System Identification and Advanced Tracking Strategies for Linear and Nonlinear Control Systems

A THESIS
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Syed Salik Hafeez

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

Dr.Desineni Subbaram Naidu

December, 2019

Acknowledgements

I would like to express my great indebtedness to my supervisor, Dr. Subbaram Naidu, who was abundantly helpful and offered invaluable assistance, support and guidance. Dr. Subbaram Naidu inspired me greatly to work on this thesis. His willingness to motivate me contributed tremendously to this work. To him, I am sincerely grateful for offering much of his precious time revising and correcting many of the shortcomings of this work. Really, I was honored to work under his supervision. Also, I was very pleased to attend his graduate courses specially "Optimal Control System" that gave me an insight knowledge of the novel advance control techniques. His professional teaching is amazing and makes the scientific material interesting and easy for understanding. I would like to extend my appreciation to Dr.Jing Bai and Dr.Arshia Khan, for carefully reading and providing useful comments and suggestions on this dissertation.

I would like to express appreciation to my parents, who was always my support in the moments when there was no one to answer my queries.At the end, I would like to dedicate this dissertation to my family. Without you, this dissertation could not be completed. May God bless everyone.

Abstract

The research in the master dissertation addresses development of an appropriate model of a dynamic system using observed data combined with basic mechanics and dynamics, prior knowledge of relationships between parameters.The main idea of system identification is studying the behavior of existing structures by recording the output or input-output in discrete time signals. The input-output description of a discrete-time system consists of a mathematical expression which explicitly defines the relation between the input and output signals. Further evaluating the key points for the model accuracy requirements to control estimate and predict according to the input. Also shedding light on different types of tools and techniques can be utilized to determine the dynamics of a system.

Next, this research presents a new and computationally efficient online technique for infinite-horizon and finite-horizon for linear and nonlinear dynamical systems. This technique is based on change of variables that converts the nonlinear differential Riccati equation to a linear Lyapunov differential equation. During online implementation, the Lyapunov equation is solved in a closed form at any given time step. Further, an online technique is presented for finite-horizon nonlinear tracking problems. The idea of the proposed technique is to integrate the Kalman filter algorithm and the finite-horizon SDRE technique. Unlike the ordinary methods which deal with the linearized system, this technique estimates the unmeasured states of the nonlinear system directly by converting into SDC (state dependent coefficient)form for each time step, and this makes the proposed technique effective for a wide range of operating points.

Further, the proposed infinite-horizon nonlinear technique is used to regulate the states of *Mathieu equation*, tracking of *force damped pendulum* system states and regulating the angle of an inverted pendulum on a *cart pole* system. Moreover, finite-horizon nonlinear tracking technique is used to regulate the ball position and gear angle of a *ball and beam* system and angle tracking of the flight dynamics and control of *vertical lift-off vehicle* system to demonstrate the effectiveness of the developed technique. Regulation and tracking of the its roll and pitch angles keeping the yaw constant are further presented to demonstrate the effectiveness of the developed technique.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 System, Models and Prediction

In modern day life we are highly dependent on various Model system to predict the near future changes like weather forecast, how the certain aircraft model is going to behave in a given set of conditions. Lindskog [21] describes an early ancestors made models.The hunter's survival, e.g.,was tightly coupled to his hunting skills and his ability to avoid predators. Without pondering on the model concept, he no doubt learned to distinguish between a prey and a man-eater, how to hunt the prey and how to defend himself against the predator. When confronted with an animal species never seen before he used earlier gained experience to classify it and predict its behavior. Sometimes the prediction turned out to be incorrect,i.e., the animal reacted different than expected, and the internal model had to be refined. But as the number of such experiences increased the hunter's predictability improved although it was never perfected - the model still had its limited range of validity.Thus the hunter's model describes the behavior of animals - the true system, and it was developed through education, practice and experience - the performed experiments.

1.1.1 Model Evolution

Selection of the correct and appropriate Model structure is one of the challenge in system Identification that is addressed by Ljung and Glad [45] it categorise physical, mathematical and verbal models.

- Verbal Model: A verbal model is qualitative in nature, a model is described in words for example the growth rate of the population and rising of water levels due to rain.

- Mathematical Model: such a model the relationships between the variables observable in the system are described by mathematical relations (often equations). To achieve a suitable generality we will deal with equations that are dynamic, i.e., the quantities may depend on earlier values on one or more variables. Notice, however, that this does not exclude the presence of purely algebraic, or static, relations.

- Physical Model: Physical models are often small scale, models of ships, aircraft etc. are developed with the purpose to investigate the behavior of the real system under realistic conditions.

There are many reasons for developing mathematical models of already existing systems. Besides the sheer curiosity to know and understand mother Nature, some of the most common are given by the following list (which without doubt is incomplete):

- Because of safety and/or economical reasons it might be impractical or impossible to perform experiments on the real system. Thus experiment with the model instead.

- We are interested in physical states which must be monitored but that are not directly available through measurements, and therefore we try to deduce their values using a model.

- More advanced control structures require a model of the system.

- The model can be used in an operator decision support system. Before making a decision the operator can ask the model questions such as "What will happen if I choose this setting?".

It is imperative to choose the correct design Model structure and go through number of iterations to construct a number of models of the intended system. A verity of tools and methods are discussed to evaluate the model [23]

1.1.2 Optimal Control for Linear and Nonlinear Systems

Many systems, physical, chemical, and economical, can be modeled by mathematical relations, such as deterministic and/or stochastic differential and/or dierence equations. These systems then change with time or any other independent variable according to the dynamical relations. It is possible to steer these systems from one state to another state by the application of some type of external inputs or controls [28]. If this can be done at all, there may be different ways of doing the same task. If there are different ways of doing the same task, then there may be one way of doing it in the best way. This best way can be minimum time to go from one state to another state, or maximum thrust developed by a rocket engine. The input given to the system corresponding to this best situation is called optimal control. The measure of best way or performance is called performance index or cost function. Thus, we have an optimal control system, when a system is controlled in an optimum way satisfying a given performance index.

Optimal tracking control algorithm and strategies [28] are applied to the linear [44] and nonlinear systems.In the case of infinite-horizon optimal tracking control, the system is not tracked until the time reaches infinity (on large time), while for the finite case, the system must be tracked to a reference trajectory for a finite duration of time[46]. Since many limitations exist in traditional optimal tracking control approaches, such as plant inversion [47]and linearization, it is necessary to design direct optimal tracking control schemes for nonlinear systems. Numerous design methodologies exist for the control design of nonlinear systems such as feedback linearization technique and slidng mode control (Structure variable control) which are one of the most used techniques in nonlinear control dynamical systems[7]. One of the recently developed techniques for the optimal control of nonlinear systems is the State Dependent Riccati Equation (SDRE) [18].

1.2 Background Knowledge

The advanced control theory concerned with multiple inputs and multiple outputs (MIMO) is based on state variable representation in terms of a set of rst-order dierential equations. Here, the system (plant) is characterized by state variables in linear time-invariant form as

$$\dot{x}(t) = Ax(t) + Bu(t), \tag{1.1}$$

$$y(t) = Cx(t) + Du(t), \tag{1.2}$$

where dot denotes dierentiation with respect to (w.r.t.) t, x(t), u(t), and y(t) are n, r, and m dimensional state, control, and output vectors respectively, and A is nn state, B is nr input, C is mn output, and D is mr transfer matrices. Similarly, a nonlinear dynamical system is characterized by

$$\dot{x}(t) = f(x(t), u(t), t), \tag{1.3}$$

$$y(t) = g(x(t), u(t), t) \tag{1.4}$$

where f and g are n and m dimensional vectors respectively.

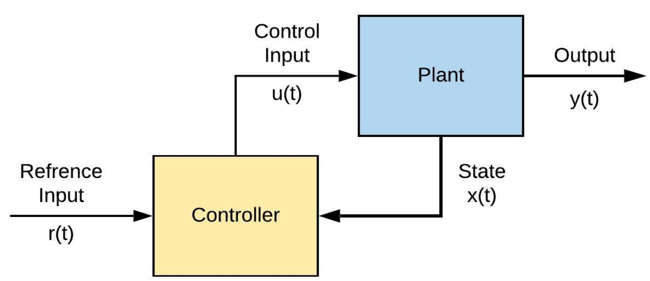

Figure 1.1: Advanced Control Configuration

The advanced theory dictates that all the state variables should be fed back after suitable weighting. We see from Fig.1.1 that in advanced control conguration, the input u(t) to the plant is determined by the controller driven by system states x(t) and reference signal r(t), all or most of the state variables are available for feedback control, and it depends on well-established matrix theory, which is amenable for large scale computer simulation [28]

Optimization is a very desirable feature in day-to-day life. The main objective of optimal control is to determine control signals that will cause a process (plant) to satisfy some physical constraints and at the same time extremize (maximize and minimize) a chosen performance criterion (performance index or cost function).

The formulation of optimal control problem requires

1. a mathematical description (or model) of the process to be controlled (generally in state variable form),

2. a specication of the performance index, and

3. a statement of boundary conditions and the physical constraints on the states and/or controls.

In modern control theory, the optimal control problem is to nd a control which causes the dynamical system to reach a target or follow a state variable (or trajectory) and at the same time extremize a performance index. A performance index in general form can be written as

$$J = x^T(t_f)Fx(t_f) + \int_{t_0}^{t_f} [x^T(t)Qx(t) + u^T(t)Ru(t)]dt, \tag{1.5}$$

where, $x(t_0)$ is fixed or given initial time, $x(t_f)$ is free (or unspecified in advance) final time, $x(t_f)$ is a specied final state, F is a positive semi-definite matrix, $x(t)$ is the error between the desired and the actual values, Q is a weighting matrix, which is positive semi-definite matrix, R is a positive definite matrix and (T) denotes the transpose. Note that the matrices Q and R may be time varying. The particular form of performance index 1.5 is called quadratic form.

Also, the performance index can be in the general (non-quadratic) form as

$$J = S(X(t_f), tf) + \int_{t_0}^{t_f} V(x(t), u(t), t)dt, \tag{1.6}$$

where, $S(x(tf), tf)$ is called the terminal cost function; $S(x(t), u(t), t) = x0(tf)Fx(tf)$ and $V(x(t), u(t), t)$ is called the integral cost function; $V(x(t), u(t), t) = x0(t)Qx(t) + u0(t)Ru(t)$. There are many other forms of cost functions depending on our performance specications. However, the above mentioned performance indices (with quadratic forms) lead to some very elegant results in optimal control systems [28]

1.3 Organization of Thesis

- Chapter 1 introduces the analytic goals, background and the contributions of the work pursued in this thesis.

- Chapter 2 briefly presents the System Identification, Methods, Model Set and criterion of fit, explain dynamics of physical models.
- In Chapter 3 presents the infinite-horizon nonlinear regulation using SDRE and simulation results presented using Mathieu equation.
- In Chapter 4 describes the strategy of infinite-horizon nonlinear Tracking using SDRE and presents simulation results using Force damped pendulum and cart-pole system.
- Chapter 5 The development of the finite-horizon nonlinear regulating for systems and verification of results through Ball and Beam and Three DOF Hover system dynamics.
- Chapter 6 The development of the finite-horizon nonlinear Tracking for systems is presented and results are presented utilising an electro-mechanical model of DC motor in this Chapter.
- Chapter 7 Finally, conclusions and future directions of research are detailed in this chapter.

Chapter 2

System Identification

System identification is the art and science of building mathematical models of dynamic systems from observed input-output data[22]. It can be seen as the interface between the real world of applications and the mathematical world of control theory and model abstractions. As such, it is an ubiquitous necessity for successful applications. System identification is a very large topic, with different techniques that depend on the character of the models to be estimated: linear, nonlinear, hybrid, non-parametric etc. At the same time, the area can be characterized by a small number of leading principles, e.g. to look for sustainable descriptions by proper decisions in the triangle of model complexity, information contents in the data, and effective validation.

The purpose with the chapter is to introduce main concepts, algorithms and ideas detail of Model study, more comprehensive details on System identification are given in Ljung [45] Soderstorm and Stoica [14].

2.1 Model Structure and Selection Criterion

The system identification procedure is highly iterative in nature and show three main ingredients, all of which in some degree are subject to personal judgments:

- **The data Z_N:** To be able to estimate models we first need data, which will depend both on the true system and on the experimental conditions. We will by Z_N denote a data set containing N measurements that are organized as

$$Z^N = [y^N u^N], \tag{2.1}$$

$$y^N = [y(1)y(2)...y(N)]^T, \tag{2.2}$$

$$u^N = [u(1)u(2)...u(N)]^T, \tag{2.3}$$

with y^N being a sequence of ordered outputs and u^N a sequence of observable inputs. Sometimes there is no observable input at all, which means that the models must be based only on measured outputs ($Z^N = y^N$). In the literature this particular situation is labeled time series modeling.

- **The model structure** $M*$: It is generally agreed upon that the single most difficult step in identification is that of model structure selection (and this is the very reason for why we later spend so much time on it). Roughly speaking the problem can be divided into three subproblems. **The first one** is to specify the type of model set to use. To be more specific, this involves selection between linear and nonlinear models, between black boxes, semi-physical and physically parameterized models, and so forth. **Second** is the size of the model set must be decided. This includes the choice of possible variables and combination of variables to use in the model description. It also involves fixing orders and degrees of the model types, usually to some intervals. Once these two issues are settled we in principle have determined a model set $M*$ over which the search for a model can be conducted. Notice though that M can be a very large subset, but by using o priori knowledge it can often be reduced significantly. The **last item** to consider is how to parameterize the model set $M*$ so that the estimation algorithms stand as good chances as possible to find reasonable parameter values. To classify the thesis, the latter issue is the main topic of Part II, while Part III largely is concerned with the former two matters. Assuming from now on that the members of $M*$ can be parameterized by a finitely dimensional parameter vector $\theta \in D_M \subset R^d$, a particular model corresponding to θ is denoted $M(\theta)$. The model structure to which the model belongs is defined by the mapping

$$M : \theta \in D_M M(\theta) \in M*, \tag{2.4}$$

Instead of using this system theoretic notation, it can be represented by a function

$$\hat{y}(t|\theta) = g(t, \theta, \psi(t)), \tag{2.5}$$

where $\hat{y}(t|\theta)$ accentuates that the function $g(\cdot)$ is predicted. The search for a model is thus conducted over the parameters θ, and is based on $\psi(t)$ which is composed of input signals up to index t and output signals up to index $t-1$.

- **The selection criterion** $V_N(\theta, Z^N)$: The purpose of the selection criterion is to rank different models according to some pre-determined cost function (hence each model is assigned a quality mark). The criterion can come in several shapes, although we shall here adopt a scalar measure of the fit between the predicted and the measured value, i.e., a measure based on

$$V_N(\theta, Z^N) = \frac{1}{N}\sum_{t=t^o}^{N} l(y(t) - \hat{y}(t|\theta)), \tag{2.6}$$

where $l(\cdot)$ is a positive scalar valued function, typically chosen to be quadratic, i.e., $l(\cdot) = |\cdot|^2$. Sometimes $l(\cdot)$ is modified to increase "slower" than quadratic for large errors, since the criterion then becomes more robust (i.e. to large seldom occurring measurement errors).

