Imperial College London

Final Year Project

Optimal Control and its Role in the Treatment of Cancer

Student: Mr. Max Yang
CID: 01085301
Project Supervisor: Dr Mylvaganam
Date: 28/05/2019

Department: Department of Aeronautics
Course: A401 MEng in Aeronautics
Module: Individual Project
Academic year: 2018/2019

Abstract

In the following thesis, four methods of nonlinear optimal control for the application of cancer treatment on a chosen model have been presented, in which, the performance of a proposed dynamic control law have been studied and analysed extensively. The State Dependent Riccati Equation (SDRE) is a trialed and tested method which provided the main benchmark for performance measurement, whilst Linear Quadratic Regulator (LQR) and Direct Collocation presented two ends of the spectrum in terms of nonlinear control optimality and have been applied for the purpose of demonstration. It has been shown that the LQR's linear approximation was too crude for this optimal control problem and failed to control the system upon initial implementation. Following the modification of the objective, an acceptable performance was attained but was still inferior to the SDRE. The solution to the dynamic control law was much more difficult to obtain compared to the other methods but its greater design flexibility and offline control mechanism allowed the use of iterative procedures to further minimise its approximations and optimise the control, beyond capabilities of LQR and SDRE. The accuracy of the final result was shown to be within 0.004% to the optimal solution obtained using Direct Collocation. It's guaranteed local asymptomatic stability, strong performance and minimal computational effort makes a superior controller to the other methods and a very attractive option for general nonlinear optimal control problems.

Acknowledgements

I would like to give a special thanks to my project supervisor Dr Mylvaganam for her support throughout this project and her undying faith in my abilities. It is no doubt that her continuous guidance and comprehensive feedback have made the entire process very enjoyable and a tremendous way to conclude my Master's degree. I would also like to thank Jonny Tilford for his motivational support during the bad times. Finally, I would like to thank all the friends I have made in the four years of my degree and their support. It has been a pleasure to be part of this educational journey and I will be eternally grateful.

Contents

List of Figures

List of Tables

1 Introduction

Cancer is a leading cause of death across the globe and for many decades, researchers have continued to push the boundaries of limitation in order to stop this deadly disease. In the modern age, a variety of treatment methods are available to patients, including common procedures such as chemotherapy, immunotherapy, radiotherapy and surgery. The most effective treatment can be highly dependent on the cancer type and the patient under treatment. In spite of these development, new cancer cases are growing at an unprecedented rate, far beyond researchers' capabilities to develop new forms of treatment. It is therefore becoming increasingly important to develop upon the existing treatment methods and translate our understanding of cancer into more therapies.

Chemotherapy is a widespread form of treatment due to its effectiveness amongst many different types of cancer cells. However, the advantages can bring their drawbacks. During the treatment, healthy cells are also under risk, leaving the patient vulnerable to infections and other diseases. The ability to balance the trade-off between eliminating the cancer cells whilst maintaining the health of the patient presents a challenging optimal control problem. Throughout literature, various studies have demonstrated superior treatment of cancer via the optimal control of drug administration as compared to periodical protocols ([14],[2]). Therefore, utilizing the mathematical modelling of the different cancer types can be viewed as a powerful tool in developing improved treatment procedures.

Many of the real world systems are highly nonlinear and solving their corresponding optimal control problem presents a difficult challenge. The application of linear assumptions within nonlinear control problems have been implemented extensively within industries and often is able to offer satisfactory performance [9]. However, this enforces a critical linear assumption that is not reliable and for many applications can lead to a breakdown of the controls. These cases therefore necessitates the development of a reliable and robust nonlinear control technique. This provides the motivation for the implementation of a proposed dynamic control law [43] which has the potential to outperform the current industry-standard techniques. The feasibility of this nonlinear control strategy in the field of cancer treatment and the implementation process will make up the majority of this thesis.

In the following chapters, a chosen tumor model with chemotherapy treatment developed by De Pillis et al. in [15] will be studied under the optimal control problem formulated by Itik et al. in [22]. The literature review will entail an initial insight into the mathematical models and an overview of some well-established control methods that have been previously explored. The background section will contain the key theories needed for the implementation of each method, including the derivation of the *Hamiltonian-Jacobi-Bellman* (HJB) equation which will provide the core concept within optimal control theory. Here, a dynamic control law will also be introduced, along with Linear Quadratic Regulator (LQR), State Dependent Riccati Equation (SDRE) and Direct Collocation, which all have been previously explored in this domain (Direct Collocation has been previously used through commercial software). The comparison of their relative performance will be one of the key focus of this project.

A brief preliminary analysis of the tumor model will be shown in order to identify its behaviour in the drug-free state. The majority of this analysis have already been presented in [11], so only the key information will be re-emphasized in order to formulate the optimal control objectives. A simple LQR control will be applied to demonstrate the inadequacy of the linear assumption. Following this, the proven performance of a more reliable control method (SDRE) will be analysed and provide the standard to be matched when implementing the dynamic control law. This will conclude the background studies and lead to the individual contributions of the project.

In the following chapters, the search process and the mathematical derivation of an algebraic $\bar{P}$ solution to the dynamic control law will be presented. The difficulties and issues involved in finding the optimal control parameters will be discussed in detail along with potential alleviating strategies. Here, an optimisation problem has been formulated from the dynamic control law and the application of a Monte Carlo search method and Genetic Algorithm, which are closely related, have been utilized to offer insights into the performance of different control parameter settings. With this information, a systematic guideline for solving the optimal dynamic control problem will be devised and potential

improvements of the program will be applied where possible. The final result of the dynamic control law will be compared with SDRE and Direct Collocation (which is assumed to be optimal) with the expectation of its performance to lie in between the two.

The conclusion will discuss the performance of each control strategy as well as their drawbacks, with a focus on the dynamic control law and it's suitability for this particular application. It should be stressed that the biological feasibility of a continuous chemotherapy administration will not be considered in this thesis and the benefit of the following study offers a potential form of treatment in future applications when the necessary technological advancements are in place.

2 Literature Review

In the following literature review, an overview of widely used cancer models amongst acedemic researchers will be presented and the most appropriate model for the application of a nonlinear dynamic control will be selected and studied in more detail. The aim is to provide some initial insights about the types of cells and chemotherapy that the model considers and the assumptions involved. Common optimal control methods employed in previous research papers will also be outlined and the best performing controller will be carried forward to allow a comparison with the proposed dynamic control law.

2.1 The Mathematical Model

2.1.1 Model Selection

Various cancer models have been developed over the years, including [15],[16],[5],[7]. One particular set of models that have been repeatedly used and refined throughout research for optimal control are those proposed by De Pillis et al. initially in [15], which is a further extension of the lower dimension model described in [25]. These consists of the states; normal cells (N), tumor cells (T), immune cells (I), and chemotherapy concentration in the blood (M). Further model refinement explicitly represented the immune system by replacing the normal cell population with a circulating lymphocytes in [12], providing an improved measure of the patient's health. Therefore, in this model there are two immune components: immune cells and circulating lymphocytes, serve as a way to monitor the additional damaging side effects of chemotherapy.

One of the most recent advancement upon the model described so far is to include immunotherapy in addition to chemotherapy treatment as done so in [13]. This is a more recent treatment development that has gained much interest in the cancer research community due to its effectiveness. The goal of immunotherapy is to strengthen the body's own natural ability to combat cancer by enhancing the immune system and is often applied along chemotherapy to improve resistance. Here, the immune cells that are actively destroying the tumor cells are now modelled by a combination NK and $CD8^{+}T$ cells, both of which has the capabilities of killing cancer cells. The immunotherapy administration is modelled in a similar fashion to that of chemotherapy but only to boost the body's immune system by stimulating the recruitment of $CD8^{+}T$ cells. In [13], the paper demonstrated the ability of an optimal controlled mixed chemotherapy and immunotherapy treatment strategy, outperforming the capabilities of their individual treatments, both of which failed to control the tumor growth on their own.

Although the improved accuracy and scientific advancements of the latters model are undoubtedly clear, the complex drug-free behaviour of these systems indicate that they can be difficult to control, especially in [13] where it was shown that the tumor-free equilibrium is unstable. This implies that unless every tumor cell has been eliminated, the system will return to the high tumor equilibrium if treatment were to be stopped. In addition to this, these models are not as widely adopted in the research of optimally control cancer treatment due to immunotherapy being a relatively modern technique, therefore providing limited control performance for comparison. This therefore does not seem like an attractive option to introduce the dynamic control law to the field of cancer treatment.

For the reasons mentioned above, an earlier iteration of the models developed by De Pillis et al. [14] will be studied in this project. This model presents a great level interest due to the large variety of control methods that are associated with it and its ease of interpretation, offering a great starting point to examine the applicability of a new nonlinear control method. However, the more advanced models do present an interesting challenge and potential topics for future works following the successful implementation of the dynamic control law.

2.1.2 System of Equations

The model represents different cell populations as interacting species consisting of the states; normal cells $N(t)$, immune cells $I(t)$, tumor cells $T(t)$, and the drug concentration $M(t)$ at the tumor site and the entire system is modelled under the influence of chemotherapy administration $u(t)$. The interaction between different cells can be interpreted as follows; the immune cell growth may be stimulated by the presence of the tumor cells and aims to destroy in a predator and prey fashion, whilst the normal cells and tumor cells compete for resources, no competition term exists between the normal and immune cells, and the only harmful cells are the homogeneous tumor cells. The behaviour of the cells in a drug-free system can be modelled by the following nonlinear differential equations,

$$\begin{aligned} \dot{N} &= r_2 N(1 - b_2 N) - c_4 TN \\ \dot{T} &= r_1 T(1 - b_1 T) - c_2 IT - c_3 TN \\ \dot{I} &= s + \frac{\rho IT}{\alpha + T} - c_1 IT - d_1 I \end{aligned} \tag{1}$$

A logistic growth rate for tumor and normal cells have been chosen in the model. The normal cell population has a growth rate of r_2 with a maximum carrying capacity of b_2^{-1}. Similarly, the tumor cell population has a growth rate of r_1 with a maximum carrying capacity of b_1^{-1}. The competition terms between tumor and immune cells are represented by the death rates with constants c_2 and c_1 in the tumor and immune equations respectively. For normal and tumor cell interactions, the death rates are represented by the constants c_4 and c_3 in normal and tumor equations respectively. The source of immune cell is considered to be outside of the systems with a constant influx rate of s and in the absence of tumor cells, the immune cells will die off at a per capita rate d_1, resulting in a long-term population of s/d_1 cells. Under the presence of tumor cells, the growth of immune cells are stimulated by the nonlinear growth term $\frac{\rho IT}{\alpha+T}$, where α and ρ are constants.

Chemotherapy is a type of anti-cancer drug treatment that circulates in the blood stream and can be used to treat cancer cells anywhere in the body. The drug targets cells that are in the growing process of dividing which means it affects not only cancer cells but also the normal and immune cell population. In this study, it is assumed that chemotherapy kill all cells at different rates using mass action terms which have been previously devised by Itik et al. in [22], allowing for a simplified optimal control analysis. The full nonlinear system incorporating the effect of chemotherapy drug becomes

$$\begin{aligned} \dot{N} &= r_2 N(1 - b_2 N) - c_4 TN - a_3 MN \\ \dot{T} &= r_1 T(1 - b_1 T) - c_2 IT - c_3 TN - a_2 MT \\ \dot{I} &= s + \frac{\rho IT}{\alpha + T} - c_1 IT - d_1 I - a_1 MI \\ \dot{M} &= -d_2 M + u(t) \end{aligned} \tag{2}$$

where the constants a_1, a_2, a_3 are the killing effect of chemotherapy concentration on the immune, tumor and normal cells respectively. The drug concentration in the blood $M(t)$ has a decay rate of d_2 and is affected by the drug dosage given $u(t)$. Here, it is worth noting that the control input $u(t)$ only directly affects the drug concentration in the blood and this state further influence the other states. This can have major implications on the characteristics of the controlled system as will be seen later.

The behaviour of this system is highly dependent on the constants which can ultimately determine whether this cancer is curable (i.e. system is controllable). This is also a generalized cancer model which doesn't focus on a particular type of cancer. As mentioned by the author, the model is a simplified dynamics of the true tumor behaviour, assuming a homogeneous tumor site, which does not significantly affect the qualitative behaviour of the overall system backed by clinical data. This was assumed to be sufficient for the optimal control study and any in-depth analysis of the model parameters will not be considered in this report. A controllable system with a set of parameter given in [14] is shown in Table 1 of chapter 4.

2.2 Optimal Control

2.2.1 Basic Theory

Optimal control theory [4] is the idea of determining the control such that an optimality criterion is achieved, following a path described by the differential equations. The objective can be defined in numerous ways and it is down to the designer to select the most suitable objective function to achieve a desired performance. The generality of the theory allows it to be applied to a wide range of problems with varying degree of complexity. The generalized continuous time optimal control problem can be defined as follows,

$$\begin{aligned} \min_{x,u} \quad & J(x(t),u(t)) = \phi(x(t_f)) + \int_0^{t_f} L(x(t),u(t),t)\ dt \\ \text{subjected to} \quad & \dot{x} = f(x,u) \\ & x(0) = x_0 \end{aligned} \tag{3}$$

where $[0,\ t_f]$ is the time interval of interest, x is a state vector, u is a ontrol vector. $J(x,u)$ is the cost function to be minimised consisting of the terminal cost $\phi(x(t_f))$ and running intermediate cost $L(x,u,t)$. The minimisation is subjected to the dynamical system $f(x)$ (a set of ordinary differential equations (ODEs)) and the initial condition x_0.

Optimal control strategies have been previously explored in cancer treatment in [12],[14],[2],[3]. The motivation for applying this theory is due to the implicit understanding of chemotherapy and its side effects during treatment. The objective of the optimal control problem can be formulated to successfully eliminate the cancer cells whilst minimising the amount of drug administered, thereby avoiding the weakening of the patient's immune system to an unhealthy state. It has already been shown in [14] that a simple optimally applied treatment strategy in the form of a bang-bang controller can offer significant improvements on the traditional periodic protocols. This encourages the further development of more advanced control mechanisms that hopefully can provide enhanced treatment procedures than previously explored techniques.

2.2.2 Control Methods

Linear system are those that can be written in the form of $\dot{x} = Ax + Bu$ and the optimal control of these systems can be solved analytically using the classic approach, Linear Quadratic Regulator (LQR). However for the case of nonlinear systems, this is still an open topic with much advancements in the recent years due to increased computational capabilities. Many real world systems are inherently nonlinear and solving such optimal control problems with a linear control is often inadequate due to their complex nature. A linearisation of the cancer model can be highly inaccurate and produce undesirable results, often causing detrimental effects on the patients' health during treatment. With this in mind, many researchers in this domain handle the optimal cancer treatment problem as a nonlinear problem with the objective to minimise tumor levels and chemotherapy administration throughout the treatment. The LQR method will therefore only be used as a demonstration in this study.

One technique that has received much attention recently in the domain of nonlinear control is the State Dependent Riccati Equation (SDRE) which has been successfully applied in theory [8] and experimental practices in a wide range of applications [9]. It was first proposed by Pearson [37] and has been studied extensively in aerospace applications [44]. Due to its simplicity and satisfactory performance in nonlinear applications, this method have since been widely adopted in biological systems and more recently in cancer treatment [2]. The application of SDRE was attempted in [22], specifically for our model of interest, and the results showed a fast and easy derivation of a sub-optimal control to the cancer treatment problem. In addition, its ability to offer locally asymptotically stability for a neighbourhood about the origin also provides a very desirable control property. A similar control technique was also implemented in [21] for the same problem which uses Linear-Time-Varying (LTV) approximations but failed to outperform the SDRE.

The SDRE uses a psuedo-linear representation which allows it to utilize the well-established LQR methodology. A more recent control technique has been proposed by Astolfi et al. [43]. This is a dynamic control law which aims to quantify and minimise the approximations that exist in SDRE. In theory, this has the potential to outperform the SDRE or other Linear-Quadratic (LQ) approximations of the nonlinear problems. This have been explored previously by Sassano et al. in [42] and Mylvaganam et al. in [35] for the case of LQ approximation. In this study, the aim is to successfully introduce this dynamic control law to the cancer treatment problem and offer a close comparison with the trialed and tested SDRE control as well as the basic LQR. To fully understand the optimality of this dynamic control's performance, the use of an Nonlinear Programming (NLP) solver via an optimal control transcription method (direct collocation) will be applied to the same problem and its result will be used to provide an indication of the relative control performances of each method.

2.3 Objective Summary

Due to the similarity of the SDRE approach to the dynamic control law, the results presented in [22] will be utilized as the benchmark for performance comparison. The key objectives of the project are as follows,

1. Gain understanding of tumor model
2. Implement LQR and SDRE control and analyse their performance
3. Implement dynamic control law and optimise its performance
4. Implement direct collocation
5. Compare overall results of all the applied control techniques

This report will focus on the optimal control aspect of cancer treatment, with a strong emphasis on control optimality. Although important, the stability of the controlled systems and any treatment conditions have not been explicitly studied in this project which is required for the complete analysis of the dynamic control law. This provides potential future topics to be explored and will be discussed in future works in section 9.2.

3 Background and Theory

This section will cover the underlying theory behind optimal control and the different control strategies under investigation. To allow an unbiased comparison of these methods, the same quadratic cost function will be used throughout this report. The new optimal control problem is formulated as,

$$\begin{aligned} \min_{x,u} \quad & J(x(t),u(t)) = \frac{1}{2}\int_0^\infty q(x) + u^T R_c u \; dt \\ \text{subjected to} \quad & \dot{x} = f(x) + g(x)u \\ & x(0) = x_0 \end{aligned} \tag{4}$$

where $q(x)$ is a positive quadratic function that relates to the cost of the states and $R_c \in \mathbb{R}^{m\times m}$ is a positive definite matrix that penalizes the control. This creates a smooth objective function, allowing the derivation of the *Hamilton-Jacobi-Bellman* (HJB) equation which can be solved to provide the solution to the optimal control problem. This formulation is also considered as an infinite horizon problem, as $t_f \to \infty$, $\phi(x(t_f)) \to 0$, which simplifies the controller design by ignoring a terminal cost. The penalty gains can then be chosen such that the optimal controller aims to eliminate the tumor cells whilst maintaining minimal amount of control input.

3.1 Hamilton-Jacobi-Bellman Equation

The *Hamilton-Jacobi-Bellman* (HJB) equation provides the fundamental concept for solving a dynamic optimisation problem. It relies on Bellman's *Principle of Optimality* [24] and a mathematical optimisation technique called *Dynamic Programming* [41] to complete its formulation. A continuous time formulation of the HJB equation will be presented which will provide the main theory behind the optimal control methods that are investigated in the following sections.

Consider the general optimal control case presented in (3). The value function $V(x)$ is defined as the optimal cost remaining until destination at every point in the state space starting at the state x_0,

$$V(t_0, x_0) = \int_{t_0}^{t_f} L(t, x^*, u^*) \; dt \tag{5}$$

where the optimal trajectory composes of the optimal control u^* and optimal states x^*. In order to retain a meaningful optimal control problem, the value function has been defined such that $V(x) > 0$ for all $x \neq 0$ and $V(0) = 0$. The *Principle of Optimality* states that if the entire trajectory is optimal, whatever the initial state and initial control are, the subsequent control must also be optimal given those initial conditions. In other words, the subsection of an optimal control policy must also be optimal for that sub-problem. This can be formulated as,

$$V(t_0, x_0) = \int_{t_0}^{t_0+\Delta t} L(t, x^*, u^*) \; dt + \int_{t_0+\Delta t}^{t_f} L(t, x^*, u^*) \; dt \tag{6}$$

This implies that for the time interval $[t_0, t_f]$ to be optimal, the time intervals $[t_0, t_0 + \Delta t]$ and $[t_0 + \Delta t, t_f]$ also have to be optimal. Assuming that $L(t, x, u)$ is smooth over the interval $[t_0, t_0 + \Delta t]$ and Δt is sufficiently small, the first integral can be approximated as an initial cost and can be written as,

$$\begin{aligned} J_0(x^*, u^*) &= \int_{t_0}^{t_0+\Delta t} L(t, x^*, u^*) \\ &= \Delta t \cdot L(t_0 + \Delta t, x^*(t_0 + \Delta t), u^*(t_0 + \Delta t)) \end{aligned} \tag{7}$$

Utilizing the *Principle of Optimality*, the second integral in (6) can be represented as a slice of the value function and expanded with a Taylor expansion,

$$
\begin{aligned}
V(t_0 + \Delta t, x_0(t_0 + \Delta t)) &= \int_{t_0+\Delta t}^{t_f} L(t, x^*, u^*)\ dt \\
&= V(t_0, x_0) + \left[\frac{\partial V(t_0, x_0)}{\partial t_0}\right] \Delta t + \left[\frac{dV(t_0, x_0)}{dx_0}\right]^T \Delta x_0 + O(\Delta t^2, \Delta x^2)
\end{aligned} \tag{8}
$$

The higher order terms $O(\Delta t^2, \Delta x^2)$ are assumed to be small and Δx_0 can be approximated as $\dot{x}_0 \Delta t$ using a finite difference approximation, where $\dot{x}_0 = f(t_0, x_0, u_0)$ are the dynamic equations. Substituting (7) and (8) back into (6) gives

$$
\begin{aligned}
V(t_0, x_0) = &\ \Delta t \cdot L(t_0 + \Delta t, x^*(t_0 + \Delta t), u^*(t_0 + \Delta t)) \\
&+ V(t_0, x_0) + \left[\frac{\partial V(t_0, x_0)}{\partial t_0}\right] \Delta t + \left[\frac{\partial V(t_0, x_0)}{\partial x_0}\right]^T f(t_0, x_0, u_0) \Delta t
\end{aligned} \tag{9}
$$

Since the initial condition can be anywhere in the state space, the subscript '0' can be dropped. Simplifying the above equation and taking the limit $\Delta t \to 0$, the HJB equation for the problem stated in (3) can be evaluated,

$$
-\dot{V} = L(t, x^*, u^*) + V_x^T \cdot f(t, x, u) \tag{10}
$$

where $V_x \equiv \lambda$ is also known as the *Lagrangian* multiplier. The right hand side of the equation is defined as the *Hamiltonian*, $H_{opt} = min(L + \lambda^T f)$, and corresponding optimal control can be obtained by evaluating $H_u = 0$. For the case of a quadratic cost function in problem statement (4) and considering the case of an infinite horizon problem in which case $\dot{V} = 0$, the HJB equation reduces to the *Hamiltonian*,

$$
H \equiv \frac{1}{2}(q(x) + u^T R_c u) + V_x^T (f(x) + g(x)u) = 0 \tag{11}
$$

and the corresponding optimal control is

$$
u^* = -R_c^{-1} g(x)^T V_x^T \tag{12}
$$

and substituting the control back into (11) gives the final HJB equation,

$$
0 = \frac{1}{2} q(x) + V_x^T f(x) - \frac{1}{2} V_x g(x) R_c^{-1} g(x)^T V_x^T \tag{13}
$$

If it is possible to solve the above partial differential equation (PDE) and obtain a value function that is differentiable with continuous derivatives, then an optimal control policy can be evaluated using (12). However as it will be shown, solving this PDE can be extremely complicated, especially when the system is nonlinear. For the following sections, the analytical solution to the linear case is first derived and then different techniques used to expand the theory for nonlinear problems will be shown.

