5-2016

A radial basis function method for solving optimal control problems.

Hossein Mirinejad
University of Louisville

A RADIAL BASIS FUNCTION METHOD FOR SOLVING OPTIMAL CONTROL PROBLEMS

By

Hossein Mirinejad
B.Sc. Electrical Engineering, Iran Univ. of Sci. & Tech.
M.Sc. Mechatronics Engineering, K.N.T. Univ. of Tech.

A Dissertation
Submitted to the Faculty of the
J.B. Speed School of Engineering of the University of Louisville
in Partial Fulfillment of the Requirements
for the Degree of

Doctor of Philosophy
in Electrical Engineering

Department of Electrical and Computer Engineering
University of Louisville
Louisville, Kentucky

May 2016

A RADIAL BASIS FUNCTION METHOD FOR SOLVING OPTIMAL CONTROL PROBLEMS

By

Hossein Mirinejad
B.Sc. Electrical Engineering, Iran Univ. of Sci. & Tech.
M.Sc. Mechatronics Engineering, K.N.T. Univ. of Tech.

A Dissertation Approved On

April 15, 2016

by the following Dissertation Committee:

Tamer Inanc (Dissertation Director)

Gail W. DePuy

Ibrahim Imam

Michael McIntyre

Jacek M. Zurada

DEDICATION

To all who love, support, and help human beings to make the earth a better place for us…

ACKNOWLEDGMENTS

This work would not have been possible without the help of many people. I would like to thank my advisor, Dr. Tamer Inanc for his guidance, advice, patience, and understanding throughout my Ph.D. work. I am thankful for his support and also for giving me the freedom as well as flexibility in my research. I would also like to express my gratitude to Dr. Jacek Zurada, a member of my dissertation committee, for his great help and strong support during my research. The joint seminars he organized were great opportunities to share ideas and develop presentation skills. I extend my thanks to the rest of my committee members, Dr. McIntyre, Dr. Imam, and Dr. Depuy, for agreeing to be on my committee and also for providing helpful suggestions during my research, especially those suggested on my proposal exam date. I also acknowledge Dr. Adam Gaweda and Dr. Mike Brier from School of Medicine for their helpful discussions and for providing required data and software codes for the anemia management project.

I am sincerely indebted to my parents for all their relentless support, patience, encouragement, prayer, and guidance over the years. Mom and dad, I forever love you, and I am eternally grateful for your sacrifice to make me a better person. I would also like to thank all my friends who made this journey enjoyable. You may have not realized, but your friendship and support helped me a lot in my personal and professional life. Lastly, I want to thank Maryam, my best friend and beloved wife, who walked beside me through my moments of doubt and discouragement. You have always been my greatest source of

energy to finish this work, and I am really thankful for all your love, support, encouragement, sacrifice, and dedication to our family.

ABSTRACT

A RADIAL BASIS FUNCTION METHOD FOR SOLVING OPTIMAL CONTROL PROBLEMS

Hossein Mirinejad

April 15, 2016

This work presents two direct methods based on the radial basis function (RBF) interpolation and arbitrary discretization for solving continuous-time optimal control problems: *RBF Collocation Method* and *RBF-Galerkin Method*. Both methods take advantage of choosing any global RBF as the interpolant function and any arbitrary points (meshless or on a mesh) as the discretization points. The first approach is called the RBF collocation method, in which states and controls are parameterized using a global RBF, and constraints are satisfied at arbitrary discrete nodes (collocation points) to convert the continuous-time optimal control problem to a nonlinear programming (NLP) problem. The resulted NLP is quite sparse and can be efficiently solved by well-developed sparse solvers. The second proposed method is a hybrid approach combining RBF interpolation with Galerkin error projection for solving optimal control problems. The proposed solution, called the RBF-Galerkin method, applies a Galerkin projection to the residuals of the optimal control problem that make them orthogonal to every member of the RBF basis functions. Also, RBF-Galerkin costate mapping theorem will be developed describing an exact equivalency between the Karush–Kuhn–Tucker (KKT) conditions of the NLP problem resulted from the RBF-Galerkin method and discretized form of the first-order necessary conditions of the optimal control problem, if a set of conditions holds. Several examples are provided to verify the feasibility and viability of the RBF method and the RBF-Galerkin approach as means of finding accurate solutions to general optimal control problems. Then, the RBF-Galerkin method is applied to a very important drug dosing application: anemia management in chronic kidney disease. A multiple receding horizon control (MRHC) approach based on the RBF-Galerkin method is developed for individualized dosing of an anemia drug for hemodialysis patients. Simulation results are compared with a population-oriented clinical protocol as well as an individual-based control method for anemia management to investigate the efficacy of the proposed method.

TABLE OF CONTENTS

CHAPTER I

CHAPTER II

CHAPTER III

CHAPTER IV

CHAPTER V

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION AND MOTIVATION

Optimal control problems are extensively used in various areas of science, technology and even medicine. While the traditional applications of optimal control problems including control and navigation of aerospace systems and robot motion planning are still of great interest, their new applications in economics, bioengineering, medicine and other fields of science have received significant attention from researchers recently. For example, minimizing the cost of production and storage of a given product could be treated as an optimal control problem [1]. Similarly, in medicine, minimizing the number of cancer cells over a fixed therapy interval can be formulated as an optimal control problem with the number of cancer cells as *states (x)* and drug dosage as the *control (u)* [2].

In a general optimal control problem, the goal is to efficiently optimize a cost function subject to different constraints including system dynamics (e.g. linear or nonlinear dynamics), boundary conditions (e.g. initial and final constraints), path constraints (e.g. obstacle avoidance), and actuator constraints (e.g. limit in force, torque). Except for simplified systems with linear dynamics and constraints, seeking for minimizers of the optimal control problem cannot be analytically accomplished due to the existence of both differential and algebraic equations imposing a nonlinear infinite dimensional search space. Instead, numerical approaches are usually applied for solving these problems by introducing some level of approximation to the problem [3], [4].

Numerical methods for solving optimal control problems are either indirect or direct methods. Indirect methods apply Pontryagin’s Maximum Principle (PMP) and calculus of variations to find the optimality conditions for a given problem, eventually leading to a Hamiltonian Boundary Value Problem (HBVP) that can be solved using one of the various numerical methods [5]–[7]. Analytical differentiation, which is necessary to derive the optimality conditions, might be long and quite tedious for complex optimal control problems. Also, indirect methods suffer from difficulties in finding the initial guess of costates (adjoint variables) for an HBVP problem making them numerically unreliable. In addition, they need a priori knowledge of the constrained and unconstrained arcs in case of having inequality path constraints in the optimal control problem [3].

On the other hand, direct methods are extensively being used for solving optimal control problems by applying parameterization-then-discretization to the original problem. In a direct method, the states and/or controls are approximated by a specific function with unknown coefficients, and the optimal control problem is discretized using a set of proper nodes (collocation points) to eventually transcribe it into a nonlinear programming (NLP) problem. The NLP can then be solved by commercial off-the-shelf NLP solvers. While the solution of a direct transcription may not be as accurate as that of an indirect method as a result of approximation and discretization, there would be some advantages in solving optimal control problems with direct methods. Perhaps, one advantage is that they do not use the PMP to solve an optimal control problem. Hence, there is no need to analytically differentiate the expressions for the cost and constraints. Also, they have larger radii of convergence than indirect methods, in general. In addition, direct methods are more robust to the initial guess of parameters, do not need the guess of costates, and can be modified quite easily without reformulating the whole problem.

Direct methods are mostly based on polynomial interpolation whether local or global polynomial(s). In local polynomial methods, such as Hermite-Simpson (HS) approaches [8] and spline-based approximation methods [9], states and controls are usually discretized into a set of nodes (equally or unequally spaced) in the time domain, and segments between each node are approximated by low-order polynomials. Global polynomial methods, on the other hand, use a high-degree global polynomial (Chebyshev or Lagrange polynomials of degree N) for parameterization, and a set of orthogonal nodes associated with the family of the polynomial for discretization [10]-[15]. Global polynomial methods have an exponential (spectral) convergence rate and hence, converge faster than local polynomial methods with algebraic convergence rate for a small number of nodes. However, they are limited by being tied to a certain grid of nodes. This drawback becomes more important when an interpolating function is not smooth enough making a pseudospectral method less efficient [16]. Although polynomials are computationally easy to use and have simple forms, other types of approximation, including RBF interpolation may lead to more efficient results, depending on the problem under consideration [17], [18].

In this dissertation, an RBF-based framework is developed for solving optimal control problems. The first method proposed is called the RBF collocation method, using arbitrary global RBFs for parameterizing the states and controls along with arbitrary discretization nodes to transcribe the optimal control problem into an NLP problem with RBF coefficients as decision variables of the NLP. The second method proposed is a hybrid method, called RBF-Galerkin approach, combining RBFs as trial functions with Galerkin error projection to efficiently solve an optimal control problems numerically. Another contribution of the current work is costate estimation using the RBF-Galerkin method. A set of conditions are provided for the equivalency of Karush-Kuhn-Tucker (KKT) multipliers of the NLP resulted from the RBF-Galerkin method and discretized form of the costates of the optimal control problem to eventually develop a costate mapping theorem for the RBF-Galerkin method.

The fact that both RBF collocation method and RBF-Galerkin approach do not require a specific grid of nodes for the discretization, provides more flexibility in choosing the collocation points, which would be really helpful for approximating nonsmooth functions with discontinuities. It is emphasized that the work presented here is substantially different from [19]. The problem considered in this dissertation is a general optimal control problem in which the final optimization time may be either known or unknown. In a free-final-time problem (e.g. time-optimal problem), the final optimization time and possibly some of the final states are unknown making the optimization problem more challenging. In addition, the problem considered here may include path constraints which are of vital importance in motion planning and navigation problems. Nonlinear path constraints are of great importance in guidance and navigation, so that motion planning with obstacle avoidance cannot be solved without considering those constraints. Moreover, the proposed approaches (RBF and RBF-Galerkin methods) may employ any global RBF (e.g. Gaussian RBFs, multiquadrics, inverse multiquadrics, …) for parameterization and any arbitrary points (e.g. equally-spaced nodes, orthogonal nodes, …) for discretization. The practical importance of the proposed work is that a variety of RBF functions can be applied for the interpolation of states and controls (instead of being limited to a specific type of polynomial as in polynomial-based methods), and also a wide range of discretization nodes can be easily employed for the discretization, providing a flexible RBF framework for solving optimal control problems.

The rest of the dissertation is organized as follows. In Chapter II, the continuous-time optimal control problem is formulated, also previous solutions as well as related work in surveys are reviewed. In Chapter III, the RBF collocation method, a computational approach based on the RBF interpolation and arbitrary discretization, is proposed to solve optimal control problems numerically. Chapter IV explains the RBF-Galerkin approach for costate estimation and direct trajectory optimization. Also, numerical examples and performance comparisons are provided in Chapter III and Chapter IV to investigate the efficiency of the RBF method as well as the RBF-Galerkin approach for solving optimal control problems. Chapter V describes the application of the proposed solutions to the anemia management problem. Anemia management is formulated as a constrained optimal control problem and a multiple receding horizon controller based on the RBF-Galerkin method designed for individualized dosing of an anemia drug in patients with chronic kidney disease. Finally, conclusions and suggested future work are given in Chapter VI.

CHAPTER II

OPTIMAL CONTROL PROBLEM FORMULATION AND EXISTING SOLUTIONS

Optimal control of nonlinear systems is an active research area in control systems theory. In a general optimal control problem, the goal is to minimize a cost function with respect to various types of constraints, including system equations, boundary conditions, path constraints, and actuator constraints. Finding the minimizers of an optimal control problem subject to the constraints is usually called optimal trajectory generation (trajectory optimization), or motion planning -the latter is more popular term in robotics- which is an open research problem in many different fields. In this Chapter, a continuous-time optimal control problem is formulated. Also, a summary of previous and existing solutions to optimal control problems is reviewed.

2.1 Problem Statement

Consider the following constrained optimal control problem. Determine the state $\boldsymbol{x}(\tau)\in\mathbb{R}^n$, control $\boldsymbol{u}(\tau)\in\mathbb{R}^m$, and possibly final time τ_f to minimize the Bolza cost function

$$J=\Gamma\left(\boldsymbol{x}\left(\tau_1\right),\tau_1,\boldsymbol{x}\left(\tau_f\right),\tau_f\right)+\int_{\tau_1}^{\tau_f} L\left(\boldsymbol{x}\left(\tau\right),\boldsymbol{u}\left(\tau\right)\right)d\tau \tag{2-1}$$

subject to dynamic constraints,

$$\dot{\boldsymbol{x}}(\tau) = \boldsymbol{f}\left(\boldsymbol{x}(\tau), \boldsymbol{u}(\tau)\right) \tag{2-2}$$

boundary conditions,

$$\boldsymbol{\gamma}(\boldsymbol{x}(\tau_1), \tau_1, \boldsymbol{x}(\tau_f), \tau_f) = \mathbf{0} \in \mathbb{R}^{\gamma} \tag{2-3}$$

and path constraints,

$$\boldsymbol{q}\left(\boldsymbol{x}(\tau), \boldsymbol{u}(\tau)\right) \leq \mathbf{0} \in \mathbb{R}^{q} \tag{2-4}$$

Among different types of optimal control problems, fixed-final-time problems are those having a known final optimization time τ_f, and hence the cost function should be minimized during a fixed amount of time. One example is a fixed-final-time guidance problem, where a dynamical system is controlled to satisfy the final conditions while maintaining any path and actuator constraints imposed on the system [20].

On the other hand, in a free-final-time problem, the final optimization time and possibly some of the final states are unknown making the optimization problem more challenging. A typical example of a free-final-time problem would be a time-optimal problem in which the system trajectory needs to be developed in a minimum time possible. From the point of view of PMP, free-final-time problems are relatively difficult compared to other types of optimal control problems, since the final optimization time is unknown, and therefore adjoint variables in the associated HBVP problem do not have any boundary conditions. These problems are usually addressed either by searching for switching times or by using phase plane analysis (limited to second order problems) [21], [22]. However, these algorithms are limited to the problem under consideration and usually require the detailed knowledge of the system. An alternative approach is direct transcription of an optimal control problem into an NLP problem, known as a direct method [8]-[15], [19], [23]-[26].