Once these items are settled we have implicitly defined the searched for model. It then "only" remains to estimate the parameters θ and to decide upon whether the model is good enough or not. If the model cannot be accepted some or even all of the entities above have to be reconsidered; in the worst case we must start from the very beginning and collect new data. Thus system identification is iterative, and the model acceptance criterion show personal taste characteristics

2.2 Black Box Linear Model Structure

"A black box structure is one where the parameterization in terms of θ is chosen so that the family of models $g(t, \theta, \psi(t)), \theta \in D_M$ covers as "many common and interesting" ones as possible" Quoting [23]

The most common class of model structures used for black box modeling is the following linear one:

$$y(t) = G(q, \theta)u(t) + H(q, \theta)e(t), \tag{2.7}$$

where q is the shift operator, and $G(q, d)$ and $H(q, \theta)$ are rational transfer functions, both assumed to be proper and $H(q, \theta)$ additionally assumed to be monic. Here, $e(t)$ is the disturbance contribution, which for analysis purposes normally is assumed to be white noise.

$$\hat{y}(t|\theta) = H^{-1}(q, \theta)G(q, \theta)u(t) + [1 - H^{-1}(q, \theta)], \tag{2.8}$$

and hence $g(\cdot)$ becomes

$$g(t, \theta, \psi(t)) = H^{-1}(q, \theta)G(q, \theta)u(t) + [1 - H^{-1}(q, \theta)]y(t), \tag{2.9}$$

A straightforward and general parameterization of $Equation$(2.7) is given by (dropping for simplicity the θ argument)

$$y(t) = q^{-nk} \frac{B(q)}{A(q)F(q)} u(t) + \frac{C(q)}{A(q)D(q)} e(t), \tag{2.10}$$

with nk representing the delay between the input signal $u(t)$ and the output signal $y(t)$.and with polynomials defined as

$$A(q) = 1 + a_1 q^{-1} + ... + a_{na} q^{-na}, \tag{2.11}$$

$$B(q) = b_1 + b_2 q^{-1} + ... + b_{nb} q^{-nb+1}, \tag{2.12}$$

$$C(q) = 1 + c_1 q^{-1} + ... + c_{nc} q^{-nc}, \tag{2.13}$$

$$D(q) = 1 + d_1 q^{-1} + ... + d_{nd} q^{-nd}, \tag{2.14}$$

$$F(q) = 1 + f_1 q^{-1} + ... + f_{nf} q^{-nf}, \tag{2.15}$$

Applying Equation (2.8) to this model structure gives the predictor

$$y(t) = q^{-nk} \frac{B(q)}{A(q)F(q)} u(t) + \left[1 - \frac{A(q)D(q)}{C(q)}\right] y(t), \tag{2.16}$$

Few Model structures are described below:

$$ARX: y(t) = q^{-nk}\frac{B(q)}{A(q)}u(t) + \frac{1}{A(q)e(t)}, \tag{2.17}$$

$$FIR: y(t) = q^{-nk}B(q)u(t) + e(t), \tag{2.18}$$

$$ARMAX: y(t) = q^{-nk}\frac{B(q)}{A(q)}u(t) + \frac{C(q)}{A(q)e(t)}, \tag{2.19}$$

$$ARMA: y(t) = \frac{C(q)}{A(q)}e(t), \tag{2.20}$$

$$OE: y(t) = q^{-nk}\frac{B(q)}{F(q)}u(t) + e(t), \tag{2.21}$$

$$BJ: y(t) = q^{-nk}\frac{B(q)}{F(q)}u(t) + \frac{C(q)}{D(q)e(t)}, \tag{2.22}$$

2.3 Non Linear Model Structure

Nonlinear models when a linear model provides a poor fit to the measured output signals and cannot be improved by changing the model structure or order. Nonlinear models have more flexibility in capturing complex phenomena than the linear models of similar orders.[31][3]. Nonlinear Dynamic models have states, where a state vector contains the information of the past.
The general form of a model in discrete time is:

$$y(t) = f(u(t-1), y(t-1), u(t-2), y(t-2), ...), \tag{2.23}$$

Such a model is nonlinear if the function f is a nonlinear function. f may represent arbitrary non-linearities.

2.3.1 Structure of Nonlinear ARX Model

A nonlinear ARX model consists of model regressors and a nonlinearity estimator[14]. The nonlinearity estimator comprises both linear and nonlinear functions that act on the model regressors to give the model output. This block diagram represents the structure of a nonlinear ARX model in a simulation scenario.
The Matlab software computes the nonlinear ARX model output y in two stages:

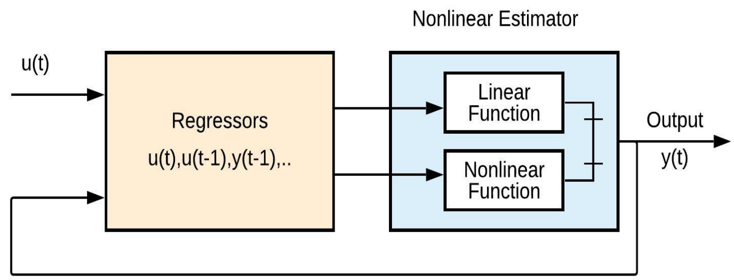

Figure 2.1: Nonlinear ARX Model Structure

1. It computes regressor values from the current and past input values and past output data.In the simplest case, regressors are delayed inputs and outputs, such as $u(t-1)$ and $y(t-3)$. These kind of regressors are called standard regressors.

2. It maps the regressors to the model output using the nonlinearity estimator block. The nonlinearity estimator block can include linear and nonlinear blocks in parallel. For example:

$$F(x) = L^T(x-r) + d + g(Q(x-r)), \tag{2.24}$$

Here, x is a vector of the regressors, and r is the mean of the regressors x. $L^T(x)+d$ is the output of the linear function block and is affine when $d0$. d is a scalar offset. $g(Q(x-r))$ represents the output of the nonlinear function block. Q is a projection matrix that makes the calculations well conditioned. The exact form of $F(x)$ depends on your choice of the non-linearity estimator.

2.4 System Modeling

Identification is an exercise which is used for explaining the relation between the input and output of the system. The algorithm for system identification process which are useful for understanding the whole process. According to this algorithm for any plant we collect experiment from experimental design and after collect data we choose an appropriate model structure which is helpful for system and perform more than other. After choosing model employ a criterion which is fit for the system after that estimate

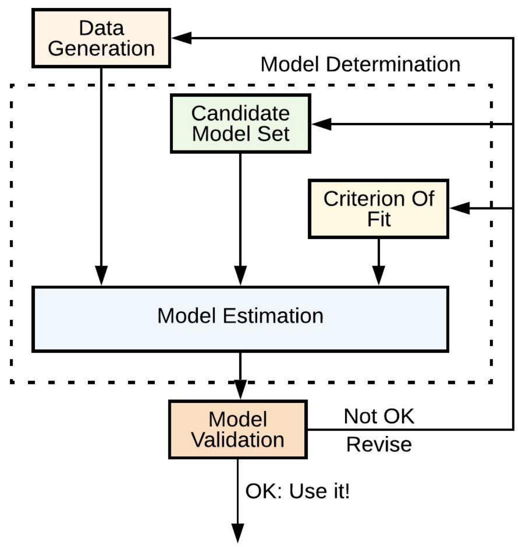

Figure 2.2: System Identification Loop

model parameter. After taking all parameter access the model and check the parameters of the system and compare the data with known response, errors in parameters estimation. After all this process system check the process if it satisfies then we use the system if it not satisfied then process go back previous step and check the parameter and perform the process again until the system not in use, Using this algorithm we can easily identify the system[24]

2.5 System Identification Using Matlab

System Identification Toolbox provides MATLAB functions, Simulink blocks, and an app for constructing mathematical models of dynamic systems from measured input-output data. It lets you create and use models of dynamic systems not easily modeled from first principles or specifications.

In this section, a plant model from input-output data, use the identified model to design a controller, and implement it. The workflow includes the following steps: acquiring data, identifying linear and nonlinear plant models[43], designing and simulating feedback controllers, and implementing these controllers on an embedded microprocessor for real-time testing.

2.5.1 DC Servo Motor Setup

The physical system is a DC motor connected to an Arduino Uno board via a motor driver. It is required to design a feedback controller for this motor to track a reference position. The controller will generate the appropriate voltage command based on the motor position reference data. When applied to the motor, this voltage will cause the motor to generate the torque that turns the motor shaft. We will use a potentiometer to measure the angle of rotation of the motor shaft, and feed this angle back to the controller.

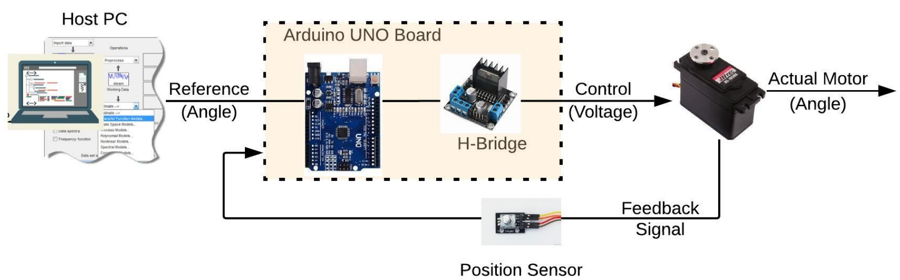

Figure 2.3: Control System Setup

The motor driver Dual H-bridge circuit increases the current capability and can drive the motor in both directions. We receive the motor position data through an Analog Input pin on the Arduino board and compute the error between the reference and actual data (the controller input). A voltage command (the controller output) to two Analog Output pins on the board as PWM signals. These signals are fed to the Dual H-bridge circuit that provides the motor with the appropriate drive currents.

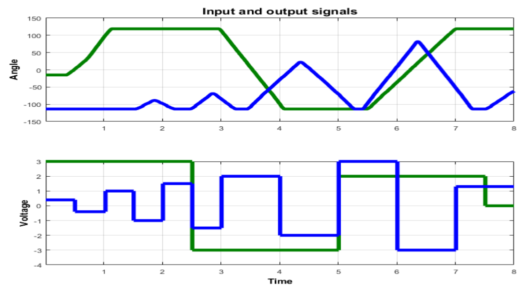

Figure 2.4: Input (Volatge) and Output (Angle)

Fig: 2.4 presents the input voltage to the motor and corresponding angle of the motor shaft.Matlab System identification toolbox utilise that data to extract the Linear and Non-linear Model, based on designer choice and further allows to estimate the parameters of a user-defined model and use the identified model for system response prediction and plant modeling in Simulink [40].

```
  From input "Voltage" to output "Angle":
                                1593 s - 676
  exp(-0.24*s) * ----------------------------------------------
                 s^4 + 8.69 s^3 + 45.34 s^2 + 16.04 s + 6.25e-05
Name: tf1
Continuous-time identified transfer function.

Parameterization:
   Number of poles: 4   Number of zeros: 1
   Number of free coefficients: 6
   Use "tfdata", "getpvec", "getcov" for parameters and their uncertainties.

Status:
Estimated using TFEST on time domain data "mergedEss".
Fit to estimation data: [50.52 75.75 -10.84 81.09]% (simulation focus)
FPE: 542.7, MSE: [601.2 432.9 850 379.1]
```

```
Nonlinear ARX model with 1 output and 1 input
 Inputs: Voltage
 Outputs: Angle
 Standard regressors corresponding to the orders
   na = 4, nb = 3, nk = 8
 Custom regressor:   max(min(Angle(t-1),114),-114)
 Nonlinear regressors:
   Angle(t-1)
   Angle(t-2)
   Angle(t-3)
   Voltage(t-8)
   Voltage(t-9)
   Voltage(t-10)
   max(min(Angle(t-1),114),-114)
 Nonlinearity: sigmoidnet with 10 units
Loss function: 381.5203
Sampling interval:  0.01
Estimated by PEM
```

Figure 2.5: Linear and Nonlinear ARX Model

2.5.2 Validation

Validation test is performed on both Linear and Non-linear model to satisfy that each model reacts to the input excitation, it can be seen in fig 2.6 that each model displays the accuracy percentage based on each input excitation.

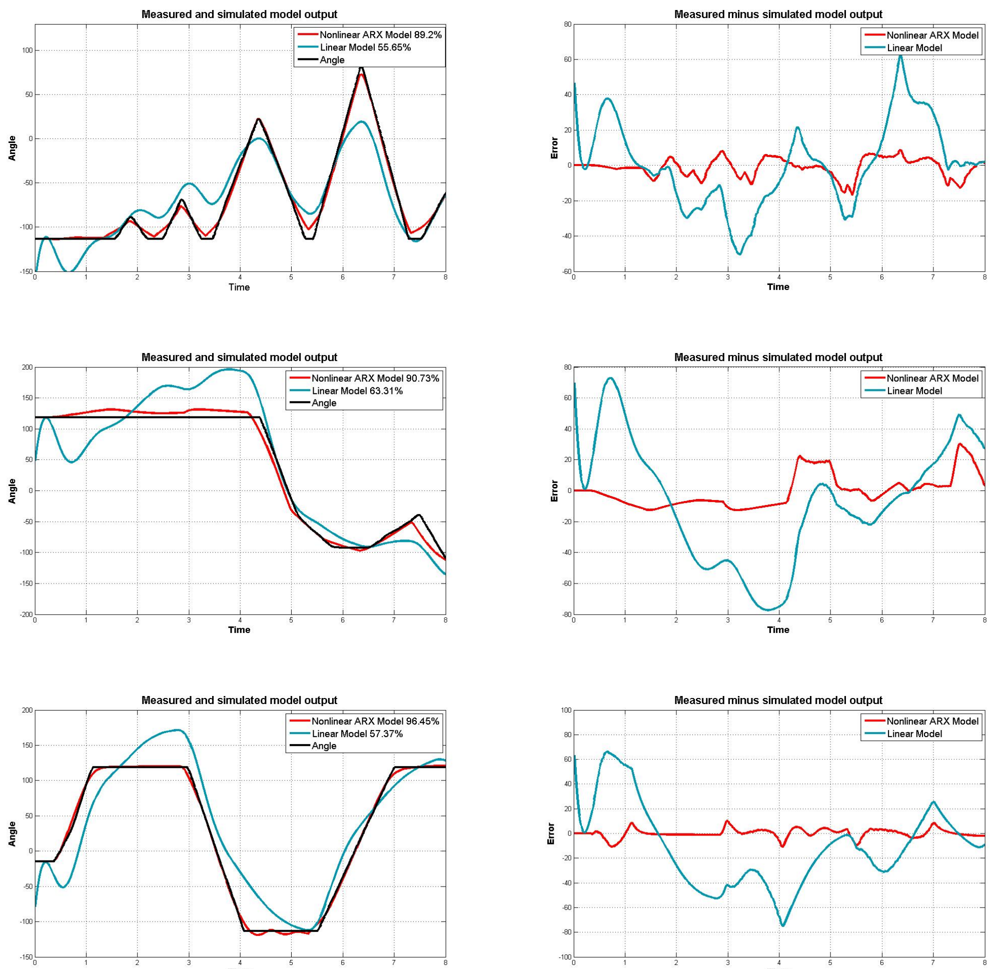

Figure 2.6: Linear and Nonlinear Model with Reference Angle

2.5.3 Final Response

Model results are further validated through the simulation of the Non-linear ARX Model with actual Motor to satisfy that model captured the system dynamics as evident from fig 2.7 and 2.8