3.2 Linear Quadratic Regulator

For a linear system, given a state penalty Q and control penalty R_c in the quadratic cost function, the solution to the HJB equation can be derived directly, resulting in a global optimal control via Linear Quadratic Regulator (LQR) [26]. It is a common procedure to set the control penalty matrix as the identity matrix $R_c = I$ and vary the state penalties since only the relative difference of the penalty matrices affect the final control. Consider a controllable linear system with a stable zero-equilibrium ($f(0) = 0$) under the infinite horizon optimal control problem (4),

$$
\dot{x} = Ax + Bu \tag{14}
$$

where $A \in \mathbb{R}^{n \times n}$ is the dynamic matrix and $B \in \mathbb{R}^{n \times m}$ is the control matrix. The HJB equation in (13) can be re-written as,

$$
0 = \lambda^T A x - \frac{1}{2} \lambda B B^T \lambda^T + \frac{1}{2} x^T Q x \tag{15}
$$

where the state penalty function $q(x)$ has been redefined as x^TQx with a positive definite matrix $Q \in \mathbb{R}^{n\times n}$. Lets assume there exist a solution in the form of $\lambda = Px$, where P is a symmetric positive definite matrix. This property can be exploited to give $PA = \frac{1}{2}(PA + A^TP)$ and substituting this with the *Lagrangian* back into the HJB (15) gives

$$\frac{1}{2}x^T(PA + A^TP - PBB^TP + Q)x = 0 \tag{16}$$

where x can be factorized out of the equation. Since this equation must hold for any initial condition $x(0) = x_0$ even for those states where $x \neq 0$, the factorized term must be zero, resulting in the *Algebraic Ricatti Equation* (ARE),

$$PA + A^TP - PBB^TP + Q = 0 \tag{17}$$

and the optimal feedback control with gains $K = -B^TP$,

$$u^* = -B^TPx \tag{18}$$

Now instead of solving a PDE explicitly, the optimal control policy can be evacuated by solving the ARE (17) and (18) provides the global optimal control.

3.3 State Dependent Riccati Equation

As already seen, the optimal control problem for the linear case is relatively straightforward with the analytic solution to the HJB equation obtained through solving the ARE. The same cannot be achieved for the nonlinear case and this presents one of the most difficult challenges in control theory. The solution to the HJB equation for a nonlinear system for many cases are impossible to evaluate analytically which has led to many methods existing in literature for obtaining sub-optimal controls though linear approximations.

The State Dependent Riccati Equation (SDRE) [17] is a relatively established approach closely related to the LQR methodology which requires the non-unique factorization of the dynamic equations known as the State Dependent Coefficient (SDC) parameterisation [36]. This captures the non-linearity of the system whilst representing the model in a linearly state dependent structure, allowing the nonlinear system to be solved as a series of Linear Time Invariant (LTI) systems. This implies that a sub-optimal control can be obtained through an online optimisation procedure by solving the ARE at each time step during the simulation.

Consider again a system of equations under the optimal control problem in (4) that has a zero-equilibrium ($f(0) = 0$) The dynamical system can be re-written in a psuedo-linear representation

$$\dot{x} = f(x) + g(x)u = A(x)x + B(x)u \tag{19}$$

where $A(x) \in \mathbb{R}^{n\times n}$ and $B(x) \in \mathbb{R}^{n\times m}$ are non-zero SDC matrices. The minimisation of the objective function will be designed to allow the control to drive all the states to zero and for SDRE, it can re-written as,

$$\min_{x,u} \quad J(x,u) = \frac{1}{2}\int_0^\infty x^TQ(x)x + u^Tu \; dt \tag{20}$$

where $Q(x) \in \mathbb{R}^{n\times n}$ is the positive definite state dependent penalty matrix that is user-defined and can vary over the course of the trajectory, providing an extra degree of freedom when solving the nonlinear problem. The user should be aware that whilst the cost in this form is quadratic in u, it is non-quadratic in x unless $Q(x)$ is constant.

By representing the control problem in this manner, the well-established LQR technique can be utilized in the process. The solution to the optimal control problem can be obtained by solving the algebraic SDRE during the trajectory,

$$P(x)A(x) + A(x)^TP(x) - P(x)B(x)B(x)^TP(x) + Q(x) = 0 \tag{21}$$

where $P(x)$ is a symmetric positive definite matrix. The corresponding control law is,

$$u(x) = -B(x)^T P(x)x \tag{22}$$

For the above control law to exist, A, B, Q and R must be continuous matrix-valued functions and

- $\{A(x), B(x)\}$ is required to be pointwise stabilizable pair.
- $\{A(x), Q(x)\}$ is required to be pointwise detectable pair.

The latter can be guaranteed by ensuring $Q(x)$ is positive definite for all $x \neq 0$. The SDC matrix $A(x)$ can be obtained an infinite number of ways if $n > 1$ and can therefore be a design choice. Depending on the parameterisation, this can have an effect on the size of the stable region of the system which can be further optimised for.

It has been shown in [10], SDRE is capable of ensuring a controlled system that is locally asymptotically stable and locally asymptotically optimal for a small neighbourhood near the origin. This makes intuitive sense since the SDRE will converge to the LQR solution as the system converges to the origin. The method therefore provides a systematic and effective approach in obtaining a sub-optimal solution to the nonlinear optimal control problems. The number of design parameters also offer user to adapt the controller to best suit the nature of the problem. However, the SDRE approach remains a linear approximation of the nonlinear problem whose accuracy depends on the time intervals of the LTI systems and the method can be computationally expensive since an the solution to the ARE is required at each time step which often is numerically obtained.

3.4 Dynamic Control Law

A more recent approach to the nonlinear optimal control problem have been proposed by Astolfi et al. [43] where the level of approximation of the nonlinear system is exactly quantifiable and in theory, can be minimized to offer greater performance. In this section, the methodology of a dynamic control law will be formulated which aims to construct an exact solution to the HJB equation dynamically without explicitly solving any PDE. The technique will rely on the *Lyapunov Stability* theory and the notion of algebraic $\bar{P}$ solutions. For the following formulation and the remainder of the thesis, the notation of the inequality $M < 0$ defines the matrix M as a negative definite matrix and $M \leq 0$ as negative semi-definite matrix. The same applies for positive definite and semi-definite matrices.

3.4.1 Lyapunov Stability

A *Lyapunov Function* is a function that can provide information of the asymptotic stability of a equilibrium point without explicit knowledge of the system under control. Consider the system

$$\dot{x} = f(x) + g(x)u \tag{23}$$

with a zero-equilibrium $f(0) = 0$. If there exist a continuous differentiable function $V(x)$ in a neighbourhood $\mathcal{U}$ satisfying the conditions; $V(x) > 0$ for all $x \in \mathcal{U}$, $V(0) = 0$ at the origin, and

$$\dot{V} = V_x(f(x) + g(x)u) < 0 \tag{24}$$

then the trajectory tends to zero and by *Lyapunov Stability* theory, the zero-equilibrium is locally asymptotically stable. For the case of a non-strict inequality $\dot{V} \leq 0$, *LaSalle's Invariant Principle* [27] provides an additional criterion for which the system's stability holds. This function $V(x)$ is known a *Lyapunov Function* [28].

3.4.2 Optimal Control

The SDRE mimics the LQR methodology and is able to provide an approximated value function that can be related back to the original optimal control problem and the HJB equation (13) via the solution to algebraic SDRE (21),

$$V(x) = \frac{1}{2}x^T P(x)x \tag{25}$$

by assuming a piece-wise linear system $\dot{x} = F(x)x + g(x)u$, given that there exist a zero-equilibrium, $f(0) = 0$. However, this value function is vaguely attached to the exact solution of the HJB equation in the nonlinear sense and can cause over-approximation. In this section, the notion of algebraic $\bar{P}$ solution will be introduced which are defined as matrix-valued functions satisfying certain conditions and this will provide the main component for deriving a dynamic control law that aims to explicitly approximate the solution to the optimal control problem.

In place of SDRE and the HJB equation, consider including some matrix-valued function $\Sigma(x)$ to the algebraic SDRE (21) where $\Sigma(x) = \Sigma(x)^T$ and $\Sigma(0) = \bar{\Sigma} > 0$, resulting in the algebraic matrix equation,

$$P(x)F(x) + F(x)^T P(x) + Q(x) + \Sigma(x) - P(x)g(x)g(x)^T P(x) = 0 \tag{26}$$

where $P(x) : \mathbb{R}^n \to \mathbb{R}^{n\times n}$ is a symmetric matrix-valued function of size n by n and $P(0) = \bar{P}$ is the solution to the new algebraic Riccati equation $\bar{P}\bar{F} + \bar{F}^T\bar{P} + Q - \bar{P}BB^T\bar{P} = 0$ where $F(0) = \bar{F}$, satisfying the conditions $\bar{P} = \bar{P}^T > 0$. $F(x)$ is some factorization of the dynamics $f(x) = F(x)x$, similar to the SDC matrix $A(x)$ but without any any additional requirements. This algebraic matrix equation becomes a condition that needs to be satisfied and its corresponding solution is known as the algebraic $\bar{P}$ solution to the HJB equation (13).

Further exploiting the property of this algebraic $\bar{P}$ solution, it can be shown that a value function exists in the following form,

$$V(x,\xi) = \frac{1}{2}x^T P(\xi)x + \frac{1}{2}(x-\xi)^T R(x-\xi) \tag{27}$$

where $R = R^T \in \mathbb{R}^{n\times n}$ is a penalty matrix, $\xi \in \mathbb{R}^n$ is an augmented state vector of size n and $P(\xi)$ is the solution to (26). This value function solves the following partial differential (PD) inequality exactly,

$$\mathcal{H}_k(x,\xi) \triangleq V_x f(x) - \frac{1}{2}V_x g(x)g(x)^T V_x^T + \frac{1}{2}q(x) - \kappa V_\xi V_\xi^T \leq 0 \tag{28}$$

for some $\kappa \in [\kappa^*, \infty]$ under the augmented system

$$\begin{aligned} \dot{x} &= f(x) + g(x)u \\ \dot{\xi} &= \kappa\left(\frac{1}{2}\nabla_\xi (P(\xi)x)^T x - R(x-\xi)\right) \equiv -\kappa V_\xi^T \end{aligned} \tag{29}$$

where ∇_ξ is a jacobian operator. The corresponding dynamic control law is defined as

$$u = -g(x)^T(P(\xi)x + R(x-\xi)) \equiv -g(x)^T V_x^T \tag{30}$$

The proof has been presented fully in [43] and will not shown here. More importantly, it was shown that for any positive definite $R > 0$, there exist a neighbourhood in which $V(x,\xi)$ is positive. The implies that by satisfying the inequality (28), if $R > 0$, $V(x,\xi)$ becomes a non-strict *Lyapunov Function* for the closed-loop system and by the conditions of *Lyapunov Stability* and *LaSalle's Invariant Principle*, the control guarantees local stability for a neighbourhood about the origin. The augmented system has been designed to achieve this, as $t \to \infty$, $\xi \to x$ which converges to the origin. The additional design parameters κ, R and ξ_0 can then be tuned to ensure the system remains in this stable neighbourhood throughout the trajectory.

Additionally, the PD inequality (28) is shown to be the exact HJB equation for the augmented system with a modified cost, therefore the value function (27) provides the solution for this extended problem. The inequality can also be rewritten as an equality,

$$\mathcal{H}_k + h(x,\xi) = 0 \tag{31}$$

where $h(x,\xi) \geq 0$. The original optimal control problem has been transformed into a different one with a new instantaneous running cost $L(x,\xi,u) = q(x) + 2h(x,\xi) + u^T u$. But it can be seen that as the system is driven to the zero-equilibrium, the original HJB equation is retained and the value

function will correspond to the solution to the classical optimal control problem in an extended state space (x, ξ). The additional cost $h(x, \xi)$ can therefore be recognized as the level of approximation to be minimised through the dynamic control. Since the original objective has been changed under the new system, the solution is no longer optimal with respect to the original problem. However, the additional freedom offered by the parameters κ, R and ξ_0 can be designed in a manner that can lead to a lower overall approximation and a near optimal control strategy with respect to the real system, offering opportunities for further optimisation. This is one of the key advantages of this dynamic control law.

Solving the HJB inequality provides a significant easier challenge than solving the original HJB equation since it does not require an explicit solution of a PDE. As $\Sigma(x)$ does not need to be explicitly defined, the theory allows (26) to be also written as an inequality,

$$P(x)F(x) + F(x)^T P(x) + Q(x) - P(x)g(x)g(x)^T P(x) < 0 \tag{32}$$

and therefore without the need to obtain an exact matrix-valued function $P(x)$. The difficulty of obtaining such solution can be dependent on the system dynamics. Multiple algebraic $\bar{P}$ solution can exist and their corresponding local region of stability can vary, but this can also be compensated for through the extended system tuning parameters.

This presents the methodology for a powerful nonlinear optimal control tool which solves the HJB equation through a dynamic control law and is able to provide guaranteed stability and performance. In addition to the freedom of parameters, the control mechanism is also a less computationally expensive option since the algebraic $\bar{P}$ solution can be obtained offline.

3.5 Direct Collocation

An alternate method for solving optimal control problem is to transcribe the problem into a nonlinear program and utilize a general Nonlinear Programming (NLP) solver to optimise for the control policy. One commonly used transcription method is Direct Collocation [34], which has been previously mentioned and applied in [14] for cancer treatment to compute the optimal control. This transcription method involves approximating the states and control using piece-wise continuous polynomials and enforcing the dynamics to be satisfied at a series of collocation points during the optimisation, so that the complete trajectory can be smoothly represented.

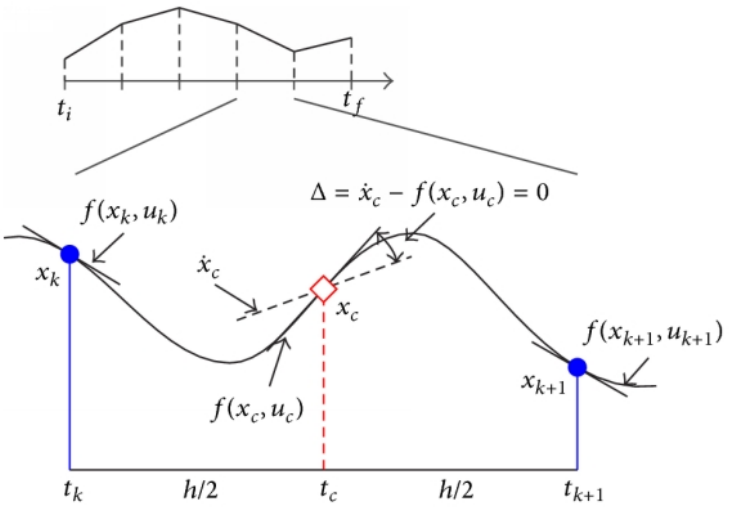

Figure 1: The collocation point (subscript 'c') conditions between two intervals of time t_k and t_{k+1}. [45].

In order to apply the transcription, time is first discretised into N intervals as $t = [t_1 ... t_k ... t_N]$ with a constant discretisation $h = t_{k+1} - t_k$. A *Hermite-Simpson* collocation [39] has been used for the

following formulation which involves approximating the states as a cubic polynomial and the control as piece-wise linear between the intervals of the trajectory, and the collocation point will be taken at the mid-point as shown in Figure 1. The dynamics can now be approximated between the interval $[t_k, t_{k+1}]$. Time and control at the collocation point are relatively easy to derive as they follow a linear function,

$$t_{k,c} = \frac{t_k + t_{k+1}}{2} \quad \text{and} \quad u_{k,c} = \frac{u_k + u_{k+1}}{2}$$

The states and the dynamics can then be approximated as a third order polynomial,

$$\begin{aligned} x(t) &= a_{k,0} + a_{k,1}t + a_{k,2}t^2 + a_{k,3}t^3 \\ \dot{x}(t) &= a_{k,1} + 2a_{k,2}t + 3a_{k,3}t^2 \end{aligned} \tag{33}$$

where $a_{k,i}$ are polynomial coefficients. For clarity, the interval have been shifted from $[t_k, t_{k+1}]$ to between $[0, h]$ to calculate the coefficients.

$$\begin{bmatrix} x(0) \\ \dot{x}(0) \\ x(h) \\ \dot{x}(h) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & h & h^2 & h^3 \\ 0 & 1 & 2h & 3h^2 \end{bmatrix} \begin{bmatrix} a_{k,0} \\ a_{k,1} \\ a_{k,2} \\ a_{k,3} \end{bmatrix} \quad \text{where} \quad \begin{aligned} x(0) &= x_k \\ \dot{x}(0) &= \dot{x}_k = f(x_k, u_k) \\ x(h) &= x_{k+1} \\ \dot{x}(h) &= \dot{x}_{k+1} = f(x_{k+1}, u_{k+1}) \end{aligned} \tag{34}$$

Taking the inverse reveals the coefficients in terms of h, the states (x_k) and the dynamics ($\dot{x}_k$). The coefficients can then be substituted back into (33) with the rates replaced by $\dot{x}_k = f(x_k, u_k)$, finally giving the approximated states and dynamics at the collocation point,

$$\begin{aligned} x_c = x(h/2) &= \frac{1}{2}(x_k + x_{k+1}) + \frac{h}{8}[f(x_k, u_k) - f(x_{k+1}, u_{k+1})] \\ \dot{x}_c = \dot{x}(h/2) &= -\frac{3}{2h}(x_k - x_{k+1}) - \frac{1}{4}[f(x_k, u_k) + f(x_{k+1}, u_{k+1})] \end{aligned} \tag{35}$$

The collocation condition can now be enforced to ensure that the approximated time derivatives at the collocation point match with the real system which is enforced as a defect factor Δ_k at each k^{th} interval,

$$\begin{aligned} \Delta_k &= \dot{x}_c - f(x_c, u_c) \\ &\rightarrow x_k - x_{k+1} - \frac{h}{6}\left[f(x_k, u_k) + f(x_{k+1}, u_{k+1}) + 4f(x_c, u_c)\right] = 0 \end{aligned} \tag{36}$$

For the case of a quadratic cost, a trapezoidal integration method can be used to transcribe the optimal control objective function. The full static optimisation objective can be written as

$$\begin{aligned} \min_{x,u} \quad & J(x(t), u(t)) = \phi(x_N, u_N) + \frac{1}{2}\sum_{k=1}^{N-1}(x_k^T Q x_k + u_k^T R_c u_k + x_{k+1}^T Q x_{k+1} + u_{k+1}^T R_c u_{k+1})h \\ \text{s.t.} \quad & \Delta_k = 0 \\ & u_{min} \leq u_k \leq u_{max} \\ & x_{min} \leq x_k \leq x_{max} \\ & C_{eq}(x_k, u_k) = 0 \\ & C(x_k, u_k) \leq 0 \end{aligned} \tag{37}$$

where ϕ is the final cost which in the quadratic case is $\phi(x_N, u_N) = x_N^T Q x_N + u_N^T R_c u_N$. The initial condition can be enforced in the equality constraint $C_{eq} = x(0) - x_0 = 0$, along with other state or control constraints. The entire trajectory can then be optimised using a NLP solver. This is a very robust optimisation method that can deal with many variations of the optimal control problem and returns a guaranteed local optimal solution. However, it is an extremely computationally demanding option where large amount of constraints are introduced in a high dimensional problem and the accuracy can be highly dependent on N. This therefore is a great technique for confirming the existence of an optimal solution or comparing with other sub-optimal strategies to analyse their relative control optimality which will be the purpose of this tool for the project.

4 Model Analysis

A preliminary analysis of the drug-free model, described in the literature review, will be performed in this section to provide some initial insights about its behaviour and the difficulties that may arise when obtaining the desired control strategy.

$$\begin{aligned}
\dot{N} &= r_2 N(1 - b_2 N) - c_4 TN - a_3 MN \\
\dot{T} &= r_1 T(1 - b_1 T) - c_2 IT - c_3 TN - a_2 MT \\
\dot{I} &= s + \frac{\rho IT}{\alpha + T} - c1IT - d_1 I - a_1 MI \\
\dot{M} &= -d_2 M + u(t)
\end{aligned} \tag{38}$$

This model does not focus on a specific type of cancer and for our study, the parameters has been taken from [14] (displayed in Table 1). The cells are normalized so that the carrying capacity of normal cell is 1 (i.e. $b_2 = 1$). The reference paper provided justification for these values and its relevance to clinical data, therefore the parameters have been kept constant throughout the control analysis. Since the model has been repeatedly used in literature without modification for optimal control, it is therefore assumed that the accuracy of the model is sufficient to carry out the implementation of the dynamic control law. The remainder of the report will focus on the performance of the discussed control strategies on this particular cancer model.

Parameters	Description	Value
a_1	Fractional normal cell kill by chemotherapy	0.2
a_2	Fractional tumor cell kill by chemotherapy	0.3
a_3	Fractional immune cell kill by chemotherapy	0.1
b_1^{-1}	Tumor cell carrying capacity	1.0
b_2^{-1}	Normal cell carrying capacity	1.0
c_1	Fractional tumor cell kill by immune cells	1.0
c_2	Fractional immune cell kill by tumor cells	0.5
c_3	Fractional tumor cell kill by normal cells	1.0
c_4	Fractional normal cell kill by tumor cells	1.0
d_1	Death rate of immune cell	0.2
d_2	Death rate of chemotherapy drug	1.0
r_1	Tumor cell growth rate	1.5
r_2	Normal cell growth rate	1.0
s	Steady source for immune cells	0.33
α	Immune threshold rate	0.3
ρ	Immune response rate	0.01

Table 1: Model parameters from [14].

4.1 Drug-free Equilibria

Equilibrium points constitute the state in which the system will remain constant with time without further control inputs. The drug-free equilibrium points are evaluated from a system without any drug inputs ($u(t) = 0$) and with zero initial drug concentration in the blood ($M_0 = 0$). The patient is considered cured if the system remains in a tumor-free equilibrium or a co-existing equilibrium where there are only harmlessly small amount of tumor cells left in the body.