Due to the nature of constraints, usually a mixture of linear/nonlinear algebraic and differential equations, it is often impossible or at least impractical to find an analytical solution to the optimal control problem of Eqs. (2-1)—(2-4). Therefore, numerical approaches are usually applied to solve these problems that are divided into two main classes [3], [4]: indirect and direct methods

2.2 Indirect Methods

Indirect methods apply PMP and calculus of variations to the optimal control problem leading to a HBVP problem solved by numerical methods [5]-[7], [27]. The name of "indirect" comes from the fact that these methods solve the HBVP problem resulted from optimality conditions, thus indirectly solve the optimal control problem. While indirect methods have the high accuracy, and also guarantee to satisfy the first-order optimality conditions, their radii of convergence are small compared to direct methods. Also, they are highly sensitive to the initial guess of adjoint variables making them numerically unreliable, especially for the large scale problems. Moreover, to derive a HBVP problem, it is required to analytically differentiate the expressions for the cost and constraints, which could be highly tedious for the complicated problems.

2.2.1 Indirect Shooting Methods

The shooting method could be considered the most basic indirect method [27]. The first step in the indirect shooting is to make a guess for the unknown boundary conditions at one end of the interval of the optimal control problem. Using the guess and initial conditions, the HBVP can be integrated to the other end of the interval. As the integration reaches the other end of the interval, the terminal conditions found are compared to the known terminal conditions (known terminal conditions are those boundary conditions as well as transversality conditions obtained from the first-order necessary conditions). By setting a maximum tolerance between the two, one can decide how close is acceptable. If the obtained terminal conditions found are not within the maximum tolerance, then the unknown initial conditions can be adjusted, and the integration and comparison process would be repeated. The overall process is repeated until within the maximum tolerance [4]. The simplicity of this method makes writing a code to solve the problem relatively simple as well. A simple "Do-while" loop can be used to loop the guessing and integration while the specified tolerance is not met. Although this method is simple and straightforward, its major difficulty comes from its sensitivity to the initial guess. In fact, as the integration of dynamics moves in either direction of time, errors made in the unknown boundary conditions amplifies [4]. Another problem with the shooting method is that changes have a more substantial impact the earlier they are introduced in the trajectory, due to the integration starting at one end of the interval and ending at the other. As a result, small changes introduced at the beginning can eventually turn into the big nonlinear changes at the other end of the interval [3].

2.2.2 Indirect Multiple Shooting Methods

The multiple shooting method [28], [29], which is a slight variation of the shooting method, is used as an improvement to the shooting method, since it accounts for the expansion and contraction within typical Hamiltonian systems [3], [4]. The multiple shooting method seeks to break the full interval into smaller segments, then apply the shooting method to each segment. Also, a number of additional conditions are introduced to enforce the continuity condition at the joints. The multiple shooting method is a significant improvement in regards to the sensitivity issues with the simple shooting method. However, the size of the problem would be increased as a result of extra variables introduced (values of states and adjoints at the interface points), compared to the simple shooting method.

2.2.3 Indirect Collocation Methods

Indirect collocation [30], also called indirect transcription, is a common method for solving two point boundary value problems (TPBVP). Similar to the multiple shooting method, the collocation method breaks the overall interval into smaller, sub-intervals. Also, in a transcription method, states and controls are parameterized using piecewise polynomials with the polynomial coefficients as unknown parameters found by using appropriate root-finding techniques such as Newton's method [31]. The most common method to satisfy the defect constraint is the Hermite-Simpson method [32]. Collocation methods are most effective when used for solving multipoint boundary value problems, such as simple trajectory optimization. The major downfall to this method, which is the key downfall to all indirect methods as a whole, is that they cannot be applied without solving the adjoint equations. In real-world, complex problems this can be a relatively difficult task. Also, if the inequality path constraints are present, a priori knowledge is required to predetermine the sequence of constrained and unconstrained arcs to correctly formulate the TPBVP.

2.3 Direct Methods

In direct methods, states and/or controls of the optimal control problem are approximated in such an appropriate manner that the original problem can be well transcribed into an NLP problem. The resulting NLP can be solved by gradient-based methods such as sequential quadratic programming or by using heuristic approaches like genetic algorithms and particle swarm optimization techniques [4]. When the state equations are left in the form of original differential equations and only controls are approximated, it is called *control parameterization method* in which the control is parameterized using a specific function, and the cost is evaluated by solving the initial value problem of the system differential equations. One example is direct multiple shooting method [23], where the time interval $[\tau_1 \ \ \tau_f]$ is divided into M subintervals, and the state at the beginning of each subinterval as well as the unknown coefficients in the control parameterization are used as optimization parameters (decision variables) of the NLP. On the other hand, in a *control-state parameterization method*, both state variables and controls are parameterized simultaneously using a specific function with unknown coefficients. While the total parameterization leads to a larger NLP, it is not required to solve the initial value problem at every single iteration of the NLP problem which is a great advantage of the control-state parameterization method.

Direct methods for numerically solving optimal control problems are mostly based on polynomial interpolation whether local or global polynomial(s). There also exist some non-polynomial methods in literature that will be described later in this Chapter.

2.3.1 Direct Shooting and Direct Multiple Shooting Methods

A direct shooting method [33], which is classified as a control parameterization method, is typically much more complex than its indirect counterpart. In the direct shooting, controls are parameterized with a specific function with unknown coefficient and dynamic constraints integrated by using a time-marching algorithm [34], [35]. In a shooting method, the phase propagation of a trajectory is accomplished by using an ordinary differential equations (ODE) initial value method for implicitly defined control, or differential algebraic equations (DAE) initial value method for explicitly defined controls [3]. Direct shooting applications are most successful when used for problems that have a limited number of NLP variables. However, problems that result in a large NLP cannot be succefully solved using a direct shooting approach. Another issue with the direct shooting is its high computational cost of evaluating finite difference gradients for each NLP iteration [3].

Similar to how direct shooting relates to indirect shooting, direct multiple shooting [23] relates to indirect multiple shooting. The time interval is divided into subintervals, in order to create a more detailed NLP compared to simple shooting method. The state values at the beginning of each subinterval as well as the unknown coefficients in the control parameterization are decision variable of the NLP. Direct multiple shooting produces a much larger NLP problem; however, it is still an improvement to the direct shooting method since it does not have as many sensitivity issues. The sensitivity issues are decreased simply because the integration is performed over several smaller intervals versus one large interval, and therefore the error propagation would be limited to each subinterval.

2.3.2 Local Polynomial Methods

One of the most common approaches using control-state parameterization is direct collocation method (local polynomial method), first introduced by Hargraves et al. [24], in which both states and controls are discretized into a set of points (equally or unequally spaced) in time (collocation points or nodes) and segments between each node are approximated with low-order polynomials (usually a cubic polynomial). The parameterization along with the discretization converts the optimal control problem to an NLP problem with the polynomial coefficients as the NLP decision variables. The aforementioned approach is sometimes called direct collocation with nonlinear programming (DCNLP) which has been addressed to solve the constrained optimal control problem in [24], [25]. DCNLP is a well-established method for solving optimal control problems having the local support as well as numerical robustness to the initial guess. However, it suffers from major defects: first of all, it has a low algebraic convergence rate, compared to global polynomial methods with spectral accuracy. In addition, the way both state variables and controls are parameterized in a direct collocation method leads to a very large NLP problem. For the real-time trajectory optimization, it may not be possible to find a fast solution to a large NLP problem.

Some efforts have been done to reduce the size of the NLP resulted from direct transcription by mapping the states and controls to a lower dimensional space instead of directly parameterizing all states and controls. For example, Milam et al. [26] introduced the nonlinear trajectory generation (NTG) method in which states and controls were mapped to a lower dimensional space using differential flatness [36]. They were able to provide a rapid solution to optimal control problems with differentially flat dynamics by parameterizing the flat outputs (lower-dimensional variables in differentially flat systems) with piecewise low-order polynomials represented by B-splines [26].

In splined-based methods, piecewise low-order polynomials are used for the interpolation of states and controls. Trajectory optimization using B-splines and NURBS (non-uniform rational B-splines) are two examples described here.

A. Trajectory Optimization using B-splines

In trajectory optimization with B-splines, the search space for minimizers of an optimal control problem is restricted to the vector space of all piecewise polynomial functions represented by B-spline basis functions. Consequently, the states and controls can be parameterized by B-splines, and an even discretization along the time is performed to completely transcribe the optimal control problem into an NLP problem [9], [26]. One example is the NTG method, designed at Caltech by Mark Milam et al. [26], combining the concept of differential flatness [36] with B-spline representation for direct transcription of an optimal control problem into an NLP problem. The NTG algorithm has three steps:

- Determination of outputs, such that system dynamics can be mapped to a lower dimensional output space

- Parameterization of the outputs in terms of B-spline basis functions
- Even discretization using equally-spaced nodes and transcription of the optimal control problem into an NLP problem

The first step is to find outputs such that Eq. (2-2) can be mapped to a lower dimensional output space. The reason for this transformation is that it would be easier and more efficient to solve a lower dimensional problem. In addition, solving an optimization problem in a lower dimensional space needs a less computational time helping make the real-time trajectory generation feasible. For this purpose, it is needed to find outputs $\boldsymbol{z} = (z_1, z_2, ..., z_m)^T$ of the form

$$\boldsymbol{z} = \mathrm{A}\left(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{u}^{(1)}, \ldots, \boldsymbol{u}^{(r)}\right) \quad \boldsymbol{z} \in \mathbb{R}^m \tag{2-5}$$

such that all the states $\boldsymbol{x}$ and controls $\boldsymbol{u}$ can be recovered from the outputs $\boldsymbol{z}$ and their derivatives:

$$(\boldsymbol{x}, \boldsymbol{u}) = \mathrm{B}(\boldsymbol{z}, \boldsymbol{z}^{(1)}, ..., \boldsymbol{z}^{(s)}) \tag{2-6}$$

These outputs are called *flat outputs*. In general, if the number of outputs $\boldsymbol{z}$ required for representing the states $\boldsymbol{x}$ and controls $\boldsymbol{u}$ is exactly equal to the number of controls (inputs) $\boldsymbol{u}$, the system is differentially flat, and outputs $\boldsymbol{z}$ are called flat outputs [36]. The necessary conditions for the system's differential flatness were discussed in [36], [37]. When the flat outputs are defined, no explicit dynamic constraints remain in the transformed optimal control problem (removal of Eq. (2.2)), since the flat outputs implicitly contain all the information about the dynamics of the system. However, depending on the type of the optimal control problem, system dynamics may not be differentially flat. Also, determining the flat outputs is not always possible even if it is proven to exist [37]. In the case that a flat output cannot be determined or that no flat output exists, the system dynamics in Eq. (2.2) can still be mapped to the lowest dimensional space possible using the outputs $\boldsymbol{z} = \mathrm{A}\left(\boldsymbol{x},\boldsymbol{u},\boldsymbol{u}^{(1)},\ldots,\boldsymbol{u}^{(r_1)}\right)$, $\boldsymbol{z}\epsilon\,\mathbb{R}^{p+m}$. Therefore, $(\boldsymbol{x},\boldsymbol{u})$ will be completely determined by the reduced-order model

$$\begin{aligned}(\boldsymbol{x},\boldsymbol{u}) &= \mathrm{B}_1(\boldsymbol{z},\boldsymbol{z}^{(1)},\cdots,\boldsymbol{z}^{(s_1)})\\ \mathbf{B}_2(\boldsymbol{z},\boldsymbol{z}^{(1)},\cdots,\boldsymbol{z}^{(s_2)}) &= \mathbf{0}\end{aligned} \tag{2-7}$$

In this case, an additional constraint function needs to be added to the optimal control problem. To find a feasible solution, it is assumed that outputs $\boldsymbol{z}$ are in the finite dimensional vector space. The assumption is necessary to avoid seeking for minimizers in the space of all k-times continuously differentiable functions, an infinite dimensional space, making the optimization problem intractable. The space of all piecewise polynomial functions with a prescribed number of polynomial pieces, order, and smoothness has been selected for the NTG method. This space can be well represented by B-splines defined in [38]. Thus, outputs $\boldsymbol{z}$ are parameterized in terms of B-spline basis functions as

$$\begin{aligned}z_1 &= \sum_{i=1}^{q_1} B_{i,\,k_1}\left(\tau\right)\, C_i^1\\ z_2 &= \sum_{i=1}^{q_2} B_{i,\,k_2}\left(\tau\right)\, C_i^2\\ &\vdots\\ z_{p+m} &= \sum_{i=1}^{q_{p+m}} B_{i,\,k_q}\left(\tau\right)\, C_i^{p+m}\\ &\textit{and}\\ q_i &= l_i\left(k_i - s_i\right) + s_i\end{aligned} \tag{2-8}$$

where $B_{i,k_j}(t)$ are the B-spline basis functions with the polynomial order k_i. Also, C_i^j are the coefficients of the B-splines, l_i is the number of piecewise polynomials (knot intervals), s_i is the number of smoothness conditions at the knots (breakpoints), and q_i is the number of coefficients.

A balanced discretization of the optimal control problem is needed to transform the original continuous-time problem into an NLP problem. For this purpose, the collocation points $\tau_1 = t_0 < \ldots < t_{N_t - 1} = \tau_f$ are defined, where N_t is the total number of collocation points. Collocation points are the points in the time interval in

which the constraints are enforced. Finally the integral cost in Eq. (2-1) is approximated by a trapezoidal quadrature rule to transcribe the optimal control problem of Eqs. (2-1)—(2-4) into the following NLP problem:

$$min \quad J = \Gamma\left(z(t_0), t_0, z(t_{N_t-1}), t_{N_t-1}\right) + \sum_{j=0}^{N_t-2}\sum_{i=0}^{N} \alpha_i \mathrm{L}(z(t_i)) \tag{2-9}$$

subject to

$$\begin{aligned} &\mathbf{B}_2\left(z(t_i), z^{(1)}(t_i), \ldots, z^{(s_2)}(t_i)\right) = \mathbf{0} \qquad i = 0,1,\ldots,N_t - 1 \\ &\boldsymbol{\gamma}\left(z(t_0), t_0, z(t_{N_t-1}), t_{N_t-1}\right) = \mathbf{0} \\ &\mathbf{q}\left(z(t_i)\right) \leq \mathbf{0} \qquad\qquad\qquad\quad i = 0,1,\ldots,N_t - 1 \end{aligned} \tag{2-10}$$

where Γ and L are terminal cost and running cost, respectively. Also, α_i are the quadrature weights. The resulting NLP is solved by NPSOL [39], a dense NLP solver, designed at Stanford.