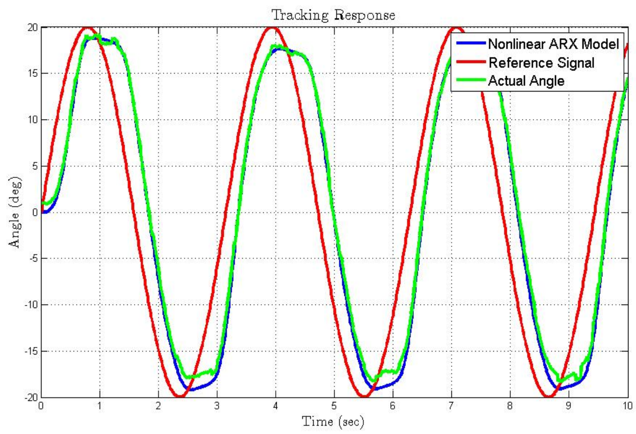

Figure 2.7: Nonlinear ARX Model with Actual Motor Angle

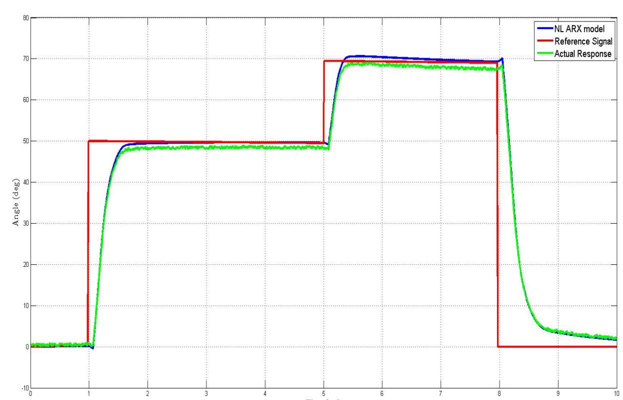

Figure 2.8: Nonlinear ARX Model with Actual Motor Angle

Chapter 3

Infinite-Horizon Regulation with State Dependent Riccati Equation (SDRE)

The optimal control of nonlinear systems is one of the most challenging and difficult topics in control theory. It is well known that the classical optimal control problems can be characterized in terms of Hamilton-Jacobi equations[4].The solution to the (HJE) gives the optimal performance value function and determines an optimal control[49][41][34].There exist many nonlinear control design techniques, each has benefits and weaknesses. Most of them are limited in range of applicability, and use of certain nonlinear control technique for a specic system usually demands choosing between different factors, e.g., performance, robustness, optimality, and cost. Some of the well-known nonlinear control techniques are feedback linearization, adaptive control, nonlinear predictive control, sliding mode control, and approximating sequence of Riccati Equations. One of the highly promising and rapidly emerging techniques for nonlinear optimal controllers designing is the State Dependent Riccati Equation (SDRE) technique also known Frozen time Riccati Equation (FTRE).

3.1 History of SDRE

The State Dependent Riccati Equation (SDRE) controller design is a relatively new practical approach for nonlinear control problems. This method was first proposed by Pearson[33] in 1962 and later research was carried and expanded by Werneli Cook [48]in 1975. The theoretical contribution in Cloutier, D'Souza & Mracek (1996) and Mracek & Cloutier (1998) has initiated an increasing use of SDRE technique in a wide variety of nonlinear control applications.These includes Nonlinear problems in aviation and aerospace using SDRE Nonlinear regulation [11][10]. Advanced guidance law development[12],skid-to-turn missile autopilot design[13] (Cloutier,2001),various benchmark problems [26][27] (Doyle, Mracek & Cloutier,1998),integrated guidance and control design [32] (Palumbo,1999), robotic manipulator with non-linearities[42] (Ruderman,2011), Under actuated robot[15] (EB Erdem, AG Alleyne,2001), feedback control of systems with parasitic effects [16] (Friedland,1997) and biomedical field for cancer treatment and HIV feedback control [19][6] as his approach provided an effective algorithm for synthesizing nonlinear optimal feedback control which is closely related to the classical linear quadratic regulator.

Due to its computational simplicity and its satisfactory simulation/experimental results, SDRE optimal control technique becomes an attractive control approach for a class of non linear systems. A wide variety of nonlinear control applications using the SDRE techniques are exposed in literature. Survey of State-Dependent Riccati Equation in nonlinear optimal feedback control [8] (Tayfun Cimen, 2012) briefly describes and classify the SDRE application in numerous fields like Missiles, Aircraft, Unmanned Aerial Vehicles, Satellites and Spacecraft, Ships, Autonomous Underwater Vehicles, Automotive Systems, Biological and Biomedical Systems, Process Control and Robotics.(S. N. Balakrishnan & Ali Heydari, 2013) produced a renowned path-planning problems for the approach and landing (A&L) phase of a reusable launch vehicle (RLV), such that the vehicle lands in a fixed and specified down range with the least possible vertical velocity and flight-path angle[17]. There are various contributors to develop and enhance SDRE technique in various application of sceeince but a handful of them are mentioned here.

3.2 Algebraic State Dependent Riccati Equation Overview

State-dependent Riccati equation (SDRE) techniques are rapidly emerging as general design and synthesis methods of nonlinear feedback controllers and estimators for a broad class of nonlinear regulator problems. In essence, the SDRE approach involves mimicking standard linear quadratic regulator (LQR) formulation for linear systems. In particular, the technique consists of using direct parameterization to bring the nonlinear system to a linear structure having state-dependent coefficient matrices. Theoretical advances have been made regarding the nonlinear regulator problem and the asymptotic stability properties of the system with full state feedback [6]

The SDRE method involves factorization of the nonlinear dynamics into product of a matrix-valued function (which depends on the states) and state vector. Thus, the SDRE algorithm captures the nonlinearities of the system, transfornonlinear system to a linear-like structure with state dependent coecient (SDC) matrices, and minimizing a nonquadratic performance index with a quadratic-like structure[25]. The Riccati equation using the SDC matrices is then solved online to give the sub-optimum control law. Moreover, with enough sample points, the suboptimal solution can be made to be very close to the optimal solution of the original nonlinear system. The coefficients of this Riccati equation vary with each point in state space. The algorithm thus involves solving, at a given point in state space, a SDRE whose pointwise stabilizing solution during state evolution yields the SDRE nonlinear feedback control law.

As the SDRE depends only on the current state, the computation can be carried out online, in which case the SDRE is dened along the state trajectory. In addition, a primary advantage offered by SDRE to the control designer is the opportunity to make trade offs between control effort and state errors by tuning the SDC.

The process of factorizing a nonlinear system into a linear-like structure that contains SDC matrices is called extended linearization. Its well known that for single- variable (scalar) systems, the SDC parameterization is unique, On the other hand, in the multi variable case, the SDC parameterization is not unique. In fact, there are an infinite number of ways to bring a nonlinear system to SDC form. For example, consider the two-dimensional nonlinear system in the form

For example, consider the two-dimensional nonlinear system in the form

$$x'(t) = f(x) + B(x)u(t), \tag{3.1}$$

where $f(x) = [x_2, x_1^2]'$ we can write this system in SDC form as

$$x'(t) = A(x)x(t) + B(x)u(t), \tag{3.2}$$

with the obvious SDC parameterization $A_1(x) = \begin{bmatrix} 0 & 1 \\ x_1^2 & 1 \end{bmatrix}$. We can evaluate other SDC parameterizations:

$$A(x) = A_2(x) = \begin{bmatrix} -x^2 & 1 + x_1 \\ x_1^2 & 0 \end{bmatrix}, \text{ or } A(x) = A_3(x) = \begin{bmatrix} \frac{x_2}{x_1} & 0 \\ x_1^2 & 0 \end{bmatrix}$$

To choose the one of the correct parameterizations for $A(x)$ and $B(x)$, one should consider that the matrices $A(x)$ and $B(x)$ must be chosen in such a way that the nonlinear system is controllable or at least stabilizeable. It is well known that the solution of the SDRE cannot be found analytically, except for very limited nonlinear systems. It is given that Taylor series and interpolation methods can be used to approximate the offline solution of the SDRE. However, it is still hard to solve the SDRE with these methods when the dynamics of the nonlinear system become very complex or of high-order.

3.3 Infinite Horizon Regulation for Non-Linear Systems

An infinite time horizon, full state feedback with following dynamics can be represented as follows:

$$\dot{x}(t) = f(x) + B(x)u(t), \tag{3.3}$$

with state vector $x(t) \in \mathbb{R}^n$, input vector $u(t) \in \mathbb{R}^m$.Such that $f(x) : \mathbb{R}^n \to \mathbb{R}^n$, $B(x) : \mathbb{R}^n \to \mathbb{R}^{n*m}$,with $B(x) \neq 0 \vee x(t)$.In this context, the minimization of an infinite-time horizon performance criterion is given as

$$J(x, u) = \frac{1}{2} \int_0^\infty [x'(t)Q(x)x(t) + u'(t)R(x)u(t)]dt, \tag{3.4}$$

The state and input weighting matrices (design parameters) are assumed to be state dependent, such that $Q(x)$ is a symmetric *positive semi-definite* matrix, and $R(x)$ is a symmetric *positive definite* matrix. Moreover, $x'(t)Q(x)x(t)$ is a measure of state accuracy and $u'(t)R(x)u(t)$ is a measure of control effort [9].
Further using SDC parameterization, Nonlinear system dynamics of equation (3.3) can be converted into a linear like structure which contains SDC matrices. Under the assumption $f(0) = 0$, a continuous nonlinear matrix-valued function $A(x)$ exist such that

$$f(t) = A(x)x(t), \tag{3.5}$$

where $A(x) \in \mathbb{R}^{\ltimes * \ltimes}$, is found by mathematical factorization and is non-unique when $n > 1$. Therefore, the nonlinear system (3.3) can now be represented in the SDC form

$$\dot{x}(t) = A(x)x(t) + B(x)u(t), \qquad x(0) = x_o, \tag{3.6}$$

which has a linear structure with SDC matrices $A(x)$ and $B(x)$

3.4 Algebraic SDRE Regulation Controller Structure

The SDRE procedure uses *extended linearization* as the basic design concept in formulating the nonlinear optimal control problem (3.3) and (3.4).The basic linear control synthesis method in this case is the *LQR* method. Motivated by the LQR problem, which is charachterized by an *Algebric Riccati Equation* (ARE), the algebric SDRE feedback control is an extended linearization control method that provides a similar approach to the nonlinear regulation problem for matheiw equation dynamics (3.3) with the cost function (3.4). The controller design must meet the following criterion.
Condition 1: $f(x)$ is a continuously differential vector-valued function of $x(t)$.
Condition 2 The origin $x = 0$ is an equilibrium point of the system with $u = 0$.
Condition 3: **Q(x)**) is a symmetric positive semi-definite matrix, and $R(x)$ is a symmetric positive definite matrix.
Condition 4: $A(x)$ and $B(x)$ must be chosen in such a way that the nonlinear system is controllable or at least stabilizable.

Utilising the LQR formulation, the optimal state-feedback controller is obtained in the form

$$u(x) = -K(x)x(t), \tag{3.7}$$

Such that the nonlinear state-feedback gain $K(x)$ for minimizing (3.4) becomes

$$K(x) = R^{-1}(x)B'(x)P(x), \tag{3.8}$$

and hence the state-feedback controller can be written as

$$u(x) = -R^{-1}(x)B'(x)P(x)x(t) \tag{3.9}$$

Here, $P(x)$ is the unique, symmetric, positive-difinite solution of the continuous time algebric SDRE

$$P(x)A(x) + A'(x)P(x) - P(x)B(x)R^{-1}(x)B'(x)P(x) + Q(x) = 0, \tag{3.10}$$

The resulting algebric SDRE-controlled trajectory is the solution of the closed loop dynamics

$$\dot{x}(t) = \left[A(x) - B(x)R^{-1}(x)B'(x)P(x)\right]x(t). \tag{3.11}$$

The algebraic SDRE, strictly speaking it could be called State Dependent Algebraic Riccati Equation (SDARE), solution to the infinite-time horizon autonomous nonlinear regulator problem (3.3) and (3.4) is a true generalization of the infinite-time horizon *time-invarient* LQR problem, where all the coefficient matrices are *state-dependent*. At each instant, the method treats the SDC matrices as being constant,and computes a control action by solving an LQ optimal control problem. The main advantage of the algebraic SDRE algorithm is its simplicity and its obvious eciency, since there is no attempt to solve the Hamilton Jacobi Bellman (HJB) equation.Thats applicable for nonlinear optimal control problems. When the coecient and weighting matrices are constant, the nonlinear regulator problem converts to the LQR problem and the algebraic SDRE control method converts to the steady-state linear regulator.

Fig.3.1 shows a ow chart of the infinite-horizon algebraic SDRE regulator. At each sample time, the following procedure is accomplished. First, the current state vector x(t) is used to calculate numerical values for A(x) and B(x). Then, using the LQR equations,

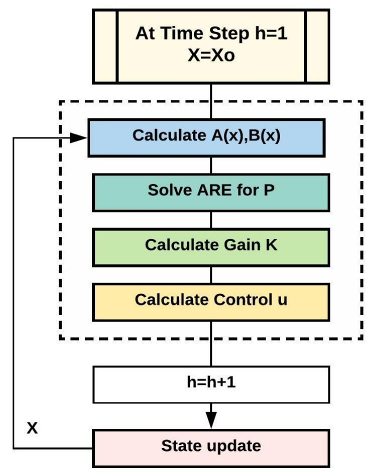

Figure 3.1: Flow Chart of Infinite-Horizon Algebraic SDRE Regulator

P(x) and K(x) are calculated. Control input u(x) is then calculated and applied to the system to calculate the state of the next step of time. This LQR procedure is then repeated at the next sample time. For the algebraic SDRE technique, the ARE is solved at every sample time for each new value of A(x) and B(x) at each step of time. This procedure views the nonlinear system to be approximated as a series of linear systems. Therefore, shorter time increments increase the accuracy of the control law, because the change in nonlinear dynamics over shorter time increments is more like a linear change. Because of its approximating nature,the algebraic SDRE technique is considered a sub optimal solution. However, with the proper choices for the A(x) and B(x) matrices, and with the proper amount of sample times, the algebraic SDRE technique can provide a very adequate optimal solution.

3.4.1 Mathieu Equation

Consider the stabilizaton of Mathieu equation

$$\ddot{q}(t) + (\alpha + \beta cos(wt))q(t) = bu(t). \tag{3.12}$$

The open loop system is described by

$$x(t) = \begin{bmatrix} q(t) \\ \dot{q}(t) \end{bmatrix}, \quad A(t) = \begin{bmatrix} 0 & 1 \\ -(\alpha + \beta cos(wt) & 0 \end{bmatrix}, \quad B(t) = \begin{bmatrix} 0 \\ b \end{bmatrix}, \tag{3.13}$$

Let $q(0) = q_o$ and $\dot{q}(0) = \dot{q}_o$ denote the initial conditions. [3;0],[-5;1].The task is to bring the states to the equilibrium [0;0]. Parameter values are described in the table as.