By equating each of the cell growth rate to zero, the equilibrium points can be found by re-arranging and solving the dynamic equations in (38),

$$\dot{N} = 0 \rightarrow \begin{cases} N = 0 \\ N = \frac{1}{b_2} - \frac{c_4}{r_2} T \quad = g(T) \end{cases} \tag{39}$$

$$\dot{T} = 0 \rightarrow \begin{cases} T = 0 \\ T = \frac{1}{b_1} - \frac{c_2}{r_1 b_1} I - \frac{c_3}{r_1 b_1} N \end{cases} \tag{40}$$

$$\dot{I} = 0 \quad \rightarrow \quad I = \frac{s(\alpha + T)}{(c_1 T + d_1)(\alpha + T) - \rho T} = f(T) \tag{41}$$

and this results in three equilibrium types,

- Tumor-free: $(N, T, I) = (\frac{1}{b_2}, 0, \frac{s}{d_1})$
- Dead: $(N, T, I) = (0, 0, \frac{s}{d_1})$
  $(N, T, I) = (0, T, f(T))$
- Co-existing: $(N, T, I) = (g(T), T, f(T))$

Depending on the parameters of the model, the coexisting equilibrium can contain numerous solutions. The desired equilibria are therefore the tumor-free equilibrium or the co-exisitng equilibrium where T is small and $g(T)$ is close to 1. To identify the stability of each equilibrium, the linearised model or the jacobian matrix is required,

$$J = \begin{bmatrix} r_2(1 - 2b_2 N) - c_4 T & -c_4 N & 0 \\ -c_3 T & r_1(1 - 2b_1 T) - c_2 I - c_3 N & -c_2 T \\ 0 & \frac{\rho I}{\alpha + T} - \frac{\rho I T}{(\alpha + T)^2} - c_1 I & \frac{\rho I}{\alpha + T} - c_1 I - d_1 \end{bmatrix} \tag{42}$$

and its eigenvalues evaluated at each equilibrium point can determines their stability. This study can be found in [11]. It was shown that with the given model parameters, five equilibrium points existed, two of which are locally stable and are of significance to the cancer treatment procedure, the desired tumor-free state $(N, T, I) = (1, 0, 1.65)$, and an unhealthy coexisting state $(N, T, I) = (0.44, 0.56, 0.44)$.

4.2 Drug-free Simulation

To visually interpret these stable equilibrium points, simulations using ode45 function in MATLAB [33] have been run with two initial condition cases, a high and low tumor cell concentration.

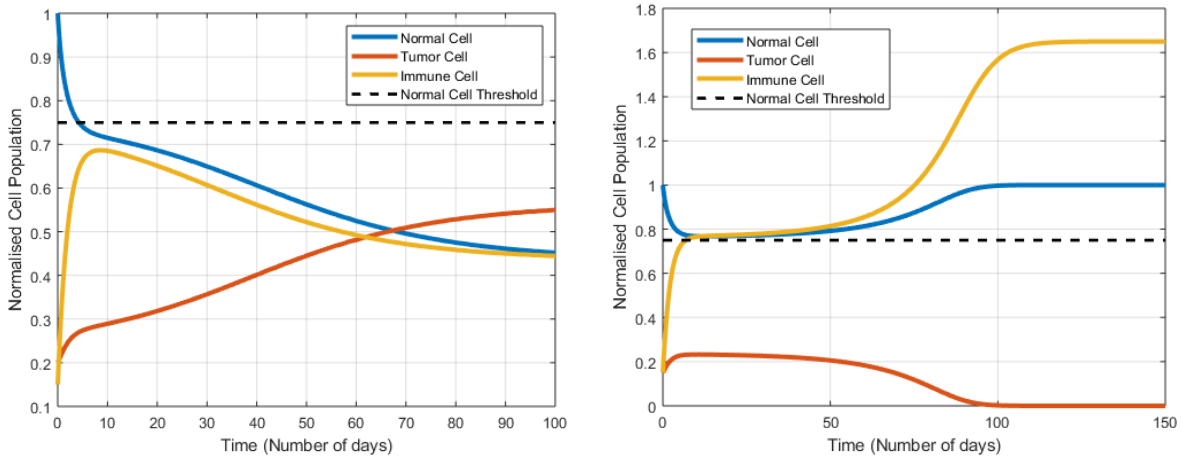

Figure 2: Drug-free simulations with two initial conditions, $T_0 = 0.2$ on the left, $T_0 = 0.15$ on the right.

For the unhealthy outcome on the left, the initial tumor concentration was too high or insufficient immune response existed to react fast enough and cure the patient. This further demonstrate the importance of the immune system and the benefit of high immune concentration in the process of cancer treatment. However, to avoid a self-sustained system seen on the right, the initial condition under consideration for this optimal control problem will be the high tumor case $(N_0, T_0, I_0) = (1, 0.2, 0.15)$, provided by [14], requiring the definite need for chemotherapy. The aim of the controller is to drive

this system to the healthy end condition with minimal drug administration to reduce the toxicity effect of chemotherapy and avoid the deterioration of normal cells.

This toxicity level has been incorporated by the author in the form of control bounds between $0 \leq u \leq 1$ and maintaining a normal cell level above 0.75. It can be seen on the right of Figure 2 that even under the uncontrolled self-curing system, the normal cell concentration is bordering on the threshold limit and is likely to fall below this desired constraint with any drug input. These constraints therefore offer additional challenges in searching for the optimal control and a smooth continuous control law may not exist under such conditions. The ability of each control strategy to satisfy state and control constraint has not been a key objective of this project but is another important aspect of optimal control theory which will be discussed in future works in section 9.2.

5 Initial Control

Prior to implementing control, the system must be in the correct form. For most control strategies, it is required that the system contains a zero-equilibrium, namely $f(0) = 0$. The system is therefore shifted in the error coordinates, $\{x_1, x_2, x_3, x_4\} = \{N - 1/b_2, T, I - s/d_1, M\}$ with respect to the tumor-free equilibrium $(N, I, T, M) = (1/b_2, 0, s/d_1, 0)$. The new system of equations become

$$\begin{aligned}
\dot{x_1} &= -r_2 x_1(1 + x_1 b_2) - \frac{c_4}{b_2} x_2 - c_4 x_1 x_2 - a_3 x_1 x_4 - \frac{a_3}{b_2} x_4 \\
\dot{x_2} &= r_1 x_2(1 - b_1 x_2) - \left(\frac{c_2 s}{d_1} + \frac{c_3}{b_2}\right) x_2 - c_3 x_1 x_2 - c_2 x_2 x_3 - a_2 x_4 x_2 \\
\dot{x_3} &= -\frac{c_1 s}{d_1} x_2 - d_1 x_3 - \frac{a_1 s}{d_1} x_4 + \frac{\rho s}{d_1} \frac{x_2}{\alpha + x_2} + \rho \frac{x_2 x_3}{\alpha + x_2} - c_1 x_2 x_3 - -a_1 x_4 x_3 \\
\dot{x_4} &= -d_2 x_4 + u(t)
\end{aligned} \tag{43}$$

The complete optimal control problem can now be formulated with the objective discussed in the previous chapter. To achieve the desired performance, a quadratic cost function has been chosen,

$$\min_{x,u} \quad J(x(t), u(t)) = \frac{1}{2} \int_0^\infty x^T Q x + u^T R_c u \; dt \tag{44}$$

where the state penalty matrix Q and control penalty R_c are

$$Q = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 100 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.1 \end{bmatrix} \quad \text{and} \quad R_c = 1$$

which have been provided by [22] in the analysis of SDRE proven to give guaranteed performance. This penalty consists of a combination of tumor cells and drug concentration in the blood and will be adopted as a performance index for different control strategies.

5.1 Linear Quadratic Regulator

For this project, the *lqr* function in MATLAB [32] has been applied to solve the ARE in (17) which uses a numerical algorithm described in [1]. The LQR method returns global optimal and stable control for the linear case. It's application in a nonlinear system requires the linear approximation and in many cases, this provides a crude approximation and often results in poor control. The procedure for applying the LQR method for the tumor model in (43) will involve linearising the system about the origin and obtaining the optimal control gains that offer local stability.

The state and control matrix of the linearised system (jacobian) evaluated at the origin are

$$A = \begin{bmatrix} A_{11} & A_{12} & 0 & A_{14} \\ A_{21} & A_{22} & A_{23} & A_{24} \\ 0 & A_{32} & A_{33} & A_{34} \\ 0 & 0 & 0 & A_{44} \end{bmatrix}\Bigg|_{(0,0,0,0)} = \begin{bmatrix} -1 & -1 & 0 & -0.1 \\ 0 & -0.325 & 0 & 0 \\ 0 & -1.595 & -0.2 & -0.33 \\ 0 & 0 & 0 & -1 \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \tag{45}$$

where

$$
\begin{aligned}
A_{11} &= -r_2(1+2b_2x_1) - c_4x_2 - a_3x_4 \\
A_{12} &= -\frac{c_4}{b_2} - c_4x_1 \\
A_{14} &= -a_3x_1 - \frac{a_3}{b_2} \\
A_{21} &= -c_3x_2 \\
A_{22} &= r_1(1-2b_1x_2) - \left(\frac{c_2s}{d_1} + \frac{c_3}{b_2}\right) - c_3x_1 - c_2x_3 - a_2x_4 \\
A_{23} &= -c_2x_2 \\
A_{24} &= -a_2x_2 \\
A_{32} &= -\frac{c_1s}{d_1} + \frac{\rho s/d_1 + \rho x_3}{(\alpha + x_2)} - \frac{(\rho s/d_1 + \rho x_3)x_2}{(\alpha + x_2)^2} - c_1x_3 \\
A_{33} &= -d_1 + \frac{\rho x_2}{\alpha + x_2} - c_1x_2 - a_1x_4 \\
A_{34} &= -\frac{a_1s}{d_1} - a_1x_3 \\
A_{44} &= -d_4
\end{aligned}
$$

and substituting this linear system with the cost function (44) into the ARE (17) results in the P solution with the corresponding feedback control

$$
P = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 153.8462 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.0488 \end{bmatrix} \quad \rightarrow \quad K = \begin{bmatrix} 0 & 0 & 0 & 0.0488 \end{bmatrix} \quad \rightarrow \quad u(t) = -0.0488x_4(t) \tag{46}
$$

This particular P solution is a diagonal matrix and resulted in a control that only has a dependency on state x_4 which is the drug concentration in the blood. By considering the closed-loop controlled dynamics $A_{cl} = (A - BK)x$,

$$
\dot{x} = A_{cl}x = \begin{bmatrix} A_{11} & A_{12} & 0 & A_{14} \\ A_{21} & A_{22} & A_{23} & A_{24} \\ 0 & A_{32} & A_{33} & A_{34} \\ 0 & 0 & 0 & A_{44} - 0.0488 \end{bmatrix} x \tag{47}
$$

It can be seen that the only effect of this control is in the $\dot{x}_4$ dynamics and if the initial drug concentration in the blood is zero (already at equilibrium for this state), the resulting control is also zero with the dynamics of the other cells unaffected. The controller therefore only aims to drive x_4 to equilibrium and disregards the other states. This suggests that near the origin, the system is stable enough in the tumor state such that a control depending on this state is not required given this particular cost function. Evidently, an optimal control obtained from a linearised system at origin cannot be extended to the full nonlinear problem and illustrates the poor performance of the LQR method.

In order to gain control and provide a P solution with non-diagonal terms, a penalty in the normal cell x_1 has been included to introduce a control dependency on the other states. The new state penalty matrix becomes

$$
Q = \begin{bmatrix} Q_{11} & 0 & 0 & 0 \\ 0 & 100 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.1 \end{bmatrix} \tag{48}
$$

where Q_{11} is a variable that can be changed to find the best controller performance. By doing so, the resulting P solution and the control are in the form of

$$P = \begin{bmatrix} P_{11} & P_{12} & 0 & P_{14} \\ P_{12} & P_{22} & 0 & P_{24} \\ 0 & 0 & 0 & 0 \\ P_{14} & P_{24} & 0 & P_{44} \end{bmatrix} \quad \rightarrow \quad u(t) = -P_{14}x_1(t) - P_{24}x_2(t) - P_{44}x_4(t) \tag{49}$$

with the corresponding closed-loop system,

$$\dot{x} = A_{cl}x = \begin{bmatrix} A_{11} & A_{12} & 0 & A_{14} \\ A_{21} & A_{22} & A_{23} & A_{24} \\ 0 & A_{32} & A_{33} & A_{34} \\ -P_{14} & -P_{24} & -P_{34} & A_{44} - P_{44} \end{bmatrix} x \tag{50}$$

The difference now is that the closed-loop dynamics of $\dot{x}_4$ (only state that responds to chemotherapy directly) has a dependency on other states but at the cost of altering state penalty matrix. The resulting control is studied under the effect of the parameter Q_{11} and the corresponding control and tumor levels during the treatment have been plotted with a summary of their control performance tabulated in Table 2.

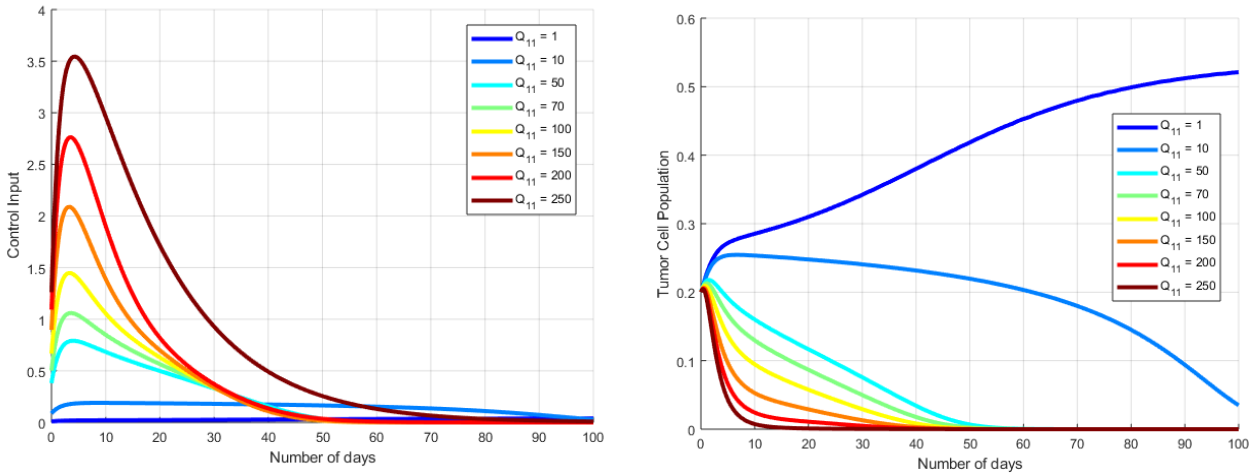

Figure 3: Plots of the control (on the left) and tumor response (on the right) for different Q_{11} in the LQR optimal control problem.

Indeed by introducing a normal cell penalty, a stabilising control was established through LQR. The exact effect of different Q_{11} on the LQR's performance is much more apparent in Table 2.

Q_{11}	0	1	10	50	70	100	150	200	250
Final Cost (J)	-	-	234	73.0	72.5	80.8	110.4	171	369

Table 2: The effect of varying Q_{11} on the LQR control performance.

For the cases $Q_{11} = 0$ or 1, there are none or insufficient control in order to successfully treat the patient, causing the system to be driven to the unhealthy equilibrium. As the penalty gain increased, it resulted in a larger peak in the control input, eliminating the tumor cells in a shorter time and hence lowering the cost. However, since the cost now considers the normal cells, it eventually rises again due to a reduction in normal cell levels and heavy doses of chemotherapy. This resulted in an optimal performance achieved approximately around $Q_{11} = 70$, producing a final cost of $J = 72.5$. The resulting performance of this controller has been displayed in Figure 4,

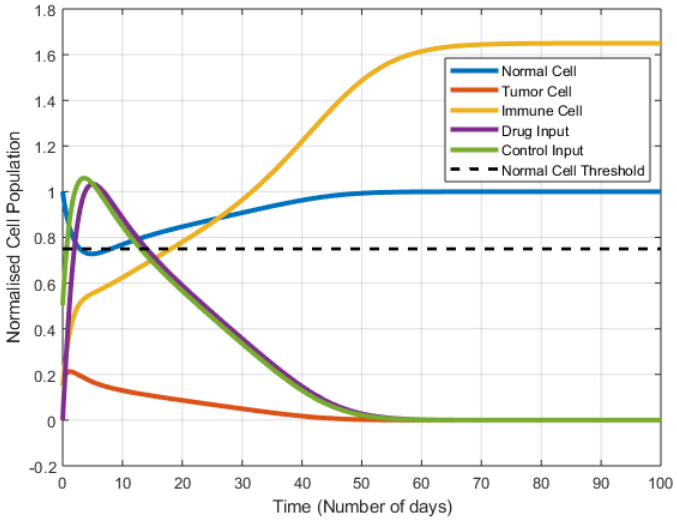

Figure 4: The evolution of the states with time for $Q_{11} = 70$ using LQR.

In this case, the controller obtained for a linearised system about the origin was able stabilise the nonlinear system. However, this could only be achieved by modifying the objective function which resulted in a poor final cost. The LQR control provides local asymptotic stability for a region about the origin but it does not guarantee stabilising control elsewhere, as already shown in the uncontrolled case. Here, it has been demonstrated that a linear approximation within a highly nonlinear system caused the control theory to breakdown. Contrarily, the SDRE can provide a much improved approximation and builds upon the LQR methodology which can result in a more optimally controlled system.

5.2 State Dependent Riccati Equation

The performance of the SDRE technique applied to the dynamical system in (43) has already be extensively studied by Itik et al. in [22]. The main result will be summarized in this section. Prior to implementing the SDRE control, selecting the SDC matrices $A(x)$ and $B(x)$ are required. For a nonlinear system, the SDC matrices are not unique and there exist infinitely many possibilities for the parametrisation, shown in [36]. However, the system in the linear sense must be point-wise controllable in order to apply the LQR methodology. Since the control matrix is constant, there is only one option for $B(x)$ which is equivalent to $g(x) = [0\ ,0\ ,0\ ,1]^T$. One version of the SDC matrix $A(x)$ that satisfies the controllability condition was shown in the study and will be adopted in this analysis,

$$A(x) = \begin{bmatrix} -r_2(1+b_2x_1) & -c_4(x_1+\frac{1}{b_2} & 0 & -a_3(x_1+\frac{1}{b_2}) \\ -c_3x_2 & r_1(1-b_1x_2) - \left(\frac{c_2s}{d_1}+\frac{c_3}{b_2}\right) & -c_2x_2 & -a_2x_2 \\ 0 & \frac{\rho(x_3+\frac{s}{d_1})}{\alpha+x_2} - c_1(x_3+\frac{s}{d_1}) - x_4 & -d_1 & -a_1(x_3+\frac{s}{d_1}) + x_2 \\ 0 & 0 & 0 & -d_2 \end{bmatrix} \tag{51}$$

The SDRE technique is implemented by solving the ARE (17) at every time step. At each time step, the $A(x)$ matrix of the dynamics will be updated with current states and the optimal control will be obtained for that local LTI system to forward the dynamics for the next time step. This procedure is repeated until the final time or the equilibrium point have been reached. As the time step is reduced, the series of LTI systems converges to the nonlinear system.

To simulate the SDRE control, the MATLAB function ode45 has been utilized which takes variable time steps depending the rate of change of the states. The same time step can also be suitable for the sampling rate at which the $A(x)$ matrix is updated since if the problem is near constant, the system can be represented as approximately linear and an update is not required. This time step is

an additional parameter that can be tuned to increase accuracy. The same MATLAB function *lqr* has been utilized to solve the ARE.

For the optimal control problem, the penalty matrix $Q(x)$ and penalty $R_c(x)$ have been kept the same as in (44) and remained constant throughout the trajectory.

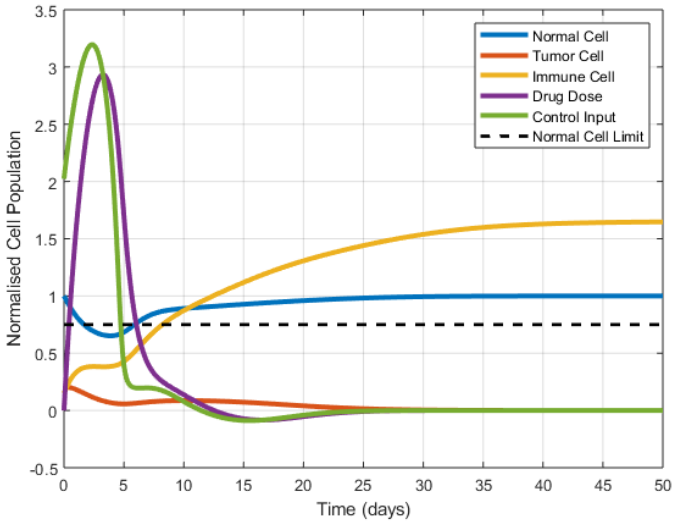

Figure 5: The evolution of the the states with time for SDRE with constant penalty matrices.

The result for SDRE shown in Figure 5 display similar characteristics as the modified LQR controller, both of which creates a sharp rise in the drug input early on in the treatment process. However, SDRE was able to provide a more optimally controlled strategy where the control effort is much more rigorous, rising rapidly to a heavy dose before dropping quickly again to near zero. The effect of this on the cells is evident where the rise in control caused a dip in all cells. However, in doing so, the tumor cell population was able to drop sufficiently low to allow the killing effect of the growing immune cell to take over, thus completing the treatment without the need for further control effort.

Without considering any constraints, the SDRE approach provided a final cost of $J = 27.16$ which is a significant improvement upon the LQR control due to a more valid approximation of the nonlinear system. However, this control strategy is designed by considering the optimal control for a local LTI system and therefore remains a sub-optimal approach.

Summary of LQR and SDRE

In the analysis of LQR, it is clear that for this optimal control problem, the linear approximation of the nonlinear system was inadequate which has led to a failed control strategy. Stable control was obtained after modification to the objectives by incorporating a normal cell penalty, but this resulted in an unsatisfactory performance with a high cost.

On the other hand, SDRE's simplicity, degree of flexibility and guaranteed performance makes it an excellent controller and a widely used technique in the domain of optimal control for cancer treatment. By approximating the nonlinear system as a series of LTI systems and adopting the LQR methodology, it is able to extend the region stability beyond the capabilities of LQR. The result provides the benchmark for performance comparison of further control strategies applied to this problem. In the following chapters, a more recent technique will be explored which aims to quantify the level of approximation that exists in SDRE and minimise it through a dynamic control law.

6 Dynamic Control Law

The formulation of a dynamic control law with respect to the cancer treatment problem will be presented in this chapter. This method has the ability to improve upon the SDRE performance by quantifying the approximations and providing additional control freedom which can be tuned to further optimise the control. This method requires the problem to be solved offline prior to its implementation which also means that it's a less computationally expensive option. The methodology of obtaining such a solution and its implementation will be presented in sequential order and the details of the challenges involved will also be discussed.