B. *Trajectory Optimization using NURBS Basis Functions*

This method was developed by Melvin Flores et al. [40] at Caltech in 2006. Similar to the NTG, the local approximation is restricted to the vector space of all piecewise polynomial functions, but instead of the B-spline parameterization, a linear combination of non-uniform rational B-splines (NURBS) basis functions is used for the approximation such that,

$$z^{(r)}(\tau) = \sum_{j=0}^{N_C-1} \mathcal{R}_{j,d}^{(r)}\left(\tau, w_0, \ldots, w_{N_C-1}\right) \mathbf{P}_j \tag{2-11}$$

where $z^{(r)}(\tau)$ is the r^{th} time derivative of the interpolating function (states and/or controls), $\mathcal{R}_{j,d}^{(r)}$ is the r^{th} time derivative of the j^{th} NURBS basis function of degree d, depending on time $\tau \in \mathbb{R}$ and weights $w_j \in \mathbb{R}$. Also, $\mathbf{P}_j \in \mathbb{R}^d$ is the j^{th} control point (coefficient of the linear combination), and N_C is the total number of active decision variables. The NURBS functions are expressed in terms of B-splines as

$$\mathcal{R}_{j,d}(\tau) = \frac{B_{j,d}(\tau) w_j}{\sum_{i=0}^{N_c-1} B_{i,d}(\tau) w_i} \tag{2-12}$$

where $w_j > 0$ is the j^{th} weight corresponding to the j^{th} control point. NURBS basis functions are non-negative with local support satisfying partition-of-unity properties. In addition, they depend on two sets of parameters: control points $\mathbf{P}_j$ and weights w_j. In trajectory optimization using NURBS, one set of parameters ($\mathbf{P}_j$) was used to specify a region of space that automatically satisfies the path constraints (removal of path constraints Eq. (2-4)). The rest of parameters (i.e. weights as well as remaining control points) are used as decision variables of the transcribed NLP to find the optimal trajectory. In summary, the trajectory generation for differentially flat systems via NURBS basis functions consists of four steps:

- Rewriting the optimal control problem in terms of the flat outputs and their derivatives to map the system dynamics to a lower dimensional space (removal of dynamic constraints)
- Parameterizing flat outputs with piecewise polynomial functions using a linear combination of NURBS basis functions
- Fixing the control points of parametric NURBS paths in such a way that they describe regions free from obstacles in their respective path spaces and contain their own initial and final path conditions (removal of path constraints)
- Transcribing the modified optimal control problem into an NLP problem with the active weights and control points as decision variables of the NLP

2.3.3 Pseudospectral Methods (Global Polynomial Methods)

Another important class of control-state parameterization methods is the class of *pseudospectral methods*. They were originally developed for computational fluid dynamics applications [41] and extensively used in the optimal control community since 1990s [10]-[15]. In a *pseudospectral method*, states and controls are approximated by Chebyshev or Lagrange polynomials of degree N, and the optimal control problem is discretized using orthogonal collocation points. For instance, Chebyshev-Gauss-Lobatto (CGL) or Legendre-Gauss-Lobatto (LGL) points are two popular choices for discretization nodes in pseudospectral methods [10], [11]. Orthogonal nodes are unequally spaced in the time domain offering higher accuracy of interpolation with fewer nodes, compared to equally-spaced nodes [42]. Also, they take advantage of highly accurate quadrature rule (e.g. Gauss quadrature), used for approximating the integral cost of Eq. (2-1) [43], [44].

The orthogonal nodes are chosen differently based on the family of the orthogonal polynomial and the discretization scheme. Some typical schemes are roots of Legendre polynomials (Legendre-Gauss or LG nodes), extrema of Legendre polynomials (Legendre-Gauss-Lobatto or LGL nodes), roots of Chebyshev polynomials (Chebyshev-Gauss or CG nodes), and extrema of Chebyshev polynomials (Chebyshev-Gauss-Lobatto or CGL nodes). They can also be some other nodes related to the orthogonal polynomials. Pseudospectral methods are usually called *global orthogonal collocation* methods, because they approximate states and controls with a single high-degree polynomial (global approximation) with orthogonally collocated points (e.g. roots of Legendre polynomials). While pseudospectral methods have simple structures and converge exponentially for smooth well-behaved optimal control problems, they have difficulties in finding the solution for nonsmooth problems. In fact, the exponential convergence rate (spectral property) only holds for smooth functions, so the convergence rate is extremely slow for nonsmooth problems, even using a high-degree polynomial [45], [46].

In all polynomial-based methods, whether global or local, the approximation is being limited to the polynomials, so there is no variability among the basis functions. While polynomials have desired properties for function approximation, in some cases, based on the type of the optimal control problem, representing the states and controls with exponential, trigonometric and other types of functions may lead to more accurate and more efficient results than approximating them with polynomials [18].

2.3.4 Neural Network Based Methods

Particular characteristics of neural networks architecture and the way they process information make them superior to many conventional techniques on certain classes of optimization problems. In fact, a wealth of work exists on using neural networks in optimization problems [47]-[60]. The basic principle of using neural networks for trajectory optimization is to model the system dynamics over a small, time period and to chain these segments together in order to build an optimal trajectory. The primitive idea was started by Dennis [47] at MIT in 1957, where he provided a fast solution to a quadratic programming problem by implementing it with analog electrical networks. The idea of using a neural network as a dynamic system whose equilibrium point is the solution to the linear/nonlinear programming problem came out of Dennis's work. Later, Chua and Lin [48] designed an NLP circuit, using the Kuhn-Tucker conditions from mathematical programming theory, to simulate general NLP problems. Tank and Hopfield [49] developed a highly interconnected network of simple analog processors for solving linear programming problems using the computational properties of analog-processor networks. Kennedy and Chua [50] extended the results of Tank and Hopfield method to the general NLP problems. Lillo et al. [51] introduced a continuous nonlinear neural network model architecture based on the penalty method to solve constrained optimization problems. The idea behind the penalty method is to approximate a constrained optimization problem by an unconstrained problem (see [52] for more details). Effati and Baymani [53] proposed a method for quadratic programming, a technique used for solving NLP problems, by using a Hopfield neural network model. The network is treated as a dynamic system converging to the optimal solution of the problem. Reifman and Feldman [54] provided a method of transcribing the NLP problem into an unconstrained form by applying a multilayer feed-forward neural network as a model for system dynamics. The converted problem was then solved with the bisection method. Yeh [55] provided an experimental model for the strength of the concrete mixture, given various ingredients and experimental data, using a neural network and then used the network in an NLP problem to optimize the mix.

Another application of neural networks in optimization problems is the approximation of optimal controllers, known as neural dynamic optimization (NDO). For example, Niestroy [56] presented a method of creating an approximate optimal feedback controller for a constrained nonlinear system via parameterizing the optimal control problem with adaptive weights and biases of a neural network and using an NLP solver to find the optimum values. The optimized weights and biases were used online to control the system. Other examples of designing optimal controllers with the NDO were discussed by Seong and Barrow [57] and Peng et al. [58]. An NDO model is often a multilayer feed-forward sigmoidal neural network generating the optimization parameters online. While the training process is quite fast and the resulting controller is robust to modelling errors, it is not easy to update the controller parameters without retraining the network, in case of changing dynamics and/or cost function.

In [59], Inanc et al. used a neural network to approximate the *signature* and *probability detection* functions of an unmanned aerial vehicle (UAV) for a low-observable trajectory generation under the presence of multiple radars. They used a three-layer feed-forward neural network, in which the first two layers employed a sigmoidal transfer function, and the output layer simply took a weighted sum of its inputs, to approximate tabular data for signature and probability detection of a UAV. Having approximated the signature and probability detection with differentiable functions, a spline-based optimization method was then applied to find a low-observable trajectory. The authors showed that the neural network approximation increased the smoothness of the resulting trajectory, compared to the B-spline approximation of signature and probability detection functions. According to their results, the main reason was the simple and straightforward structure of the neural network requiring fewer parameters than B-splines for the same approximation. In addition, while the neural network approximation performs an unconstrained minimization, spline approximation enforces the continuity conditions at the interfaces, thereby may provide a less smooth interpolating curve.

In an attempt to combine neural networks with direct collocation methods to solve a trajectory optimization problem for UAVs, Geiger et al. [60] used a feed-forward neural network to approximate system dynamics as well as the cost function of an optimal control problem. In their work, the approximating trajectory was discretized into n equal segments, and a feed-forward neural network was then used to recursively approximate states, controls, and cost values on each segment of the trajectory. The optimal trajectory was built recursively and treated as an NLP problem. In each segment, a linear interpolation was used for the control time history

$$u(t) = u_0 + (u_1 - u_0)\frac{t - t_0}{t_1 - t_0}. \tag{2-13}$$

Also, equations of motion were integrated in each segment to find the state values at the end of the segment

$$x(t_0+\tau)=\int_{t_0}^{t_0+\tau} f\left(x(t),u(t)\right)dt. \tag{2-14}$$

As a result, the neural network approximations of states and the cost function were represented as

$$\begin{aligned} &x_{i+1}=Y_d\left(x_i,u_i,u_{i+1}\right) \qquad\qquad i=0,1,\ldots,n-2 \\ &J=\sum_{i=0}^{n-2}J_i=\sum_{i=0}^{n-2}Y_J(x_i,u_i,u_{i+1}) \end{aligned} \tag{2-15}$$

where Y_d , Y_J are neural network approximations of the final state and the cost function in each segment, respectively. Instead of finding both states and controls, this approach only finds the optimal controls using the NLP solver (control parameterization method). It should be noted that the collocation problem is no longer available, since the approximating functions have been replaced by the neural network. The aforementioned neural network method was applied to a UAV surveillance trajectory optimization problem. Authors showed that the approximated trajectory with the neural network was comparable with those generated by direct collocation and pseudospectral methods, while it was required much less computational time because of offline training of the neural network parameters [60].

Based on the literature review, performance of neural networks in constrained optimization problems looks promising. In particular, results of Inanc et al. [59] and Geiger et al. [60] demonstrated successful application of feed-forward neural networks for real-time trajectory generation problems. Some advantages of neural networks in trajectory optimization can be summarizes as follows [61]:

- Trajectory optimization with neural networks is fast, and the calculation time is consistent, thus making the real-time control feasible.
- Neural networks are fault tolerant, hence able to operate in hazardous environments with more reliability.
- Approximating the optimal trajectory can be learned from a few examples of the optimal paths (easy training).
- A neural network based control system can be easily built by integrating neural networks trained for several tasks

CHAPTER III

RBF COLLOCATION METHOD FOR SOLVING OPTIMAL CONTROL PROBLEMS

A new approach based on the RBF interpolation and arbitrary discretization is proposed to solve the optimal control problems. The proposed approach is called the RBF collocation method or simply the RBF method. The RBF method is a direct method combining RBF parameterization of states and controls with proper discretization at arbitrary nodes to transcribe the optimal control problem into an NLP optimization problem with RBF coefficients as decision variables of the NLP. Regardless of the number and type of nodes (discretization scheme), the RBF interpolation for global RBFs is always unique [18]. Therefore, discretization nodes can be arbitrarily chosen in the proposed method for solving optimal control problems (they do not even need to be on a mesh of points, in general). This property makes the RBF method a truly mesh-free method, compared to pseudospectral methods that are being tied to a certain gird of nodes. The ability to arbitrarily select the collocation points is particularly useful for interpolating non-smooth functions, in that case the RBF method can provide more dense points near those discontinuities for the better approximation, a feature that is not possible in a classic pseudospectral method without using mesh refinement techniques [4].

In addition to providing great flexibility in choosing the discretization nodes, the proposed method also offers flexibility in choosing the type of basis functions (trial functions) for parameterization. Instead of being limited to a specific type of local/global polynomial with a certain degree, RBF method uses a broad class of global RBF functions (e.g. Gaussian RBFs, multiquadrics, inverse multiquadrics, etc.) for parameterizing the optimal control problems. Having continuous derivatives, global RBF functions have local support leading to numerical stability for solving optimal control problems.

3.1 Continuous-Time Optimal Control Problem

Without loss of generality, consider the following continuous-time optimal control problem. Determine the state $\boldsymbol{x}(\tau) \in \mathbb{R}^n$, control $\boldsymbol{u}(\tau) \in \mathbb{R}^m$, and possibly final time t_f to minimize the cost function

$$J = \Gamma\left(\boldsymbol{x}(-1), t_0, \boldsymbol{x}(1), t_f\right) + \frac{t_f - t_0}{2} \int_{-1}^{1} L\left(\boldsymbol{x}(\tau), \boldsymbol{u}(\tau)\right) d\tau \tag{3-1}$$

subject to dynamic constraints,

$$\dot{\boldsymbol{x}}(\tau) = \frac{t_f - t_0}{2} \boldsymbol{f}\left(\boldsymbol{x}(\tau), \boldsymbol{u}(\tau)\right) \tag{3-2}$$

boundary conditions,

$$\boldsymbol{\gamma}(\boldsymbol{x}(-1), t_0, \boldsymbol{x}(1), t_f) = \mathbf{0} \in \mathbb{R}^{\gamma} \tag{3-3}$$

and path constraints,

$$\boldsymbol{q}\left(\boldsymbol{x}(\tau), \boldsymbol{u}(\tau)\right) \leq \mathbf{0} \in \mathbb{R}^{q}. \tag{3-4}$$

The optimal control problem of Eqs. (3-1)—(3-4) can be transformed form the time interval $\tau \in [-1,1]$ to the time interval $\mathrm{t} \in [t_0, t_f]$ using an affine transformation,

$$t = \frac{t_f - t_0}{2}\tau + \frac{t_f + t_0}{2} \tag{3-5}$$

where t_0 and t_f are initial and final optimization time, respectively.

3.2 RBF Introduction and Definition

RBF functions were first studied by Rolland Hardy [62] in 1971 and then introduced by Kansa [63], [64] for solving partial differential equations using the collocation method in 1990. RBF functions were originally developed for the function approximation. They are naturally defined as multivariate functions, so they can be applied almost in any dimension [18]. An RBF is a real-valued function whose value depends only on the distance from a fixed point, called center point or center,

$$\varphi(\boldsymbol{x}, \boldsymbol{c}) = \varphi\left(\|\boldsymbol{x} - \boldsymbol{c}\|\right) \tag{3-6}$$

where φ is the RBF and $\boldsymbol{c}$ is the center. The norm $\| \ \|$ is usually the Euclidean norm, but other distance functions could be used, as well. Any function satisfying Eq. (3-6) is called an RBF function. Some classical RBFs are shown in Table 3.1.

In general, the RBF could be piecewise smooth like Polyharmonic Splines (PS) or infinitely smooth (global RBF) such as Gaussian (GA), multiquadric (MQ), or inverse multiquadric (IMQ) RBFs. Infinitely smooth RBFs usually have a shape parameter ε to tune the overall shape of the RBF. Increasing ε produces sharper RBFs while decreasing it leads to more flat RBFs.

