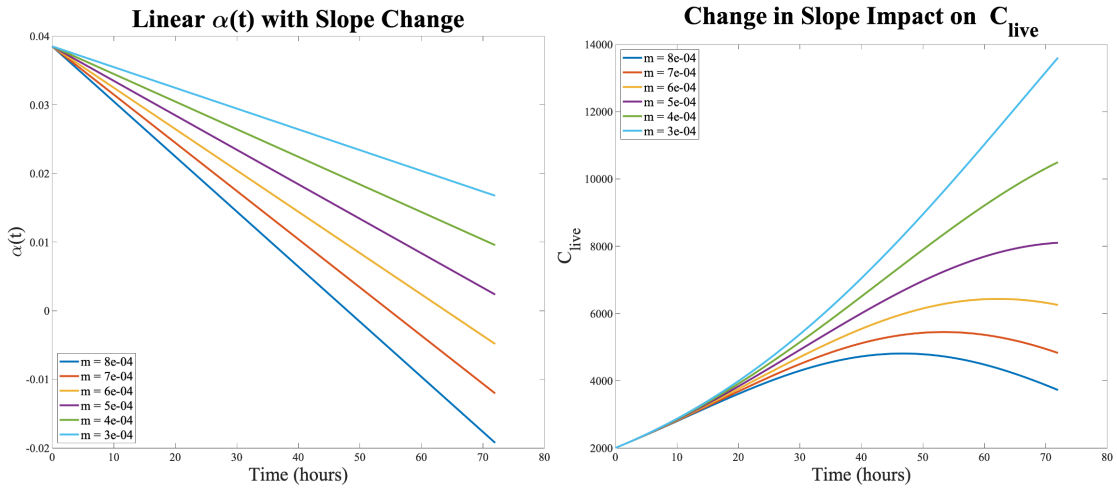


Figure 4.16: *Left*: Graph of linear $\alpha(t)$, as depicted in Eq. (3.2), with $\alpha_0 = \frac{ln(2)}{18}$ over the timespan, $0 \leq t \leq 72$ hours. The hyperparameter, m, slope values are within the following range: $3 \times 10^{-4} \leq m \leq 8 \times 10^{-4}$. *Right*: Solution to Equation (3.5) with the linear $\alpha(t)$ variations shown in left image, $\gamma = 0.001$, and initial values $C_{live}(0) = 2000, C_{dead}(0) = 200$.

We can see that changing the slope of the linear $\alpha(t)$, drastically changes the growth of C_{live}. As the slope becomes less steep, the live cell population, C_{live}, will continue to grow in an almost exponential growth pattern. This result was expected, but the large range of final live cell population values for the different slopes suggests that even an estimation that does not capture the true slope will overestimate the live cell population. Figure 4.16 suggests any change in the slope changes the growth of the live cell population. Thus, implying if the EnKF is producing an estimation of a linear $\alpha(t)$ with a less steep slope than the true value, then an overestimation of C_{live} will occur. Figure 4.17 shows the result of taking the red line prediction of the linear $\alpha(t)$ from the left-most image in Figure 4.15 and solving our model with this prediction.

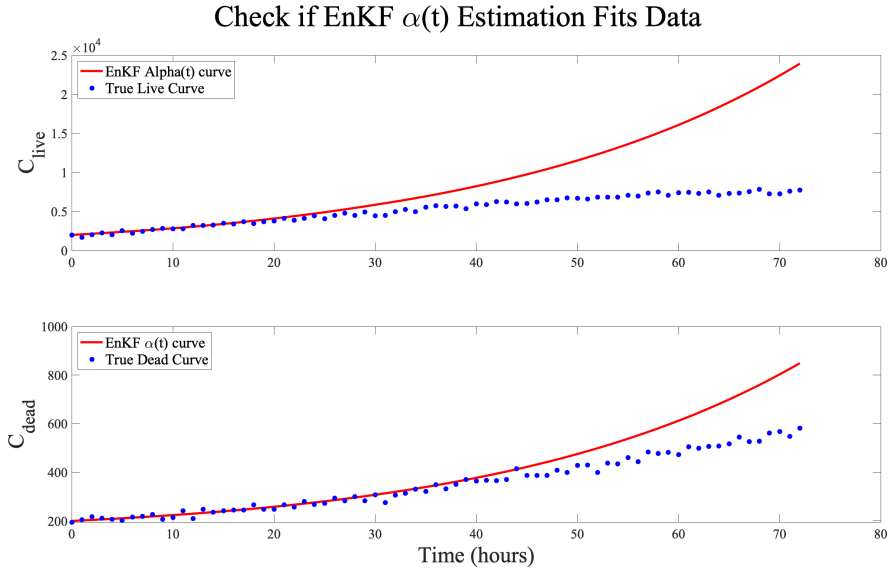


Figure 4.17: Results of using filter's parameter estimation for linear growth parameter (left-most image in Figure 4.15) to solve Eq. (3.5) and (3.6). Further supports the notion that the EnKF estimation results in an overestimation of both C_{live} and C_{dead}.