6.1 Algebraic $\bar{P}$ Solution

A key component of this control law is obtaining the solution to the inequality (32) which provides the algebraic $\bar{P}$ solution and is the tool used to solve the HJB inequality (28). To do so requires satisfying the properties of a positive definite matrix. Therefore, before presenting the solution, the concept of Diagonal Dominance and Schur Complement will be introduced which will be utilized in the solution formulation.

6.1.1 Positive Definiteness

A real symmetric matrix $M \in \mathbb{R}^{n\times n}$ is said to be positive definite if the scalar $z^T M z$ is positive for all nonzero vector $z \in \mathbb{R}^n$ [46]. To confirm the positive definiteness of a symmetric matrix, its corresponding eigenvalues should be strictly positive. There are various techniques that can be used to demonstrate positive definiteness. Two of which used in this formulation are Diagonal Dominance and Schur Complement.

i) A **diagonally dominant** square matrix M satisfies the property,

$$|M_{ii}| \geq \sum_{i\neq j} |M_{ij}| \tag{52}$$

for all i. In other words, if the absolute sum of the non-diagonal entries of the matrix is smaller or equal to the absolute diagonal entries in each of its row, then the matrix is diagonally dominant. Then, if the diagonal entries are non-negative, then the matrix is said to be positive definite [20].

ii) To utilize the concept of **Schur Complement**, let a square symmetric matrix M be represented by a 2×2 block matrix,

$$M = \begin{bmatrix} A & B \\ B^T & D \end{bmatrix} \tag{53}$$

where A, D are symmetric matrices. If any of the following properties are true, then by schur complement, M is said to be a positive definite matrix [6].

- if D is invertible, $D > 0$ and $A - BD^{-1}B^T > 0$.
- if A is invertible, $A > 0$ and $D - BA^{-1}B^T > 0$

6.1.2 Problem Formulation

The inequality (32) can be represented and solved in two ways, either in the following form,

$$p(x)f(x) + \frac{1}{2}x^T Q x - \frac{1}{2}p(x)g(x)g(x)^T p(x) < 0 \tag{54}$$

solving a scalar inequality equation involving the dynamics $\dot{x} = f(x)$ and a $\mathcal{C}^1$ mapping $p(x) : \mathbb{R}^n \to \mathbb{R}^{1\times n}$ with the corresponding matrix-valued algebraic $\bar{P}$ solution obtained through factorization $p(x) = P(x)x$, or in the factorized form,

$$-M(x) \triangleq P(x)F(x) + F(x)^T P(x) + Q - P(x)g(x)g(x)^T P(x) < 0 \tag{55}$$

solving a matrix inequality equation to obtain the matrix-valued function $P(x) : \mathbb{R}^n \to \mathbb{R}^{n\times n}$, involving a $F(x)$ parameterisation where $\dot{x} = F(x)x$. Either way, the following properties must be true; the inequality has to be satisfied for all $x \in \mathcal{X}$ in some neighbourhood about the origin, $P(x) = P(x)^T$ and

$$-\bar{M} \triangleq \bar{P}\bar{F} + \bar{F}^T\bar{P} + Q - \bar{P}BB^T\bar{P} < 0 \tag{56}$$

where $P(0) = \bar{P} = \bar{P}^T > 0$ defines the solution at the origin. These conditions provide some freedom in choosing the algebraic $\bar{P}$ solution. But mainly, the quality of the control will be determined by the size of the neighbourhood $\mathcal{X}$ in which the stability conditions are satisfied. The ability to satisfy these conditions for all x would allow for a larger region of stability and reduce the difficulty in searching for the optimal dynamic control parameters.

For the system in (43), it was found that the search for an algebraic $\bar{P}$ solution in the factorized form of the inequality was a less stringent process. For the following solution formulation, the matrix inequality equation in (55) will be represented by the matrix $M(x)$ and the objective is to find the solution for which $M(x)$ is positive definite.

6.1.3 Solution Formulation

First it is required to obtain the parametrisation $F(x)$. Similar to the $A(x)$ matrix of SDRE, $F(x)$ too is non-unique but they are not equivalent since $F(x)$ has no additional requirements. The linear representation should be chosen such that the positive definiteness of $M(x)$ can be easily proven either by diagonal dominance or schur complement without complication. Generally, diagonal dominance is a desired property for positive definite matrices and therefore in order to keep the non-diagonal entries zero when possible, the following $F(x)$ matrix was chosen,

$$F(x) = \begin{bmatrix} F_{11} & F_{12} & 0 & F_{14} \\ 0 & F_{22} & 0 & 0 \\ 0 & F_{32} & F_{33} & F_{34} \\ 0 & 0 & 0 & F_{44} \end{bmatrix} \tag{57}$$

where

$$\begin{aligned}
F_{11} &= -r_2(1 + b_2x_1) - c_4x_2 - a_3x_4 \\
F_{12} &= -\frac{c_4}{b_2} \\
F_{14} &= -\frac{a_3}{b_2} \\
F_{22} &= r_1(1 - 2b_1x_2) - \left(\frac{c_2s}{d_1} + \frac{c_3}{b_2}\right) - c_3x_1 - c_2x_3 - a_2x_4 \\
F_{32} &= -\frac{c_1s}{d_1} + \frac{\rho s}{d_1(\alpha + x_2)} \\
F_{33} &= -d_1 + \frac{\rho x_2}{\alpha + x_2} - c_1x_2 - a_1x_4 \\
F_{34} &= -\frac{a_1s}{d_1} \\
F_{44} &= -d_2
\end{aligned}$$

The penalty function Q is designed by the user and $P(x)$ is the unknown matrix-valued function to be solved with 10 unknown entries,

$$Q = \begin{bmatrix} Q_{11} & 0 & 0 & 0 \\ 0 & Q_{22} & 0 & 0 \\ 0 & 0 & Q_{33} & 0 \\ 0 & 0 & 0 & Q_{44} \end{bmatrix} \quad \text{and} \quad P(x) = \begin{bmatrix} P_{11} & P_{12} & P_{13} & P_{14} \\ P_{12} & P_{22} & P_{23} & P_{24} \\ P_{13} & P_{23} & P_{33} & P_{34} \\ P_{14} & P_{24} & P_{34} & P_{44} \end{bmatrix} \tag{58}$$

where $Q > 0$ and $P(0) > 0$. Now the optimal control problem can be substituted into inequality (55), resulting in an algebraic matrix inequality to be solved which can be written as (for clarity, model parameters in Table 1 have been substituted in),

$$M(x) = \begin{bmatrix} M_{11} & M_{12} & M_{13} & M_{14} \\ M_{12} & M_{22} & M_{23} & M_{24} \\ M_{13} & M_{23} & M_{33} & M_{34} \\ M_{14} & M_{24} & M_{34} & M_{44} \end{bmatrix} > 0 \tag{59}$$

where

$$\begin{aligned}
M_{ii}: \quad M_{11} &= P_{14}^2 + 2P_{11}(x_1 + x_2 + 0.1x_4 + 1) - Q_{11} \\
M_{22} &= P_{24}^2 + 2P_{12} - P_{23}\left(\frac{0.033}{(x_2 + 0.3)} - 3.3\right) + 2P_{22}(x_1 + 1.5x_2 + 0.5x_3 + 0.3x_4 + 0.325) - Q_{22} \\
M_{33} &= P_{34}^2 + 2P_{33}\left(x_2 + 0.2x_4 - \frac{0.01x_2}{(x_2 + 0.3)} + 0.2\right) - Q_{33} \\
M_{44} &= P_{44}^2 + 2P_{44} + 0.2P_{14} + 0.66P_{34} - Q_{44}
\end{aligned}$$

$$\begin{aligned}
M_{\substack{ij \\ i \neq j}}: \quad M_{12} &= P_{11} + P_{12}(x_1 + 1.5x_2 + 0.5x_3 + 0.3x_4 + 0.325) - P_{13}\left(\frac{0.0165}{(x_2 + 0.3)} - 1.65\right) + \\
&\qquad P_{12}(x_1 + x_2 + 0.1x_4 + 1) + P_{14}P_{24} \\
M_{13} &= P_{13}\left(x_2 + 0.2x_4 - \frac{0.01x_2}{(x_2 + 0.3)} + 0.2\right) + P_{13}(x1 + x2 + 0.1x4 + 1) + P_{14}P_{34} \\
M_{14} &= 0.1P_{11} + 0.33P_{13} + P_{14}(x_1 + x_2 + 0.1x_4 + 2) + P_{14}P_{44} \\
M_{23} &= P_{13} + P_{23}(x_1 + 1.5x_2 + 0.5x_3 + 0.3x_4 + 0.325) + P_{23}(x_2 + 0.2x_4 - \frac{0.01x_2}{(x_2 + 0.3)} + 0.2) \\
&\qquad - P_{33}\left(\frac{0.0165}{(x_2 + 0.3)} - 1.65\right) + P_{24}P_{34} \\
M_{24} &= 0.1P_{12} + P_{14} + 0.33P_{23} + P_{24}(x_1 + 1.5x_2 + 0.5x_3 + 0.3x_4 + 1.325) \\
&\qquad - P_{34}\left(\frac{0.0165}{(x_2 + 0.3)} - 1.65\right) + P_{24}P_{44} \\
M_{34} &= 0.1P_{13} + 0.33P_{33} + P_{34}\left(x_2 + 0.2x_4 - \frac{0.01x_2}{(x_2 + 0.3)} + 1.2\right) + P_{33}P_{44}
\end{aligned}$$

One possible algebraic $\bar{P}$ solution exist by setting the non-diagonal entries of $P(x)$ equal to zero, leaving the remaining diagonal terms and $P(0)$ as,

$$\begin{aligned}
P_{11} &= K_1(x_1 + x_2 + 0.1x_4 + 1) \\
P_{22} &= K_2(x_1 + 1.5x_2 + 0.5x_3 + 0.3x_4 + 0.325) \\
P_{33} &= K_3\left(x_2 + 0.2x_4 - \frac{0.01x_2}{(x_2 + 0.3)} + 0.2\right) \\
P_{44} &= K_4
\end{aligned} \quad \rightarrow \quad P(0) = \begin{bmatrix} K_1 & 0 & 0 & 0 \\ 0 & 0.325K_2 & 0 & 0 \\ 0 & 0 & 0.2K_3 & 0 \\ 0 & 0 & 0 & K_4 \end{bmatrix} \tag{60}$$

where $K_i > 0$ are some positive constants satisfying certain conditions. This fulfills the properties of $P(x) = P(x)^T$ and $P(0) > 0$. The diagonal entries of matrix $M(x)$ becomes,

$$\begin{aligned}
M_{11} &= 2K_1(x_1 + x_2 + 0.1x_4 + 1)^2 - Q_{11} \\
M_{22} &= 2K_2(x_1 + 1.5x_2 + 0.5x_3 + 0.3x_4 + 0.325)^2 - Q_{22} \\
M_{33} &= 2K_3\left(x_2 + 0.2x_4 - \frac{0.01x_2}{(x_2 + 0.3)} + 0.2\right)^2 - Q_{33} \\
M_{44} &= K_{44}^2 + 2K_{44} - Q_{44}
\end{aligned} \tag{61}$$

The reasoning behind this particular algebraic $\bar{P}$ solution is that it should be possible to design the K_i constants large enough for the diagonals entries to overpower the penalty terms Q_{ii} and the non-zero non-diagonal terms of $M(x)$ to achieve positive definiteness by the argument of schur complement or

diagonal dominance. It is beneficial to achieve a positive definite $M(x)$ for all x to provide a larger region of stability but this can be difficult to satisfy. Since the underlying theory only guarantees stability for a neighbourhood about the origin, the inequality is only required to be satisfied here. By substituting this algebraic $\bar{P}$ solution back into (59) and evaluating at the origin, the following algebraic matrix inequality can be obtained,

$$\mathcal{M}_1 = M(0) = \begin{bmatrix} 2K_1 - Q_{11} & K_1 & 0 & 0.1K_1 \\ K_1 & 0.2113K_2 - Q_{22} & 0.319K_3 & 0 \\ 0 & 0.319K_3 & 0.08K_3 - Q_{33} & 0.066K_3 \\ 0.1K_1 & 0 & 0.066K_3 & K_{44}^2 + 2K_{44} - Q_{44} \end{bmatrix} > 0 \tag{62}$$

where the constants K_i are yet to be defined. It is now useful to obtain the explicit values of K_i in terms of the penalty terms Q_{ii} such that a positive definite $M(0)$ can be achieved.

6.1.4 Parameter Extraction

The following formulation will determine the explicit conditions required for the constants K_i to achieve a positive definite $M(0)$. Usually a range of solutions will exist when solving inequalities. For this reason, a set of positive tuning parameters γ will also be included which will offer an extra degree of freedom in the algebraic $\bar{P}$ solution. There are three general scenarios that occur when satisfying inequalities in this analysis. To present this, n, m and Q in the following scenarios have been defined as arbitrary positive constants and x is the unknown to be solved.

Scenario 1: The inequality is in the linear form,

$$nx - m > 0 \tag{63}$$

and a solution can be found by re-arranging the equation,

$$x > \frac{m}{n} \quad \rightarrow \quad x = \frac{1}{n}(m + \gamma) \tag{64}$$

Scenario 2: The inequality is in the quadratic form,

$$nx^2 + 2mx - Q > 0 \tag{65}$$

and a quadratic solution can be obtained by taking the positive part,

$$x > -m + \sqrt{m^2 + nQ} \quad \rightarrow \quad x = (-m + \sqrt{m^2 + nQ}) + \gamma \tag{66}$$

Scenario 3: The inequality is in the quadratic form,

$$-\frac{x^2}{n} + 2mx - Q > 0 \quad \text{or} \quad \frac{x^2}{n} - 2mx + Q < 0 \tag{67}$$

and a range of valid solution will always exist within the bound

$$n\left(m - \sqrt{m^2 - \frac{Q}{n}}\right) < x < n\left(m + \sqrt{m^2 - \frac{Q}{n}}\right) \tag{68}$$

if the additional condition $m^2 > \frac{Q}{n}$ can be satisfied to avoid imaginary solutions. Since any x within this bound is valid, an operator has been designed to return a varying average based on the parameter γ,

$$x = n\left[\gamma\left(m - \sqrt{m^2 - \frac{Q}{n}}\right) + (1-\gamma)\left(m + \sqrt{m^2 - \frac{Q}{n}}\right)\right] \tag{69}$$

where the design parameter γ is within the range $0 < \gamma < 1$. (Note, setting $\gamma = 0.5$ will return x equal to the centre of the bound in (68).) These three scenarios are true for any positive constants $n, m, Q > 0$ and will be used to explicitly define the constants K_i.

The block matrix representation in (53) will be used where the subscript 'i' of matrices A_i, B_i, D_i will represent the number of recursive iterations of schur complement that have been applied to matrix $\mathcal{M}_i$. First, satisfying $\mathcal{M}_1 > 0$ in (62) can be done by taking its schur complement and representing the block matrix $D_1 = \mathcal{M}_1(4,4)$ as the last diagonal entry of $\mathcal{M}_1$. By utilizing **Scenario 2**, K_4 can be obtained,

$$\begin{aligned} D_1 &= K_4^2 + 2K_4 - Q_{44} > 0 \\ &\rightarrow \quad K_4 = -1 + \sqrt{Q_{44}+1} + \gamma_4 \\ &\rightarrow \quad D_1 = \gamma_4^2 + 2\gamma_4\sqrt{Q_{44}+1} \end{aligned} \tag{70}$$

where $\gamma_4 > 0$ and the block matrices A_1, B_1, D_1 forms the second condition of schur complement,

$$\mathcal{M}_2 = A_1 - B_1 D_1^{-1} B_1^T = \begin{bmatrix} 2K_1 - Q_{11} - \frac{0.01K_1^2}{D_1} & K_1 & -\frac{0.0066K_1K_3}{D_1} \\ K_1 & 0.2113K_2 - Q_{22} & 0.319K_3 \\ -\frac{0.0066K_1K_3}{D_1} & 0.319K_3 & -\frac{0.0044K_3^2}{D_1} + 0.08K_3 - Q_{33} \end{bmatrix} > 0 \tag{71}$$

In order to satisfy the second condition, the schur complement of the matrix $\mathcal{M}_2$ is again taken by defining the block matrix $D_2 = \mathcal{M}_2(3,3)$ and utilizing **Scenario 3** (Note for this scenario, the symbol n_i has been defined for clarity of representation) to obtain K_3,

$$\begin{aligned} D_2 &= -\frac{K_3^2}{n_1} + 0.08K_3 - Q_{33} > 0 \quad \text{where} \quad n_1 = \frac{D_1}{0.0044} \\ &\rightarrow K_3 = n_1\left[\gamma_3\left(0.04 - \sqrt{0.0016 - \frac{Q_{33}}{n_1}}\right) + (1-\gamma_3)\left(0.04 + \sqrt{0.0016 - \frac{Q_{33}}{n_1}}\right)\right] \end{aligned} \tag{72}$$

where $0 < \gamma_3 < 1$ and an additional condition of $0.0016n_1 > Q_{33}$. The second schur complement condition for this is

$$\mathcal{M}_3 = A_2 - B_2 D_2^{-1} B_2^T = \begin{bmatrix} 2K_1 - Q_{11} - \frac{0.01K_1^2}{D_1}(+\approx 0) & K_1\left(1 + \frac{0.0021K_3^2}{D_1D_2}\right) \\ K_1\left(1 + \frac{0.0021K_3^2}{D_1D_2}\right) & 0.2113K_2 - Q_{22} - \frac{0.102K_3^2}{D_2} \end{bmatrix} > 0 \tag{73}$$

Similarly, to satisfy the above, a final schur complement is applied to the matrix $\mathcal{M}_3$ by defining $D_3 = \mathcal{M}_3(2,2)$ and utilizing **Scenario 1** to obtain K_2,

$$\begin{aligned} D_3 &= 0.2113K_2 - \frac{0.102K_3^2}{D_2} - Q_{22} > 0 \\ &\rightarrow K_2 = \frac{1}{0.2113}\left(Q_{22} + \frac{0.102K_3^2}{D_2} + \gamma_2\right) \\ &\rightarrow D_3 = \gamma_2 \end{aligned} \tag{74}$$

where $\gamma_2 > 0$, producing the final scalar inequality which can be solved via **Scenario 3** to obtain K_1,

$$\begin{aligned} A_3 - B_3 D_3^{-1} B_3^T &= -\frac{K_1^2}{n_2} + 2K_1 - Q_{11} > 0 \quad \text{where} \quad n_2 = \left(\frac{0.01}{D_1} + \frac{1}{D_3}\left(1 + \frac{0.0021K_3^2}{D_1D_2}\right)^2\right)^{-1} \\ &\rightarrow K_1 = n_2\left[\gamma_1\left(1 - \sqrt{1 - \frac{Q_{11}}{n_2}}\right) + (1-\gamma_1)\left(1 - \sqrt{1 + \frac{Q_{11}}{n_2}}\right)\right] \end{aligned} \tag{75}$$

where $0 < \gamma_1 < 1$ with an additional condition $n_2 > Q_{11}$.

In order to avoid imaginary solutions, additional conditions were introduced in (72) and (75). These can be satisfied by introducing bounds on the free parameters γ_4 and γ_2. Consider the additional

condition in (72), a bound for γ_4 can be provided by substituting in the resulting D_1 in (70) into this condition and utilizing **Scenario 2** to solve for an updated γ_4,

$$\begin{aligned} 0.0016 - \frac{Q_{33}}{n_1} &= \gamma_4^2 + 2\gamma_4\sqrt{Q_{44}+1} - 2.7225Q_{33} > 0 \\ &\rightarrow \gamma_4 = (-\sqrt{Q_{44}+1} + \sqrt{Q_{44}+1+2.7225Q_{33}}) + \bar{\gamma_4} \end{aligned} \tag{76}$$

where $\bar{\gamma}_4 > 0$ is the updated tuning parameter. The same can be achieved for (75) by substituting the resulting D_3 in (74) back into the condition, re-arranging and utilizing **Scenario 1**,

$$\begin{aligned} \frac{1}{Q_{11}} - \frac{1}{n_2} \Rightarrow \gamma_2\left(\frac{1}{Q_{11}} - \frac{0.01}{D_1}\right) - \left(1 + \frac{0.0021K_3^2}{D_1D_2}\right)^2 > 0 \\ \rightarrow \gamma_2 = \frac{Q_{11}\left(1 + \frac{0.0021K_3^2}{D_1D_2}\right)^2}{1 - \frac{0.01Q_{11}}{D_1}} + \bar{\gamma_2} \end{aligned} \tag{77}$$

which produces an additional condition of $D_1 > 0.01Q_{11}$ that can be easily satisfied by a large enough γ_4 and therefore will not be explicitly defined. The parameters γ_4 and γ_2 have now been redefined with known constants and the flexible parameters $\bar{\gamma}_4$ and $\bar{\gamma}_2$. This changes the definition of D_1 and D_3 but does not conflict with any of the original conditions of schur complement ($D_1 > 0$ and $D_3 > 0$) if $\bar{\gamma}_4, \bar{\gamma}_2 > 0$. Substituting the new tuning parameters back into the solution and dropping the overbar of $\bar{\gamma}_4$ and $\bar{\gamma}_2$, the final algebraic $\bar{P}$ solution can be written as,

$$P(x) = \begin{bmatrix} P_{11} & 0 & 0 & 0 \\ 0 & P_{22} & 0 & 0 \\ 0 & 0 & P_{33} & 0 \\ 0 & 0 & 0 & P_{44} \end{bmatrix} \quad \text{where} \quad \begin{aligned} P_{11} &= K_1(x_1 + x_2 + 0.1x_4 + 1) \\ P_{22} &= K_2(x_1 + 1.5x_2 + 0.5x_3 + 0.3x_4 + 0.325) \\ P_{33} &= K_3\left(x_2 + 0.2x_4 - \frac{0.01x_2}{(x_2+0.3)} + 0.2\right) \\ P_{44} &= K_4 \end{aligned} \tag{78}$$

where

$$\begin{aligned} K_4 &= -1 + \sqrt{Q_{44}+1+2.7225Q_{33}} + \gamma_4 \\ &\rightarrow D_1 = K_4^2 + 2K_4 - Q_{44} \\ K_3 &= n_1\left[\gamma_3\left(0.04 - \sqrt{0.0016 - \frac{Q_{33}}{n_1}}\right) + (1-\gamma_3)\left(0.04 + \sqrt{0.0016 - \frac{Q_{33}}{n_1}}\right)\right] \\ &\rightarrow D_2 = -\frac{K_3^2}{n_1} + 0.08K_3 - Q_{33} \quad \text{where} \quad n_1 = \frac{D_1}{0.0044} \\ K_2 &= \frac{1}{0.2113}\left(Q_{22} + \frac{0.102K_3^2}{D_2} + \frac{Q_{11}\left(1 + \frac{0.0021K_3^2}{D_1D_2}\right)^2}{1 - \frac{0.01Q_{11}}{D_1}} + \bar{\gamma_2}\right) \\ &\rightarrow D_3 = 0.2113K_2 - \frac{0.102K_3^2}{D_2} - Q_{22} \\ K_1 &= n_2\left[\gamma_1\left(1 - \sqrt{1 - \frac{Q_{11}}{n_2}}\right) + (1-\gamma_1)\left(1 + \sqrt{1 - \frac{Q_{11}}{n_2}}\right)\right] \\ &\text{where} \quad n_2 = \left(\frac{0.01}{D_1} + \frac{1}{D_3}\left(1 + \frac{0.0021K_3^2}{D_1D_2}\right)^2\right)^{-1} \end{aligned}$$

where the parameter are limited to the range $\gamma_2, \gamma_4 > 0$ and $0 < \gamma_1, \gamma_3 < 1$. Here, one possible variation of the algebraic $\bar{P}$ solution have been presented by using schur complement, starting from the last diagonal entry $\mathcal{M}_1(4,4)$ to provide a definition for K_4 and progressively applying schur complement until the first diagonal entry $\mathcal{M}_3(1,1)$ has been satisfied to finally obtain K_1, with each constant calculated sequentially. The constants K_i have now been defined as a function of the penalty matrix Q and the algebraic $\bar{P}$ solution tuning parameters γ and these will be carried forward to the next stage of the controller design.