Table 3.1 Classical RBFs, $r = r_i = \|\boldsymbol{x} - \boldsymbol{c}_i\|$

Name of the RBF	$\varphi(r)$
Gaussian (GA)	$e^{-(\varepsilon r)^2}$, $\varepsilon > 0$
Multiquadric (MQ)	$\sqrt{1 + (\varepsilon r)^2}$, $\varepsilon > 0$
Inverse quadrics (IQ)	$1/(1 + (\varepsilon r)^2)$, $\varepsilon > 0$
Inverse multiquadric (IMQ)	$1/(\sqrt{1 + (\varepsilon r)^2})$, $\varepsilon > 0$
Polyharmonic splines (PS)	$\begin{cases} r^k & \text{for } k = 1,3,5,... \\ r^k \ln(r) & \text{for } k = 2,4,6,... \end{cases}$

RBFs are typically used for the function approximation of the form

$$f(\boldsymbol{x}) = \sum_{i=1}^{N} \omega_i \, \varphi\left(\|\boldsymbol{x} - \boldsymbol{c}_{\mathrm{i}}\|\right) \tag{3-7}$$

or in the vector form

$$\begin{aligned} & f(\boldsymbol{x}) = \boldsymbol{\varphi}_x^T \, \boldsymbol{\omega} \\ & \boldsymbol{\varphi}_x^T = \left(\varphi\left(\|\boldsymbol{x} - \boldsymbol{c}_1\|\right) \varphi\left(\|\boldsymbol{x} - \boldsymbol{c}_2\|\right) ... \, \varphi\left(\|\boldsymbol{x} - \boldsymbol{c}_N\|\right)\right) \\ & \boldsymbol{\omega} = \left(\omega_1 \, \omega_2 \, ... \omega_N\right)^T \end{aligned} \tag{3-8}$$

where the interpolating function $f(\boldsymbol{x})$ is represented as a sum of N RBFs φ, each associated with an RBF center $\boldsymbol{c}_i \in \mathbb{R}^n$, and weighted by an appropriate RBF weight $\omega_i \in \mathbb{R}$.

3.3 RBF Collocation Method

The RBF collocation method for solving optimal control problems is based on interpolating global RBFs on arbitrary collocation points. To provide a more flexible framework, various sets of collocation points including equally and unequally spaced nodes could be arbitrarily chosen for discretization. For example, a set of Chebyshev-Gauss (CG), Chebyshev-Gauss-Lobatto (CGL), Legendre-Gauss (LG), and Legendre-

Gauss-Lobatto (LGL) nodes, each could be selected as a set of unequally-spaced orthogonal nodes to discretize the problem [44], [46]. Similarly, a set of equally-spaced nodes in the time can also be employed for discretization. Two most popular choices are sets of CGL and LGL nodes distributed over the interval $[-1,1]$ including both -1 and 1. The former set minimizes the max-norm of the interpolation error, while the latter one minimizes the L^2-norm of the interpolation error. CGL and LGL nodes, shown by τ_j, $j=1,2,...,N$, are zeros of the derivative of Chebyshev and Legendre polynomials of degree N-1, respectively (also add $\tau_1=-1$, $\tau_N=1$ to obtain the whole set).

Now, consider the optimal control problem of Eqs. (3-1)—(3-4). The state $\boldsymbol{x}(\tau)$ and control $\boldsymbol{u}(\tau)$ are approximated using N RBF functions as

$$\boldsymbol{x}(\tau)\approx\boldsymbol{x}^R(\tau)=\sum_{i=1}^{N}\boldsymbol{\alpha}_i\,\varphi\left(\left\|\tau-\tau_i\right\|\right)=\sum_{i=1}^{N}\boldsymbol{\alpha}_i\,\varphi_i(\tau) \tag{3-9}$$

$$\boldsymbol{u}(\tau)\approx\boldsymbol{u}^R(\tau)=\sum_{i=1}^{N}\boldsymbol{\beta}_i\,\varphi\left(\left\|\tau-\tau_i\right\|\right)=\sum_{i=1}^{N}\boldsymbol{\beta}_i\,\varphi_i(\tau) \tag{3-10}$$

where $\boldsymbol{x}^R(\tau)$ and $\boldsymbol{u}^R(\tau)$ denote the RBF interpolation of $\boldsymbol{x}(\tau)$ and $\boldsymbol{u}(\tau)$, respectively. Also, $\varphi_i(\tau)$ is the RBF and $\boldsymbol{\alpha}_i$, $\boldsymbol{\beta}_i$ are RBF weights related to $\boldsymbol{x}^R(\tau)$, $\boldsymbol{u}^R(\tau)$, respectively. Differentiating the expression in Eq. (3-9) with respect to τ yields

$$\dot{\boldsymbol{x}}(\tau)\approx\dot{\boldsymbol{x}}^R(\tau)=\sum_{i=1}^{N}\boldsymbol{\alpha}_i\,\dot{\varphi}\left(\left\|\tau-\tau_i\right\|\right)=\sum_{i=1}^{N}\boldsymbol{\alpha}_i\,\dot{\varphi}_i(\tau) \tag{3-11}$$

For the purpose of clarity and brevity, the RBF method is derived for the GA RBFs and LGL nodes here. The procedure would be similar for other types of RBFs and nodes. GA RBFs are represented as

$$\varphi_i(\tau)=\varphi\left(\left\|\tau-\tau_i\right\|\right)=\exp(-\varepsilon^2(\tau-\tau_i)^2) \tag{3-12}$$

Without loss of generality, assume $\varepsilon=1$. Now, substituting Eq. (3-12) in Eq. (3-11) and evaluating at LGL nodes results in

$$\dot{\boldsymbol{x}}(\tau_j)\approx\dot{\boldsymbol{x}}^R(\tau_j)=\left.\frac{d\boldsymbol{x}^R}{d\tau}\right|_{\tau=\tau_j}=\sum_{i=1}^{N}D_{ji}\,\boldsymbol{\alpha}_i \qquad j=1,2,...,N \tag{3-13}$$

where $D_{ji}=\dot{\varphi}_i(\tau_j)$ are entries of $N\times N$ GA differentiation matrix $\boldsymbol{D}$

$$\boldsymbol{D}:=[D_{ji}]:=\begin{cases}-2(\tau_j-\tau_i)\varphi_i(\tau_j) & i\neq j\\ 0 & i=j\end{cases} \tag{3-14}$$

and

$$\varphi_i(\tau_j)=\varphi\left(\left\|\tau_j-\tau_i\right\|\right)=\exp(-(\tau_j-\tau_i)^2)=\varphi_j(\tau_i). \tag{3-15}$$

Therefore, dynamic constraints of Eq. (3-2) are transcribed into the following algebraic equations:

$$\sum_{i=1}^{N}D_{ji}\boldsymbol{\alpha}_i-\frac{t_f-t_0}{2}\boldsymbol{f}\left(\boldsymbol{\alpha}_i,\boldsymbol{\beta}_i,\varphi_i(\tau_j)\right)=\mathbf{0} \qquad j=1,2,...,N \tag{3-16}$$

Next, the continuous cost function of Eq. (3-1) is approximated by the Gauss-Lobatto quadrature as

$$J=\Gamma\left(\boldsymbol{\alpha}_i,\varphi_i(-1),\,\varphi_i(1),t_0,\,t_f\right)+\frac{t_f-t_0}{2}\sum_{j=1}^{N}w_j\,L\left(\boldsymbol{\alpha}_i,\boldsymbol{\beta}_i,\varphi_i(\tau_j)\right) \tag{3-17}$$

where w_j are LGL weights corresponding to LGL nodes $\tau_j\in[-1,1]$, given by

$$w_j = \frac{2}{N(N-1)[P_{N-1}(\tau_j)]^2} \qquad j = 1,2,...,N \tag{3-18}$$

where P_{N-1} is the Legendre polynomial of degree $N-1$. Evaluating the path constraint of Eq. (3-4) at LGL nodes and applying Eqs. (3-16)—(3-17), the optimal control problem of Eqs. (1-4) is discretized into the following NLP problem:

Find $\boldsymbol{A} = (\boldsymbol{\alpha}_1\ \boldsymbol{\alpha}_2\ ...\ \boldsymbol{\alpha}_N)^T{}_{N\times n}$, $\boldsymbol{B} = (\boldsymbol{\beta}_1\ \boldsymbol{\beta}_2\ ...\ \boldsymbol{\beta}_N)^T{}_{N\times m}$, and possibly t_f, to minimize the cost function of

$$J = \Gamma\left(\boldsymbol{\alpha}_i, \varphi_i(-1),\ \varphi_i(1), t_0,\ t_f\right) + \frac{t_f - t_0}{2}\sum_{j=1}^{N} w_j\ L\left(\boldsymbol{\alpha}_i\ , \boldsymbol{\beta}_i\ , \varphi_i(\tau_j)\right) \tag{3-19}$$

subject to

$$\begin{aligned} &\boldsymbol{d}_j - \frac{t_f - t_0}{2}\boldsymbol{f}\left(\boldsymbol{\alpha}_i\ , \boldsymbol{\beta}_i\ , \varphi_i(\tau_j)\right) = \boldsymbol{0} \\ &\boldsymbol{\gamma}\left(\boldsymbol{\alpha}_i, \varphi_i(-1),\ \varphi_i(1), t_0,\ t_f\right) = \boldsymbol{0} \qquad j = 1,2,...,N \\ &\boldsymbol{q}\left(\boldsymbol{\alpha}_i\ , \boldsymbol{\beta}_i\ , \varphi_i(\tau_j)\right) \le \boldsymbol{0} \end{aligned} \tag{3-20}$$

where $\boldsymbol{d}_j = \sum_{i=1}^{N} D_{ji}\boldsymbol{\alpha}_i$. The $N \times N$ RBF Gram matrix $\boldsymbol{\Phi}$ is a symmetric matrix defined as

$$\boldsymbol{\Phi} = \begin{bmatrix} \varphi_1(-1) & \varphi_2(-1) & \cdots & \varphi_N(-1) \\ \varphi_1(\tau_2) & \varphi_2(\tau_2) & \cdots & \varphi_N(\tau_2) \\ \vdots & \vdots & \ddots & \vdots \\ \varphi_1(1) & \varphi_2(1) & \cdots & \varphi_N(1) \end{bmatrix} \tag{3-21}$$

that results in

$$\boldsymbol{A} = \boldsymbol{\Phi}^{-1}(\boldsymbol{x}^R)^T \quad , \qquad \boldsymbol{B} = \boldsymbol{\Phi}^{-1}(\boldsymbol{u}^R)^T \tag{3-22}$$

where

$$\begin{aligned}\boldsymbol{x}^R &= [\boldsymbol{x}^R(-1)\ \ \boldsymbol{x}^R(\tau_2)\ \dots\ \boldsymbol{x}^R(1)] \\ \boldsymbol{u}^R &= [\boldsymbol{u}^R(-1)\ \ \boldsymbol{u}^R(\tau_2)\ \dots\ \boldsymbol{u}^R(1)].\end{aligned} \tag{3-23}$$

For infinitely smooth RBFs with nonsingular matrix $\boldsymbol{\Phi}$, including GA, IQ, and IMQ RBFs, a unique solution can be readily obtained for the aforementioned NLP. Also, for RBFs with positive semi-definite $\boldsymbol{\Phi}$ such as MQ RBFs, the number of nodes as well as RBF centers can be selected in such a way to uniquely approximate the optimal trajectory.

Unlike pseudospectral methods being tied to a specific set of nodes related to the family of the global polynomial, it is emphasized that the RBF collocation method uses a set of arbitrary nodes. Also, any type of global RBF can be arbitrarily chosen for parameterization, instead of a specific type of polynomial with a certain degree as in polynomial-based methods. Deriving the RBF method for the GA RBFs and LGL nodes is an example to show how the method works. Clearly, other types of global RBFs including IQ, MQ, and IMQ can also be applied for parameterization, and various types of collocation points including CG, LG, CGL and even equally-spaced nodes can be arbitrarily chosen for discretization (discretization nodes do not even need to be on meshes of points, in general). Since the integral nodes are assumed to be the same as the discretization nodes, the quadrature rule needed for approximating the integral cost of Eq. (3-1) needs to vary according to the type of nodes used for discretization. For example, if the equally-spaced nodes are employed, the Gauss-Lobatto quadrature rule of Eq. (3-17) needs to be replaced by an appropriate quadrature rule such as trapezoidal or Simpson’s rule. Similarly, the Clenshaw-Curtis quadrature should be used for numerical integration in case of using CGL discretization nodes. For more information about the quadrature rules and related nodes see [65].

The described solution is called the *RBF collocation method*, developed as a set of MATLAB functions to transcribe the optimal control problem of Eqs. (3-1)—(3-4) into an NLP optimization problem, and then call SNOPT [66], a sparse NLP solver, to find the optimal trajectory.

3.4 Optimality Conditions

The Lagrangian of the Hamiltonian (augmented Hamiltonian), for the optimal control problem of Eqs. (3-1)—(3-4) is given by

$$\bar{H}(\boldsymbol{\mu},\boldsymbol{\lambda},\boldsymbol{x},\boldsymbol{u},\tau) = L(\boldsymbol{x},\boldsymbol{u},\tau) + \boldsymbol{\lambda}^T(\tau)\boldsymbol{f}(\boldsymbol{x},\boldsymbol{u},\tau) + \boldsymbol{\mu}^T(\tau)\boldsymbol{q}(\boldsymbol{x},\boldsymbol{u},\tau) \tag{3-24}$$

where $\boldsymbol{\lambda}(\tau) \in \mathbb{R}^n$ is the costate and $\boldsymbol{\mu}(\tau) \in \mathbb{R}^q$ is the Lagrange multiplier associated with the path constraint. The first-order optimality conditions for the continuous-time optimal control problem are derived as

$$\begin{aligned}
&\dot{\boldsymbol{x}} = \frac{d\boldsymbol{x}}{d\tau} = \frac{t_f - t_0}{2}\boldsymbol{f}(\boldsymbol{x},\boldsymbol{u},\tau) = \frac{t_f - t_0}{2}\frac{\partial \bar{H}}{\partial \boldsymbol{\lambda}} \\
&\dot{\boldsymbol{\lambda}} = \frac{d\boldsymbol{\lambda}}{d\tau} = -\frac{t_f - t_0}{2}\left(\frac{\partial L}{\partial \boldsymbol{x}} + \boldsymbol{\lambda}^T\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}} + \boldsymbol{\mu}^T\frac{\partial \boldsymbol{q}}{\partial \boldsymbol{x}}\right) = -\frac{(t_f - t_0)}{2}\frac{\partial \bar{H}}{\partial \boldsymbol{x}} \\
&\frac{\partial L}{\partial \boldsymbol{u}} + \boldsymbol{\lambda}^T\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}} + \boldsymbol{\mu}^T\frac{\partial \boldsymbol{q}}{\partial \boldsymbol{u}} = \frac{\partial \bar{H}}{\partial \boldsymbol{u}} = \boldsymbol{0} \\
&\boldsymbol{\gamma}(\boldsymbol{x}(\tau_1), t_0, \boldsymbol{x}(\tau_f), t_f) = \boldsymbol{0} \\
&\boldsymbol{\lambda}(\tau_1) = -\left.\frac{\partial \Gamma}{\partial \boldsymbol{x}}\right|_{\tau=\tau_1} - \boldsymbol{\upsilon}^T\left.\frac{\partial \gamma}{\partial \boldsymbol{x}}\right|_{\tau=\tau_1} \\
&\boldsymbol{\lambda}(\tau_f) = \left.\frac{\partial \Gamma}{\partial \boldsymbol{x}}\right|_{\tau=\tau_f} + \boldsymbol{\upsilon}^T\left.\frac{\partial \gamma}{\partial \boldsymbol{x}}\right|_{\tau=\tau_f} \\
&\mu_j = \begin{cases} 0 \\ \geq 0 \end{cases} \quad if \quad \begin{matrix} q_j(\boldsymbol{x},\boldsymbol{u}) < 0 \\ q_j(\boldsymbol{x},\boldsymbol{u}) = 0 \end{matrix} \qquad j = 1,2,...,q
\end{aligned} \tag{3-25}$$

where $\boldsymbol{\upsilon} \in \mathbb{R}^q$ is the Lagrange multiplier associated with the boundary condition $\boldsymbol{\gamma}$. Necessary conditions of Eq. (3-25) are used as a test on the optimality of a candidate solution $(\boldsymbol{x}^*, \boldsymbol{u}^*)$ after a successful run of the RBF method. It is emphasized that these conditions are necessary (not sufficient) for the optimality, and hence used to verify the feasibility of the solution.