This same experiment was conducted for both the logistic and step forms of $\alpha(t)$. The results also suggest that the EnFK is unable to fully capture the true dynamics of $\alpha(t)$, resulting in overestimation of $\alpha(t), C_{live}$ and C_{dead}. Thus, our proposed model with only a time-varying growth parameter, cannot be estimated using the EnKF algorithm without a certain degree of overestimation. With further consideration, our team believes the filter's inability to capture the true decline in $\alpha(t)$ is most likely the result of one or a combination of the following two issues.

The first could be due to the dynamics of the $\alpha(t)$ parameter being faster than the system as a whole. For the parameter tracking adaptation of the EnKF algorithm to capture the movement of the parameters, it is imperative that the parameter changes at a slower rate than the state variables. When compared to our cancer cell model, the mass-spring model (Section 2.3) demonstrates a system with faster state variable dynamics. Our data depicts a system with exponential growth in dead cell counts with little known about the path of the live cell population. Our proposed model given with both a constant and time-varying natural growth parameter is not a system with fast moving dynamics. By adding a parameter with faster dynamics than the state variables, the random walk model might not be able to fully capture this movement when provided only state variable data. If this is the case, the next steps to combat this explanation would be to further develop our model by adding more dynamics to the live cell population or experimenting with the observation and system errors at the later time points.

The second possible explanation for the overestimation of $\alpha(t)$, could be due to the filter's extreme confidence in the γ parameter. No matter the form of $\alpha(t)$, provided the same initial guess of γ, the filter almost instantaneously gains insight and confidence for the γ parameter. Our team would like to further develop our model with a time-varying drug-induced death parameter, $\gamma(t)$. This change might allow for more dynamics in the state variables or allow more movement in $\alpha(t)$ through a larger random walk coefficient. Though we do not explicitly experiment with resolutions to either of these two issues in this paper, we do comment on how our team would further explore the solutions in Chapter 5.

Chapter 5

Discussion and Conclusions

In this project, we have presented a coupled ODE system to model the interactions of cancer cells and cancer treatments, and utilized parameter estimation techniques as a mechanism to provide researchers with deeper insight into potential time-varying components, moments of importance for data collection, and experimental data collection for optimized estimation of events. We have implemented the Ensemble Kalman Filter for parameter estimation as it provided a measure of confidence for both state and parameter variable estimations and is easily adapted to accommodate time-invariant and dynamic parameters.

5.1 Discussion of Results

We conducted numerical experiments on synthetically generated data, meant to mimic the patterns of the provided experimental data, with both constant and time-varying growth parameters. When considering a model with strictly constant parameters, our team discovered the need for observational data to be collected for both live and dead cell counts over the time span of the experiment. Though the frequency at which this data is collected varies when a time-varying natural growth parameter is introduced, the system with a constant natural growth parameter requires a minimum of three observed points on both live and dead cell counts to maintain a confident and accurate estimation of both state and parameter values over time. This frequency varies as a time-varying growth parameter is implemented. This is due the relationship between the natural growth parameter and the state variable describing the live cell population. To utilize the Ensemble Kalman Filter effectively on a system with a time-varying natural growth parameter, the number of observations for live cell count is increased, especially at the later hours of the experiment. We also discovered that even with a large number of observations for live cell count, a time-varying growth parameter that decreased over time cannot be accurately estimated using the Ensemble Kalman Filter. The true decline of the proposed time-varying parameters cannot be fully captured by the filter, resulting in an overestimation of the live and dead cell counts.

The results of our numerical experiments are meant to provide researchers with future experimental guidelines, while also exploring the dynamics between cancer cells and potential treatments. The results presented in this paper have all utilized a set of synthetically generated data meant to mimic the growth pattern of cancer cells when interacting with one type of cancer treatment. Our team was provided data for three cancer treatments, but due to the time-limitations of our study, we only present the results of our proposed model with the parameter values associated with one of the cancer treatments.

5.2 Future Work

This paper highlights the initial work done to develop a system of ODEs to model the dynamics of interactions between cancer cells and treatments. Though our initial findings outlined in this paper provide potential guidelines for future experimental data collection and utilize an uncommon parameter estimation technique for the cancer research field, we do still suggest further work be done to develop a model that is easily adaptable to estimate the interactions between cancer cells and new cancer treatments.