6.2 Control Law Implementation

The algebraic $\bar{P}$ solution will be chosen with the initial default parameters $\gamma_2 = \gamma_4 = 1$ and $\gamma_1 = \gamma_3 = 0.5$, providing the corresponding constants $K_1 = 0.455$, $K_2 = 821$, $K_3 = 28.4$, $K_4 = 1.05$. The control law (30) can now be implemented into the augmented system (29). The following vector-valued function,

$$P(\xi)x = \begin{bmatrix} K_1(\xi_1 + \xi_2 + 0.1\xi_4 + 1)x_1 \\ K_2(\xi_1 + 1.5\xi_2 + 0.5\xi_3 + 0.3\xi_4 + 0.325)x_2 \\ K_3\left(\xi_2 + 0.2\xi_4 - \frac{0.01\xi_2}{(\xi_2+0.3)} + 0.2\right)x_3 \\ K_4 x_4 \end{bmatrix} \tag{79}$$

and its corresponding jacobian with repect to ξ,

$$\nabla_\xi(P(\xi)x) = \begin{bmatrix} x_1K_1 & x_1K_1 & 0 & 0.1x_1K_1 \\ x_2K_2 & 1.5x_2K_2 & 0.5x_2K_2 & 0.3x_2K_2 \\ 0 & x_3K_3\left(1 - \frac{0.003}{(\xi_2+0.3)^2}\right) & 0 & 0.2x_3K_3 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{80}$$

can now be substituted into the augmented system,

$$\begin{aligned} \dot{x} &= f(x) + g(x)u \\ \dot{\xi} &= -\kappa\left(\frac{1}{2}\nabla_\xi(P(\xi)x)^T x - R(x-\xi)\right) \end{aligned} \tag{81}$$

with the dynamic control

$$u(x,\xi) = -(x_4K_4 + R_4(x_4 - \xi_4)) \tag{82}$$

where $x, \xi \in \mathbb{R}^n$ are vectors of the real and extended states and R_4 is the 4^{th} diagonal entry of the R matrix. Notice that the control function only depends on the states x_4 and ξ_4, which can induce certain interesting characteristics on the controlled system as will be seen later. This is different to the uncontrolled situation of the LQR since ξ_4 has a dependency on the other states. Given x_0, the design parameters for the dynamic control are; the initial condition for the augmented states ξ_0, the constant κ and the diagonal penalty matrix $R \in \mathbb{R}^{n\times n}$. The motivation for choosing these parameters is to ensure that the equivalent *Lyapunov Stability* conditions for this closed-loop system,

$$\begin{aligned} \mathcal{H}_k(x,\xi) = V_x f(x) - \frac{1}{2}V_x g(x)g(x)^T V_x^T + \frac{1}{2}q(x) - \kappa V_\xi V_\xi^T \leq 0 \\ V(x,\xi) = \frac{1}{2}x^T P(\xi)x + \frac{1}{2}(x-\xi)^T R(x-\xi) > 0 \end{aligned} \tag{83}$$

are both satisfied for the entire controlled trajectory and the system remains in a region where local asymptotic stability is retained. Therefore finding and optimising the set of parameters $[\xi_0, \kappa, R]$ presents the next challenge of the dynamic control problem.

6.3 Parameter Experimentation

The controlled system is simulated through ode45 with the additional conditions of (83) implemented. The set of parameters that lead to a trajectory that fails to satisfy these conditions will be labelled as failed control parameters. The objective is to find the combination of $[\xi_0, \kappa, R]^T$ that will satisfy these conditions with the minimal cost. The cost function has been chosen the same as the original problem in (44), since this is the ultimate goal of the controller. It is already known that $\kappa > 0$ and $R > 0$ from the theory. However, ξ_0 is not bounded and can be of any magnitude.

With $R = I$ as the identity matrix and $\kappa = 1$, initial experimentation of different random combinations of ξ_0 showed that most trials resulted in an unstable system and the parameters that led to a stable system were often difficult to find by chance. The regions where $\mathcal{H}_k < 0$ are different to those where $V > 0$ and finding the regions where both conditions are satisfied can be very challenging. By

analysing sections of the trajectory where stability were sustained, it was seen that for many of these trials the control inputs were negative and any attempts to reverse the signs of some of the parameters proved unsuccessful, failing to make any difference to the overall performance (the controls remained negative). This demonstrated the nonlinearity of problem and analysing the unstable systems did not offer any additional information about the successful parameters.

6.3.1 Monte Carlo Simulation

The Monte Carlo technique is a method of stochastic simulation which uses random sampling and can be applied to characterize systems with many coupled degrees of freedom [40]. The initial observations indicated that there is a degree of randomness involved when searching for the successful parameter values and the highly coupled effect of these parameters on the real system makes Monte Carlo an attractive option in order to obtain some initial results. The method has been adopted by generating random samples of $\xi_0 = [\xi_{1,0}, \xi_{2,0}\, \xi_{3,0}, \xi_{4,0}]$ within the range $[-20, 20]$ for a particular set of κ and R, and the successful parameters as well as the failed ones will be displayed on a scatter plot to estimate the regions where solutions may exist.

By using a random generator for ξ_0, the accuracy of the simulation can be improved by increasing the sample size to provide a more complete search. The number of samples was chosen to be 1000 which offered consistent best cost with each run, indicating that the optimal accuracy has been reached. It was found by slowly reducing κ, the simulation produced more successful samples and the value that resulted in the largest proportion of successful samples occurred at $\kappa = 0.05$ and $R = I$, with around 15% success rate. For one simulation of randomly generated $\xi_{i,0}$, the results have been displayed on two separate 2-D scatter plots for demonstration but it should be noted that the both plots make up a single simulation. The successful control parameters have also been colour coded with their corresponding cost.

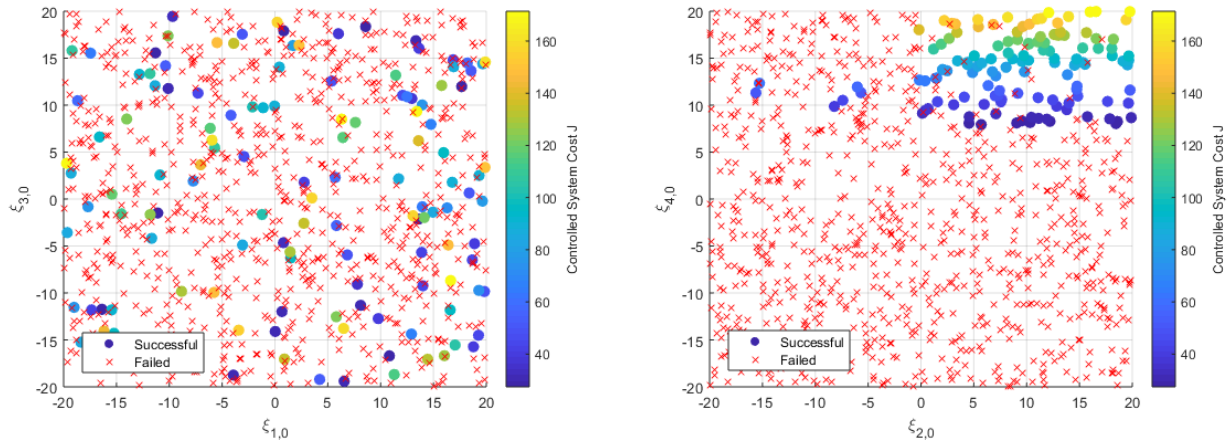

Figure 6: Successful and failed trials for 1000 random samples of $\xi_{i,0}$ between range $[-20, 20]$, at $\kappa = 0.05$ and $R = I$.

From Figure 6, it appears that there are many successful control parameters with varying costs. For this particular setting of κ and R, a clear pattern can be seen on the plane of $(\xi_{2,0}, \xi_{4,0})$ where successful parameters mostly lie in the upper quadrant $\xi_{2,0}, \xi_{4,0} > 0$, and the those with the lowest cost exist somewhere in the range of $8 < \xi_{4,0} < 12$. However, it is not as clear on the $(\xi_{1,0}, \xi_{3,0})$ plane where solutions exist over the entire region and those with the lowest cost are also difficult to locate. The nonlinearity of the problem has been further demonstrated here where the distance between successful and failed parameters can be extremely small and shifting their values by as small as 2 decimal places in some cases can result in the breakdown of the control.

Different values of κ were investigated in the search space and similar patterns existed on the $(\xi_{1,0}, \xi_{3,0})$ plane where successful parameters were randomly distributed within the entire search space. Some trends can be deduced on the $(\xi_{2,0}, \xi_{4,0})$ plane and have been shown in Figure 7.

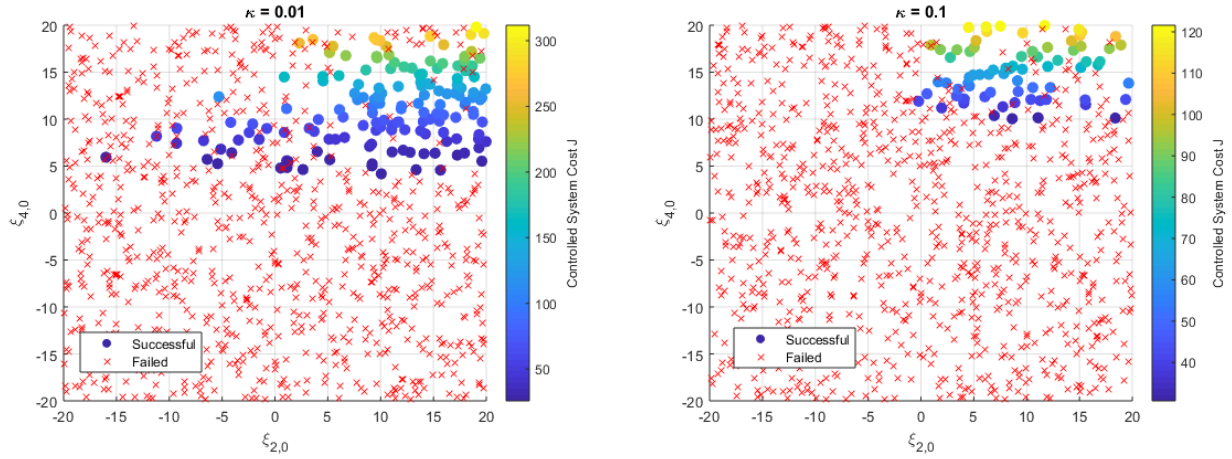

Figure 7: Successful and failed trials for 1000 samples of ξ_0 between range $[-20, 20]$, at $R = I$.

There is a clear reduction in the number of solutions and the minimum cost from the whole sample also increased with larger κ. For the case of $\kappa = 0.01$, solutions with a negative $\xi_{2,0}$ began to appear. Combining this with the results in Figure 6 and 7, it can be noticed that the lower threshold for valid $\xi_{4,0}$ increased with κ.

The effects of varying R have also been investigated by setting $R = I \times C$ for some varying constant C. The results for both analysis have been tabulated in Table 3. From this, the outcome of successful samples seems to be more sensitive to the effect of varying κ than it is to R (graphical study in appendix B) and in the search space of $[-20, 20]$ for ξ_0, the number of successful samples quickly reduced to 0 as κ increased beyond 0.25. With increasing R, although the number of solutions remained approximately the same, the minimum cost worsened.

At $R = I$			At $\kappa = 0.05$		
κ	No. Successful Samples	Min Cost	R	No. Successful Samples	Min Cost
0.01	98	32.5	0.5	109	24.9
0.03	143	25.8	**1.0**	**150**	**24.8**
0.05	**150**	**24.8**	2.0	147	26.4
0.10	88	30.4	3.0	127	31.4
0.15	45	41.2	4.0	118	46.9
0.20	6	63.7	5.0	163	55.1

Table 3: Effects on the number of successful samples and minimum cost obtained out of all successful samples with changing κ and R in a search space of ξ_0 in $[-20, 20]$.

From the figures above, it is possible that by varying κ and R, the region of successful parameters can be shifted out of the search space on the $(\xi_{2,0}, \xi_{4,0})$ plane. For this reason, the search space was increased to $[-100, 100]$. Previously, at $R = 1$ and $\kappa = 0.2$ produced few valid parameters with poor costs. But by increasing the search space for this case, the simulation was able to offer more solutions with improved minimum cost (plots in appendix C). Once again, only solutions on the $\xi_{4,0}$ line were bounded whilst the remaining parameters produced successful samples distributed everywhere. This was tested for even larger space and the same pattern can be witnessed. It is therefore deduced that the parameters that led to successful or even optimal control depends on the relative difference of their values and similar control strategies can be obtained by changing their relative magnitude. By varying the parameters κ and R, it shifts the probable region of optimal solutions elsewhere on the $\xi_{i,0}$ space. However, as the search space is increased, this space is searched less thoroughly or increase effort (more samples) is required to find the optimal parameter values. Additionally, this can also

causes the general quality of the solutions to worsen with respect to the objective, creating a large spread of costs. These effect will be reiterated and studied more extensively under the analysis in the next chapter.

By utilizing the Mone Carlo method, initial experimentation provided a lowest cost of $J = 24.8$ and the corresponding controlled system have been plotted in Figure 8.

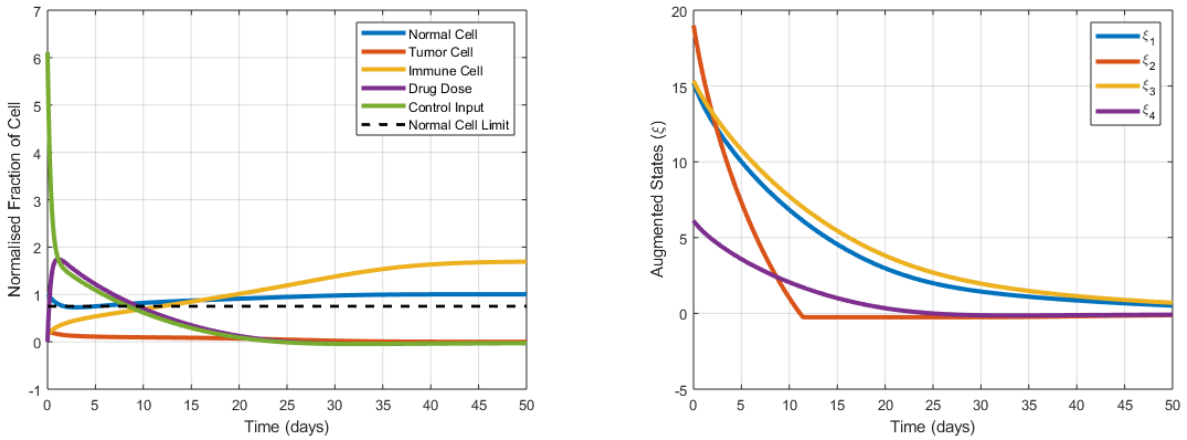

Figure 8: The controlled trajectory with the parameters, $\kappa = 0.05$, $R = I$, $[\xi_{1,0}, \xi_{2,0}, \xi_{3,0}, \xi_{4,0}] = [15.2, 19.0, -15.3, 7.11]$ and the corresponding cost of 24.8.

Comparing this result with LQR in Figure 4 and SDRE in Figure 5 and their corresponding costs, it is clear that the dynamic control law already provides significant a improvement on both control methods without any in-depth optimisation.

6.3.2 Preliminary Findings

Recall that the dynamic control law aims to solve a regional dynamic optimal control problem with the new instantaneous cost $L(x, \xi, u)$. This means that the control policy obtained through the dynamic control law may not necessarily be the optimal control for the original problem which is the true objective. This was observed from the variety of costs that resulted from successful samples in the Monte Carlo simulations. To obtain an optimal control with respect to the real system requires further tuning the control parameters, $[\xi_0, \kappa, R]$.

Since the control parameters under experimentation mainly affects the augmented states ξ, it is difficult to predict their impact on the real system. Their effects are highly nonlinear and coupled, resulting in similar trajectories obtained from multiple different controller parameter settings of different magnitudes. The Monte Carlo simulations demonstrated this irregularity and offered an initial look at the performance of the derived dynamic control law. Figure 8 provided one possible dynamically controlled system but it's not the optimal control strategy. It is already recognised that many valid parameters can exist in the optimisation problem over an infinite search space and there can be countless local optimal solutions. Constraining R and κ may be able to bound the valid parameters to a certain region and help find the local optimal solution. But due to the number of free variables available in this optimisation problem, to continue experimenting with Monte Carlo simulations will not be very efficient.

This concludes the Monte Carlo analysis and presents the motivation for the application of Genetic Algorithm which offers a more effective and elegant search technique for optimisation. The purpose of the program will be to obtain dynamic control parameters that will lead to a close-to-optimal control strategy with respect to the original control objective. The findings presented in this chapter will be carried forward to improve the search performance of this program.

7 Genetic Algorithm

Genetic Algorithm (GA) is a method for solving constrained or unconstrained optimisation problems [19]. This method differs to the classical optimisation algorithms and is especially powerful for problems in which the objective function is discontinuous, nondifferentiable, stochastic, or highly nonlinear. This was shown to be the case for the dynamic control problem where the effect of the parameters ξ_0, κ and R on the final controller performance is not clear. In this chapter, the theory of genetic algorithm will be briefly introduced and a systematic guideline for incorporating it with the original optimal control problem will be presented. Further analysis of the options within the program will be studied to improve its performance. Finally, the effects of the control parameters on the optimal solution obtained from the program will be discussed to gain an understanding of the optimisation problem.

7.1 The Algorithm

Some terminologies will be introduced to describe genetic algorithm.

- *Fitness function* is the the objective function for the optimisation problem. It defines the *fitness* of a particular individual.
- An *individual* is a particular set of values or variables that the problem is optimising for.
- *Population* is an array of individuals.
- *Generation* is each successive iteration of a newly generated population from the previous one.
- *Fitness value* of an individual is its value obtained from the fitness function. The *Best Fitness Value* is therefore the individual with the lowest fitness value in the population.
- *Parents* are defined as a selection of individuals used to compute the individuals of the next generation which are the *Children*.

Genetic algorithm's progression is based on the process of natural selection, the fittest individuals and their corresponding characteristics are carried forward through each successive generation. The key components of it's inner working involve the concept of; elitism (selection of the fittest individuals to be carried forward directly as children of the next generation); crossover (selected parents share information in order to create children); and mutation (selective few parent individuals are randomly modified based on their associated probability). These procedures allow the algorithm to strike a balance between exploitation of the fit individuals and exploration of the search space in order to converge to a close-to-optimal solution. An outline of the algorithm is presented below.

Algorithm 1 The Genetic Algorithm procedure from [30]

```
1: function GeneticAlgorithm()
2:     Population ← generate initial population of size N
3:     while Not Stopping Criteria do
4:         Score ← evaluate fitness of each individual in Population
5:         ScaledScore ← scale the raw fitness Score
6:         Elite ← best scored individuals in ScaledScore are selected
7:         Parents ← select parents based on ScaledScore
8:         ChildCrossover ← children created from Parents by crossover
9:         ChildMutation ← children created from Parents by mutation
10:        Population ← new population generated from [Elite, ChildC, ChildM]
11: return Best individual from Population
```

Algorithm 1 gives highly fit individuals a higher probability to reproduce and their children eventually will converge to an optimal solution through characteristic inheritance. The method is inherently random but performs significantly better than a pure random search as it exploits historical information.

There are various options within GA that will allow the algorithm to more thoroughly explore its search space; population size, initial population creation function, scaling function of the raw fitness scores, selection operator, number of elites, crossover fraction rate and its operator, mutation fraction rate and its operator, and the stopping criteria. The selection of these options or operators depends on the characteristics of the problem and can be determined through experimentation. The best settings should offer a balance between the exploit and explore trade-off. Later, these options will be studied to improve the search for the optimal solution.

Genetic algorithm's simplicity provides an intuitive understanding of its inner working and allows the method to be adapted and applied to a variety of problems. Its capabilities make it an ideal tool for the dynamic control parameter optimisation due to the problem's non-smooth and probabilistic nature. Next, an objective function for the program will be formulated to successfully find the optimal parameters.

7.2 Implementation

The new problem is defined as finding the best control parameters that will result in an optimally controlled system under the dynamic control law, hence the original control problem can now be formulated as a time-independent optimisation problem. It was previously found that two situations arise when simulating the tumor model with different ξ_0, κ and R, either the inequalities (83) are not satisfied resulting in an unstable system, or the inequalities are satisfied resulting in a stable system with a defined cost. The fitness function will therefore mostly remain the same as the original problem with a minor modification to account for the failed controls,

$$\begin{aligned} \min_{\xi(0),\kappa,R} J(\xi(0),\kappa,R) = &\begin{cases} \frac{1}{2}\int_0^\infty x^T Q(x)x + u^T R_c(x)u \; dt & \text{if} \quad \mathcal{H}_k(x,\xi) \le 0 \quad \text{and} \quad V(x,\xi) > 0 \\ \mathcal{U} & \text{if} \quad \text{otherwise} \end{cases} \\ \text{subjected to} \quad &\dot{x} = f(x,\xi,u) \\ &x(0) = x_0 \end{aligned} \tag{84}$$

where the penalty $Q(x)$ and $R_c(x)$ are the same as (44), $\mathcal{U}$ is an upper bound to quantify the cost of failed control and $f(x,\xi,u)$ is the augmented tumor system in (29) with a dynamic control law in (30). In this problem, an individual is defined as a vector containing the parameters $[\xi_0,\kappa,R]^T$ and its corresponding fitness value can be obtained by simulating the entire trajectory of the dynamically controlled system with the chosen parameters. Following the simulation, either the actual cost is returned as the individual's fitness value or if the inequalities have not been satisfied, an upper bound $\mathcal{U}$ is returned instead.