3.5 Numerical Examples

In this section, the RBF method is applied to four different examples. The first one is a temperature control problem [67] where it is desired to heat a room using the least amount of energy. The second example being considered is the brachistochrone problem in which the goal is to find the shape of the curve when a bead, sliding from rest and accelerated by gravity, slips from one point to another in the least amount of time. For the temperature control problem and brachistochrone example, the exact solutions are available from analytical approaches, so it is possible to make a comparison between the numerical result and the exact solution to evaluate the accuracy of the method. Then, the RBF method is applied to the Vanderpol example, and the results are compared with a spline-based method (NTG) [26] for the accuracy of the solution and with a Legendre pseudospectral method (DIDO) [68] for the computational efficiency. Finally, the proposed method is applied to a motion planning problem with obstacle avoidance in a 2D space. The RBF solution is compared with the numerical solutions obtained from PROPT [69] and DIDO, two commercially available optimal control software packages, to evaluate the performance of the proposed method.

3.5.1 Temperature Control Problem

Consider the optimal control problem, minimize the cost function

$$J = \frac{1}{2}\int_0^1 u^2 \, dt \tag{3-26}$$

subject to the system equation and boundary conditions

$$\begin{aligned} \dot{x}(t) &= \text{-}3\,x(t) + 2\,u(t) \\ x(0) &= 0 \quad , \quad x(1) = 10 \end{aligned} \tag{3-27}$$

where $u(t)$ is the rate of heat supply to the room and $x(t)$ is the difference between the room temperature and the ambient temperature outside. The problem was considered in [67]. To control the room temperature on the time interval $[0, 1]$ with the least possible energy, it is needed to find the optimum value for the control $u(t)$ to minimize the cost function of Eq. (3-26) subject to the constraints of Eq. (3-27). The exact solutions for $x(t)$ and $u(t)$ can be obtained from an analytical approach as

$$x(t) = 10\frac{\sinh 3t}{\sinh 3} \quad , \quad u(t) = 15\frac{e^{3t}}{\sinh 3} \tag{3-28}$$

The RBF collocation method with three different RBFs, including GA RBFs, IMQs, and third-order PS is applied to transcribe the Eqs. (3-26), (3-27) into the NLP problem. The NLP is solved by SNOPT with default feasibility and optimality tolerances, called from MATLAB via a mex interface. The number of RBF centers are defined by the user in advance based on the number of desired RBF weights (optimization parameters). Also, RBF centers are distributed in the time domain $[0, 1]$ based on the type of collocation points. For the GA RBFs and IMQ, the cost value will be tuned using the shape parameter ε. In addition, the type and total number of collocation points are changed to optimize the cost value. The results were summarized in Table 3.2. The calculations were performed in MATLAB (version 8.3) on an Intel Pentium 2.4 GHz machine running Windows 8.1. The exact value of the cost function obtained from the analytical method is $J \approx 75.186$. According to the results, overall, all three RBFs perform very well to minimize the cost function. As expected, increasing the number of RBF weights leads to more accurate approximation and hence less cost value at the expense of increased computation time. Figure 3.1 shows the exact and approximated values of $x(t)$ and $u(t)$ using three different RBFs (GA, IMQ, and PS), each with 7 RBF weights and 21 nodes.

Table 3.2 Cost function, shape parameter ε, and number of weights for three types of RBFs used in the temperature control example

RBF type	Weights	ε	Comput. Time (s)	Cost
GA	3	4	0.12	76.610
	7	0.55	0.19	75.242
	25	0.17	0.63	75.189
IMQ	3	5	0.11	76.674
	7	1.2	0.16	75.243
	25	0.2	1.43	75.191
PS	3	—	0.09	75.889
	7	—	0.16	75.238
	25	—	1.00	75.194

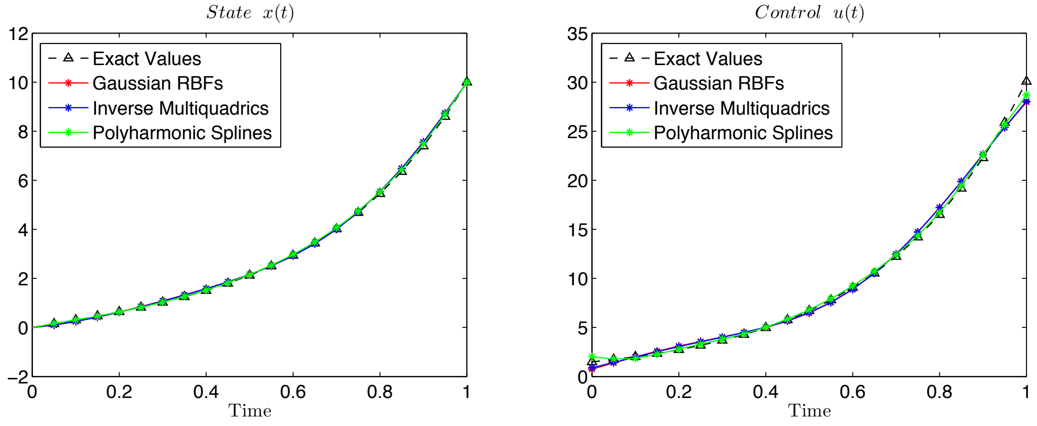


Figure 3.1 Exact and approximated values of $x(t)$ and $u(t)$ using three different RBFs, each with 7 weights and 21 nodes, temperature control example

3.5.2 Brachistochrone Problem

The proposed method is applied to the brachistochrone problem where it is desired to find the shape of the curve when a bead, sliding from rest and accelerated by gravity, slips from one point to another in the least amount of time. The optimal control problem is formulated as to minimize the final optimization time τ_f subject to dynamic constraints

$$\begin{aligned} \dot{x} &= v\sin\theta \\ \dot{y} &= -v\cos\theta \\ \dot{v} &= g\cos\theta \end{aligned} \tag{3-29}$$

and boundary conditions

$$\begin{aligned} x(0) &= 0 \\ y(0) &= 0 \\ v(0) &= 0 \\ x(\tau_f) &= 10 \\ y(\tau_f) &= -3 \\ v(\tau_f) &= \text{Free} \end{aligned} \tag{3-30}$$

25 Brachistochrone is a time-optimal control problem with an analytical solution. It turns out that the optimal trajectories are the equations of a cycloid with the control θ as a linear function of time.

The RBF method with GA, MQ, and IMQ RBFs along with the LGL nodes is used to solve the problem. The number of nodes N is set to be 5, 10, and 25 for each type of RBF. Table 3.3 shows the minimum cost as well as the computation time of the RBF method for different number of nodes (for $g = 9.81$).

Table 3.3 Comparison of cost and computation time of GA, MQ, and IMQ RBFs for the brachistochrone problem for *N*=[5, 10, 25]

Method	*N*	*J*	$\lvert J\text{-}J_{\text{analytic}}\rvert$	Time (s)
Analytic Solution		1.878940330		
GA	5	1.877877421	0.001062909	0.47
MQ	5	1.877983233	0.000957097	0.39
IMQ	5	1.879932113	0.000991783	0.36
GA	10	1.878940392	0.000000062	1.28
MQ	10	1.878940342	0.000000012	1.37
IMQ	10	1.878940412	0.000000082	1.22
GA	25	1.878940336	0.000000006	6.82
MQ	25	1.878940331	0.000000001	7.19
IMQ	25	1.878940332	0.000000002	6.83

As expected, higher accuracy is obtained by increasing the number of nodes, at the expense of increased computation time. While the IMQ RBF is computationally more efficient for this numerical example, the MQ RBF shows more accurate results. Comparing the minimum cost values with the analytical solution reveals that overall, the RBF method has an acceptable accuracy even for a low number of discretization nodes ($N = 5$). Figure 3.2 shows the solutions obtained from the MQ RBF method for the states and control against the exact solutions for $N = 25$. The maximum absolute errors in the states and control (over all nodes) for $N = 25$ are approximately 4.5×10^{-10} and

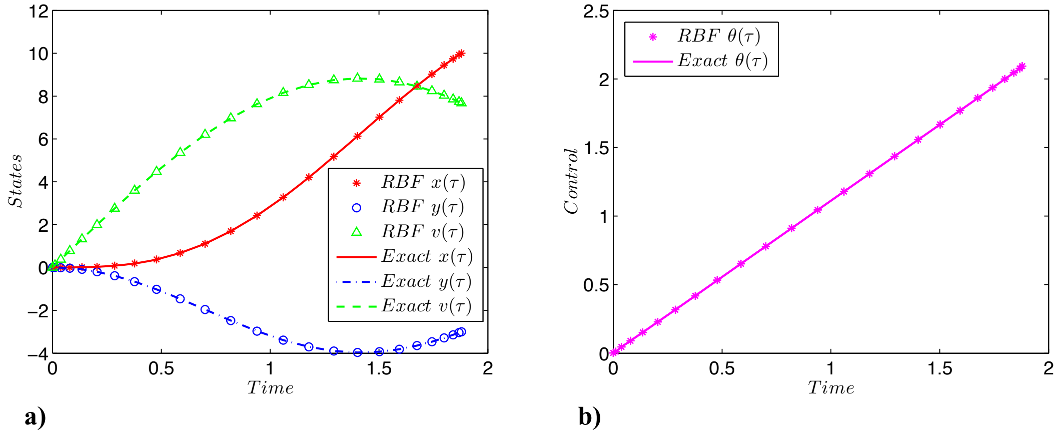


Figure 3.2 Solution to the brachistchrone example using MQ RBFs for $N = 25$
a) states x, y, v b) control

2.6×10^{-10}, respectively. The Hamiltonian for the brachistochrone problem can be written as

$$H(\boldsymbol{\lambda},\boldsymbol{x},u)=\lambda_x v\sin\theta+\lambda_y v\cos\theta+\lambda_v g\cos\theta \qquad \text{(3-31)}$$

where $\boldsymbol{\lambda}=[\lambda_x,\lambda_y,\lambda_v]$, $\boldsymbol{x}=[x,y,v]$, and $u=\theta$. Applying the necessary conditions of optimality from Eq. (3-25) holds that Hamiltonian value must be -1, and costates λ_x and λ_y should be constants, i.e.

$$\begin{aligned}&H(\boldsymbol{\lambda},\boldsymbol{x},u)=-1\\&\lambda_x=\mathrm{a}\ ,\ \lambda_y=\mathrm{b}\quad ,\ \mathrm{a,b}\ \text{ are constants.}\end{aligned} \qquad \text{(3-32)}$$

Figure 3.3.a shows estimated costates alongside the exact values of costates for $N=25$. The Hamiltonian derived from estimated costates is demonstrated in Figure 3.3.b. It is evident from the graphs that results are in complete agreement with Eq. (3-32). The maximum absolute error in the costates for $N=25$ is approximately 2.5×10^{-8}. This demonstrates that costates are accurate enough to partially verify the optimality of the solution.

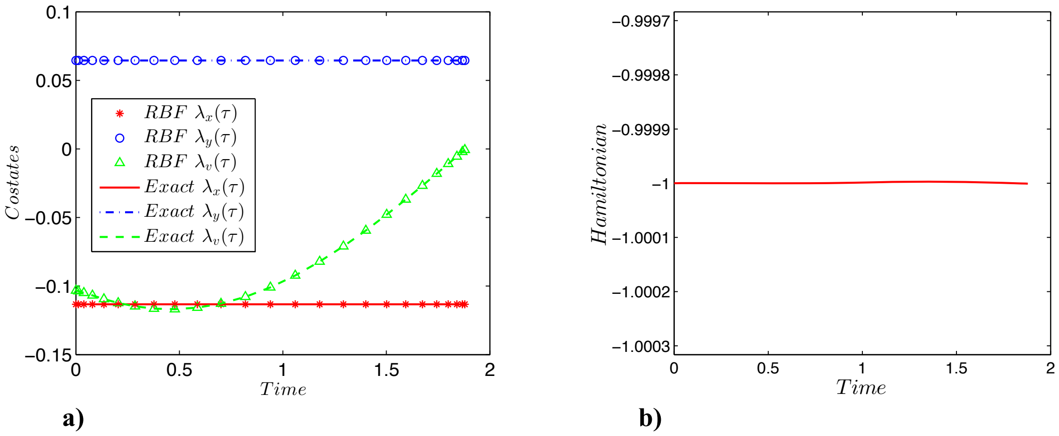


Figure 3.3 Costates and Hamiltonian for the brachistochrone example for $N=25$
a) costates $\lambda_x,\lambda_y,\lambda_v$ b) Hamiltonian H