Variable	**Value**
α	1
β	1
b	1
w	1

Table 3.1: Mathieu Equation Parameters

and the weighted matrices are

$$R = \begin{bmatrix} 0.001 - 1 \end{bmatrix}, \quad Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \tag{3.14}$$

The simulations are performed for a time interval of 5 sec with 500 unit time steps.The resulting state trajectories and Optimal control are shown in in fig(3.2) and fig(3.3). This clearly illustrates the infinite-horizon algebraic SDRE nonlinear regulator algorithm.

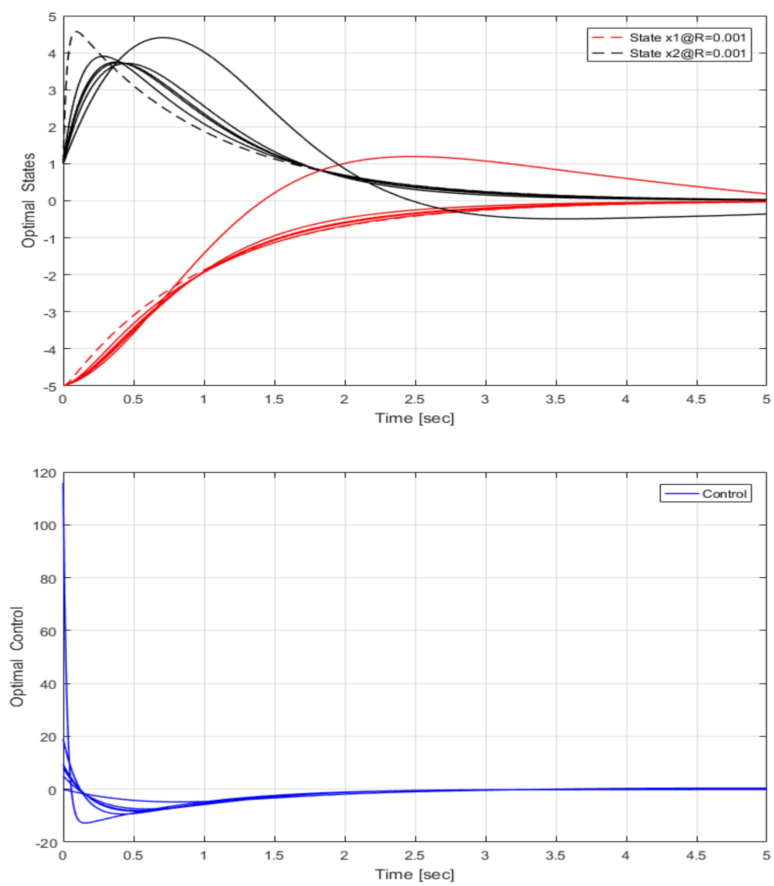

Figure 3.2: Optimal State and Control with initial conditions [-5;1]

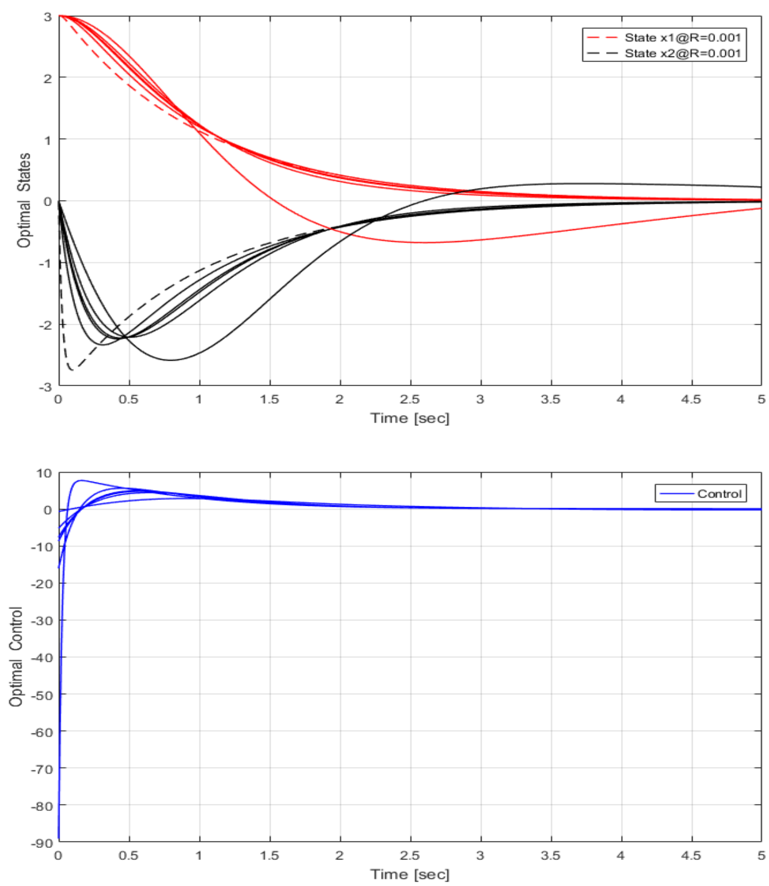

Figure 3.3: Optimal State and Control with initial conditions [3;0]

Chapter 4

Infinite-Horizon Tracking with State Dependent Riccati Equation

4.1 SDRE Tracking For Nonlinear System

For Nonlinear tracking problem consider a continuous time, state feedback and infinite-time horizon posses the following dynamics

$$\dot{x}(t) = f(x) + B(x)u(t) \tag{4.1}$$

$$y(t) = h(x) \tag{4.2}$$

with state vector $x(t) \in \mathbb{R}^n$, input vector $u(t) \in \mathbb{R}^m$, and the output $y(t) \in \mathbb{R}^p$.Such that $f(x) : \mathbb{R}^n \to \mathbb{R}^n$, $B(x) : \mathbb{R}^n \to \mathbb{R}^{n*m}$, and $h(x) : \mathbb{R}^n \to \mathbb{R}^{p*n}$
Nonlinear system can be expressed in a state-dependent like linear form, as:

$$\dot{x}(t) = A(x)x(t) + B(x)u(t), \tag{4.3}$$

$$y(t) = C(x)x(t) \tag{4.4}$$

where $f(x) = A(x)x(t)$, $h(x) = C(x)x(t)$. $A(x) \in \mathbb{R}^{\ltimes*\ltimes}$, $B(x) \in \mathbb{R}^{\ltimes*\gtrdot}$, and $C(x) \in \mathbb{R}^{\shortmid*\ltimes}$

While $Z(t)$ is the desired output. The prime task of the control system output $y(t)$ follows, as close as possible, the command $z(t)$. This objective can be accomplished by using minimizing the cost function.

$$J(x,u) = \frac{1}{2}\int_0^\infty [e'(t)Q(x)e(t) + u'(x)R(x)u(x)]dt, \tag{4.5}$$

Where $e(t) = z(t) - y(t)$,$Q(x)$ is a symmetric *positive semi-definite* matrix, and $R(x)$ is a symmetric *positive definite* matrix.$e'(t)Q(x)e(t)$ is a measure of tracking error and accuracy and $u'(x)R(x)u(x)$ is a measure of control effort [9]

4.2 Algebraic SDRE Tracking Controller Structure

The algebraic SDRE, strictly speaking it could be called State Dependent Algebraic Riccati Equation (SDARE), control theory has only been developed for the infinite time non-linear optimal regulation (stabilization) problem, for which $z(t) = 0$ and $C(x) = I$ nxn. This is because the method requires solving the infinite-time algebraic Riccati equation. Unfortunately, the developed theory of the infinite-time LQ optimal tracking problem has hindered its application for solving non-linear trajectory tracking problems, unless an integral servomechanism is used [25], which increases the number of states and thus the computation time required for solving algebraic Riccati equations. Regardless of the developed theory of infinite-time LQ optimal tracking control, a good approximation can be developed for excessively large terminal time[18]. The derived results are approximate in nature and are valid for very large values of the terminal time.
Condition 1: $f(x)$ is a continuously differentiable vector-valued function of $x(t)$.
Condition 2 The origin $x = 0$ is an equilibrium point of the system with $u = 0$.
Condition 3: **Q(x)**) is a symmetric positive semi-denite matrix, and $R(x)$ is a symmetric positive denite matrix.
Condition 4: $A(x)$ and $B(x)$ must be chosen in such a way that the nonlinear system is controllable or at least stabilizable. $A(x)$ and $C(x)$ must be chosen in such away that the nonlinear system is observable or at least detectable.

By following the LQR formulation, the state-feedback controller is obtained in the form

$$u(x) = -R^{-1}(x)B'(x)[P(x)x(t) - g(x)]. \tag{4.6}$$

P(x) is a positive-denite solution of the continuous-time algebraic SDRE

$$P(x)A(x) + A'(x)P(x) - P(x)B(x)R^{-1}(x)B'(x)P(x) + C'(x)Q(x)C(x) = 0, \tag{4.7}$$

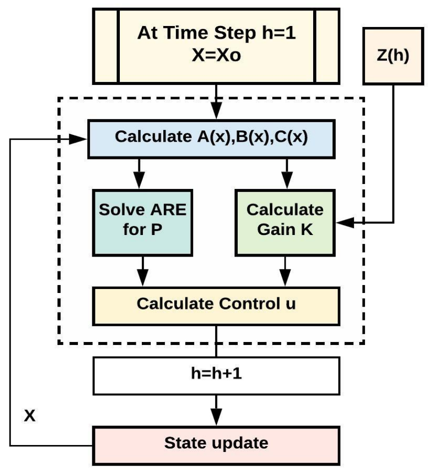

Figure 4.1: Flow Chart of Infinite-Horizon Algebraic SDRE Tracking

and $g(x)$ is a solution of the continuous-time state dependent non-homogeneous equation

$$g(x) = -\big([A(x) - B(x)R^{-1}(x)B'(x)P(x)]'\big)^{-1}C'(x)Q(x)z(x). \tag{4.8}$$

The resulting algebraic SDRE-derived trajectory is the solution of the closed-loop dynamics

$$\dot{x}(t) = [A(x) - B(x)R^{-1}(x)B'(x)P(x)]x(t) + B(x)R^{-1}(x)B'(x)g(x). \tag{4.9}$$

Fig.2.4 shows a ow chart of the innite-horizon algebraic SDRE tracking. At each sample time, the following procedure is accomplished. First, the current state vector $x(t)$ is used to calculate numerical values for $A(x)$, $B(x)$, and $C(x)$. Then, using the LQT equations, $P(x)$ and $g(x)$ are calculated. Control input $u(x)$ is then calculated and applied to the system. This procedure is then repeated at the next sample time.

4.3 Infinite-Horizon Algebraic SDRE Tracking Simulation

This section presents the non-linear state space models and simulations with the infinite-time horizon optimal tracking controller.

4.3.1 Forced Damped Pendulum

The dynamic equation for forced damped pendulum is:

$$ml^2\ddot{\theta} = mglsin(\theta) - k\dot{\theta} + T, \tag{4.10}$$

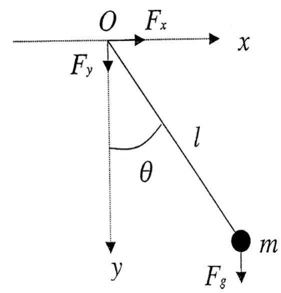

Figure 4.2: Force Damped Pendulum

where, θ is the angle of pendulum, l is the length of rod, m is the mass of pendulum, g is the gravitational constant, k is the damping (friction) constant, T is the driving torque. The system nonlinear state equations can be written in the form:

$$\dot{x_1} = x_2, \tag{4.11}$$

$$\dot{x_2} = -\frac{g}{l}sin(x_1) - \frac{k}{ml^2}x_2 + \frac{1}{ml^2}u, \tag{4.12}$$

$$y = x_1, \tag{4.13}$$

where; $\theta = x_1$,$\dot{\theta} = x_2$, and $T = u$ or in state dependent form:

$$\begin{bmatrix} \dot{x_1} \\ \dot{x_2} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{-5sin(x_1)}{x_1} & -0.25 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0.25 \end{bmatrix} u, \tag{4.14}$$

$$y(t) = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \tag{4.15}$$

where $A = \begin{bmatrix} 0 & 1 \\ -5sin\frac{(x_1)}{x_1} & -0.25 \end{bmatrix}$,$B = \begin{bmatrix} 0 \\ -0.25 \end{bmatrix}$,$C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$.
The weighted matrices are

$$Q = diag(100, 0), \qquad R = 0.01...R = 0.1, \tag{4.16}$$

and, let the reference output as

$$z(t) = \sin(t) + \cos(t) \tag{4.17}$$

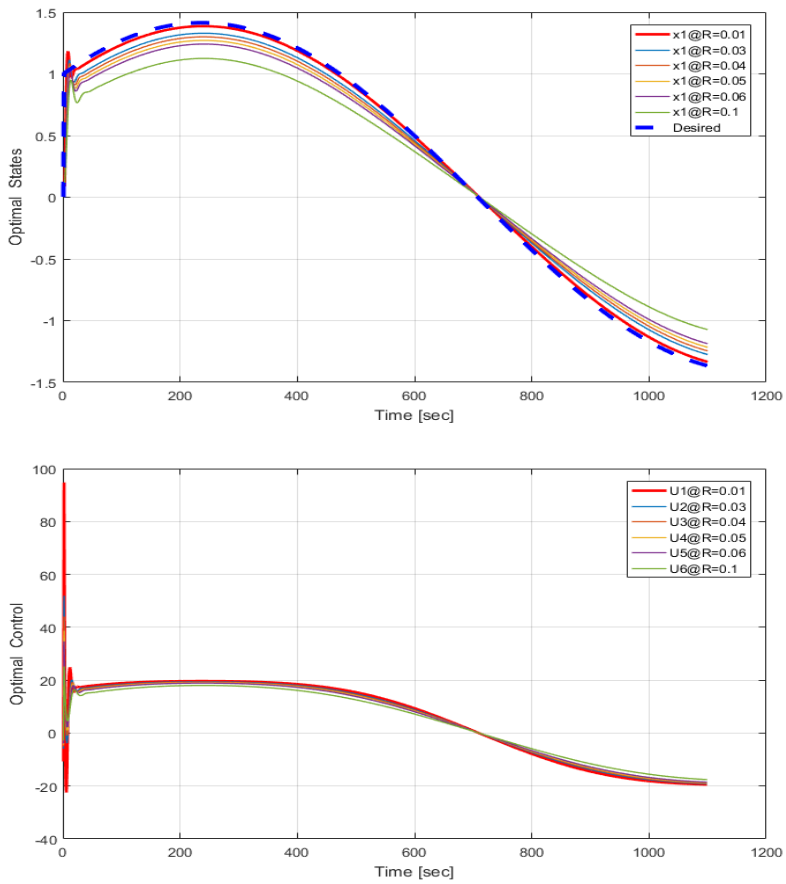

Figure 4.3: Optimal State and Control

The simulations are performed for 1100 time steps for different input weight matrix R and the resulting output trajectory and the optimal control is shown in Fig. 3.3. In Fig. 3.3,the solid line denotes the actual output trajectory and the doted line denotes the reference output trajectory. Comparing trajectories in Fig. 3.3, its clear that the infinite-horizon algebraic SDRE nonlinear tracking algorithm gives very good results as the actual optimal output is making very good tracking to the reference output.