This upper bound should be set such that during the optimisation, any individuals that led to failed control are penalized much more heavily than the worst performing individuals that provided successful control so any desired characteristics can be captured and inherited. An attractive value for this upper bound is ∞. However, in order to gain an understanding of the algorithm's optimisation process, a finite value of $\mathcal{U} = 10000$ was chosen. This value does not have any major effects on the final converged solution, but it can affect the ranking of the individuals and parent selection process.

There is also a degree of flexibility when choosing the variables to be optimized. For example, the matrix $R \in \mathbb{R}^{4\times4}$ can be set as

$$R = \begin{bmatrix} R_1 & 0 & 0 & 0 \\ 0 & R_2 & 0 & 0 \\ 0 & 0 & R_3 & 0 \\ 0 & 0 & 0 & R_4 \end{bmatrix}$$

where R_1, R_2, R_3, R_4 are all free variables to be optimised or $R_1 = R_2 = R_3 = R_4 = R$ which then result in the single variable R. However, during an analysis of the program, it was found that increasing the freedom of R did not improve the final converged solution. Therefore, in order to keep the dimension of the optimisation problem low, R has been chosen as a single variable to be optimised.

7.3 Genetic Algorithm Options

The *ga* function in the MATLAB global optimisation toolbox [30] has been utilized for the optimisation and the options will be explored in this section. For more details of the options, please refer to [31].

As already seen before, the values of successful control parameters can be quite random. Since genetic algorithms are non-deterministic methods, the converged solution can vary each time the program is run. The algorithm converges to a near-optimal solution and its optimally can be significantly affected by the initial population, especially for this problem where there are many local regions of solution. By running the program for multiple iterations with randomised initial populations and with sufficient mutation and crossover, a consistent final cost that is comparable with the Monte Carlo simulation results often suggests that the returned parameters are very close to the global optimal.

Option	Selected
Population Size	200
Initial Population Creation	Function: Uniform Random
Fitness Scaling	Function: Proportional
Selection	Operator: Remainder

Table 4: GA chosen options from [30]

The program options that did not require any in-depth study or had no major effects on the performance have been displayed in Table 4. The population size correlates to the completeness of the search and was chosen to provide a satisfactory run time and accuracy. A uniform random creation function was used and its initial population range was set around the best performing parameters obtained from the Monte Carlo approach in order to provide good initial individuals. A scaling function that scales the fitness values proportional to its raw scores has been used to hold information about the quality of individuals, especially those that failed to control the system. To support this, a remainder selection operator was chosen which provided a good balance between a deterministic and probabilistic parent selection procedure.

The fitness scaling and selection process are examples of the trade-off between exploration and exploitation. If not appropriately chosen, the algorithm can select individuals with high scaled values and quickly converges to a particular range of solutions that may not be optimal. Whilst on the other hand, poorly scaled values or selection can cause excessive exploration and the algorithm to progress very slowly.

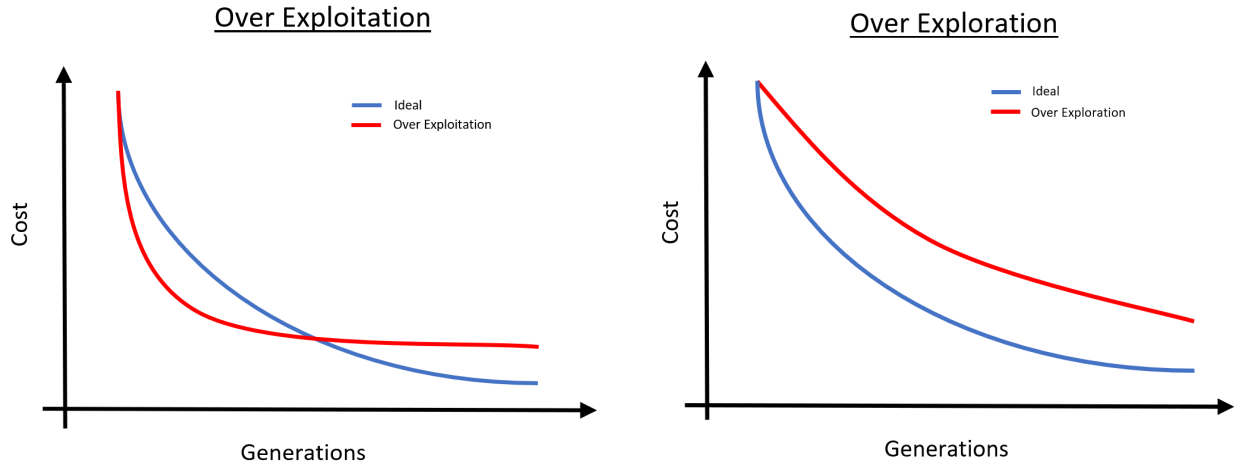

Figure 9: The effect of over exploiting versus over exploring.

Figure 9 demonstrate the effects of over exploiting and over exploring on the algorithm convergence

capabilities. Further options that can have a large impact on this are

- **Elite selection**, which is exploitation mechanism by retaining the fittest individuals for each generation. The number of elites can be varied to control this.
- **Crossover and Mutation**, which are both mechanisms of exploration. Whilst crossover shares information between individuals by combining certain variables to make a new child, mutation randomly changes the individual's variables.

Whilst elite selection aims to purely exploit the fit individuals, crossover offers limited exploration in a space that is dependent on the initial population and mutation aims to purely explore. It is therefore important that the number of elites and the relative ratio between crossover and mutation are experimented and chosen appropriately. A summary of this study has been tabulated in Table 5.

Elite No.	5	10	15	**20**	25	30	35
Avg Min Cost	24.13	24.12	24.13	**24.08**	24.09	24.11	24.24
Crossover Fraction	0.6	0.7	0.8	0.85	**0.9**	0.95	1
Avg Min Cost	24.13	24.15	24.12	24.12	**24.09**	24.15	24.23

Table 5: The effect of changing elite number and crossover fraction on the final converged cost.

The crossover fraction defines the proportion of the selected parents that are applied the crossover operator to produce children whilst the remaining proportion are mutated. The program was run several times for each setting and an average cost was recorded. The non-deterministic nature of genetic algorithm can be seen here where satisfactory results with small variations lie over a range of settings and there is no clear trend. The best setting had an elite number of 20 and crossover to mutation fraction of 0.9. This however is not the guaranteed best option since it can be seen that all the settings provided similar best fitness values, but this study does share some information on the settings that should be avoided which clearly caused early convergence.

Since the mutation ratio had been set very low, the mutation functions have not been studied further. The crossover function that performed the best was 'heuristic' [31], which returns a child that lies on a line containing the two parents.

$$child = parent_2 + R(parent_1 - parent_2)$$

where R is a ratio. Due to the random nature of the problem, by setting $R > 1$, this crossover function was able to offer a degree of mutation, allowing the program to effectively explore other regions outside the initial population range whilst still inheriting characteristics of parent individuals. R was chosen to be 1.2. The stopping criteria for the program was satisfied if the average relative change in the best fitness value was smaller than a tolerance of 1e-6 for more than 50 stall generations. This was sufficient to assume the program had converged to a local optimal.

7.4 Results and Discussion

The algorithm uses a list of stochastic procedures and operators which does not guarantee optimality, but the appropriate selection of methods for a particular optimisation problem can return solutions close to the optimal. This also requires a good understanding of the optimisation problem which is crucial for the effective application of genetic algorithm and obtaining accurate results. A consistent result from different random initial populations is a desirable program property and can be a good indicator of optimality. To investigate this further, a study on the variation of cost with different parameter values will be presented. It is beneficial to constrain some of the parameters to reduce the number of valid local optimal solutions and improve the effectiveness of the optimisation algorithm. A way to achieve this will be explored.

Recall that the Monte Carlo simulations found large amounts of valid parameters existing over an infinite space, all producing similar best fitness values and the result can be sensitive to the parameter κ. It can be demonstrated that κ is an effective deciding factor in controlling the particular region (relative magnitude) where the fit individuals lie. By keeping κ fixed in the optimisation process, it was possible to constrain the problem to an approximate local optimal cost that was consistent with each run of the *ga* program. The value of $\xi_{4,0}$ was also found to be a major indicator of the relative magnitude of parameters. The results for each κ setting have been tabulated in Table 6.

κ	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
R	0.421	0.201	0.132	0.0979	0.0779	0.0647	0.0553	0.0483	0.0429
$\xi_{4,0}$	12.2	22.9	35.7	47.5	59.3	71.0	82.9	94.6	107
Cost	24.29	24.15	24.11	24.08	24.08	24.07	24.06	24.06	24.06

Table 6: A summary of the effect of changing κ on the final converged cost outputted by the GA program.

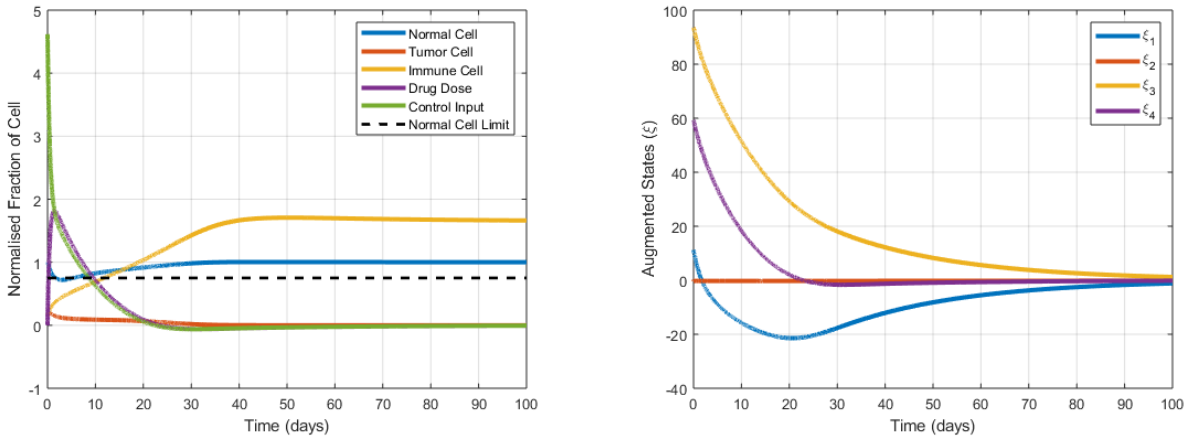

Figure 10: The controlled trajectory with small ξ_0 values, $\kappa = 0.5$, $R = 0.0779$, $[\xi_{1,0}, \xi_{2,0}, \xi_{3,0}, \xi_{4,0}] = [11.2, 0.133, 93.6, 59.3]$ and the corresponding cost of 24.08.

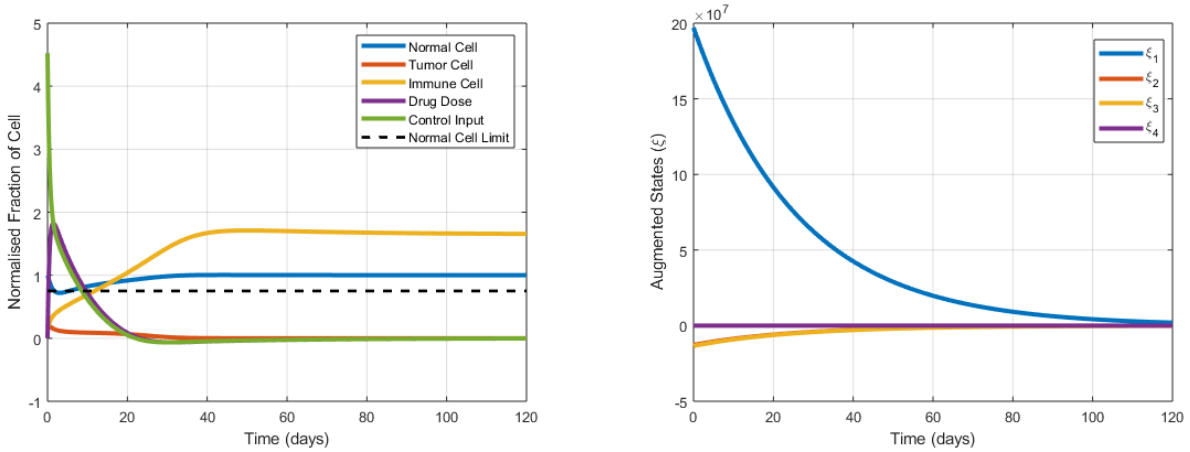

Figure 11: The controlled trajectory with large ξ_0 values, $\kappa = 2$, $R = 0.0192$, $[\xi_{1,0}, \xi_{2,0}, \xi_{3,0}, \xi_{4,0}] = [1.97 \times 10^8, -1.27 \times 10^7, -1.34 \times 10^7, 236]$ and the corresponding cost of 24.04.

Although, the converged values of $\xi_{1,0}, \xi_{2,0}$ $\xi_{3,0}$ still varied, their relative magnitude remained consistent with each run. By analysing Table 6, the relative magnitude of $\xi_{i,0}$ increased with κ and in turn also marginally improved the final converged cost. However, there appeared to be a diminishing return at very high magnitudes where a consistent optimal cost became more difficult to sustain. To study this effect, the controlled augmented system for a set of small parameters is compared with one of large and are both displayed in Figure 10 and 11.

It can be seen that two similar controlled tumor system were obtained for two very different extended state trajectories and their behaviour seems to have little correlation. Recall in (82), the only states important to the control strategy are x_4 and ξ_4 which are further dependent on other states. This suggests the other state extensions are of less direct relevance to the real system and as long as they exhibit a particular behaviour that causes ξ_4 to behave optimally, the optimal control of the real system can still be attained. The lack of dependency on ξ_1, ξ_2 and ξ_3 can be seen in the figures, even after the equilibrium of the real system had been reached, the non-zero ξ states had no further impact on the real system and control remained at zero. The trajectory of ξ states that cause ξ_4 to be optimal is not unique and therefore can cause the existence of multiple local optimal parameters in a infinitely large space. The significance of ξ_4 state on the control can be seen in the evolution of the augmented states in Figure 8, 10 and 12, where the ξ_4 trajectories exhibit similar path characteristics to produce near identical control strategies in the real system. This highlights the importance of $\xi_{4,0}$ and explains the pattern seem on the $(\xi_{2,0}, \xi_{4,0})$ plane in the Monte Carlo analysis. The exact relationship of ξ_4 to the control further depends on the magnitude of the other states and κ and R.

By evaluating the total additional cost described in (31) $\int_0^\infty h(x,\xi)dt$, this cost was 2.1×10^3 for the case in Figure 10 and 3.75×10^{14} for the case in Figure 11. As expected, the discrepancy in magnitude originated from the different augmented state trajectories, clearly indicating that including the additional cost in the minimisation would have been the incorrect objective for the optimisation problem. It may be possible that the cost can be further improved by searching in a even larger space. However, the minimal improvements in performance and the reduction in the convergence capabilities of the *ga* program suggest that this process is not worthwhile.

7.4.1 Algebraic $\bar{P}$ Solution Parameters

The optimisation had been previously done for a particular algebraic $\bar{P}$ solution where the parameters $\gamma_2 = \gamma_4 = 1$ and $\gamma_1 = \gamma_3 = 0.5$ in (78) have been fixed which provided an optimal cost of approximately $J = 24.04$. It is possible to improve the result further by allowing the solution parameters γ_i to become variables. The previous derivation of these free parameters (in the process of achieving positive definite $M(0)$ in section 6.1.4) now allows them to be varied without complex constraints in the optimisation problem. In addition, Genetic Algorithm has the ability to deal with high dimensional problems without affecting the run time significantly. It may result in longer convergence but this can be alleviated by constraining the search space appropriately. The additional condition in (77) which had not been explicitly satisfied did not affect operations of the program for this particular case of penalty Q.

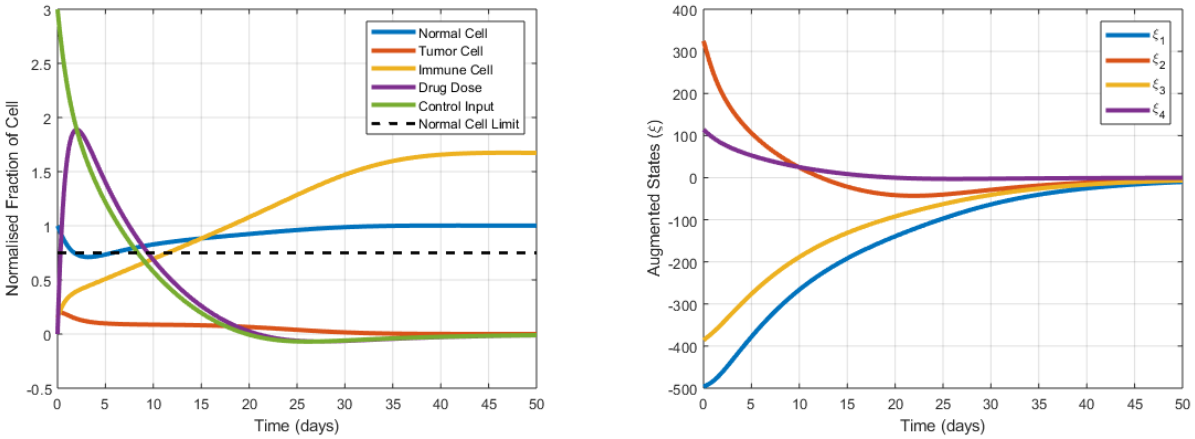

Figure 12: The controlled trajectory with the parameters, $\kappa = 3.5$, $R = 0.0263$, $[\gamma_1, \gamma_2, \gamma_3, \gamma_4] = [0.99, 1.55, 0.9, 0.05]$, $[\xi_{1,0}, \xi_{2,0}, \xi_{3,0}, \xi_{4,0}] = [-497, 325, -386, 114]$ and the corresponding cost of 23.68.

A summary of this optimisation have been tabulated in Table 7 and the best performing controller has been displayed in Figure 12. From the table, the final cost has improved significantly compared

κ	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8
$\xi_{4,0}$	7.99	8.43	10.6	12.7	15.7	18.7	21.8	24.9
Cost	23.96	23.83	23.76	23.72	23.70	23.69	23.69	23.69
κ	0.9	1	1.5	2	2.5	3	3.5	4
$\xi_{4,0}$	28.0	31.3	47.4	62.1	79.3	92.6	114	129
Cost	23.69	23.69	23.69	23.69	23.68	23.68	23.68	23.68

Table 7: A summary of the effect of changing κ on the final converged cost outputted by the GA program when including γ in the optimisation.

to the previous analysis, which suggests that the best performance of different algebraic $\bar{P}$ solutions can vary. By increasing κ, the similar trend of diminishing cost improvements can be witnessed here as before. The parameters returned by program, which led to a controlled system with an optimal cost of $J = 23.68$ had the values $[\gamma_1, \gamma_2, \gamma_3, \gamma_4] = [0.99, 1.55, 0.9, 0.05]$ and the corresponding algebraic $\bar{P}$ solution constants, $K_1 = 0.0253, K_2 = 482, K_3 = 0.191$ and $K_4 = 0.0952$.

7.5 Summary of Genetic Algorithm

The application of genetic algorithm offered a more elegant approach to optimise for the dynamic control parameters and a significant improvement on the control performance was achieved as compared to the Monte Carlo method. The ability to derive the dynamic control law offline means that the control mechanism is much less computationally demanding and allows the optimisation program to evaluate an entire population's fitness value by simulating the controlled system multiple times without run time issues. If suitable search constraints are in place, the problem can solved within 2 minutes. The algorithm's capabilities of dealing with non-smooth fitness functions is also a huge advantage that allows particular types of individuals to be excluded by modifying the fitness function (84) and assigning artificial costs. For the dynamic control problem, this objective function was found to be most effective but there is much freedom in its design to experiment on in order to achieve the best performance.

The genetic algorithm options did not offer significant changes on the final result but it did have an effect on the process the algorithm took to get to the final converged solution which impacted the convergence time. The majority of the settings seemed to have given equal emphasis on explore and exploit, providing relatively consistent final costs. The largest effect on the final converged solution was found by constraining the optimisation problem to a fixed κ but vary the algebraic $\bar{P}$ solution parameters γ_i. By increasing κ, the control can be improved, but this improvement is further limited by the optimal performance of that particular algebraic $\bar{P}$ solution.

8 Performance Comparison

The final result of each controller can now be compared to examine their relative control optimality. To offer a more complete comparison, the performance of Direct Collocation will also be included. This is a transcription method that can be coupled with an NLP solver to guarantee a local optimal solution and if initialised appropriately, the solution can be assumed to be the global optimal or very close to it. The result can be used to demonstrate the optimality of the dynamic control law and the convergence accuracy of the genetic algorithm program.

8.1 Direct Collocation

The method of direct collocation for solving the optimal control problem can be considered as a numerical approach where the entire trajectory is discretised into segments and is optimized altogether by the NLP solver. The input matrix consists of the time (t), states (x) and control (u) over the entire

trajectory, discretised by $h = t_{k+1} - t_k$ where the subscript 'k' is the k^{th} interval of the trajectory and N is the total number of intervals.

$$input = \begin{bmatrix} t_1 & x_1 & u_1 \\ \vdots & \vdots & \vdots \\ t_k & x_k & u_k \\ \vdots & \vdots & \vdots \\ t_N & x_N & u_N \end{bmatrix} \tag{85}$$

The solver aims to solve the discrete optimal control problem in (37) to find the optimal t, $x(t)$ and $u(t)$. The entire input matrix is therefore the variable, making this a high dimensional and highly constrained optimisation problem. Additionally, the size of h largely dictates the accuracy of the piece-wise approximation and the program run time. The time for convergence can also be dependent on the initial input matrix. Altogether, implying that an accurate model simulation under direct collocation can be extremely computationally expensive and a solution can be time consuming to obtain. However, the method is able to return a relatively consistent optimally controlled system with each successful run of the program.

The NLP solver utilized for this was the *fmincon* function in MATLAB [29] which uses a interior point algorithm [38]. This is an iterative method that finds the local optimal solution to convex optimisation problems. Using different initial input matrices to explore different regions of solution can help prevent being stuck at a local optimal and return a global optimal solution. A constant mesh size with 300 meshes was found to be sufficient for accurate simulations and different uniformly spaced ramp functions were used for the initialization. The stopping criterion consisted of a function tolerances which terminated the program if the change in the objective function for a step is smaller than 1e-6.