3.5.3 Vanderpol Problem

The third problem under consideration is the Vanderpol oscillator. A comparison will be made between the RBF method and two other optimal control software packages, NTG and DIDO. NTG is a local polynomial method based on the B-Spline approximation developed in [26], whereas DIDO is a global polynomial method based on the Legendre polynomials approximation developed in [68], [70]. The optimal control problem for the Vanderpol oscillator is written as

$$\min_{x_1, x_2, u} J = \frac{1}{2}\int_0^5 x_1^2 + x_2^2 + u^2 \, dt \tag{3-33}$$

subject to dynamic and boundary constraints

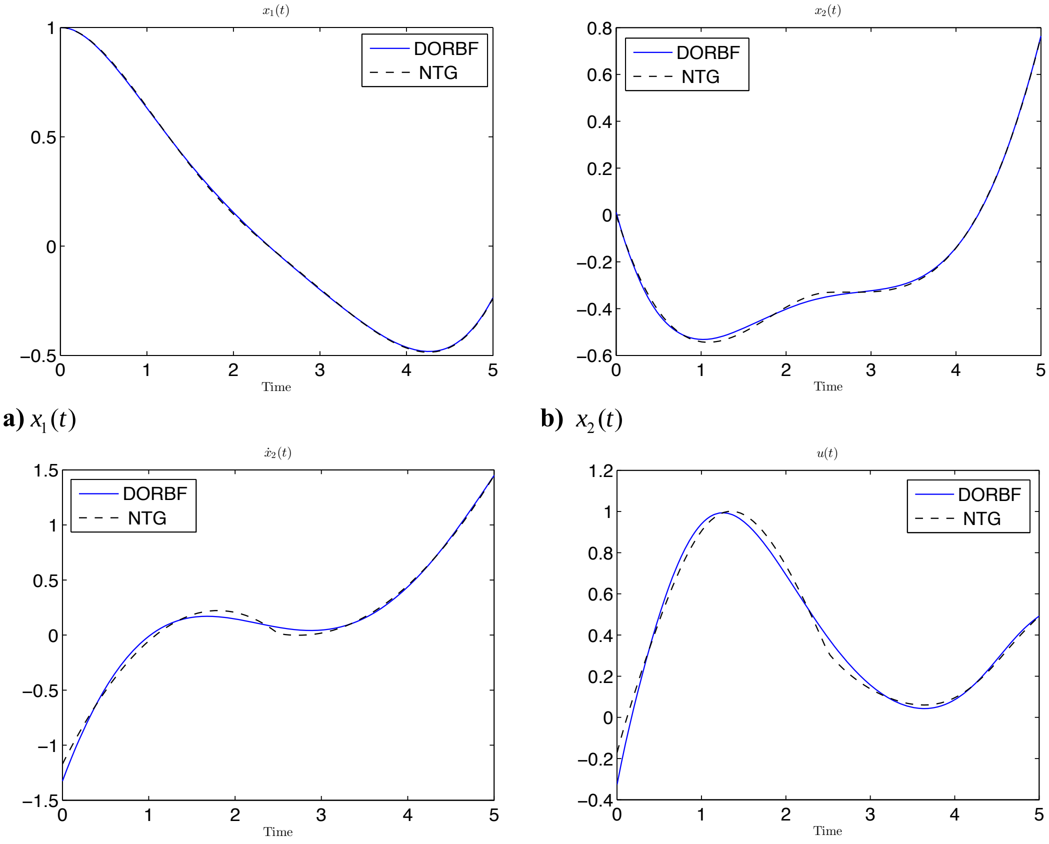
$$\begin{aligned} &\dot{x}_1(t) = x_2(t) \\ &\dot{x}_2(t) = -x_1(t) + (1 - x_1^2(\mathrm{t}))\, x_2(t) + u(t) \\ &x_1(0) = 1 \;\;,\;\; x_2(0) = 0 \;\;,\;\; -x_1(5) + x_2(5) = 1 \end{aligned} \tag{3-34}$$

For the RBF method, GA RBFs with equally-spaced centers are used for the approximation. Also LG nodes are used as the collocation points. For NTG, the smoothness and order of B-splines are considered to be three and five, respectively as in [26]. NTG needs the number of time intervals for the local approximation, so it is set to 1, 2, and 15 intervals corresponding to 5, 7, and 33 coefficients (optimization parameters), respectively. Also, the number of collocation points is set to be four times the number of coefficients as in [26]. According to Table 3.4, as the number of coefficients increases, the cost function value decreases, leading to more accurate results. In addition, considering the equal number of coefficients, the proposed approach shows better performance in minimizing the cost function than NTG (higher accuracy). In fact, the

RBF method has the flexibility to tune the RBF functions by changing the shape parameter ε to get the best cost value for each case. For example, for 7 coefficients, the cost would be $J = 1.6902$ for $\varepsilon = 4$, also $J = 1.69412$ for $\varepsilon = 5$, and the lowest cost is $J = 1.6882$ for $\varepsilon = 5.5$. Figure 3.4 shows $x_1(t)$, $x_2(t)$, $\dot{x}_2(t)$, and $u(t)$ for the RBF method and NTG, each with 7 coefficients.

Table 3.4 RBF method and NTG cost and computation time comparison, Vanderpol example

Method	Coefficients	Cost	Time (s)
RBF	5	1.9025	0.12
NTG	5	1.9127	0.04
RBF	7	1.6882	0.21
NTG	7	1.6982	0.04
RBF	33	1.6858	0.99
NTG	33	1.6859	0.1



a) $x_1(t)$

b) $x_2(t)$

c) $\dot{x}_2(t)$

d) $u(t)$

Figure 3.4 $x_1(t)$, $x_2(t)$, $\dot{x}_2(t)$, **and** $u(t)$ **for the Vanderpol example using the RBF method and NTG each with 7 coefficients**

As seen in Figure 3.4, especially in the bottom two graphs for $\dot{x}_2(t)$ and $u(t)$, the RBF method provides the smoother curve with the same number of optimization parameters. According to Table 3.4, the computation time of the NTG method is faster than the RBF approach, since the proposed method runs in MATLAB that is not as fast as NTG's compiled code. To make a fair comparison in terms of computation time, the RBF collocation method is compared with DIDO, another MATLAB-based tool.

In DIDO, the number of optimization parameters is automatically set by the program, and the user does not have any control on the precision of the algorithm. Therefore, it is not possible to compare the cost values of DIDO and the RBF method for the equal number of optimization parameters. However, the number of collocation points is adjustable, making it possible to compare the computation time of those methods for similar cost values. For DIDO, the number of collocation points is increased from one, and the cost value, along with the computation time, is recorded for each set of nodes. For the RBF method, GA RBFs and LG nodes are used for parameterization and collocation points, respectively. Also, the number of collocation points is changed between 5 to 20 times the number of coefficients to get the cost value similar to DIDO cost. The results are summarized in Table 3.5.

Table 3.5 Computation time comparison between RBF method and DIDO, Vanderpol example

Method	nodes (N_T)	Cost	Time (s)
RBF	25	1.9603	0.11
DIDO	4	1.9604	0.70
RBF	80	1.6875	0.28
DIDO	8	1.6873	1.17
RBF	220	1.6857	0.59
DIDO	10	1.6857	1.48

According to the results, the proposed method has better computation time for similar cost values. For example, the RBF collocation method is about 84% faster than DIDO for the cost value of $J = 1.9603$ and about 60% faster for the cost value of $J = 1.6857$. This clearly demonstrates the efficiency of the proposed method in terms of computation time, as compared to another MATLAB-based tool.

3.5.4 Robot Motion Planning Problem

In this example, the RBF method is applied to solve a 2D navigation problem with nonlinear path constraints. In an area of $[80 \times 80]$, a mobile robot needs to go from the initial point $(40,5)$ to the final point $(55,70)$ in the time interval $[0,20]$ using the minimum kinetic energy and avoiding three circular obstacles in the path. The obstacles are located at $(40,20)$, $(55,40)$, and $(45,65)$ with the radius $r = 10$. Also, the maximum horizontal and vertical speeds of the robot are allowed to be 10. The optimal control problem is to minimize the cost function

$$J = \int_0^{20} (\dot{x}^2(\tau) + \dot{y}^2(\tau))\, d\tau \tag{3-35}$$

subject to box constraints and boundary conditions

$$\begin{aligned} &\left|\dot{x}(\tau)\right| \le 10 \ , \ \left|\dot{y}(\tau)\right| \le 10 \\ &0 \le x(\tau) \le 80 \ , \ 0 \le y(\tau) \le 80 \\ &x(0) = 40 \ , \ y(0) = 5 \\ &x(20) = 55 \ , \ y(20) = 70 \end{aligned} \tag{3-36}$$

and nonlinear path constraints

$$\begin{aligned}10^2 &\le (x(\tau)-40)^2+(y(\tau)-20)^2 \le 80^2\\ 10^2 &\le (x(\tau)-55)^2+(y(\tau)-40)^2 \le 80^2\\ 10^2 &\le (x(\tau)-45)^2+(y(\tau)-65)^2 \le 80^2\end{aligned} \tag{3-37}$$

corresponding to three circular obstacles in the path. To solve this problem, IMQ RBFs with $\varepsilon = 5.5$ are used for parameterization. $x(\tau)$ and $y(\tau)$ each are parameterized with the following number of IMQ RBFs: *N*= [10, 30, 60]. Also, discretization nodes and RBF centers are chosen using LGL nodes. Table 3.6 shows the cost and computation time obtained from the RBF method along with those obtained from PROPT and DIDO for *N*= [10, 30, 60]. PROPT and DIDO are both MATLAB-based software packages using pseudospectral methods to solve optimal control problems.

According to Table 3.6, increasing the number of optimization parameters, *N,* leads to more accurate solutions at the expense of increased computation time. For this example, the accuracy of the RBF method is higher than either of the other methods for different values of *N*. Also, the computation time of the method is competitive with two other methods. In fact, the computation time of the RBF method is much less than that of DIDO, but slightly higher than the computation time of PROPT for the same values of *N*. Figure 3.5 shows the optimal trajectory found by the RBF method for *N*=30.

Table 3.6 Comparison of cost and computation time of the RBF method with PROPT and DIDO for *N*=[10,30,60]

Method	*N*	Cost	Time (s)
RBF	10	255.97	0.95
PROPT	10	256.21	0.16
DIDO	10	278.43	2.50
RBF	30	254.36	1.26
PROPT	30	254.38	0.75
DIDO	30	254.37	44.63
RBF	60	254.30	6.74
PROPT	60	254.31	6.30
DIDO	60	254.32	334.39

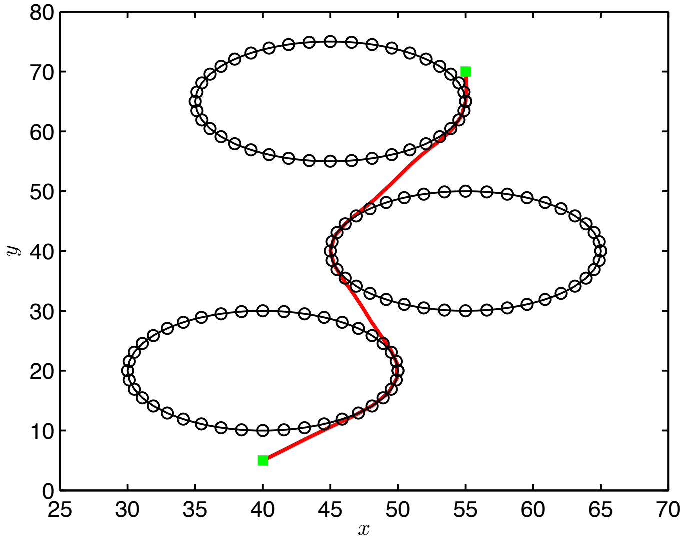


Figure 3.5 Optimal trajectory obtained from the RBF method for the robot motion planning with obstacle avoidance for *N*=30

CHAPTER IV

RBF-GALERKIN METHOD FOR TRAJECTORY OPTIMIZATION AND COSTATE ESTIMATION

Direct methods are extensively used for solving nonlinear optimal control problems mainly because of their ability to efficiently handle path constraints, robustness to initial guess of parameters, and greater radii of convergence compared to indirect methods. A direct method, sometimes called direct transcription, is based on approximating states and/or controls by a specific function with unknown coefficients, and discretizing the optimal control problem with a set of proper nodes to eventually transcribe it into an NLP problem. The resulting NLP then can be solved by efficient NLP solvers available. Most of the direct methods are collocation-based approaches which could be either local or global collocation methods depending on the type of function used for approximation. Runge-Kutta methods [71], B-spline approaches [72], and direct collocation methods [73] are examples of local collocation methods using low-degree local polynomials for approximating states and controls in each subinterval. The optimal trajectory is built by stitching those local trajectories together and enforcing some continuity conditions at joints. The main drawback of local polynomial methods is their algebraic convergence rate, so the accuracy of solution obtained from these methods is not that impressive. Pseudospectral methods [10]-[15], on the other hand, use a high-degree global polynomial (e.g. Legendre or Chebyshev polynomials) for the approximation and a set of orthogonal nodes associated with the family of the polynomial for the discretization.

Spectral (exponential) accuracy along with ease of implementation made them so popular for trajectory optimization problems. However, it was proven that the spectral accuracy only holds for sufficiently smooth functions [16], so if the solution contains irregular smoothness or switching times, a traditional pseudospectral method will converge slowly even with a very high-degree polynomial [46]. Pseudospectral methods are tied to a specific mesh of points; for example a Legendre pseudospectral method can only use a set of Legendre-Gauss-Lobatto (LGL) nodes, whereas a Gauss pseudospectral method only applies Legendre-Gauss (LG) nodes for the discretization. This limitation becomes even more important when there are some discontinuities in the optimal solution and a classic pseudospectral method would not be able to provide more dense points near those discontinuities [4]. Some research work was conducted to apply variations to the classic pseudospectral methods to capture those discontinuities, for instance, in [74], [75] a pseudospectral method along with finite elements were used to combine both benefits of global collocation (high accuracy) and local collocation (more flexibility in choosing discretization points as well as sparser NLP problem resulted from direct transcription) for solving optimal control problems. However, these modified methods impose a number of limitations to the mathematical formulation of the problem and are incapable of finding a solution to non-sequential optimal control problems [76].

A different approach is proposed to extend the discretization of the optimal control problem into any arbitrary points. Global RBFs are used as the basis functions for parameterization of states and controls. Regardless of the number of points and how they are selected (meshless points or on a mesh of points), the RBF interpolation for global RBFs are always unique [18]. Therefore, the discretization points can be arbitrarily chosen and do not even need to be on a mesh of points, in general. Also, since global RBF contains a broad class of interpolating functions including GA RBFs, MQ, and IMQ, the method offers a great flexibility in choosing basis functions for parameterizing a given problem, which is another advantage of the RBF-based approach.

In Chapter III, an RBF collocation method was developed for direct trajectory optimization. It has been demonstrated that the RBF method is very efficient for solving both fixed and free-final-time optimal control problems, as compared to both B-spline and Legendre pseudospectral methods. While the numerical examples have shown very promising results, there would still remain questions about the optimality of the proposed method. In particular, one may ask if the proposed approach would satisfy the first-order necessary conditions of the optimal control problem, in general. Can the optimality of the proposed method be verified mathematically? Is there any way to successfully estimate the costates (Lagrange multipliers) of the optimal control problem using the RBF method?