5.2.1 Potential ODE Model Developments

As mentioned at the end of of Chapter 4, our initial model with a time-varying growth parameter $\alpha(t)$ overestimates the number of live and dead cells at the end of the experimental time span of 72 hours. We suggest further simulations should be run on a model that incorporates a time-varying drug induced death parameter, $\gamma(t)$. Other experiments suggest that when a cancer treatment is introduced to a cancer cell, the treatment does not take effect instantly. This phenomena is known as cancer treatment onset-time [13]. One way to potentially capture this drug onset-time, t_{os} is by having a time-varying drug induced death parameter, $\gamma(t)$ with $\gamma(0) = 0$ and increases to its true value at time t_{os}.

Another potential addition to the model would be to reintroduce the natural death parameter β, when utilizing time-varying $\alpha(t)$ and $\gamma(t)$. This could add the necessary dynamics to allow the Ensemble Kalman Filter to capture the other parameters with the parameter tracking adaptation. The same simulations done within this study could be used when exploring the results of these additions to the model.

5.2.2 Delay Differential Equation Changes

ODE models can be adapted to accommodate delays in dynamics by using time-varying parameters, but as we have seen in our results, these changes cannot always be captured by parameter estimation techniques. Most natural systems evolve over time and the impact delays have on a system vary based on the duration of the delay. For example, a car accelerating or decelerating occurs almost instantly, with a slight delay from when a foot hits the break and the message is sent to the wheel axels. This delay is so minute, it is not considered when modeling the dynamics within a car. On a more microscopic level, cell dynamics are often delayed more substantially than the time scale of the entire cell proliferation process [2]. This indicates that any delays found within a model as small as cell population dynamics could be explicitly accounted for within the model [15].

Delays in models are also considerably helpful when considering processes that have multiple stages. The cell cycle is known to have 3-4 stages, and depending on the drug, it could change the time of one or more stages. Due to the small amount of phases present in the cell cycle, delay differential equations with low variable counts can be used [15].

Delay Differential Equations (DDE) incorporate any explicit delay in a model by operating "on an infinite-dimensional space consisting of continuous functions that accommodate high-dimensional dynamics" [15]. In other words, delay differential equations are highly dependent on the function values at previous times t. The basic DDE for $x(t) \in \mathbb{R}$ is

$$\frac{dx(t)}{dt} = f(t, x_t) \tag{5.1}$$

where $x_t(\rho) = x(t + \rho)$ and $-\tau \leq \rho \leq 0$. This makes the current point ρ based on the previous time steps. DDEs can come in many shapes and forms, but they are easily solved using stepwise functions. It is also helpful to use MATLAB® -based solvers such as **dde23** to solve numerically. The exact process for utilizing **dde23** is further explained in papers such as [24].

While there are many different ways to utilize delay differential equations, it is imperative to know how they can better model certain systems. In our case, the cell cycle does not always happen at a constant or instantaneous rate. The cell cycle lasts, on average, 24 hours, but can vary dramatically based on the size and health of the cell and cell population. Cancerous cells can either slow down or speed up this process, creating delays at "check points" or delays further along the line if other phases are slowed and impact the timing of future phase durations [27]. Delays are also imperative when considering how cancer drugs impact the cell cycle and how developed the tumor environment is. Introducing a new drug to a much stronger tumor environment might take longer to see the effects than in a weaker tumor. This is also true when considering different drugs on the same tumor environment. Not all treatments work at the same rate, thus it is important to note that a model must be able to adapt for different treatments.

5.2.3 Proposed Model with Other Cancer Treatment Data

Our final suggestion for future work is to conduct our experiments on synthetic data generated to mimic the growth patterns of other cancer treatments as well as using the actual experimental data within the simulations. As we have previously mentioned, different treatments introduce different rates of cell death or cell cycle regulation. The results for parameter values and state variable growth patterns over time presented in this paper are for interactions of a cancerous environment with one cancer treatment. To better develop a model for this data, we suggest conducting the experiments outlined in this paper on a variety of cancer treatments.

Our results do not suggest the true values of our parameters within the simulations are universal parameter values meant to describe the interaction between cancer cells and every cancer treatment available today. The estimations made by the Ensemble Kalman Filter are dependent on observed data and the assumption of the underlying model. Thus, given any set of data that our proposed system could be used to model, the Ensemble Kalman Filter should provide an accurate estimation of the parameter values.

Bibliography

[1] A. Arnold and A. L. Lloyd. An approach to periodic, time-varying parameter estimation using nonlinear filtering. *Inverse Problems*, 34:1–22, 2018.

[2] G. I. Bell and E. C. Anderson. A mathematical model with applications to cell volume and distributions in mammalian suspension cultures. *Biophysical Journal*, 7:329–351, 1967.

[3] G. C. Forcina, M. Conlon, A. Wells, J. Yinuo Cao, and S. J. Dixon. Systematic quantification of population cell death kinetics in mammalian cells. *Cell Systems*, 4:1–11, 2017.