4.3.2 Cart-Pole System

The cart is a one dimensional horizontally moving base; the mass of the pendulum is evenly distributed. The following variables are specified:

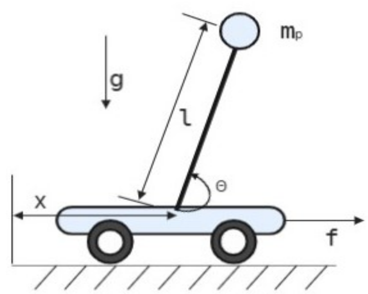

Figure 4.4: Cart-pole system

Variable	Definition
M	Mass of the cart (kg)
L	Length of the pendulums centre of mass (m)
F	Force applied to cart (N)
m	Mass of the pendulum (kg)
θ	Cart position
x	Cart horizontal displacement from origin $(kg * m^2)$

Table 4.1: Variable definitions

The state space equations of the inverted pendulum can be derived from the Lagrange-equation [39]

$$\frac{d}{dt}\left[\frac{\partial}{\partial \dot{q}}L\right] - \frac{\partial}{\partial \dot{q}}L = \tau \tag{4.18}$$

where L denotes the Lagrange-function $L = KV$ with K and V being the kinetic and potential energy, respectively. For the inverted pendulum, the kinetic energy is given by

$$K = \frac{1}{2}M\dot{x}^2 + \frac{1}{2}m\Big(\big(\dot{x} + l\dot{\theta}\cos\theta\big)^2 + \big(l\dot{\theta}\sin\theta\big)^2\Big) \tag{4.19}$$

While the $potentialenergy$ is given by

$$V = mgl\cos\theta \tag{4.20}$$

This gives the equations

$$(M+m)\ddot{x} + ml\ddot{\theta}\cos\theta - ml\dot{\theta}^2\sin\theta = F \tag{4.21}$$

$$l\ddot{\theta} + \ddot{x}\cos\theta - g\sin\theta = -f_\theta\dot{\theta} \tag{4.22}$$

Where $f_\theta\dot{\theta}$ describes the friction in the rotational link of the pendulum.
By selecting the state variable as

$$x_1 = x,\ x_2 = \dot{x},\ x_3 = \theta, \text{ and } x_4 = \theta$$

we get the equations

$$\dot{x_1} = x_2, \tag{4.23}$$

$$\dot{x_2} = \frac{ml}{M+m}\left(x_4^2 sinx_3 - \dot{x_4}cosx_3 + \frac{F}{M+m}\right) \tag{4.24}$$

$$\dot{x_3} = x_4 \tag{4.25}$$

$$\dot{x_4} = \frac{1}{l}\left(- \dot{x_2}cosx_3 - f_\theta \dot{x_4}cosx_3\right) \tag{4.26}$$

By eliminating this dependence, we finall get the non-linear equations

$$\dot{x_1} = x_2 \tag{4.27}$$

$$\dot{x_2} = \frac{-mg\sin x_3\cos x_3 + mlx_4^2\sin x_3 + f_\theta m x_4\cos x_3 + F}{M + (1-\cos^2 x_3)m} \tag{4.28}$$

$$\dot{x_3} = x_4 \tag{4.29}$$

$$\dot{x_4} = \frac{(M+m)(g\sin x_3 - f_\theta x_4) - (lmx_4^2\sin x_3 + F)\cos x_3}{l(M + (1+\cos^2 x_3)m)} \tag{4.30}$$

If both the pendulum angle θ and the cart position x are the variables of interest, then the output equation may be written as

$$y = \begin{bmatrix}\theta \\ x\end{bmatrix} = Cx\begin{bmatrix}1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0\end{bmatrix} \tag{4.31}$$

The weighted matrices, and desird trajectory are

$$Q = diag(200, 1, 500, 250), R = 1 \tag{4.32}$$

$$z(t) = 0, \tag{4.33}$$

Setting initial state of $\theta_o = 5$
This concludes that if the initial condition θ_o is set to a certain degree, the cart is going to move to bring the pendulum in its equilibrium position

Variable	**Value**
M	$2.4(kg)$
l	$0.36(m)$
L	$\pm 0.5(m)$
m	$0.23(kg)$

Table 4.2: Cart Pole System Parameters

4.4 Conclusion

The Algebraic State Dependent Riccati Equation (SDRE) provide an extremely effective algorithm for nonlinear feedback control design by allowing non-linearities in the system states while additionally offering great design flexibility through design matrices. The algebraic SDRE method involves factorization of the nonlinear dynamics into product of State Dependent Coefficient (SDC) matrices and state vector. Thus,the algebraic SDRE algorithm captures the non-linearities of the system, transforming the original nonlinear system to a linear-like structure with SDC matrices, and minimizing a non-quadratic performance index with a quadratic-like structure. The Algebraic Riccati Equation (ARE) using the SDC matrices is then solved online to give the sub optimum control law, and with enough time sample points, the sub optimal solution can be made to be very close to optimal solution. As the algebraic SDRE depends only on the current state, the computation can be carried out online,in which case the algebraic SDRE is defined along the state trajectory. In addition, the main advantage of the algebraic SDRE is the flexibility to make trade offs between control effort and state errors by tuning the SDC.

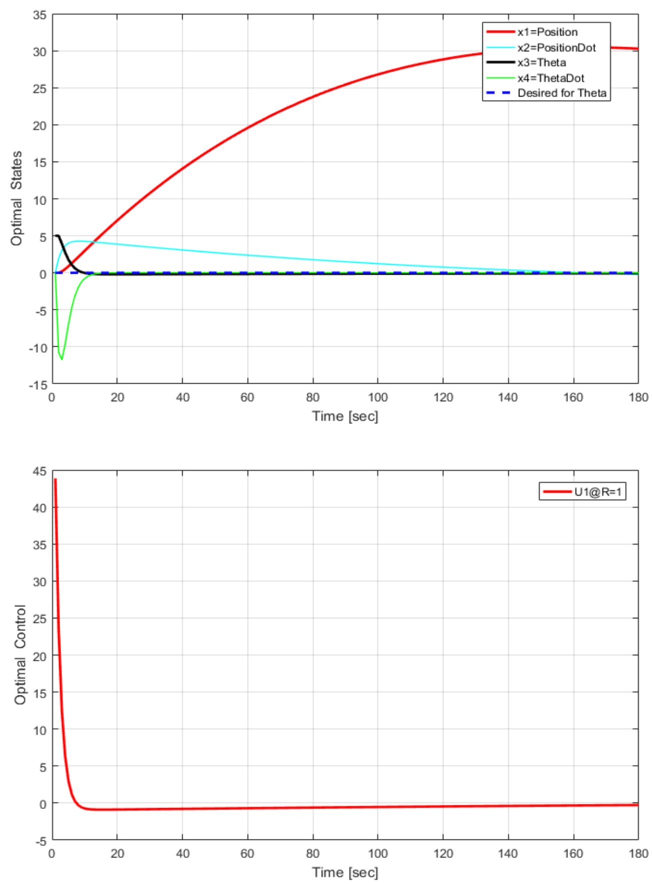

Figure 4.5: Optimal State,trajectory and Control Input

Chapter 5

Finite-Horizon Regulation with Differential State Dependent Riccati Equation (DSDRE)

In infinite-horizon optimal nonlinear control problem, the differential Riccati equation(DRE) is converted to an algebraic Riccati equation (ARE) which is easy to be solved. Finite-horizon optimal control of nonlinear systems is a challenging problem in the control field due to the complexity of time-dependency of the Hamilton Jacobi Bellman (HJB) differential equation. Also, in the finite-horizon SDRE problems, the solution of the SDRE is time dependent and a differential equation, rather than an algebraic equation in infinite-horizon SDRE.

5.1 Lyapunov Equation Approach for Differential SDRE

This section presents the (SDDRE) State Dependent Differential Riccati Equation technique for *finite horizon* optimal control of nonlinear system based on a change of variable,that converts the differential Riccati equation (DRE) to linear differential Lyaunov equation(DLE)[30]and evaluating the coefficients of the resulted equation based on the current state values at each time step and freezing these coefficients from current time to the next step. Then, the Lyapunov equation is solved in a closed form at each step during online implementation. The use of Lyapunov-type equations in solving optimal problems is given in .Because the solution is based on the differential Lyapunov equation, the new method is termed as Lyapunov equation approach.

5.2 Theorems

In this section, the relation between the proposed technique and the exact optimal solution to the finite-horizon problem will be discussed. This is done through the following Theorems

5.2.1 Theorem 1

The solution to the optimal control of the nonlinear finite-horizon problem can be approximated to an optimal solution given through finite-horizon differential SDRE. Given the nonlinear system in the form

$$\dot{x}(t) = f(x) + g(x)u(t), \tag{5.1}$$

which is expressed in a state-dependent like linear form as:

$$\dot{x}(t) = A(x)x(t) + B(x)u(t), \tag{5.2}$$

where $f(x) = A(x)x(t)$, and B(x)=g(x).

The solution to the optimal control of nonlinear system (4.2) subject to the cost function.

$$J(x,u) = \frac{1}{2}x'(t_f)Fx(t_f) + \frac{1}{2}\int_{t_0}^{t_f}[x'(t)Qx(t) + u'(t)Ru(t)]dt, \tag{5.3}$$

where $Q(x)$ and F are symmetric *positive semi-definite* matrices, and $R(x)$ is a symmetric *positive definite* matrix,is given by

$$u(x,t) = -R^{-1}(x)B'(x)[P(x,t)x(t) + \Pi], \tag{5.4}$$

where

$$\Pi = \frac{1}{2}[x'P_{x_1}x..x'P_{x_2}x...x'P_{x_n}x]', P_{xi} = \frac{\partial P(x,t)}{\partial x_i}. \tag{5.5}$$

$P(x,t)$ is a symmetric positive denite solution to the equation

$$-\dot{P}(x,t) = P(x,t)A(x) + A'(x)P(x,t) - P(x,t)B(x)R^{-1}(x)B'(x)P(x,t) + Q(x) + \Omega, \tag{5.6}$$

with the final condition $P(x, t_(f)) = F$, where

$$\Omega = \frac{1}{4} \sum_{i=1}^{n} \sum_{j=1}^{n} P x_i x [B(x) R^{-1}(x) B'(x)]_{ij} x' P_{x_j}, \tag{5.7}$$

and $[]_{ij}$ is the ith element of the jth row of that matrix: The desired finite-horizon optimal control is given by the partial dierential HJB equation

$$-J_t(x,t) = J_x^{'}(x,t) A(x) x + \frac{1}{2} x' Q(x) x - \frac{1}{2} J_x^{'}(x,t) B(x) R^{-1}(x) B'(x) J_x(x,t). \tag{5.8}$$

with the final condition $J(x, tf) = \frac{1}{2} x'(t) F x(t)$, where $J(x,t)$ represents the optimal cost function and subscript t and x denote the corresponding partial derivative J_t and J_x,respectively, of J, the optimal control is given by

$$u(x,t) = -R_{-1}(x) B'(x) J_x(x,t). \tag{5.9}$$

$J(x,t)$ is a positive definitive matrix can be written in the form

$$J(x,t) = \frac{1}{2} x'(t) P(x,t) x(t), \tag{5.10}$$

which leads to

$$J_t(x,t) = \frac{1}{2} x' P_t(x,t) x, J_x(x,t) = P(x,t) x + \Pi \tag{5.11}$$

Substituting (4.11)in (4.8)

$$\begin{aligned} -\frac{1}{2} x' P_t(x,t) x = (P(x,t)x + \Pi)' A(x) x + \frac{1}{2} x' Q(x) x \\ - \frac{1}{2} (P(x,t)x + \Pi)' B(x) R_{-1} B'(x) (P(x,t)x + \Pi), \end{aligned}$$

which is rearranged in the form

$$\begin{aligned} -\frac{1}{2} x' P_t(x,t) x = x'[P(x,t) A(x) + \frac{1}{2} Q(x) - \frac{1}{2} (P(x,t) B(x) R_{-1} B'(x) (P(x,t)] x \\ + \Pi' [A(x) x - B(x) R^{-1} B'(x) (P(x,t) x + \frac{1}{2} \Pi)] \end{aligned} \tag{5.12}$$

substituting (4.4) in (4.2)

$$\dot{x}(t) = A(x) x - B(x) R^{-1} B'(x) [P(x,t) x + \Pi], \tag{5.13}$$

substituting (4.14) in (4.13)

$$-\frac{1}{2}x'P_t(x,t)x = x'[P(x,t)A(x)+\frac{1}{2}Q(x)-\frac{1}{2}(P(x,t)B(x)R_{-1}B'(x)(P(x,t)]x+\Pi'\dot{x}+\frac{1}{2}\Pi'B(x)R^{-1}B'(x)\Pi. \tag{5.14}$$

Multiplying the transpose of (4.5) by $\dot{x}(t)$

$$\Pi'\dot{x} = \frac{1}{2}\sum_{i=1}^{n}(x'P_{x_i}x)\dot{x}_i = \frac{1}{2}x'(\sum_{i=1}^{n}(x'P_{x_i}\dot{x}_i)x. \tag{5.15}$$

Form (4.15) and (4.16)

$$-\frac{1}{2}x'(P_t+\sum_{i=1}^{n}P_{x_i}\dot{x}_i)x = x'[PA(x)+\frac{1}{2}Q-\frac{1}{2}PB(x)R^{-1}B'(x)P]x+\frac{1}{2}\Pi'B(x)R^{-1}B'(x)\Pi.$$

Calculating the total derivative of P as

$$\dot{P} = P_t + \sum_{i=1}^{n}P_{x_i}\dot{x}_i, \tag{5.16}$$

and since

$$\Pi'B(x)R_{-1}B'(x)\Pi = \frac{1}{4}x'(\sum_{i=1}^{n}\sum_{j=1}^{n})P_{x_i}x[B(x)R^{-1}B'(x)]_{ij}x'P_{x_j})x \tag{5.17}$$

substituting from (4.18) and (4.19) in (4.17)

$$\frac{1}{2}x'\dot{P}(x,t)x = x'(P(x,t)A(x)+\frac{1}{2}Q(x)-\frac{1}{2}P(x,t)B(x)R^{-1}B'(x)P(x,t)+\frac{1}{2}\omega)x. \tag{5.18}$$

It follows that (4.20) should hold good for any value of x . This clearly means that the function $P(x,t)$ should satisfy the matrix dierential Riccati equation

$$-\dot{P}(x,t) = P(x,t)A(x)+A'(x)P(x,t)+Q-P(x,t)B(x)R^{-1}B'(x)P(x,t)+\omega \tag{5.19}$$

with the final condition

$$P(c,t_f) = F \tag{5.20}$$

This proves that solving (4.6) solves the HJB equation (4.8) and gives the optimal solution to the nonlinear nite-horizon problem.

Performing some approximations by neglecting termsω in (4.6) and Π in (4.4), which leads to the optimal control given by

$$u(x,t) = -R^{-1}(x)B'(x)P(x,t)x(t), \tag{5.21}$$

resulted from solving the differential SDRE

$$-\dot{P}(x,t) = P(x,t)A(x) + A'(x)P(x,t) - P(x,t)B(x)R^{-1}(x)B'(x)P(x,t) + Q(x). \quad (5.22)$$

Using this approximation, the control will be approximated to be optimal control.