Since a less accurate numerical approximation of the trajectory was utilized in direct collocation, to examine the feasibility of this solution, the control has been interpolated and simulated with ode45. The final cost obtained here was $J = 23.58$ and its corresponding trajectory have been plotted in Figure 13.

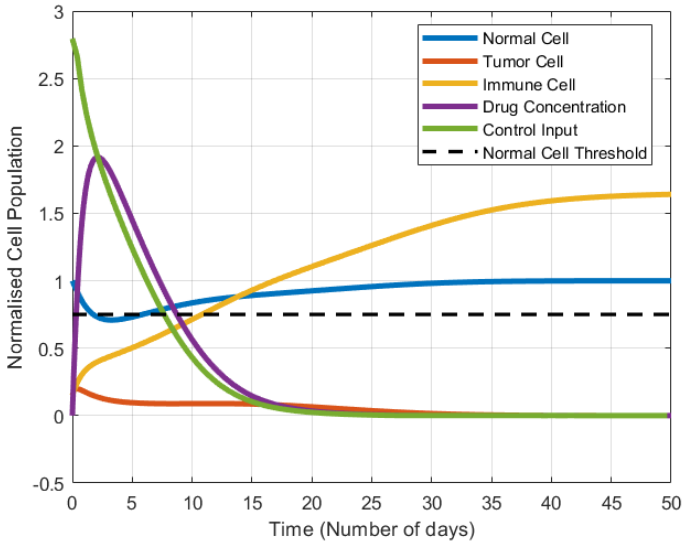

Figure 13: The optimal trajectory obtained using direct collocation with a final cost of 23.58.

By using different initial trajectories for multiple iterations, the same final converged solution was returned by the program. Additionally, observing the result in Figure 13 and the performances of other optimal control strategies, it can be assumed with a certain degree of confidence that this

solution closely resembles the global optimal control. It is not exact due to the reduced accuracy of the model simulation in the transcription method, but the approximation is assumed to be sufficient for the purpose of demonstration. The accuracy can be further improved without affecting computational effort by considering an adaptive mesh [23].

It should be aware that this method does not offer an efficient way of solving the optimal control problem as the program can take up to hours to converge and some iterations may lead to failure if not initialized correctly. The run time of the program can be dramatically improved by initializing with a near optimal solution to help convergence, such as those obtained from SDRE or the dynamic control law. The drawback of this is that it will constrain the optimisation to a particular local region of solution.

8.2 Final Result

The results of different optimal control strategies can now be compared to provide indication of their relative control optimality. These have been graphically presented in Figure 14 and their corresponding performance index J of the original objective in (44) have been tabulated in Table 8.

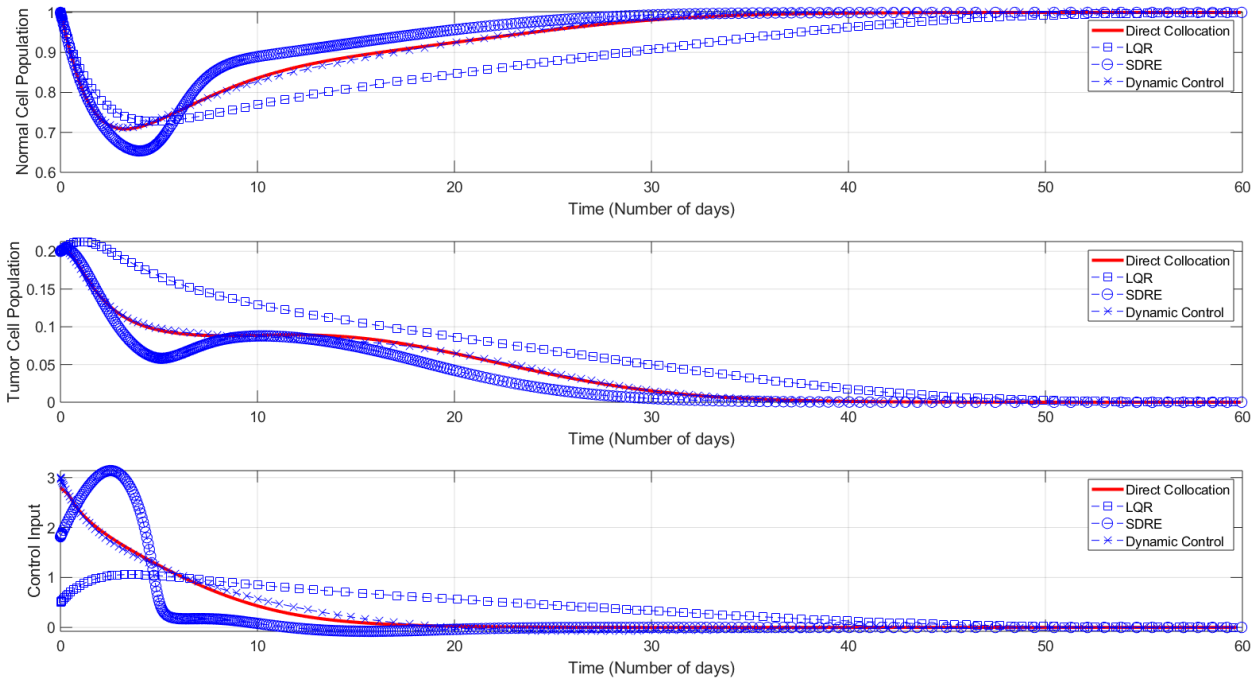

Figure 14: The evolution of normal cells, tumor cells and control input during treatment for all the methods.

Methods	Overall Cost (J)	Total Tumor Level ($\int_0^\infty T dt$)	Total Control Input ($\int_0^\infty u^2 dt$)
LQR	72.52	3.88	16.70
SDRE	27.16	1.92	33.56
Dynamic Control	23.68	2.39	23.02
Direct Collocation	23.58	2.40	22.57

Table 8: The final performance of different methods.

By assuming direct collocation provided the optimal solution, it is clear that the control mechanism that performed closest to this was the dynamic control law. This is reflected in the result of the cost J in Table 8. As expected, the LQR resulted in the worst performance due to its poor linear approximation. Although the performance of SDRE was satisfactory, its control strategy is clearly

not optimal. By optimising locally and only considering the local linearised dynamics, this has led to the unnecessary negative control at 15 seconds seen in the control plot of Figure 14, which would have been avoided if a more gradual change in the control was administered instead. From the graphs and table of results, it can be seen that even though SDRE was able to eliminate the tumor cells more quickly, the additional control effort imposed on the system outweighed this benefit in the overall cost.

In the Monte Carlo simulation without in-depth optimisation, the performance of the dynamic control seen in Figure 8 already exhibited control characteristics similar to that of the optimal solution, where the control input began at a peak and thereafter a gradual reduction in control allowed the immune cells to grow and eliminate the remaining tumor cells. Here, the flexibility of the additional control parameters was shown already to be advantageous in providing performances that are superior to the SDRE.

The dynamic control law is inherently a sub-optimal control strategy. But further optimisation using genetic algorithm, by incorporating all the available parameters in the dynamic control problem, has enabled this control to match the optimal performance of direct collocation (within 0.004%). This demonstrates the validity of the problem formulation and genetic algorithm's evolutionary process in obtaining an accurate converged solution as well as illustrating the capabilities of this dynamic control law. The additional provision of guaranteed local asymptotic stability and its strong performance gives it a significant advantage over the other methods.

9 Conclusions

9.1 Summary

In this thesis, the application of four different nonlinear control strategies on a chosen cancer model have been studied and their relative control performances compared. Their development and implementation process have also been presented and analysed, with a particular focus on a proposed dynamic control law, and relevant improvements have been made where possible.

The LQR is the simplest form of optimal control and for the particular control objective under investigation, it failed to provide any control effort, suggesting that a linear approximation was insufficient. Only once the objective had been modified to take into account the normal cells, a controlled system was attained. However, this changed the definition of the optimal control problem and resulted in a very large final cost. The SDRE is an alternative approach which creates a piece-wise linear representation of the nonlinear system and uses the LQR methodology. The simplicity of this method allowed a straightforward implementation and for one variation of the SDC matrix $A(x)$, it provided a satisfactory performance. However, it is a more computationally expensive method and its optimal performance can be quite limited.

A proposed dynamic control law shown in this thesis is a further advancement on SDRE where the approximation can be quantified and minimised through an augmented system. Implementing the control required the algebraic $\bar{P}$ solution to a matrix inequality whose existence can highly depend on the dynamical system and is non-unique. Obtaining the solution analytically can be challenging and this can be worsened with increased system complexity. Fortunately, this has not been a significant issue for this optimal control problem and a solution with variable parameters has been shown.

The experimentation of the dynamic control demonstrated the significance of the control parameters on the stability of the overall system and its sensitivity. The randomness of the successful parameters led to an exhaustive search for the optimal values using the Monte Carlo method. But once the control had been solved offline, simulations of the controlled system can be obtained with ease, allowing an iterative procedure to be considered for parameter optimisation. Both Monte Carlo and Genetic Algorithm were able to find control parameters that led to performances exceeding the SDRE. Further analysis of Genetic Algorithm provided some insight into the optimisation problem and inner workings of the dynamic control law which allowed improvements in the convergence accuracy of the program

to be made. The effectiveness of this optimisation algorithm and its convergence accuracy was further demonstrated by the near identical result of direct collocation.

The key focus of this thesis has been to demonstrate the capabilities of a dynamic control law in the application of cancer treatment and the difficulties involved when formulating a solution. A comparison of the results from different optimal control strategies has shown its ability to challenge the performance of the current state-of-the-art SDRE, at least with respect to its control optimality. There are analysis still to be performed in future works to validate its feasibility in the medical treatment environment.

9.2 Future Works

State and Control Constraints
In the original cancer model, the importance of toxicity of chemotherapy was emphasized by the author to avoid the reduction in patient's health or harmful side effects. A desired healthy threshold of $N \geq 0.75$ and control limit of $0 < u < 1$ should be fulfilled during the treatment. There conditions have not been enforced explicitly and have not been satisfied in the optimal case. Therefore, the search for this optimal solution present a separate challenge. LQR does not provide any explicit strategies for satisfying constraints whilst SDRE and the dynamic control law both offer the capability to include barrier functions to avoid a particular state. Their effectiveness in dealing with general constraints is an important part optimal control and can be further studied.

Stability
The stability of a control strategy is equally important as its optimality in an optimal control problem. The control should with stand small perturbations in the real system and not lead to any instabilities. This can be studied by experimenting with different initial conditions of the tumor model or inserting perturbations to the system during treatment, simulating unexpected behaviours such as weakening of the immune system or occasional absence of drug administration. The response of the system under these cases should be examined to ensure that the system does not enter an unstable or chaotic state and relative precautions can be put in place if these situations do occur.

Online Optimisation
The time required to solve the optimal control problem is relatively small compare to the time scale for the cancer treatment which can be weeks or months, therefore the computational efficiency of the control algorithms is not a big concern. The genetic algorithm can be incorporated with the dynamic control as a new control algorithm. The program can be run multiple times during the treatment to offer greater optimality and stability at the presence of disturbances. This approach is analogous to the Model Predictive Control algorithm [18] where the optimal control is calculated for a prediction horizon and used to forward the dynamics for short period before the control is recomputed. Since the convergence time of direct collocation can be reduced by initializing with a near optimal solution, incorporating this with the solution of the dynamic control law can also be considered as an alternative.

Model Improvements
As mentioned in the literature review, one major leap in the treatment of cancer is the application of immunotherapy with chemotherapy to boost the body's immune system which plays a crucial role in eliminating the cancer cells. The superiority of this treatment compared to chemotherapy has been demonstrated in [13]. This presents the motivation to explore a more sophisticated cancer model with greater nonlinearity and to further investigate the difficulties in obtaining a solution to the dynamic control law, especially when the model contains an unstable healthy equilibrium. This would encourage the discovery of alternative methods of obtaining the algebraic $\bar{P}$ solution if such solution is difficult or even impossible to evaluate analytically.

Bibliography

[1] William F Arnold and Alan J Laub. Generalized eigenproblem algorithms and software for algebraic riccati equations. Proceedings of the IEEE, 72(12):1746–1754, 1984.

[2] Naser Babaei, Metin U Salamci, and Tayfun Çimen. State dependent riccati equation controlled drug delivery for mixed therapy of cancer treatment. IFAC-PapersOnLine, 48(25):265–270, 2015.

[3] H Basirzadeh and S Nazari. T-lymphocyte cell injection cancer immunotherapy: an optimal control approach. Iranian Journal of Operations Research, 3(1):46–60, 2012.

[4] Arthur E Bryson. Optimal control-1950 to 1985. IEEE Control Systems Magazine, 16(3):26–33, 1996.

[5] Thalya N Burden, Jon Ernstberger, and K Renee Fister. Optimal control applied to immunotherapy. Discrete and Continuous Dynamical Systems Series B, 4(1):135–146, 2004.

[6] David Carlson. What are schur complements, anyway? Linear Algebra and its Applications, 74:257 – 275, 1986.

[7] F Castiglione and B Piccoli. Cancer immunotherapy, mathematical modeling and optimal control. Journal of theoretical Biology, 247(4):723–732, 2007.

[8] Tayfun Çimen. State-dependent riccati equation (sdre) control: A survey. IFAC Proceedings Volumes, 41(2):3761–3775, 2008.

[9] Tayfun Cimen. Systematic and effective design of nonlinear feedback controllers via the state-dependent riccati equation (sdre) method. Annual Reviews in control, 34(1):32–51, 2010.

[10] James R Cloutier, Christopher N D'Souza, and Curtis P Mracek. Nonlinear regulation and nonlinear hâĹđ control via the state-dependent riccati equation technique: Part 1, theory. In Proceedings of the First International Conference on Nonlinear Problems in Aviation and Aerospace, pages 117–130. Embry-Riddle Aeronautical Univ. Press Daytona Beach, FL, 1996.

[11] Christopher Davies. Final Year Project: Control Theory and its Role in the Optimal Treatment of Cancer. Imperial College London, 2018.

[12] Lisette G de Pillis, Weiqing Gu, K Renee Fister, T a Head, K Maples, A Murugan, T Neal, and K Yoshida. Chemotherapy for tumors: An analysis of the dynamics and a study of quadratic and linear optimal controls. Mathematical Biosciences, 209(1):292–315, 2007.

[13] Lisette G de Pillis, Weiqing Gu, and Ami E Radunskaya. Mixed immunotherapy and chemotherapy of tumors: modeling, applications and biological interpretations. Journal of theoretical biology, 238(4):841–862, 2006.

[14] Lisette G De Pillis and A Radunskaya. The dynamics of an optimally controlled tumor model: A case study. Mathematical and computer modelling, 37(11):1221–1244, 2003.

[15] Lisette G De Pillis and Ami Radunskaya. A mathematical tumor model with immune resistance and drug therapy: an optimal control approach. Computational and Mathematical Methods in Medicine, 3(2):79–100, 2001.

[16] Lisette G de Pillis, Ami E Radunskaya, and Charles L Wiseman. A validated mathematical model of cell-mediated immune response to tumor growth. Cancer research, 65(17):7950–7958, 2005.

[17] Salwa Elloumi, Ines Sansa, and Naceur Benhadj Braiek. On the stability of optimal controlled systems with sdre approach. In International Multi-Conference on Systems, Sygnals & Devices, pages 1–5. IEEE, 2012.

[18] Carlos E Garcia, David M Prett, and Manfred Morari. Model predictive control: theory and practiceâĂŤa survey. Automatica, 25(3):335–348, 1989.

[19] John H Holland. Genetic algorithms. Scientific american, 267(1):66–73, 1992.

[20] Roger A Horn and Charles R Johnson. Matrix analysis. Cambridge university press, 2012.

[21] Mehmet Itik, Metin U Salamci, and Stephen P Banks. Optimal control of drug therapy in cancer treatment. Nonlinear Analysis: Theory, Methods & Applications, 71(12):e1473–e1486, 2009.

[22] Mehmet Itik, Metin Uymaz Salamci, and Stephen Paul Banks. Sdre optimal control of drug administration in cancer treatment. Turkish Journal of Electrical Engineering & Computer Sciences, 18(5):715–730, 2010.

[23] Matthew Kelly. An introduction to trajectory optimization: how to do your own direct collocation. SIAM Review, 59(4):849–904, 2017.

[24] Donald E Kirk. Optimal control theory: an introduction. Courier Corporation, 2012.

[25] Vladimir A Kuznetsov, Iliya A Makalkin, Mark A Taylor, and Alan S Perelson. Nonlinear dynamics of immunogenic tumors: parameter estimation and global bifurcation analysis. Bulletin of mathematical biology, 56(2):295–321, 1994.

[26] Huibert Kwakernaak and Raphael Sivan. Linear optimal control systems, volume 1. Wiley-interscience New York, 1972.

[27] Joseph LaSalle. Some extensions of liapunov's second method. IRE Transactions on circuit theory, 7(4):520–527, 1960.

[28] Aleksandr Mikhailovich Lyapunov. The general problem of the stability of motion. International journal of control, 55(3):531–534, 1992.

[29] MATLAB. Matlab documentation, fmincon. `https://uk.mathworks.com/help/optim/ug/fmincon.html`. [Accessed: 2019-05-20].

[30] MATLAB. Matlab documentation, genetic algorithm. `https://uk.mathworks.com/help/gads/genetic-algorithm.html`. [Accessed: 2019-05-5].

[31] MATLAB. Matlab documentation, genetic algorithm options. `https://uk.mathworks.com/help/gads/genetic-algorithm-options.html`. [Accessed: 2019-05-5].

[32] MATLAB. Matlab documentation, lqr. `https://uk.mathworks.com/help/control/ref/lqr.html`. [Accessed: 2019-05-5].

[33] MATLAB. Matlab documentation, ode45. `https://uk.mathworks.com/help/matlab/ref/ode45.html`. [Accessed: 2018-06-5].

[34] University of Edinburgh Michael Mistry. Robot learning and sensorimotor control. `http://wcms.inf.ed.ac.uk/ipab/rlsc/lecture-notes/RLSC-MM-06.pdf`. [Accessed: 2019-05-11].

[35] Thulasi Mylvaganam and Alessandro Astolfi. Control of microgrids using a differential game theoretic framework. In 2015 54th IEEE Conference on Decision and Control (CDC), pages 5839–5844. IEEE, 2015.

[36] Sam Nazari and Bahram Shafai. Robust sdc parameterization for a class of extended linearization systems. In Proceedings of the 2011 American Control Conference, pages 3742–3747. IEEE, 2011.

[37] JD Pearson. Approximation methods in optimal control i. sub-optimal control. International Journal of Electronics, 13(5):453–469, 1962.

[38] Florian A Potra and Stephen J Wright. Interior-point methods. Journal of Computational and Applied Mathematics, 124(1-2):281–302, 2000.

[39] Anil V Rao. A survey of numerical methods for optimal control. Advances in the Astronautical Sciences, 135(1):497–528, 2009.

[40] Brian D Ripley. Stochastic simulation, volume 5. Wiley Online Library, 1987.

[41] Jean-Michel RÃľveillac. 4 - dynamic programming. In Jean-Michel RÃľveillac, editor, Optimization Tools for Logistics, pages 55 – 75. Elsevier, 2015.

[42] Mario Sassano and Alessandro Astolfi. Dynamic approximate solutions of the hj inequality and of the hjb equation for input-affine nonlinear systems. IEEE Transactions on Automatic Control, 57(10):2490–2503, 2012.

[43] Mario Sassano, Thulasi Mylvaganam, and Alessandro Astolfi. An algebraic approach to dynamic optimisation of nonlinear systems: a survey and some new results. Journal of Control and Decision, 6(1):1–29, 2019.

[44] Donald T Stansbery and James R Cloutier. Position and attitude control of a spacecraft using the state-dependent riccati equation technique. In Proceedings of the 2000 American Control Conference. ACC (IEEE Cat. No. 00CH36334), volume 3, pages 1867–1871. IEEE, 2000.

[45] Francesco Topputo and C Zhang. Survey of direct transcription for low-thrust space trajectory optimization with applications. In Abstract and Applied Analysis, volume 2014. Hindawi, 2014.

[46] A van den Bos. Appendix c: Positive semidefinite and positive definite matrices, 2007.

Appendix A MATLAB Codes

The complete MATLAB codes for the optimisation of the dynamic control parameters using Genetic Algorithm and Direct Collocation have been included in this appendix.