[4] S. Thakur, D. I. Cattoni, and M. Nöllmann. The fluorescence properties and binding mechanism of sytox green, a bright, low photo-damage dna intercalating agent. *European Biophysical Societies' Association*, 44:337–348, 2015.

[5] D. A. Charlebois and G. Balázsi. Modeling cell population dynamics. *Silico Biology*, 13:21–39, 2018.

[6] F. Billy, J. Clairambault, and O. Feroq. Optimization of cancer drug treatments using cell population dynamics. *Mathematical Methods and Models in Biomedicine*, 1:265–309, 2013.

[7] Y. Dai and S. Grant. *Cell Cycle Checkpoints*, chapter Methods to Study Cancer Therapeutic Drugs that Target Cell Cycle Checkpoints, pages 257–304. Springer, 2011.

[8] U. Ledzewicz, A. d'Onofrio, and H. Schättler. Tumor development under combination treatments with anti-angiogenic therapies. *Mathematical Methods and Models in Biomedicine*, 1:311–337, 2013.

[9] L. Galluzzi and I. Vitale. Molecular mechanisms of cell death: Recommendations of the nomenclature committee on cell death 2018. *Cell Death and Differentiation*, pages 486–541, 2017.

[10] F. Gao and L. Han. Implementing the Nelder-Mead Simplex Algorithm with Adaptive Parameters. *Comput Optim Appl*, 51:259–277, 2012.

[11] M. S. Grewal and A. P. Anders. *Kalman Filtering: Theory and Practice Using MATLAB*. Wiley, 2001.

[12] J. F. Maxnes, J. Haux, and K. Hausken. The dynamics of cell proliferation. *Medical Hypotheses*, 62:556–563, 2004.

[13] H. R Schwartz, R. Richards, R. E. Fontana, A. J. Joyce, M. E. Honeywell, and M. J. Lee. Drug grade: An integrated analysis of population growth and cell death reveals drug-specific and cancer subtype-specific response profiles. *bioRxiv*, 2020.

[14] S. I. Kabanikhin. *Theory and Applications: Inverse and Ill-Proposed Problems*, chapter Basic Concepts and Examples. De Gruyter, Inc., 2011.

[15] Y. Kuang. Delay differential equations. *Arizona State University, Tempe*, pages 163–166.

[16] R. Richards, H. R. Schwartz, M. S. Stewart, P. Cruz-Gordillo, M. E. Honeywell, A. J. Joyce, B. D. Landry, and M. J. Lee. Drug combination antagonism and single agent dominance result from differences in death activation kinetics. *bioRxiv*, 2020.

[17] M. Lassas and S. Siltanen. Foreword: Inverse problems in biology. *Mathematical Biology*, 1:67, 2013.

[18] G. Lillacci and M. Kammash. Parameter estimation and model selection in computationl biology. *PLOS Computational Biology*, 6:1–17, 2010.

[19] V. Manca and L. Marchetti. Solving dynamical inverse problems by means of metabolic p systems. *Biosystems*, 109:78–86, 2012.

[20] J Mazur and L. Kaderali. *Model Based Parameter Estimation*, chapter The Importance and Challenges of Bayesian Parameter Learning in Systems Biology. Springer, 2013.

[21] C. T. Baker, G. A. Bocharov, C. A. Paul, and F. A. Rihan. Modeling and analysis of time-lags in some basic patterns of cell proliferation. *Journal of Mathematical Biology*, 37:341–371, 1998.

[22] B. D. Landry, T. Leete, R. Richards, P. Cruz-Gordillo, H. R. Schwartz, M. E. Honeywell, G. Ren, A. D. Schwartz, S. R. Peyton, and M. J. Lee. Tumor-stroma interactions differentially alter drug sensitivity based on the origin of stromal cells. *Molecular Systems Biology*, 14:1–15, 2018.

[23] H. Shirzadfar, S. Riahi, and M. S. Ghaziasgar. Cancer imaging and brain tumor diagnosis. *Journal of Bioanalysis and Biomedicine*, 9, 2017.

[24] L. F. Shampine and S. Thompson. Solving delay differential equations with dde23. *Southern Methodist University and Radford University*, pages 1–44, 2000.

[25] Y. Zheng, H. Moore, A. Piryatinska, T. Solis, and E. A. Sweet-Cordero. Mathematical modeling of tumor cell proliferation kinetics and label retention in a mouse model of lung cancer. *American Association for Cancer Research: Cancer Research*, 73:3525–3533, 2013.

[26] K. Campbell, L. Staugler, and A. Arnold. Estimating time-varying applied current in the Hodgkin-Huxley model. *Applied Sciences*, 10, 2020.

[27] M. Villasana and A. Radunskaya. A delay differential equation model for tumor growth. *Journal of Mathematical Biology*, 47:270–294, 2003.