5.2.2 Theorem 2

The approximate optimal control given through finite-horizon differential SDRE in (4.23)resulted from solving the DRE (4.24) for the positive definitive matrix $P(x,t)$, makes the nonlinear system (4.2) a globally stable system.
Selecting the Lyapunov function

$$-V(x,t) = x'P(x,t)x, \quad (5.23)$$

where $P(x,t)$ is the symmetric positive definitive matrix, and taking the total derivative of $V(x,t)$ leads to

$$-\dot{V}(x,t) = \dot{x}P'(x.t)x + x'\dot{P}(x,t)x + x'P(x,t)\dot{x}. \quad (5.24)$$

substituting from (4.23) in (4.2)

$$\dot{x}(t) = A(x)x - B(x)R_{-1}B'(x)P(x,t)x. \quad (5.25)$$

Substituting from (4.24) and (4.27) in (4.26)

$$-\dot{V}(x,t) = \dot{x}'[-P(x,t)B(x)R^{-1}(x)B'(x)P(x,t) - Q]x. \quad (5.26)$$

Since $Q(x)$ is a symmetric positive *semi-definitive* matrix, and $P(x,t)$ and $R(x)$ are symmetric *postive definitive* matrices, it's clear that $V(x,t)$ is a *positive definitive* matrix and the total derivative $\dot{V}(x,t)$ is a *negative definitive* matrix. Hence, the finite-horizon differential SDRE method is globally stable.

5.3 Finite Horizon Regulator for Non-linear Systems

5.3.1 Problem Formulation

The *nonlinear* system considered in this chapter is presented in the following form:

$$\dot{x}(t) = f(x) + g(x)u(t), \tag{5.27}$$

$$y(t) = h(x). \tag{5.28}$$

This nonlinear system can be expressed in a state-dependent linear-like form

$$\dot{x}(t) = A(x)x(t) + B(x)u(t), \tag{5.29}$$

$$y(t) = C(x)x(t). \tag{5.30}$$

where $f(x) = A(x)x(t)$, $B(x) = g(x)$, and $h(x) = C(x)x(t)$.

The goal is to find a state feedback optimal control law of the form $u(x) = -Kx(t)$, that minimizes a cost function given by...

$$J(x,u) = \frac{1}{2}x'(t_f)Fx(t_f) + \frac{1}{2}\int_{t_0}^{t_f} [x'(t)Q(x)x(t) + u'(x)R(x)u(x)]dt \tag{5.31}$$

where $Q(x)$ and F are symmetric *positive semi-definite* matrices, and $R(x)$ is a symmetric *positive definite* matrix. Moreover, $x'Q(x)x$ is a measure of state accuracy and $u'(x)R(x)u(x)$ is a measure of control effort.

5.3.2 Solution for Finite-Horizon Differential SDRE Regulator

To minimize the above cost function (5.31), a state feedback control law is given as

$$u(x) = -Kx(t) = -R^{-1}(x)B'(x)P(x)x(t), \tag{5.32}$$

where $P(x,t)$ is a symmetric, positive-denite solution of the differential SDRE,strictly speaking it could be called State Dependent Differential Riccati Equation (SDDRE), of the form

$$-\dot{P}(x) = P(x)A(x) + A'(x)P(x) - P(x)B(x)R^{-1}(x)B'(x)P(x) + Q(x), \tag{5.33}$$

with the final condition as

$$P(x,t_f) = F. \tag{5.34}$$

The resulting differential SDRE-controlled trajectory becomes the solution of the state-dependent closed-loop dynamics

$$\dot{x}(t) = [A(x) - B(x)R^{-1}(x)B'(x)P(x)]x(t) \tag{5.35}$$

As the differential SDRE is a function of (x, t), the value of the states is unknown ahead of present time step. Therefore, the SDC State dependent coefficient can not be calculated to solve (5.33) with the final condition (5.34) by backwards integration from t_f to t_0. In order to sideline this issue,an approximate analytical approach is used [17] [29], which converts the original nonlinear differential Ricatti equation to a linear differential Lyapunov equation, which can be solved in closed form at each time step. To solve SDDRE(5.33) following steps are needed to be performed at each time step [20]

1. Solve the ARE to calculate steady state value $\mathbf{P}_{ss}(x)$

$$P_{ss}(x)A(x) + A'(x)P_{ss}(x) - P_{ss}(x)B(x)R^{-1}(x)B'(x)P_{ss}(x) + Q(x) = 0, \quad (5.36)$$

2. Use change of variable and use that

$$K(x,t) = [P(x,t) - P_{ss}(x)]^{-1} \quad (5.37)$$

3. Calculate the value of $A_{cl}(x)$ as

$$A_{cl}(x) = A(x) - B(x)R^{-1}B'(x)P_{ss}(x) \quad (5.38)$$

4. Calculate the value of D by solving the algebric Lyapunov equation...

$$A_{cl}D + DA'_{cl} - BR^{-1}B' = 0. \quad (5.39)$$

5. Solve the differential Lyapunov equation

$$\dot{K}(x,t) = K(x,t)A'_{cl}(x) + A'_{cl}(x)K(x,t) - B(x)R^{-1}B'(x) \quad (5.40)$$

   The solution of (5.40), as shown by (5.37), is given by

$$K(x,t) = e^{A_{cl}(t-t_f)}(K(x,t_f) - D)e^{A'_{cl}(t-t_f)} + D \quad (5.41)$$

6. Use change of variables procedure to calculate the value $P(x,t)$ from (5.37)

$$P(x,t) = K^{-1}(x,t) + P_{ss}(t) \quad (5.42)$$

7. Finally, calculate the value of the optimal control $u(x,t)$ as

$$u(x,t) = -R^{-1}B'(x)P(x,t)x(t) \tag{5.43}$$

Instead of solving $P(x,t)$ backward in time equation (5.33), its is solved through (5.42) in terms of $K(x,t)$ (5.41),the differential Lyapunov equation (DLE) (5.40) which requires the solution of ARE (5.36).

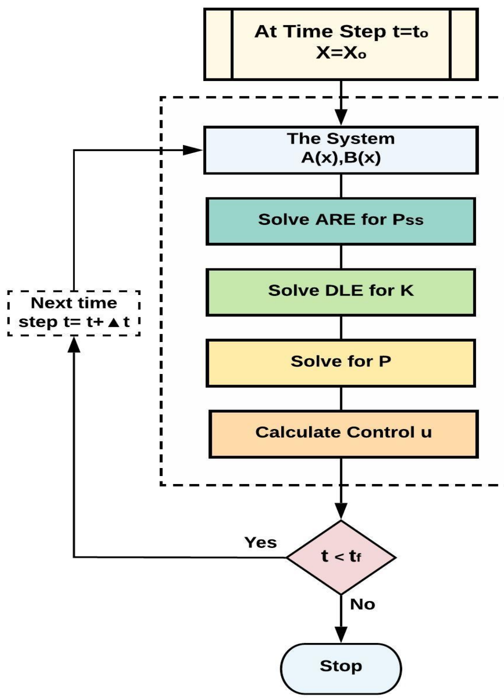

Figure 5.1: Flow Chart for Finite-Horizon Differential SDDRE Regulation Technique

5.4 Finite-Horizon Differential SDRE Regulation Simulation

In order to support the above technique, this section demonstrate the simulations with finite-horizon optimal regulator controller for linear and non-linear systems.

5.4.1 Ball and Beam System

A ball is placed on a beam in fig (5.2). A leaver arm is attached to the beam at one end and a servo gear at the other. If the servo gear turns by an angle θ, the leaver changes the angle of the beam by α. When the angle is changed from the horizontal position, gravity causes the ball to roll along the beam. A controller will be designed for this system so that the ball's position can be manipulated.

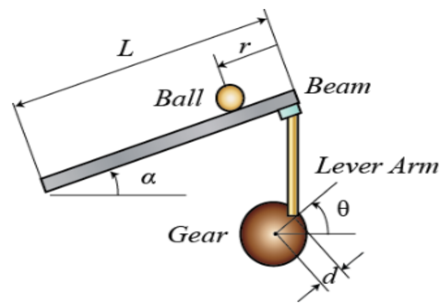

Figure 5.2: Ball and Beam System [5]

For this problem, we will assume that the ball rolls without slipping and friction between the beam and ball is negligible.The constants and variables for this example are defined as follows

Variable	**Description**	**Value**
m	mass of the ball	$0.111(kg)$
R	radius of the ball	$0.015(m)$
d	lever arm offset	$0.03(m)$
g	gravitational acceleration	$9.8(m/s^2)$
L	length of the beam	$1.0(m)$
J	ball's moment of inertia	$9.99e^{-6}(kg.m^2)$
r	ball position coordinate	
α	beam angle coordinate	
θ	servo gear angle	

Table 5.1: Ball and Beam System Parameters

The linearized system equations can also be represented in state-space form. This can be done by selecting the ball's position (r) and velocity ($\dot{r}$) as the state variable and the gear angle (θ). The state-space representation is shown below:

$$\begin{bmatrix} \dot{r} \\ \ddot{r} \\ \dot{\alpha} \\ \ddot{\alpha} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{-mg}{(\frac{J}{R^2}+m)} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} r \\ \dot{r} \\ \alpha \\ \dot{\alpha} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} u \end{bmatrix}, \tag{5.44}$$

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} r \\ \dot{r} \\ \alpha \\ \dot{\alpha} \end{bmatrix} \tag{5.45}$$

The task is to bring the states (r) and (θ) from their initial conditions to the equilibrium state.

Setting the initial states as

$$x_0 = \begin{bmatrix} 3 & 0 & 2 & 0 \end{bmatrix}^T$$

Weighting matrices are selected as below.

Q=diag([3,1,1,1])   R=0.05   F=diag([0,0,0,0])

The simulations are performed for final time of 10 seconds.The states for (r) and (θ) reaches to equlibrium as can be seen in fig (5.3) with in span of less than 3.5 sec. The cost function can further be altered by changing the weighting matrices i.e (R) and (Q).

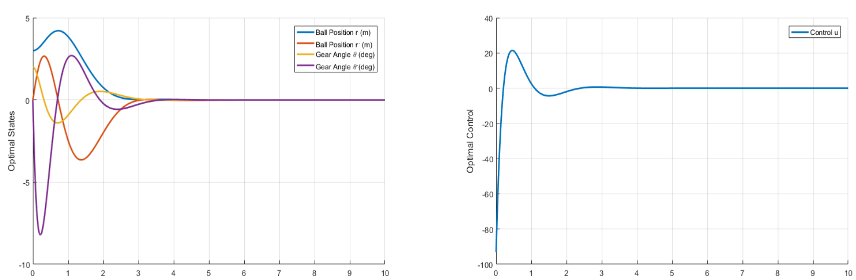

Figure 5.3: Optimal State and Control

5.4.2 Flight Dynamics and Control of 3DOF Hover

The control of vertical lift-off vehicles 3 DOF Hover [38] consist of a planar round frame with four propellers. The frame is mounted on a joint that enables the body to rotate about three axis. The propellers are driven by four DC motors. The lift force generated by the propellers is used to control the roll (ϕ) and pitch (θ) angles.Yaw control is done using the total torque generated by the propellers.The main focus is to control the roll and pitch of the hover, while maintaining the constant yaw.

Figure 5.4: 3 DOF Hover [1]

Nonlinear Dynamics

The nonlinear equations of motion of 3DH are given by [35].

$$\dot{\phi} = p + sin(\phi)tan(\theta) + cos(\phi)tan(\theta)r, \tag{5.46}$$

$$\dot{\theta} = cos(\phi)q - sin(\phi)r, \tag{5.47}$$

$$\dot{\psi} = \frac{cos(\phi)}{cos(\theta)} + \frac{sin(\phi)}{cos(\theta)}q, \tag{5.48}$$

$$\dot{p} = \frac{J_{yy} - J_{zz}}{J_{xx}}qr + \frac{1}{J_{xx}}\tau_r, \tag{5.49}$$

$$\dot{q} = \frac{J_{zz} - J_{xx}}{J_{yy}}pr + \frac{1}{J_{yy}} + \frac{1}{J_{yy}}\tau_p, \tag{5.50}$$

$$\dot{r} = \frac{J_{xx} - J_{yy}}{J_{zz}}qr + \frac{1}{J_{zz}}\tau_y, \tag{5.51}$$

$$\tag{5.52}$$

where ϕ, θ, ψ are the Euler angles, p, q, r are the angular rates in the body axes, J_{xx}, J_{yy}, J_{zz} are the moments of inertia, and τ_r, τ_p, τ_y are the torques acting on the roll, pitch, and yaw axes, respectively. Let V_f, V_b, V_r, V_l be the corresponding voltages of the front, back, right, and left motors. The torques are given by

$$\tau_r = LK_f(V_r - V_l), \tag{5.53}$$

$$\tau_p = LK_f(V_f - V_b), \tag{5.54}$$

$$\tau_y = K_t(V_r + V_l) - K_t(V_f + V_b), \tag{5.55}$$

$$\tag{5.56}$$

where Kf is the thrust-force constant, Kt is the thrust-torque constant, and L is the distance between each propeller motor and the pivot on the axis. The parameters of 3D Hover are given below [2]

Parameter	**Value**	**Units**
Pitch angle range	±37.5	deg
Yaw angle range	360	deg
Base dimension (L)	0.175	m
Moment of inertia around x-axis (J_{xx})	0.055	$kg - M^2$
Moment of inertia around y-axis (J_{yy})	0.055	$kg - M^2$
Moment of inertia around z-axis (J_{zz})	0.110	$(kg.m^2)$
Motor/propeller force-thrust constant (K_f)	0.119	N/V
Motor/propeller torque thrust constant (K_t)	0.0036	$N - m/V$

Table 5.2: 3 DOF Hover Specication

SDC Model of 3D Hover

Define the state vector $x = [\phi, \theta, \psi, p, q, r]^T$ and the control vector $u = [V_f, V_b, V_r, V_l]^T$, the dynamics of the 3D hover is given below as

$$A(x) = \begin{bmatrix} 0 & 0 & 0 & 1 & sin(\phi)tan(\theta) & cos(\phi)tan(\theta) \\ 0 & 0 & 0 & 0 & cos(\phi) & -sin(\phi) \\ 0 & 0 & 0 & 0 & \frac{sin(\phi)}{cos(\theta)} & \frac{cos(\phi)}{cos(\theta)} \\ 0 & 0 & 0 & 0 & 0 & \frac{j_{yy}-j_{zz}}{j_{xx}}q \\ 0 & 0 & 0 & \frac{j_{zz}-j_{xx}}{j_{yy}}r & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{j_{xx}-j_{yy}}{j_{zz}}p & 0 \end{bmatrix}, \tag{5.57}$$

$$B(x) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{LK_f}{j_{xx}} & -\frac{LK_f}{j_{xx}} \\ \frac{LK_f}{j_{yy}} & -\frac{LK_f}{j_{yy}} & 0 & 0 \\ -\frac{K_t}{j_{zz}} & -\frac{K_t}{j_{zz}} & \frac{K_t}{j_{zz}} & \frac{K_t}{j_{zz}} \end{bmatrix}, \tag{5.58}$$