A considerable amount of work has been conducted in surveys on relating the first-order necessary conditions of the optimal control problem and KKT optimality conditions of the NLP resulting from direct transcription [8], [71], [77], [78]. It has been shown that the Legendre pseudospectral method, the Jacobi pseudospectral method and also some Runge-Kutta methods give the exact estimation of costates from KKT multipliers of the NLP, under a set of closure conditions [77], [78]. These additional conditions are required to be added to the KKT conditions in order to fill the gap between indirect and direct methods. In this Chapter, it is aimed to answer the optimality question and provide a solution to costate estimation by introducing a new hybrid approach, called the *RBF-Galerkin method* for solving optimal control problems.

The RBF-Galerkin method combines RBF interpolation with Galerkin error projection to efficiently solve an optimal control problem numerically. Another contribution of the proposed approach is costate estimation using the RBF-Galerkin method. A set of conditions are provided for the equivalency of KKT multipliers of the NLP resulted from the RBF-Galerkin method and discretized form of the costates of the optimal control problem to eventually develop a costate mapping theorem for the RBF-Galerkin method.

4.1 RBF-Galerkin Method for Direct Trajectory Optimization

A direct method based on RBF parameterization, arbitrary discretization, Galerkin projection, and nonlinear programming is proposed to solve the optimal control problem of Eqs. (3-1)—(3-4) numerically. RBF is a real-valued function whose value depends only on the distance from a fixed point (center) [18],

$$\rho(\boldsymbol{y},\boldsymbol{c}) = \rho\left(\left\|\boldsymbol{y}-\boldsymbol{c}\right\|\right) \tag{4-1}$$

where ρ is the RBF, $\| \ \|$ is the Euclidean norm, and $\boldsymbol{c}$ is the RBF center. Any function satisfying Eq. (4-1) is called an RBF function.

In the RBF-Galerkin method, global RBFs are used as the trial functions for approximating the optimal control problem. For brevity and without loss of generality, same type of RBFs, ρ, and same number of RBFs, N, are assumed to be used for the approximation of states $\boldsymbol{x}(\tau) \in \mathbb{R}^n$ and controls $\boldsymbol{u}(\tau) \in \mathbb{R}^m$ as

$$\boldsymbol{x}(\tau) \approx \boldsymbol{x}^R(\tau) = \sum_{i=1}^{N} \boldsymbol{\alpha}_i \, \rho\left(\left\|\tau - \tau_i\right\|\right) = \sum_{i=1}^{N} \boldsymbol{\alpha}_i \, \rho_i(\tau) \tag{4-2}$$

$$\boldsymbol{u}(\tau) \approx \boldsymbol{u}^R(\tau) = \sum_{i=1}^{N} \boldsymbol{\beta}_i \, \rho\left(\left\|\tau - \tau_i\right\|\right) = \sum_{i=1}^{N} \boldsymbol{\beta}_i \, \rho_i(\tau) \tag{4-3}$$

Where $\boldsymbol{x}^R(\tau)$, $\boldsymbol{u}^R(\tau)$ denote the RBF approximation of $\boldsymbol{x}(\tau)$, $\boldsymbol{u}(\tau)$, respectively. Also, $\rho_i(\tau)$ is the RBF, and $\boldsymbol{\alpha}_i$, $\boldsymbol{\beta}_i$ are RBF weights for $\boldsymbol{x}^R(\tau)$, $\boldsymbol{u}^R(\tau)$, respectively. Taking derivative of Eq. (4-2) with respect to τ yields

$$\dot{\boldsymbol{x}}(\tau) \approx \dot{\boldsymbol{x}}^R(\tau) = \sum_{i=1}^{N} \boldsymbol{\alpha}_i \, \dot{\rho}\left(\left\|\tau - \tau_i\right\|\right) = \sum_{i=1}^{N} \boldsymbol{\alpha}_i \, \dot{\rho}_i(\tau) \tag{4-4}$$

Substituting Eq. (4-4) in Eq. (3-2), the *defect constraints* (residuals) $\boldsymbol{\psi}(\tau)$ are defined as

$$\boldsymbol{\psi}(\tau) = \frac{t_f - t_0}{2} \boldsymbol{f}\left(\boldsymbol{\alpha}_i, \boldsymbol{\beta}_i, \rho_i(\tau)\right) - \sum_{i=1}^{N} \boldsymbol{\alpha}_i \, \dot{\rho}_i(\tau). \tag{4-5}$$

A Galerkin projection [79] is applied to defect constraints in which the defect constraints are set to be orthogonal to every member of the RBF basis functions, i.e.

$$\int_{-1}^{1} \boldsymbol{\psi}(\tau) \, \rho_j(\tau) \, \mathrm{d}\tau = 0 \qquad \textit{for} \quad j = 1, 2, ..., N \tag{4-6}$$

where $\rho_j(\tau)$ is the RBF. It implies that the defect $\boldsymbol{\psi}$ converges to zero in the mean (in the limit $N \to \infty$). If $\{\boldsymbol{x}^R, \boldsymbol{u}^R\}$ satisfies the boundary conditions of Eq. (3-3), and $\boldsymbol{\psi}$

converges to zero in the mean, the approximated solution of Eq. (3-2), $\{\boldsymbol{x}^R, \boldsymbol{u}^R\}$, converges to its exact solution, $\{\boldsymbol{x}, \boldsymbol{u}\}$, in the mean, i.e.

$$\lim_{N\to\infty} \left\| \{\boldsymbol{x}^R, \boldsymbol{u}^R\} - \{\boldsymbol{x}, \boldsymbol{u}\} \right\|_2 = 0. \tag{4-7}$$

In other words, by applying the Galerkin error projection, the defect constraints are minimized in L^2-norm sense. Now, substituting Eq. (4-5) in Eq. (4-6) and approximating the integral of Eq. (4-6) by a proper quadrature yields

$$\sum_{k=1}^{N} w_k \left(\frac{t_f - t_0}{2} \boldsymbol{f}\left(\boldsymbol{\alpha}_i, \boldsymbol{\beta}_i, \rho_i(\tau_k)\right) - \sum_{i=1}^{N} \boldsymbol{\alpha}_i \dot{\rho}_i(\tau_k) \right) \rho_j(\tau_k) = \mathbf{0} \tag{4-8}$$

for $j = 1, \ldots, N$, where w_k, $k = 1, 2, \ldots, N$ are quadrature weights corresponding to the type of quadrature points used for approximating the integral.

A non-negative slack variable function $\boldsymbol{p}(\tau)$ is defined to convert the inequality path constraints of Eq. (3-4) to equality constraints, approximated using N global RBFs as:

$$\boldsymbol{p}(\tau) \approx \boldsymbol{p}^R(\tau) = \sum_{i=1}^{N} \boldsymbol{\kappa}_i \, \rho\left(\left\|\tau - \tau_i\right\|\right) = \sum_{i=1}^{N} \boldsymbol{\kappa}_i \, \rho_i(\tau) \tag{4-9}$$

where $\boldsymbol{p}^R(\tau)$ is the RBF approximation of $\boldsymbol{p}(\tau)$, and $\boldsymbol{\kappa}_i$ denote RBF weights for the. $\boldsymbol{p}^R(\tau)$. The residual of path constraints, $\boldsymbol{R}_q$ is calculated as

$$\boldsymbol{R}_q = \boldsymbol{q}\left(\boldsymbol{\alpha}_i, \boldsymbol{\beta}_i, \rho_i(\tau)\right) + \sum_{i=1}^{N} \boldsymbol{\kappa}_i \, \rho_i(\tau). \tag{4-10}$$

Similar to Eq. (4-6), a Galerkin projection is applied to the residual $\boldsymbol{R}_q$ to set it orthogonal to every member of the RBFs that can be shown in the discretized form as

$$\sum_{k=1}^{N} w_k \left(\boldsymbol{q}\left(\boldsymbol{\alpha}_i, \boldsymbol{\beta}_i, \rho_i(\tau_k)\right) + \sum_{i=1}^{N} \boldsymbol{\kappa}_i \, \rho_i(\tau_k) \right) \rho_j(\tau_k) = 0 \tag{4-11}$$

for $j = 1, ..., N$, where w_k are the same quadrature weights as used in Eq. (4-8). Using the same numerical quadrature scheme for the approximation of running cost L, the optimal control problem of Eqs. (3-1)—(3-4) is transcribed into the following NLP problem:

Determine $\boldsymbol{A} = (\boldsymbol{\alpha}_1 \; \boldsymbol{\alpha}_2 \; ... \; \boldsymbol{\alpha}_N)^T{}_{N\times n}$, $\boldsymbol{B} = (\boldsymbol{\beta}_1 \; \boldsymbol{\beta}_2 \; ... \; \boldsymbol{\beta}_N)^T{}_{N\times m}$, $\boldsymbol{K} = (\boldsymbol{\kappa}_1, \boldsymbol{\kappa}_2, ..., \boldsymbol{\kappa}_N)^T{}_{N\times q}$, t_0, and t_f that minimize the cost

$$\bar{J} = \Gamma\left(\boldsymbol{\alpha}_i, \rho_i(-1), \rho_i(1), t_0, t_f\right) + \frac{t_f - t_0}{2} \sum_{k=1}^{N} w_k \, L\left(\boldsymbol{\alpha}_i, \boldsymbol{\beta}_i, \rho_i(\tau_k)\right) \tag{4-12}$$

subject to:

$$\begin{aligned} &\sum_{k=1}^{N} w_k \left(\frac{t_f - t_0}{2} \boldsymbol{f}\left(\boldsymbol{\alpha}_i, \boldsymbol{\beta}_i, \rho_i(\tau_k)\right) - \sum_{i=1}^{N} \boldsymbol{\alpha}_i \, \dot{\rho}_i(\tau_k) \right) \rho_j(\tau_k) = 0 \\ &\boldsymbol{\gamma}\left(\boldsymbol{\alpha}_i, \rho_i(-1), \rho_i(1), t_0, t_f\right) = \boldsymbol{0} \\ &\sum_{k=1}^{N} w_k \left(\boldsymbol{q}\left(\boldsymbol{\alpha}_i, \boldsymbol{\beta}_i, \rho_i(\tau_k)\right) + \sum_{i=1}^{N} \boldsymbol{\kappa}_i \, \rho_i(\tau_k) \right) \rho_j(\tau_k) = 0 \end{aligned} \tag{4-13}$$

for $j = 1, 2, ..., N$. The proposed approach is called the RBF-Galerkin method for solving optimal control problems. Since RBF interpolation for global RBFs is always unique, regardless of the type and number of points, RBF-Galerkin method can use any arbitrary global RBF as trial functions for parameterization and any arbitrary set of points for discretization of the optimal control problem. This property makes the proposed method very flexible in terms of both interpolant function and discretization points, compared to most of the other direct methods.

4.2 Costate Estimation

It will be shown that the KKT optimality conditions of the NLP problem of Eqs. (4-12)—(4-13) are exactly equivalent to the discretized form of the first-order necessary conditions of the optimal control problem of Eqs. (3-1)—(3-4), if a set of conditions will be added to the KKT conditions.

4.2.1 KKT Optimality Conditions

Lagrangian or augmented cost of the NLP problem is written as

$$
\begin{aligned}
\bar{J}_a =& \Gamma\left(\boldsymbol{\alpha}_i, \rho_i(-1), \rho_i(1), t_0, t_f\right) + \frac{t_f - t_0}{2}\sum_{k=1}^{N} w_k L\left(\boldsymbol{\alpha}_i, \boldsymbol{\beta}_i, \rho_i(\tau_k)\right) \\
&+ \sum_{j=1}^{N} \tilde{\boldsymbol{\xi}}_j^T \sum_{k=1}^{N} w_k \left(\frac{t_f - t_0}{2} \boldsymbol{f}\left(\boldsymbol{\alpha}_i, \boldsymbol{\beta}_i, \rho_i(\tau_k)\right) - \sum_{i=1}^{N} \boldsymbol{\alpha}_i\, \dot{\rho}_i(\tau_k) \right) \rho_j(\tau_k) \\
&+ \tilde{\boldsymbol{\upsilon}}^T \boldsymbol{\gamma}\left(\boldsymbol{\alpha}_i, \rho_i(-1), \rho_i(1), t_0, t_f\right) \\
&+ \sum_{j=1}^{N} \tilde{\boldsymbol{\eta}}_j^T \sum_{k=1}^{N} w_k \left(\boldsymbol{q}\left(\boldsymbol{\alpha}_i, \boldsymbol{\beta}_i, \rho_i(\tau_k)\right) + \sum_{i=1}^{N} \boldsymbol{\kappa}_i\, \rho_i(\tau_k) \right) \rho_j(\tau_k)
\end{aligned}
\tag{4-14}
$$

Where $\tilde{\boldsymbol{\xi}}_j$, $\tilde{\boldsymbol{\upsilon}}$, $\tilde{\boldsymbol{\eta}}_j$ are KKT multipliers associated with the NLP constraints of (4-13). Differentiating $\bar{J}_a$ with respect to $\boldsymbol{\alpha}_m, \boldsymbol{\beta}_m, \tilde{\boldsymbol{\xi}}_m, \tilde{\boldsymbol{\eta}}_m, \tilde{\boldsymbol{\upsilon}}, \boldsymbol{\kappa}_m, t_0, t_f$ and setting them equal to zero give the KKT optimality conditions: To save space and make it easier to follow, shortened notation $\boldsymbol{x}_1^R \equiv \boldsymbol{x}^R(-1)$, $\boldsymbol{x}_N^R \equiv \boldsymbol{x}^R(1)$, $\boldsymbol{u}_1^R \equiv \boldsymbol{u}^R(-1)$, $\boldsymbol{u}_N^R \equiv \boldsymbol{u}^R(1)$, $\boldsymbol{f}_k \equiv \boldsymbol{f}\left(\boldsymbol{\alpha}_i, \boldsymbol{\beta}_i, \rho_i(\tau_k)\right)$, $\boldsymbol{q}_k \equiv \boldsymbol{q}\left(\boldsymbol{\alpha}_i, \boldsymbol{\beta}_i, \rho_i(\tau_k)\right)$, and $L_k \equiv L\left(\boldsymbol{\alpha}_i, \boldsymbol{\beta}_i, \rho_i(\tau_k)\right)$ are used throughout the Chapter.