A.1 Genetic Algorithm Optimisation

```
% Code: Optimising the dynamic control variables using Genetic Algorithm

clear
clc
close all

% Define ploblem
ObjectiveFunction = @cancer_opt_fun;
nvars = 10;    % Number of variables
A = [];
B = [];
Aeq = [];
Beq = [];
% kappa = 1;

% Hard variable constraints
LB = [-inf,-inf,-inf,  0,  0,   0, 0,    0, 0,    0];   % Lower bound
UB = [ inf, inf, inf,inf,  inf, inf, 1,  inf, 1,  inf];  % Upper bound

% Initial population constrains
initialLB = [-200,-200,-200,   0, 0, 0,    0,   0,    0,    0];
initialUB = [ 200, 200, 200, 200, 1, 0.1,  0.5,   5,  0.5,    5];
nonlcon = [];
intcon = []; %set all var to be integers
options = optimoptions('ga','ConstraintTolerance',1e-6','PlotFcn',{@gaplotbestf,@gaplotdistance},...
        'InitialPopulationRange',[initialLB;initialUB],'FitnessScalingFcn',...
        @fitscalingprop,'SelectionFcn',@selectionremainder,...
        'EliteCount',20,'CrossoverFraction',0.9,'CrossoverFcn',@crossoverheuristic,...
        'UseParallel',true);

% solve
[x,fval,exitflag,output,populations,scores] = ga(ObjectiveFunction,nvars,...
    A,B,Aeq,Beq,LB,UB,nonlcon,intcon,options);


%% run the dynamic control law results and plot the solution
a1 = 0.2;
a2 = 0.3;
a3 = 0.1;
b1 = 1;
b2 = 1;
c1 = 1;
c2 = 0.5;
c3 = 1;
c4 = 1;
d1 = 0.2;
d2 = 1;
r1 = 1.5;
r2 = 1;
s = 0.33;
alpha = 0.3;
rho = 0.01;

tspan = [0,350];
```

```
% initial conditions
N0 = 1;
T0 = 0.2;
I0 = 0.15;
M0 = 0;
J0 = 0;
C0 = 0;

% origin shift
x1_0 = N0 - 1/b2;
x2_0 = T0;
x3_0 = I0 - s/d1;
x4_0 = M0;


% paramters taken from GA output
xi1_0 = x(1);
xi2_0 = x(2);
xi3_0 = x(3);
xi4_0 = x(4);
k = x(5);
r_constant = x(6);
gamma1 = x(7);
gamma2 = x(8);
gamma3 = x(9);
gamma4 = x(10);

% input paramters and initial conditions to simulate system
y0_DCL = [x1_0;x2_0;x3_0;x4_0;xi1_0;xi2_0;xi3_0;xi4_0;J0;C0];
opt    = odeset('Events', @myEvent);
[t2,y2] = ode45(@(t,y) nonlinear_rhs_opt(t,y,k,r_constant,gamma1,gamma2,gamma3,...
    gamma4), tspan, y0_DCL, opt);

% getting the controls and additional cost
u_dynamic = zeros(1,length(y2));
c_x_xi = zeros(1,length(y2));

for i = 1:length(y2)
    [~,u_dynamic(:,i),c_x_xi(:,i)] = nonlinear_rhs_opt(t2(i),y2(i,:),k,r_constant,...
        gamma1,gamma2,gamma3,gamma4);
end
y2(end,9:10)

% shift back the origin
y2(:,1) = y2(:,1) + 1/b2;
y2(:,3) = y2(:,3) + s/d1;

% plot states
figure
plot(t2,y2(:,1:4),'-',t2,u_dynamic,'-','linewidth',3)
hold on
plot(tspan,[0.75,0.75],'--k','linewidth',2)
legend('Normal Cell','Tumor Cell','Immune Cell','Drug Dose','Control Input','Normal Cell Limit')
grid on
xlabel('Time (days)')
ylabel('Normalised Fraction of Cell')

% plot the augmented states
figure
plot(t2,y2(:,5:8),'-','linewidth',3)
legend('\xi_1','\xi_2','\xi_3','\xi_4')
grid on
xlabel('Time (days)')
ylabel('Augmented States (\xi)')

% plot the cost
```

```
figure
plot(t2,y2(:,9:10),'-','linewidth',3)
legend('Tumor Cost','Approximated Cost')
grid on
xlabel('Time (days)')
ylabel('Normalised Fraction of Cell')



%% Objective Function for ga
function cost = cancer_opt_fun(x)

% cancer model constants
a1 = 0.2;
a2 = 0.3;
a3 = 0.1;
b1 = 1;
b2 = 1;
c1 = 1;
c2 = 0.5;
c3 = 1;
c4 = 1;
d1 = 0.2;
d2 = 1;
r1 = 1.5;
r2 = 1;
s = 0.33;
alpha = 0.3;
rho = 0.01;

%% simulate trajectory:
%state = [Normal Cell, Tumor Cell, Immune Cell, Drug, Cost, Additional Cost]

tspan = [0,50];

% initial condtions
N0 = 1;
T0 = 0.2;
I0 = 0.15;
M0 = 0;
J0 = 0;
C0 = 0;

% origin shift
x1_0 = N0 - 1/b2;
x2_0 = T0;
x3_0 = I0 - s/d1;
x4_0 = M0;

% augmented states
xi1_0 = x(1);
xi2_0 = x(2);
xi3_0 = x(3);
xi4_0 = x(4);

% dynamic control paramters
k = x(5);
r_constant  = x(6);

%% run the nonlinear dynamic control law
y0_DCL = [x1_0;x2_0;x3_0;x4_0;xi1_0;xi2_0;xi3_0;xi4_0;J0;C0];

% algebraic P solution tuning parameters
gamma_1 = x(7);
gamma_2 = x(8);
gamma_3 = x(9);
```

```
gamma_4 = x(10);

opt     = odeset('Events', @myEvent);
[t,y] = ode45(@(t,y) nonlinear_rhs_opt(t,y,k,r_constant,gamma_1,gamma_2,gamma_3...
    ,gamma_4), tspan, y0_DCL, opt);

%% define cost
cost = y(end,10);

% if inequality not satisfied, set arbitrary large cost
if isnan(cost) || isinf(cost)
    cost = 10000;
    return
end

end


%% simulate trajectory
function [y_dot,u,c_x_xi] = nonlinear_rhs_opt(t,y,k,r_constant,gamma1,gamma2_bar,gamma3,gamma4_bar)
% define ODE dynamics here to be used in solver, states x = [N,I,T,M]'

% input y-vector
x1 = y(1);
x2 = y(2);
x3 = y(3);
x4 = y(4);

x = [x1;x2;x3;x4];

% augmented states
xi1 = y(5);
xi2 = y(6);
xi3 = y(7);
xi4 = y(8);

xi = [xi1;xi2;xi3;xi4];

% cancer model constants
a1 = 0.2;
a2 = 0.3;
a3 = 0.1;
b1 = 1;
b2 = 1;
c1 = 1;
c2 = 0.5;
c3 = 1;
c4 = 1;
d1 = 0.2;
d2 = 1;
r1 = 1.5;
r2 = 1;
s = 0.33;
alpha = 0.3;
rho = 0.01;

% penalty parameters
Q11 = 0;
Q22 = 100;
Q33 = 0;
Q44 = 0.1;
Q = diag([Q11,Q22,Q33,Q44]);

% dynamic control law parameter
R = eye(4)*r_constant;
```

```
%%  the linear parametrisation F(x) and control matrix B(x)
F11 = -r2*(x1*b2+1) - c4*x2 - a3*x4;
F12 = -c4/b2;
F13 = 0;
F14 = -a3/b2;

F21 = 0;
F22 = r1*(1-b1*x2) - (c2*s/d1 + c3/b2) - c3*x1 - c2*x3 - a2*x4;
F23 = 0;
F24 = 0;

F31 = 0;
F32 = -c1*s/d1 + rho*s/d1/(alpha + x2);
F33 = -d1 + rho*x2/(alpha+x2) - c1*x2 - a1*x4;
F34 = -a1*s/d1;

F41 = 0;
F42 = 0;
F43 = 0;
F44 = -d2;

F = [F11,F12,F13,F14;F21,F22,F23,F24;F31,F32,F33,F34;F41,F42,F43,F44];

B = [0;0;0;1];


%% algebraic P solutions paramter derivation
gamma4 = -sqrt(Q44 + 1) + sqrt(Q44+1+2.7225*Q33) + gamma4_bar;
K4 = -1 + sqrt(Q44+1) + gamma4;
D4 = K4^2 + 2*K4 - Q44;

alpha3 = 250000/1089*D4;

% average between 0 and 1
K3 =  alpha3*(gamma3*(1/25 - 1/25*sqrt(1-Q33/alpha3*25^2)) +...
    (1-gamma3)*(1/25 + 1/25*sqrt(1-Q33/alpha3*25^2)));
D3 = 2/25*K3 - Q33 - 1089/250000*K3^2*1/D4;


gamma2 = (1+0.0021*K3^2/(D4*D3))^2/(1/Q11 - 1/(100*D4)) + gamma2_bar;
K2 = 800/169*(Q22 + 0.1018/(D3)*K3^2 + gamma2);
D2 = 169/800*K2 - Q22 - 0.1018*K3^2/D3;

alpha1 = 1/(1/(100*D4) + (1+0.0021*K3^2/(D4*D3))^2/D2);

% an average between 0 and 1
K1 = alpha1*(gamma1*(1-sqrt(1-Q11/alpha1)) + (1-gamma1)*(1+sqrt(1-Q11/alpha1)));
D1 = -K1^2*(1/alpha1) + 2*K1 - Q11;

%% algebraic P solution
P_xi_jacobian_x = [ K1*x1,             K1*x1,          0,    (K1*x1)/10;
 K2*x2,                                (3*K2*x2)/2, (K2*x2)/2, (3*K2*x2)/10;
     0, K3*x3*(xi2/(100*(xi2 + 3/10)^2) - 1/(100*(xi2 + 3/10)) + 1),          0,    (K3*x3)/5;
     0,                                 0,          0,             0];

P44_xi = K4;
P33_xi = K3*(xi2 + xi4/5 - xi2/(100*(xi2 + 3/10)) + 1/5) ;
P22_xi = K2*(xi1 + (3*xi2)/2 + xi3/2 + (3*xi4)/10 + 13/40);
P11_xi = K1*(xi1 + xi2 + xi4/10 + 1);

P_xi = diag([P11_xi,P22_xi,P33_xi,P44_xi]);


%% value function derivatives
dv_dx = x'* P_xi +  (x - xi)'*R;
```

```
dv_dxi =  1/2 * x'* P_xi_jacobian_x - ( x - xi)'*R;


%% ========================================================================================
% input dynamic control law
u = - (B' * dv_dx');

%% check the inequlaity conditions and stop ode45 if not satisfied
c_x_xi = dv_dx*F*x + 1/2*x' * Q * x - ...
    1/2*dv_dx*B*B'*dv_dx'  - k* dv_dxi * dv_dxi';

if c_x_xi > 0
    y_dot = inf*ones(10,1);
    return
end

v = 1/2*x'*P_xi*x + 1/2*(x-xi)'*R*(x-xi);

if v < 0
    y_dot = inf*ones(10,1);
    return
end


%% forwarding dynamics
y_dot = F*[x1;x2;x3;x4] + B*u;
xi_dot = -k * dv_dxi';
y_dot(5:8) = xi_dot;
y_dot(9) =  1/2 * x' * Q * x + 1/2 * u^2;
y_dot(10) = - c_x_xi;


end

%% event to stop ode45
function [value, isterminal, direction] = myEvent(T, Y)

% stop the code if nan appears, which means that the controls have not
% satisfied the inequality

    if any(isnan(Y))
        condition = 0;
    else
        condition = 1;
    end

    value      = condition;
    isterminal = 1;   % Stop the integration
    direction  = 0;

end
```

A.2 Direct Collocation

```
% Code: Direct Collocation to solve the optimal control problem

clc
clear
close all

% load an optimised solution to initialise
load('DirColInitial4')
```

```
% mesh size
gridN = 300;
tic

% objective function
obj_fun = @cc_obj_fun;

% The initial parameter guess vector = [t_f; x1; x2; x3; x4 ; u]
% A ramp
% x0 = [120; ones(gridN,1); linspace(0.2,0,gridN)'; linspace(-1.5,0,gridN)';...
%     linspace(1,0,gridN)'; linspace(3,0,gridN)'];

% load a partially optimised solution to initialise
load('DirColInitial4')
x0 = [120; x1I'; x2I'; x3I'; x4I' ; x5I'];

% No linear inequality or equality constraints
A = [];
b = [];
Aeq = [];
Beq = [];

% Lower bound and upper bound
lb = [0;    ones(gridN , 1) * (0 - 1);  zeros(gridN, 1); ones(gridN , 1) * -1.5;
    zeros(gridN, 1); zeros(gridN, 1)];
ub = [Inf;  ones(gridN , 1) * inf;   ones(gridN, 1) *inf ; ones(gridN , 1) * inf;
    ones(gridN , 1)*inf ; ones(gridN, 1)*inf];

% Options for fmincon
options = optimoptions(@fmincon, 'TolFun', 0.00000001, 'MaxIter', 10000, ...
                       'MaxFunEvals', 1000000, 'Display', 'iter', ...
                       'DiffMinChange', 0.001, 'Algorithm', 'sqp');

% Solve for the best simulation time + control input
[optimal,fval] = fmincon(obj_fun, x0, A, b, Aeq, Beq, lb, ub, ...
              @cc_cons, options);

%% Plotting the solution
% Discretize the times
sim_time = optimal(1);
delta_time = sim_time / gridN;
times = 0 : delta_time : sim_time - delta_time;

% Get the states and control out of the vector
x1      = optimal(2             : 1 + gridN);
x2      = optimal(2 + gridN   : 1 + gridN * 2);
x3      = optimal(2 + gridN *2  : 1 + gridN * 3);
x4      = optimal(2 + gridN *3  : 1 + gridN * 4);
u       = optimal(2 + gridN *4  :  end);

%shift back from the origin
x1 = x1 + 1;
x3 = x3 + 1.65;

% Make the plots
figure();
plot(times, x1,'o',times, x2,'o',times, x3,'o',times, x4,'o',times,u,'o');
legend('Normal Cell','Tumor Cell','Immune Cell','Drug Concentration','Control Input')
title('Optimal Output');
xlabel('Time (s)');
ylabel('Cell Numbers  (m/s^2)');
grid on

disp(sprintf('Finished in %f seconds', toc));
```

```
%% objective function
function [cost,grad] = cc_obj_fun(X_opt)
% X_opt input is in the form of [t_final;x1;x2;x3;x4;u]

%number of grids used in discretisation
gridN = (length(X_opt) - 1)/5;

% time discretisation
t_f = X_opt(1);
t = linspace(0,1,gridN)'*t_f;
dt = t(2) - t(1);

% number of states
n_s = 4;

% reshape the states into [x1;x2;x3;x4]
x = X_opt(2:length(X_opt) - gridN);
x = reshape(x,gridN,n_s);

% get control
u = X_opt(length(X_opt) - gridN + 1:end);

% calculate cost using trapzoidal integration
Q =[0,100,0,0.1]';
y = 1/2*(x.^2*Q + u.^2);
cost = trapz(t,y);

% get gradient
grad_states = [x(1,:)*diag(Q)*dt; x(2:end-1,:)*diag(Q)*2*dt; x(end,:)*diag(Q)*dt];
grad_control = [u(1)*dt; u(2:end-1)*2*dt; u(end)*dt];

grad = ones(size(X_opt));
grad(1) = 0;
grad(2:end) = [grad_states(:);grad_control];

end

%% constraints of direct collocation
function [C , Ceq] = cc_cons(X_opt)


%number of grids used in discretisation
gridN = (length(X_opt) - 1)/5;

% time discretisation
t_f = X_opt(1);
t = linspace(0,1,gridN)'*t_f;
dt = t(2) - t(1);

% number of states
n_s = 4;

% reshape the states into [x1;x2;x3;x4]
x = X_opt(2:length(X_opt) - gridN);
x = reshape(x,gridN,n_s);

x_ll = x(1:end-1,:);
x_rr = x(2:end,:);

% get control
u = X_opt(length(X_opt) - gridN + 1:end);

% get f(x,u)_k and f(x,u)_k+1
x_dot = cc_fun(t,x,u);
x_dot_ll = x_dot(1:end-1,:);
x_dot_rr = x_dot(2:end,:);
```

```
% get collocation points time control and states
t_c = (t(1:end-1) + t(2:end))/2;
u_c = (u(1:end-1) + u(2:end))/2;
x_c = 1/2 * (x_ll + x_rr) + dt/8*(x_dot_ll - x_dot_rr);
x_dot_c = cc_fun(t_c,x_c,u_c);

% the constraint of the dynamical system
Ceq = x_ll - x_rr + dt/6*(x_dot_ll + x_dot_rr + 4*x_dot_c);
Ceq = Ceq(:);

% initial and final constraints
x1_0 = 0;
x2_0 = 0.2;
x3_0 = -1.50;
x4_0 = 0;


% input into equality constraints
Ceq = [Ceq; x(1,1) - x1_0; x(1,2) - x2_0; x(1,3) - x3_0; x(1,4) - x4_0;
    x(end,1); x(end,2); x(end,3); x(end,4)];
C = [];

end

%% calculate the rates
function [x_dot] = cc_fun(t,x,u)

x1 = x(:,1);
x2 = x(:,2);
x3 = x(:,3);
x4 = x(:,4);

% parameters
a1 = 0.2;
a2 = 0.3;
a3 = 0.1;
b1 = 1;
b2 = 1;
alpha = 0.3;
c1 = 1;
c2 = 0.5;
c3 = 1;
c4 = 1;
d1 = 0.2;
d2 = 1;
r1 = 1.5;
r2 = 1;
s = 0.33;
rho = 0.01;


% non-linear dynamics shifted
x1_dot = -r2*x1.*(1+b2*x1) - c4*x2.*x1 - c4/b2*x2 -a3*x4.*x1 - a3/b2*x4;
x2_dot = r1*x2.*(1-b1*x2) - (c2*s/d1 + c3/b2)*x2 - c3*x2.*x1 - c2*x2.*x3 - a2*x4.*x2;
x3_dot =  - c1*s/d1*x2 - d1*x3 - a1*s/d1*x4 + rho*s/d1*x2./(alpha + x2) + rho*x2.*x3./(alpha + x2) .
    - c1*x3.*x2 - a1*x4.*x3;
x4_dot = u - d2*x4;


x_dot = [x1_dot,x2_dot,x3_dot,x4_dot];

end
```

Appendix B Additional Parameter Study

This appendix contains some studies to provide some understanding behind the effect of the parameters κ and R and their sensitivity on the control input during the treatment.

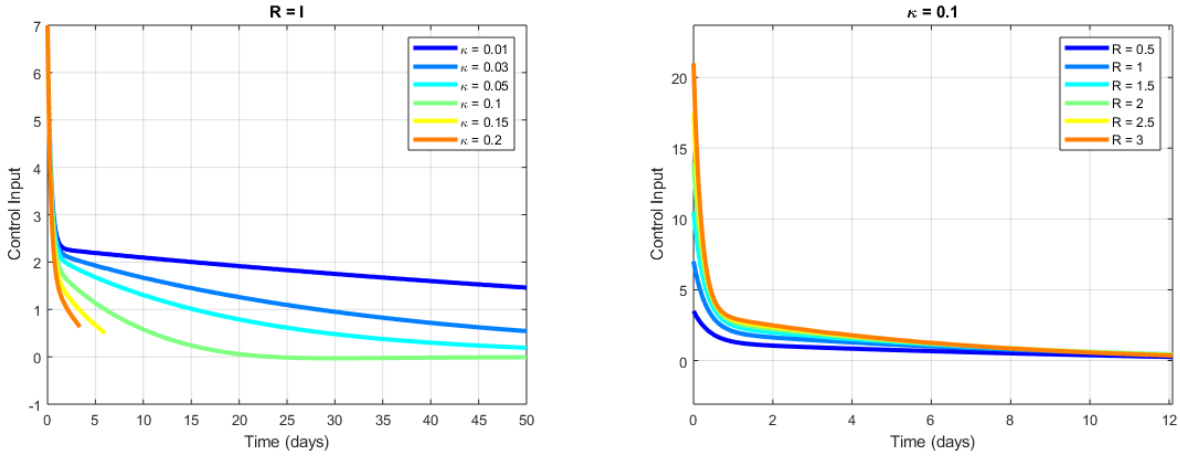

Figure B.1: The effects of varying κ and R on the control input in the controlled system.

The impact of κ on the controlled system is much more significant at later stages of the treatment whilst R affects the beginning. Increasing R has dramatic effects on the initial control dosage. Note, even though the inequality conditions may not satisfied, it can still result in a overall stable system where the unstable controls can drive the system back into stable region. This is mostly due to the natural stability within the original system and not a result the control theory.

Appendix C Monte Carlo Extended Search Space

This appendix contains some additional Monte Carlo simulations on an extended search space to demonstrate the existence of valid control parameters over a wide range of values.

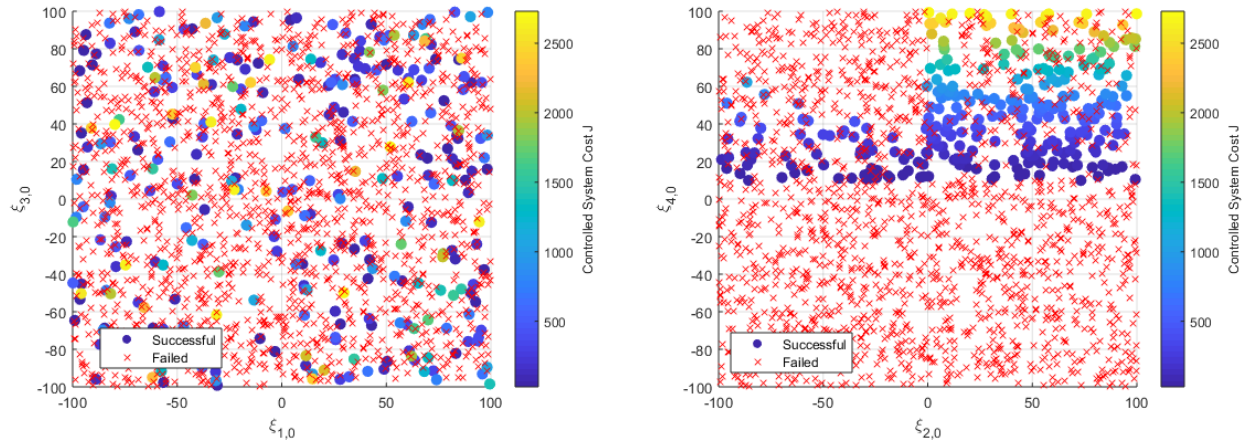

Figure C.1: Monte Carlo simulation of 2000 samples for a $\xi_{i,0}$ search space of $[-100, 100]$ at $\kappa = 0.05$ and $R = I$.

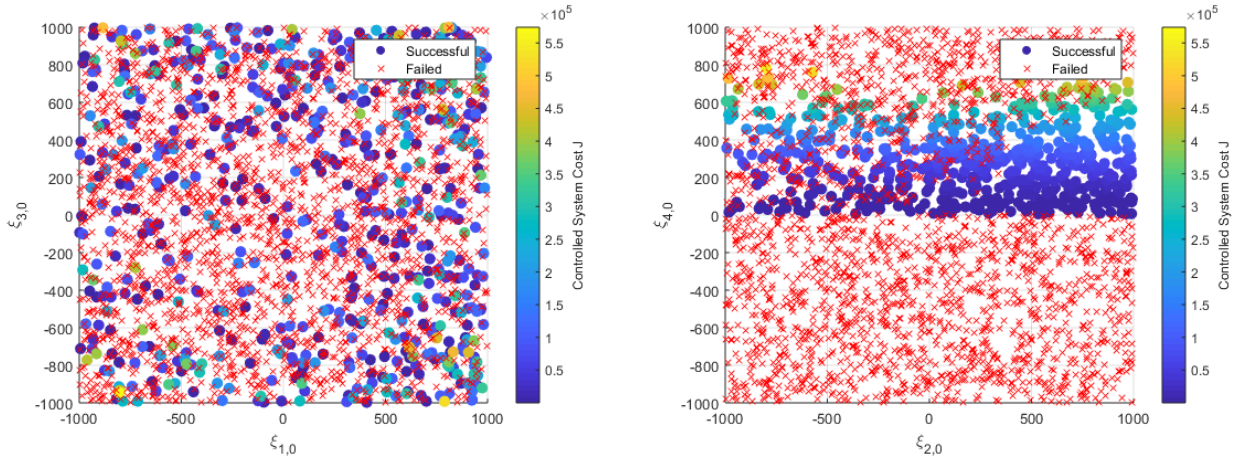

Figure C.2: Monte Carlo simulation of 3000 samples for a $\xi_{i,0}$ search space of $[-1000, 1000]$ at $\kappa = 0.05$ and $R = I$.

There are no obvious patterns on the plane of $(\xi_{1,0}, \xi_{3,0})$ other than a few noticeable patches where less solutions exist. On the $(\xi_{2,0}, \xi_{4,0})$ plane, the similar threshold for $\xi_{4,0}$ exist on these larger search spaces as for smaller search spaces.