Setting the initial condition as

$$x_0 = \begin{bmatrix} 2 & -2 & 0 & 0 & 0 & 0 \end{bmatrix}^T$$

Weighting matrices are selected as below.

$$\text{Q}=diag(11[I_2], [I_4]) \quad \text{R}=0.01[I_4] \quad \text{F}=diag(0 * [I_6])$$

Let the desired trajectory for ϕ and θ are

$$z_1(t) = 0, \tag{5.59}$$

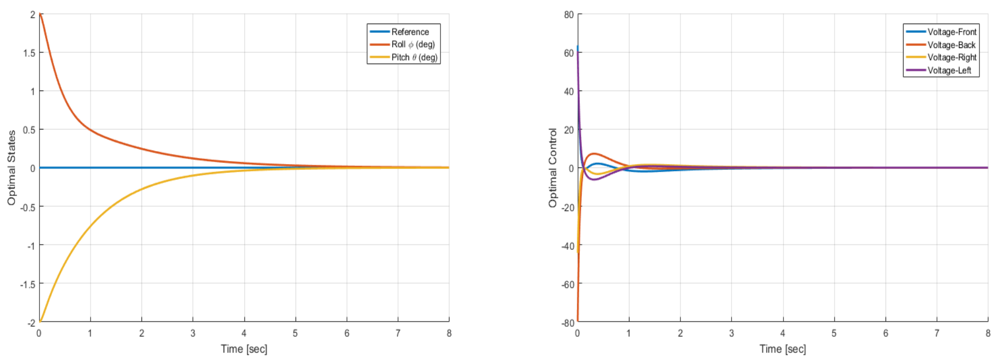

Figure 5.5: Optimal State and Control

Setting the initial condition as

$$x_0 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T$$

Let the desired trajectory for ϕ and θ are

$$z_2(t) = \begin{cases} 4 \ \ deg \text{ for } 1 < t < 6, \\ 0 \ \ deg \text{ for } 6 < t < 11, \\ 4 \ \ deg \text{ for } 11 < t < 16, \\ 0 \ \ deg \text{ for } 16 < t < 20, \end{cases} \tag{5.60}$$

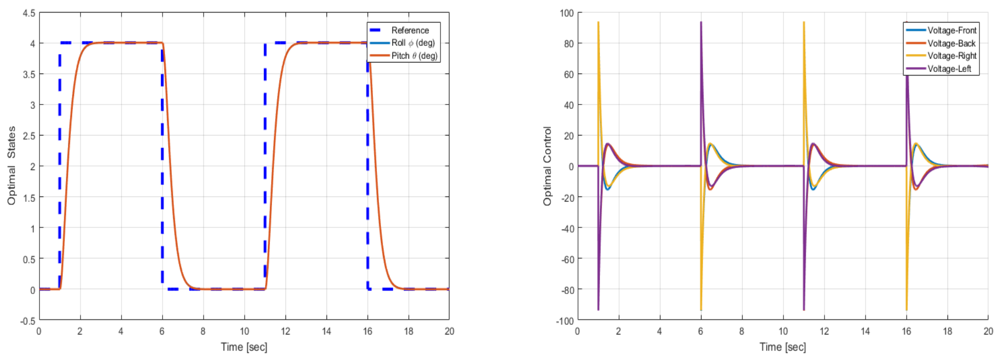

Figure 5.6: Optimal State and Control

Chapter 6

Finite-Horizon Tracking with Differential State Dependent Riccati Equation (DSDRE)

6.1 Finite-Horizon Tracking for Deterministic Nonlinear Systems

6.1.1 Problem Formulation

The nonlinear system considered in this chapter is in the form:

$$\dot{x}(t) = f(x) + g(x)u(t), \tag{6.1}$$

$$y(t) = h(x). \tag{6.2}$$

This nonlinear system can be expressed in a state-dependent *linear*-like form

$$\dot{x}(t) = A(x)x(t) + B(x)u(t), \tag{6.3}$$

$$y(t) = C(x)x(t). \tag{6.4}$$

where $f(x) = A(x)x(t)$, $B(x) = g(x)$, and $h(x) = C(x)x(t)$.The goal is to nd a state feedback optimal control law of the form u(x) = Kx(t),that minimizes a cost function given by

$$J(x,u) = \frac{1}{2}e'(t_f)Fe(t_f) + \frac{1}{2}\int_{t_0}^{t_f}[e'(t)Q(x)e(t) + u'(x)R(x)u(x)]dt \tag{6.5}$$

where $Q(x)$ and F are symmetric *positive semi-definite* matrices, and $R(x)$ is a symmetric *positive definite* matrix. Moreover, $x'Q(x)x$ is a measure of state accuracy and $u'(x)R(x)u(x)$ is a measure of control effort and the error is $e(t) = z(t) - y(t)$.

6.1.2 Solution for Finite-Horizon Tracking using Dierential SDRE

To minimize the cost function, a feedback control law is given as

$$u(x) = -R^{-1}B'(x)[P(x)x - g(x)], \tag{6.6}$$

where $P(x)$ is a symmetric, positive-definitive solution of the differential SDRE of the form

$$-\dot{P}(x) = P(x)A(x) + A'(x)P(x) - P(x)B(x)R^{-1}B'(x)P(x) + C'(x)Q(x)C(x), \tag{6.7}$$

with the final condition

$$P(x,t_f) - C'(t_f)FC(t_f), \tag{6.8}$$

and $g(x)$ is a solution of the state-dependent non-homogeneous vector differential equation

$$\dot{g}(x) = -[A(x) - B(x)R^{-1}(x)B'(x)P(x)]'g(x) - C'(x)Q(x)z(x), \tag{6.9}$$

with the final condition

$$g(x,t_f) = C'(t_f)Fz(t_f). \tag{6.10}$$

The resulting differential SDRE- controlled trajectory becomes the solution of the state dependent closed loop dynamics.

$$\dot{x}(t) = [A(x) - B(x)R^{-1}(x)B'(x)P(x)]x(t) + B(x)R^{-1}(x)B'(x)g(x). \tag{6.11}$$

Similar to Section 4.2.2, an approximate analytical approach is used and the DRE (4.3.3), and the non-homogeneous dierential equation (4.3.5), can be solved in the following steps at each time step:

1. Solve for $P(x,t)$ similar to the dierential SDRE regulator problem in Section 4.2.2, steps from 1 to 6.

2. Calculate the steady state value $gss(x)$ from the equation

$$g_{ss}(x) = [A(x) - B(x)R^{-1}(x)B'(x)P_{ss}(x)]'^{-1}C'(x)Q(x)z(x) \tag{6.12}$$

3. Use change of variables technique and assume that

$$K_g(x,t) = [g(x,t) - g_{ss}(x)]. \tag{6.13}$$

4. Solve the differential equation

$$K_g(x,t) = e^{-(A-BR^{-1}B'P)'(t-t_f)}[g(x,t_f) - g_{ss}(x)]. \tag{6.14}$$

5. Use changing of variables procedure to calculate the value of $g(x,t)$

$$g(x,t) = K_g(x,t) + g_{ss}(x). \tag{6.15}$$

6. Calculate the value of the optimal control u(x,t) as

$$u(x,t) = -R^{-1}(x)B'(x)[P(x,t)x(t) - g(x,t)]. \tag{6.16}$$

6.2 Finite-Horizon Dierential SDRE Tracking Simulation

This section presents the simulation results with finite horizon optimal tracking controller for Non-linear systems.

6.2.1 Permanent Magnet Synchronous Motor

A Non-linear mathematical model of a surface mounted permanent magnet synchronous DC motor is presented as

$$J\frac{dw}{dt} = P\psi_f i_q - Bw - cd - df, \tag{6.17}$$

$$L\frac{di_q}{dt} = -Ri_q + LPwi_d - P\psi_f w + u_q, \tag{6.18}$$

$$L\frac{di_d}{dt} = -Ri_d + LPwi_q + u_d, \tag{6.19}$$

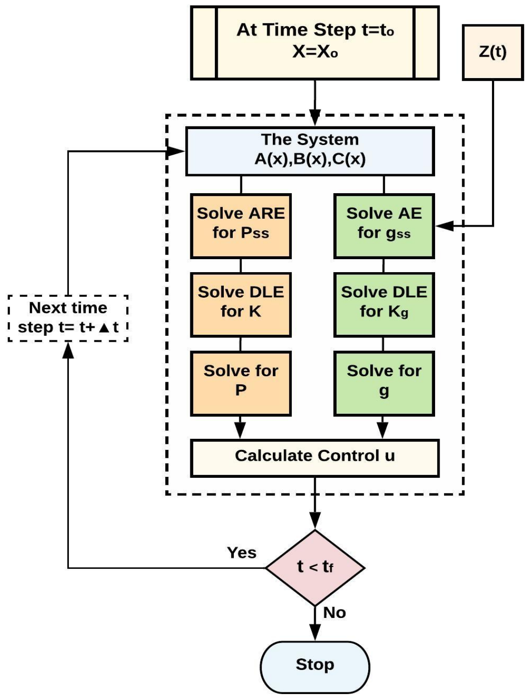

Figure 6.1: Finite-Horizon Differential SDRE Tracking technique

where R is stator resistance, L is stator winding inductance, P is pole pairs, ψ_f is rotor permanent magnet ux linkage, J is moment inertia, B is viscous friction coecient, i_d and i_q , u_d and u_q are dq axis stator current and voltage respectively, w is the motor mechanical velocity.
The nonlinear equation of the system can be written as

$$\dot{x_1} = -0.02_1 + 166.2x_2, \tag{6.20}$$

$$\dot{x_2} = -231.07x_1 - 24.2x_2 + 6x_1x_3 + 0.05u_1, \tag{6.21}$$

$$\dot{x_3} = -24.2x_3 + 6x_1x_2 + 0.05u_2, \tag{6.22}$$

$$y = x_1, \tag{6.23}$$

where $w = x_1$, $i_q = x_2$, $i_d = x_3$,$u_q = u_1$,$u_d = u_2$ and the motor parameters are showed

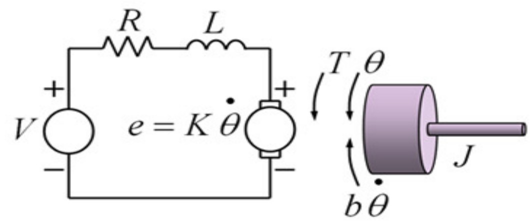

Figure 6.2: Electro-Mechanical Model of A DC Motor

in Table The system nonlinear equations can be written in the state dependent form

$$\begin{bmatrix} \dot{x_1} \\ \dot{x_2} \\ \dot{x_3} \end{bmatrix} = \begin{bmatrix} -0.02 & 166.2 & 0 \\ -231.07 & -24.2 & 6x_1 \\ 0 & 6x_1 & -24.2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0.05 & 0 \\ 0 & 0.05 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}, \tag{6.24}$$

Let the desired trajectory as

$$z_1(t) = 10 * \sin(t) \tag{6.25}$$

$$z_2(t) = \begin{cases} 200rpm \text{ for } 1 < t < 5, \\ 350rpm \text{ for } 5 < t < 8, \end{cases} \tag{6.26}$$

and the selected weighted matrices be

$$Q = diag(2000, 2000, 2000), R = diag(1, 1), F = diag(1, 1, 1) \tag{6.27}$$

Parameter	Nominal value
R	0.349ω
L	14.385mH
J	$0.02kg.m^2$
P	6
ψ_f	0.554Wb
B	0.0004N.m.sec/rad

Table 6.1: Synchronous Motor Parameters

The simulations are performed for final time of 10 seconds and the resulting motor speed trajectory and the optimal control is shown in fig (6.3), the dashed line denotes the desired speed and solid lines denotes the actual motor speed trajectory with various different weighting matrix (R).

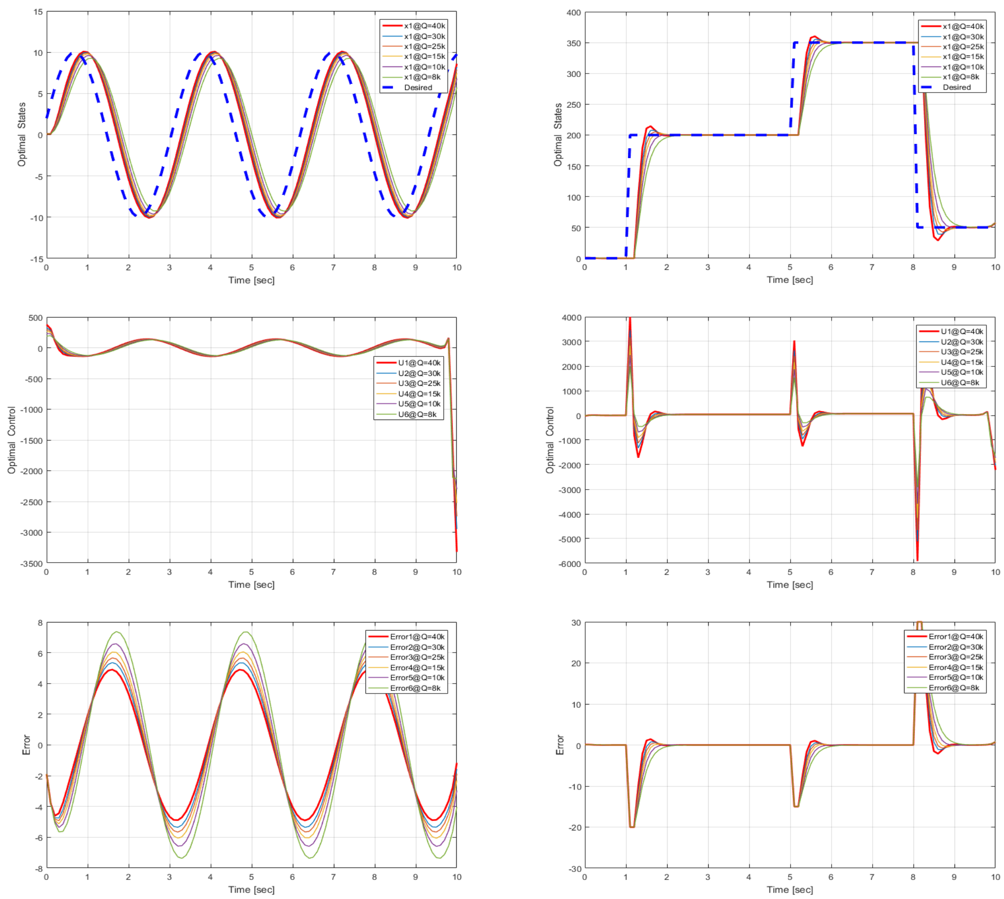

Figure 6.3: Optimal State, Control and Error

By Comparing these trajectories in fig (6.3), its clear that the finite-horizon differential SDRE nonlinear tracking algorithm gives very good results as the actual optimal speed is making very good tracking to the desired trajectory, and the developed algorithm is able to solve the differential SDRE finite-horizon nonlinear tracking problem.

6.2.2 Conclusion

Finite-horizon optimal control of nonlinear systems is a challenging problem in the control field due to the complexity of time-dependency of the Hamilton Jacobi Bell man (HJB) differential equation. The available methods for this purpose have draw backs in terms of the dependency of the solution to the pre-specified initial conditions and the lack existence of the optimal solution under all conditions. This chapter presents a novel and computationally efficient online technique used for finite-horizon nonlinear regulating and tracking problems. This technique based on change of variables that converts the differential Riccati equation to a linear Lyapunov equation. During online implementation, the Lyapunov equation is solved in a closed form at the given time step.

Chapter 7

Conclusion and Future Investigations

In this thesis, a brief investigation of advanced System Identification and tracking strategies for linear and nonlinear optimal control systems and its application has been presented. The main conclusions derived from this dissertation are discussed in this chapter.

7.1 Conclusion

In System identification toolbox input-output data has been collected from a plant of interest. To get a quick feeling of the process' behavior and complexity one often starts to view the input-output relationship via non-parametric frequency domain models, such as the spectral analysis estimate. This approach is indeed appealing, since very little about the system in question must be known beforehand. Apart from the assumption that the system is linear the only practical consideration is the choice of weighting function to smooth out the frequency response (see, for instance, Ljung [23] for computational details).After several iteration of input output data and adjustment of poles and zeroes the Mathematical Linear mathematical model was formed and further validated with actual input signal. Similarly, Non-linear ARX model is also validated with the actual motor angle.

The State Dependent Riccati Equation (SDRE) provide an effective algorithm for optimal control design for nonlinear dynamical systems. The SDRE algorithm captures the nonlinearities of the system, transforming the original nonlinear system to a linear-like structure with State Dependent Coefficient (SDC) matrices, and minimizing a non-quadratic performance index with a quadratic-like structure. The main advantage of SDRE is the ability to make trade offs between control effort and state errors by tuning the SDC.

This research presented an efficient online technique used for infinite-horizon and tracking problems. The idea of the proposed technique is the integration of the Kalman filter algorithm and the infinite-horizon algebraic SDRE technique. Unlike the ordinary methods which deal with the linearized system, this technique estimates the unmeasured states of the nonlinear system directly.

Next, this research discussed an online technique for finite-horizon regulation and tracking of nonlinear systems. The proposed technique is based on change of variables that converts the nonlinear differential Riccati equation (DRE) (to be solved backward in time using nal condition) to a linear differential Lyapunov equation (to be solved forward in time using change in variables).

7.2 Future Investigation

There is a novel approach that is still in proceedings to control the non-linear SDC model with Forward propagating Riccati equation (FPRE) [22] [36][37]can also be utilised and compared with the existing SDRE method. Dr.Dennis S. Bernstein and Anna Prach are currently working on this novel technique.

Bibliography

[1] *3 DOF Hover*. https://www.quanser.com/products/3-dof-hover/.

[2] *3 DOF Hover*. http://www.quanser.com/products.

[3] Akram Aldroubi, Carlos Cabrelli, and Ursula Molter. "Optimal non-linear models". In: *Revista de la Unifffdfffdn Matemfffdfffdtica Argentina, ISSN 0041-6932, Vol. 50, Nfffdfffd. 2, 2009, pags. 217-225* (June 2009).

[4] BDO Anderson and JB Moore. "Optimal Control Linear Quadratic Methods Prentice-Hall". In: *Englewood Cliffs. New Jersey* (1990).

[5] *Ball and Beam System*. http://ctms.engin.umich.edu/CTMS/index.php?example=BallBeam&section=SystemModeling.

[6] HT Banks, BM Lewis, and Hien T Tran. "Nonlinear feedback controllers and compensators: a state-dependent Riccati equation approach". In: *Computational Optimization and Applications* 37.2 (2007), pp. 177–218.

[7] Mourad Boufadene. *Nonlinear Control Systems using MATLABfffdfffd*. Oct. 2018. ISBN: 1138359556. DOI: 10.1201/9780429433221.

[8] Tayfun Cimen. "Survey of state-dependent Riccati equation in nonlinear optimal feedback control synthesis". In: *Journal of Guidance, Control, and Dynamics* 35.4 (2012), pp. 1025–1047.

[9] Sinan KILIfffdfffdASLAN. "Tracking control of elastic joint parallel robots via state-dependent Riccati equation". In: *Turkish Journal of Electrical Engineering and Computer Science* 23 (2015), pp. 522–538. ISSN: 1300-0632. DOI: 10.3906/elk-1301-170.

[10] James R Cloutier. “State-dependent Riccati equation techniques: an overview”. In: *Proceedings of the 1997 American Control Conference (Cat. No. 97CH36041)*. Vol. 2. IEEE. 1997, pp. 932–936.

[11] James R Cloutier, Christopher N D’Souza, and Curtis P Mracek. “Nonlinear regulation and nonlinear Hfffdfffdfffd 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*. Embry-Riddle Aeronautical Univ. Press Daytona Beach, FL. 1996, pp. 117–130.

[12] James R Cloutier, Johnny H Evers, and Joseph J Feeley. “Assessment of air-to-air missile guidance and control technology”. In: *IEEE Control systems magazine* 9.6 (1989), pp. 27–34.

[13] James R Cloutier and Donald T Stansbery. *Nonlinear, hybrid bank-to-turn/skid-to-turn missile autopilot design.* Tech. rep. AIR FORCE RESEARCH LAB EGLIN AFB FL MUNITIONS DIRECTORATE, 2001.

[14] M Deistler. “System Identification T. Sfffdfffdderstrfffdfffdm and P. Stoica Prentice Hall International, 1989”. In: *Econometric Theory* 10 (Aug. 1994), pp. 813–815. DOI: 10.1017/S026646660000880X.

[15] Evrin B Erdem and Andrew G Alleyne. “Experimental real-time SDRE control of an underactuated robot”. In: *Proceedings of the 40th IEEE Conference on Decision and Control (Cat. No. 01CH37228)*. Vol. 3. IEEE. 2001, pp. 2986–2991.

[16] Bernard Friedland. “Feedback control of systems with parasitic effects”. In: *Proceedings of the 1997 American Control Conference (Cat. No. 97CH36041)*. Vol. 2. IEEE. 1997, pp. 937–941.

[17] Ali Heydari and SN Balakrishnan. “Path planning using a novel finite horizon suboptimal controller”. In: *Journal of guidance, control, and dynamics* 36.4 (2013), pp. 1210–1214.

[18] Tayfun fffdfffdimen. “State-Dependent Riccati Equation (SDRE) Control: A Survey”. In: *IFAC Proceedings Volumes* 41.2 (2008). 17th IFAC World Congress, pp. 3761–3775. ISSN: 1474-6670. DOI: https://doi.org/10.3182/20080706-

5-KR-1001.00635. URL: http://www.sciencedirect.com/science/article/pii/S1474667016395349.

[19] Mehmet Itik, Metin Uymaz Salamci, and Stephen Paul Banks. "SDRE optimal control of drug administration in cancer treatment". In: *Turkish Journal of Electrical Engineering & Computer Sciences* 18.5 (2010), pp. 715–730.

[20] AHMED Khamis and D Naidu. "Nonlinear optimal tracking using finitehorizon state dependent Riccati equation (SDRE)". In: *Proceedings of the 4th International Conference on Circuits, Systems, Control, Signals (WSEAS)*. 2013, pp. 37–42.

[21] Peter Lindskog. "Algorithms and Tools for System Identification Using Prior Knowledge". In: 1994.

[22] Lennart Ljung. "System Identification: Design Variables and the Design Objective". In: May 1987, pp. 251–270. DOI: 10.1007/978-3-642-87516-8_16.

[23] Lennart Ljung and Rajiv Singh. "Version 8 of the Matlab System Identification Toolbox". In: *IFAC Proceedings Volumes* 45.16 (2012). 16th IFAC Symposium on System Identification, pp. 1826–1831. ISSN: 1474-6670. DOI: https://doi.org/10.3182/20120711-3-BE-2027.00061. URL: http://www.sciencedirect.com/science/article/pii/S1474667015382227.

[24] Rohan Mandloi and Pritesh Shah. "Methods for closed loop system identification in industry". In: *Journal of Chemical and Pharmaceutical Research* 7.1 (2015), pp. 892–896.

[25] Euan McGookin and David Murray-Smith. "Comment on "Nonlinear optimal tracking control with application to super-tankers for autopilot design" by T. Cimen and S.P. Banks". In: *Automatica* 42 (Dec. 2006), pp. 2223–2225. DOI: 10.1016/j.automatica.2006.06.018.

[26] CP Mracek and James R Cloutier. "A preliminary control design for the nonlinear benchmark problem". In: *Proceeding of the 1996 IEEE International Conference on Control Applications IEEE International Conference on Control Applications held together with IEEE International Symposium on Intelligent Contro*. IEEE. 1996, pp. 265–272.

[27] Curtis P Mracek and James R Cloutier. "Control designs for the nonlinear benchmark problem via the state-dependent Riccati equation method". In: *International Journal of robust and nonlinear control* 8.4-5 (1998), pp. 401–433.

[28] (Desineni S.) Naidu. *Optimal control systems*. English (US). Electrical engineering textbook series. Includes bibliographical references (pages 415-424) and index. CRC Press, 2003. ISBN: 9780849308925.

[29] J Nazarzadeh, M Razzaghi, and KY Nikravesh. "Solution of the matrix Riccati equation for the linear quadratic control problems". In: *Mathematical and computer modelling* 27.7 (1998), pp. 51–55.

[30] Thang Nguyen and Zoran Gajic. "Solving the Matrix Differential Riccati Equation: A Lyapunov Equation Approach". In: *Automatic Control, IEEE Transactions on* 55 (Feb. 2010), pp. 191–194. DOI: 10.1109/TAC.2009.2033841.

[31] Andrew P. Robinson and Jeff Hamann. "Linear and Non-linear Modeling". In: (Jan. 2011). DOI: 10.1007/978-1-4419-7762-5_6.

[32] Neil F Palumbo and Todd D Jackson. "Integrated missile guidance and control: A state dependent Riccati differential equation approach". In: *Proceedings of the 1999 IEEE International Conference on Control Applications (Cat. No. 99CH36328)*. Vol. 1. IEEE. 1999, pp. 243–248.

[33] JD Pearson. "Approximation methods in optimal control I. Sub-optimal control". In: *International Journal of Electronics* 13.5 (1962), pp. 453–469.

[34] Mihai Popescu and Alexandru Dumitrache. "On the optimal control of affine nonlinear systems". In: *Mathematical Problems in Engineering* 2005.4 (2005), pp. 465–475.

[35] A. Prach, E. Kayacan, and D. S. Bernstein. "An experimental evaluation of the forward propagating Riccati equation to nonlinear control of the Quanser 3 DOF Hover testbed". In: *2016 American Control Conference (ACC)*. July 2016, pp. 3710–3715. DOI: 10.1109/ACC.2016.7525490.

[36] A. Prach, O. Tekinalp, and D. S. Bernstein. "A numerical comparison of frozen-time and forward-propagating Riccati equations for stabilization of periodically

time-varying systems". In: *2014 American Control Conference*. June 2014, pp. 5633–5638. DOI: 10.1109/ACC.2014.6859066.

[37] Anna Prach, Ozan Tekinalp, and Dennis S Bernstein. "Faux-Riccati synthesis of nonlinear observer-based compensators for discrete-time nonlinear systems". In: *53rd IEEE Conference on Decision and Control*. IEEE. 2014, pp. 854–859.

[38] Igor Afonso Acampora Prado et al. "Experimental evaluation of HJB optimal controllers for the attitude dynamics of a multirotor aerial vehicle". In: *ISA Transactions* 77 (2018), pp. 188–200. ISSN: 0019-0578. DOI: https://doi.org/10.1016/j.isatra.2018.04.003. URL: http://www.sciencedirect.com/science/article/pii/S0019057818301496.

[39] Lal Bahadur Prasad, Barjeev Tyagi, and Hari Om Gupta. "Optimal Control of Nonlinear Inverted Pendulum System Using PID Controller and LQR: Performance Analysis Without and With Disturbance Input". In: *International Journal of Automation and Computing* 11.6 (Dec. 2014), pp. 661–670. ISSN: 1751-8520. DOI: 10.1007/s11633-014-0818-1. URL: https://doi.org/10.1007/s11633-014-0818-1.

[40] Muhammad Junaid Rabbani et al. "Model Identification and Validation for a Heating System using MATLAB System Identification Toolbox". In: *IOP Conference Series: Materials Science and Engineering* 51 (Dec. 2013), p. 012022. DOI: 10.1088/1757-899x/51/1/012022. URL: https://doi.org/10.1088%2F1757-899x%2F51%2F1%2F012022.

[41] PD Roberts and VM Becerra. "Optimal control of nonlinear systems represented by differential algebraic equations". In: *Proceedings of the 2000 American Control Conference. ACC (IEEE Cat. No. 00CH36334)*. Vol. 2. IEEE. 2000, pp. 762–763.

[42] M. Ruderman et al. "Extended SDRE control of 1-DOF robotic manipulator with nonlinearities". In: *IFAC Proceedings Volumes* 44.1 (2011). 18th IFAC World Congress, pp. 10940–10945. ISSN: 1474-6670. DOI: https://doi.org/10.3182/20110828-6-IT-1002.00518. URL: http://www.sciencedirect.com/science/article/pii/S1474667016453711.

[43] Maryam Sarostad et al. "Nonlinear Model-Free Control and ARX Modeling of Industrial Motor". In: *International Journal of Smart Home* 10 (Dec. 2016), pp. 63–76. DOI: 10.14257/ijsh.2016.10.12.07.

[44] Haihua Tan, Shaolong Shu, and Feng Lin. "An optimal control approach to robust tracking of linear systems". In: *International Journal of Control - INT J CONTR* 82 (Mar. 2009), pp. 525–540. DOI: 10.1080/00207170802187239.

[45] Popescu Theodor. "Modeling of dynamic systems: by Lennart LJUNG and Torkel GLAD; Prentice Hall Information and System Sciences Series; Prentice Hall; Englewood Cliffs, NJ, USA; 1994 ISBN: 0-13-597097-0 - Book review". In: *Control Engineering Practice* 3 (June 1995), 897fffdfffdfffd898. DOI: 10.1016/0967-0661(95)90022-5.

[46] Ding Wang, Derong Liu, and Qinglai Wei. "Finite-horizon neuro-optimal tracking control for a class of discrete-time nonlinear systems using adaptive dynamic programming approach". In: *Neurocomputing* 78.1 (2012). Selected papers from the 8th International Symposium on Neural Networks (ISNN 2011), pp. 14–22. ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2011.03.058. URL: http://www.sciencedirect.com/science/article/pii/S0925231211005030.

[47] Haiming Wang, Qingze Zou, and Hongbing Xu. "Inversion-based optimal output trackingfffdfffdfffdtransition switching with preview for nonminimum-phase linear systems". In: *Automatica* 48 (July 2012), 1364fffdfffdfffd1371. DOI: 10.1016/j.automatica.2011.11.011.

[48] Andreas Wernli and Gerald Cook. "Suboptimal control for the nonlinear quadratic regulator problem". In: *Automatica* 11.1 (1975), pp. 75–84.

[49] C-H Won and Saroj Biswas. "Optimal control using an algebraic method for control-affine non-linear systems". In: *International Journal of Control* 80.9 (2007), pp. 1491–1502.