$$
\begin{aligned}
\frac{\partial \bar{J}_a}{\partial \boldsymbol{\alpha}_m} &= \frac{\partial \Gamma}{\partial \boldsymbol{x}_1^R}\rho_m(-1) + \frac{\partial \Gamma}{\partial \boldsymbol{x}_N^R}\rho_m(1) + \frac{t_f - t_0}{2}\sum_{k=1}^{N} w_k \frac{\partial L_k}{\partial \boldsymbol{x}^R}\rho_m(\tau_k) \\
&+ \sum_{j=1}^{N} \tilde{\boldsymbol{\xi}}_j^T \sum_{k=1}^{N} w_k \left(\frac{t_f - t_0}{2}\frac{\partial \boldsymbol{f}_k}{\partial \boldsymbol{x}^R}\rho_m(\tau_k) - \dot{\rho}_m(\tau_k) \right)\rho_j(\tau_k) \\
&+ \tilde{\boldsymbol{\upsilon}}^T \frac{\partial \boldsymbol{\gamma}}{\partial \boldsymbol{x}_1^R}\rho_m(-1) + \tilde{\boldsymbol{\upsilon}}^T \frac{\partial \boldsymbol{\gamma}}{\partial \boldsymbol{x}_N^R}\rho_m(1) \\
&+ \sum_{j=1}^{N} \tilde{\boldsymbol{\eta}}_j^T \sum_{k=1}^{N} w_k \frac{\partial \boldsymbol{q}_k}{\partial \boldsymbol{x}^R}\rho_m(\tau_k)\rho_j(\tau_k)
\end{aligned} \tag{4-15}
$$

Lemma 1.

$$
\begin{aligned}
\sum_{j=1}^{N} \tilde{\boldsymbol{\xi}}_j^T \sum_{k=1}^{N} w_k\, \dot{\rho}_m(\tau_k)\rho_j(\tau_k) &= \sum_{j=1}^{N} \tilde{\boldsymbol{\xi}}_j^T\, \rho_m(1)\rho_j(1) \\
&- \sum_{j=1}^{N} \tilde{\boldsymbol{\xi}}_j^T\, \rho_m(-1)\rho_j(-1) - \sum_{j=1}^{N} \tilde{\boldsymbol{\xi}}_j^T \sum_{k=1}^{N} w_k\, \rho_m(\tau_k)\, \dot{\rho}_j(\tau_k)
\end{aligned} \tag{4-16}
$$

Proof: Using integration by parts, it can be written

$$
\int_{-1}^{1} \dot{\rho}_m(\tau)\rho_j(\tau)\,d\tau = \rho_m(1)\rho_j(1) - \rho_m(-1)\rho_j(-1) - \int_{-1}^{1} \rho_m(\tau)\dot{\rho}_j(\tau)\,d\tau. \tag{4-17}
$$

Approximating the integrals of Eq. (4-17) with a numerical quadrature where w_k, $k = 1, 2, ..., N$ are the quadrature weights and multiplying both sides of Eq. (4-17) with $\sum_{j=1}^{N} \tilde{\boldsymbol{\xi}}_j^T$ complete the proof. Now, applying *Lemma 1* to Eq. (4-15) and rearranging the equations lead to

$$
\begin{aligned}
\frac{\partial \bar{J}_a}{\partial \boldsymbol{\alpha}_m} = \frac{t_f - t_0}{2} \sum_{k=1}^{N} w_k & \left\{ \frac{\partial L_k}{\partial \boldsymbol{x}^R} + \sum_{j=1}^{N} \tilde{\boldsymbol{\xi}}_j^T \rho_j(\tau_k) \frac{\partial \boldsymbol{f}_k}{\partial \boldsymbol{x}^R} \right. \\
& \left. + \frac{2}{t_f - t_0} \sum_{j=1}^{N} \tilde{\boldsymbol{\xi}}_j^T \dot{\rho}_j(\tau_k) + \frac{2}{t_f - t_0} \sum_{j=1}^{N} \tilde{\boldsymbol{\eta}}_j^T \rho_j(\tau_k) \frac{\partial \boldsymbol{q}_k}{\partial \boldsymbol{x}^R} \right\} \rho_m(\tau_k) \\
& + \left(\frac{\partial \Gamma}{\partial \boldsymbol{x}_1^R} + \tilde{\boldsymbol{\upsilon}}^T \frac{\partial \boldsymbol{\gamma}}{\partial \boldsymbol{x}_1^R} + \sum_{j=1}^{N} \tilde{\boldsymbol{\xi}}_j^T \rho_j(-1) \right) \rho_m(-1) \\
& + \left(\frac{\partial \Gamma}{\partial \boldsymbol{x}_N^R} + \tilde{\boldsymbol{\upsilon}}^T \frac{\partial \boldsymbol{\gamma}}{\partial \boldsymbol{x}_N^R} - \sum_{j=1}^{N} \tilde{\boldsymbol{\xi}}_j^T \rho_j(1) \right) \rho_m(1) = \mathbf{0},
\end{aligned}
\tag{4-18}
$$

similarly,

$$
\begin{aligned}
\frac{\partial \bar{J}_a}{\partial \boldsymbol{\beta}_m} = \frac{t_f - t_0}{2} \sum_{k=1}^{N} w_k & \left\{ \frac{\partial L_k}{\partial \boldsymbol{u}^R} + \sum_{j=1}^{N} \tilde{\boldsymbol{\xi}}_j^T \rho_j(\tau_k) \frac{\partial \boldsymbol{f}_k}{\partial \boldsymbol{u}^R} \right. \\
& \left. + \frac{2}{t_f - t_0} \sum_{j=1}^{N} \tilde{\boldsymbol{\eta}}_j^T \rho_j(\tau_k) \frac{\partial \boldsymbol{q}_k}{\partial \boldsymbol{u}^R} \right\} \rho_m(\tau_k) = \mathbf{0}
\end{aligned}
\tag{4-19}
$$

$$
\begin{aligned}
& \frac{\partial \bar{J}_a}{\partial \tilde{\boldsymbol{\xi}}_m} = \sum_{k=1}^{N} w_k \left(\frac{t_f - t_0}{2} \boldsymbol{f}_k^T - \sum_{i=1}^{N} \boldsymbol{\alpha}_i^T \, \dot{\rho}_i(\tau_k) \right) \rho_m(\tau_k) = \mathbf{0} \\
& \frac{\partial \bar{J}_a}{\partial \tilde{\boldsymbol{\eta}}_m} = \sum_{k=1}^{N} w_k \left(\boldsymbol{q}_k^T + \sum_{i=1}^{N} \boldsymbol{\kappa}_i^T \, \rho_i(\tau_k) \right) \rho_m(\tau_k) = \mathbf{0} \\
& \frac{\partial \bar{J}_a}{\partial \tilde{\boldsymbol{\upsilon}}} = \boldsymbol{\gamma}^T \left(\boldsymbol{\alpha}_i, \rho_i(-1), \rho_i(1), t_0, t_f \right) = \mathbf{0} \\
& \frac{\partial \bar{J}_a}{\partial \tilde{\boldsymbol{\kappa}}_m} = \sum_{k=1}^{N} w_k \sum_{j=1}^{N} \tilde{\boldsymbol{\eta}}_j^T \, \rho_j(\tau_k) \, \rho_m(\tau_k) = \mathbf{0} \\
& \frac{\partial \bar{J}_a}{\partial t_0} = \frac{\partial \Gamma}{\partial t_0} + \tilde{\boldsymbol{\upsilon}}^T \frac{\partial \boldsymbol{\gamma}}{\partial t_0} - \frac{1}{2} \sum_{k=1}^{N} w_k \left(L_k + \sum_{j=1}^{N} \tilde{\boldsymbol{\xi}}_j^T \rho_j(\tau_k) \boldsymbol{f}_k \right) = 0 \\
& \frac{\partial \bar{J}_a}{\partial t_f} = -\frac{\partial \Gamma}{\partial t_f} - \tilde{\boldsymbol{\upsilon}}^T \frac{\partial \boldsymbol{\gamma}}{\partial t_f} - \frac{1}{2} \sum_{k=1}^{N} w_k \left(L_k + \sum_{j=1}^{N} \tilde{\boldsymbol{\xi}}_j^T \rho_j(\tau_k) \boldsymbol{f}_k \right) = 0
\end{aligned}
\tag{4-20}
$$

for $m = 1, 2, ..., N$. Eqs. (4-18)—(4-20) are KKT optimality conditions for the NLP problem.

4.2.2 First-Order Necessary Conditions of the Optimal Control Problem

Assuming $\boldsymbol{\lambda}(\tau) \in \mathbb{R}^n$ is the costate, and $\boldsymbol{\mu}(\tau) \in \mathbb{R}^q$ is the Lagrange multiplier associated with the path constraints, Lagrangian of the Hamiltonian (augmented Hamiltonian) of the optimal control problem of Eqs. (3-1)—(3-4) can be shown as

$$\bar{H}(\boldsymbol{x},\boldsymbol{u},\boldsymbol{\mu},\boldsymbol{\lambda}) = L(\boldsymbol{x},\boldsymbol{u}) + \boldsymbol{\lambda}^T \boldsymbol{f}(\boldsymbol{x},\boldsymbol{u}) + \boldsymbol{\mu}^T \left(\boldsymbol{q}(\boldsymbol{x},\boldsymbol{u}) + \boldsymbol{p}\right) \tag{4-21}$$

where $\bar{H}$ is the augmented Hamiltonian, and $\boldsymbol{p}$ is the slack variable function. Please note that the notation τ was removed from Eq. (4-21) for simplicity. The first-order necessary conditions of the optimal control problem are derived as

$$\begin{aligned}
&\dot{\boldsymbol{x}}^T = \frac{t_f - t_0}{2}\boldsymbol{f}^T(\boldsymbol{x},\boldsymbol{u}) = \frac{t_f - t_0}{2}\frac{\partial \bar{H}}{\partial \boldsymbol{\lambda}} \\
&\dot{\boldsymbol{\lambda}}^T = \frac{d\boldsymbol{\lambda}}{d\tau} = -\frac{t_f - t_0}{2}\left(\frac{\partial L}{\partial \boldsymbol{x}} + \boldsymbol{\lambda}^T \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}} + \boldsymbol{\mu}^T \frac{\partial \boldsymbol{q}}{\partial \boldsymbol{x}}\right) = -\frac{t_f - t_0}{2}\frac{\partial \bar{H}}{\partial \boldsymbol{x}} \\
&\frac{\partial L}{\partial \boldsymbol{u}} + \boldsymbol{\lambda}^T \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}} + \boldsymbol{\mu}^T \frac{\partial \boldsymbol{q}}{\partial \boldsymbol{u}} = \frac{\partial \bar{H}}{\partial \boldsymbol{u}} = \boldsymbol{0} \\
&\boldsymbol{\gamma}^T(\boldsymbol{x}(-1), t_0, \boldsymbol{x}(1), t_f) = \boldsymbol{0} \\
&\boldsymbol{\lambda}^T(-1) = -\left.\frac{\partial \Gamma}{\partial \boldsymbol{x}}\right|_{\tau=-1} - \boldsymbol{\upsilon}^T \left.\frac{\partial \boldsymbol{\gamma}}{\partial \boldsymbol{x}}\right|_{\tau=-1}, \boldsymbol{\lambda}^T(1) = \left.\frac{\partial \Gamma}{\partial \boldsymbol{x}}\right|_{\tau=1} + \boldsymbol{\upsilon}^T \left.\frac{\partial \boldsymbol{\gamma}}{\partial \boldsymbol{x}}\right|_{\tau=1} \\
&\boldsymbol{q}^T(\boldsymbol{x},\boldsymbol{u}) + \boldsymbol{p}^T = \boldsymbol{0} \quad , \quad \frac{\partial \bar{H}}{\partial \boldsymbol{p}} = \boldsymbol{\mu}^T = \boldsymbol{0} \\
&\bar{H}(t_0) = \frac{\partial \Gamma}{\partial t_0} + \boldsymbol{\upsilon}^T \frac{\partial \boldsymbol{\gamma}}{\partial t_0} \quad , \quad \bar{H}(t_f) = -\frac{\partial \Gamma}{\partial t_f} - \boldsymbol{\upsilon}^T \frac{\partial \boldsymbol{\gamma}}{\partial t_f}
\end{aligned} \tag{4-22}$$

where $\boldsymbol{\upsilon} \in \mathbb{R}^\gamma$ is the Lagrange multiplier associated with the boundary conditions $\boldsymbol{\gamma}$.

4.2.3 RBF-Galerkin Discretized Form of First-Order Necessary Conditions

In this section, first-order necessary conditions of Eq. (4-22) are discretized using the RBF-Galerkin method. First, costates $\boldsymbol{\lambda}(\tau) \in \mathbb{R}^n$ and Lagrange multipliers $\boldsymbol{\mu}(\tau) \in \mathbb{R}^q$ are approximated using N global RBFs as

$$\boldsymbol{\lambda}(\tau) \approx \boldsymbol{\lambda}^R(\tau) = \sum_{j=1}^{N} \boldsymbol{\xi}_j \, \rho\left(\left\|\tau - \tau_j\right\|\right) = \sum_{j=1}^{N} \boldsymbol{\xi}_j \, \rho_j(\tau) \tag{4-23}$$

$$\boldsymbol{\mu}(\tau) \approx \boldsymbol{\mu}^R(\tau) = \sum_{j=1}^{N} \boldsymbol{\eta}_j \, \rho\left(\left\|\tau - \tau_j\right\|\right) = \sum_{j=1}^{N} \boldsymbol{\eta}_j \, \rho_j(\tau) \tag{4-24}$$

where $\boldsymbol{\lambda}^R(\tau)$ and $\boldsymbol{\mu}^R(\tau)$ are RBF approximations of $\boldsymbol{\lambda}(\tau)$ and $\boldsymbol{\mu}(\tau)$, respectively. Using Eqs. (4-2), (4-3), and (4-9) along with Eqs. (4-23) and (4-24), first-order necessary conditions are parameterized with global RBFs. Now, applying Galerkin projection to the residuals and approximating the Galerkin integral with a numerical quadrature discretize the necessary conditions of the optimal control problem as

$$\sum_{k=1}^{N} w_k \left(\frac{t_f - t_0}{2} \boldsymbol{f}_k^T - \sum_{i=1}^{N} \boldsymbol{\alpha}_i^T \, \dot{\rho}_i(\tau_k) \right) \rho_m(\tau_k) = 0 \tag{4-25}$$

$$\begin{aligned} &\frac{t_f - t_0}{2} \sum_{k=1}^{N} w_k \left\{ \frac{\partial L_k}{\partial \boldsymbol{x}^R} + \sum_{j=1}^{N} \boldsymbol{\xi}_j^T \rho_j(\tau_k) \frac{\partial \boldsymbol{f}_k}{\partial \boldsymbol{x}^R} \right. \\ &\left. + \sum_{j=1}^{N} \boldsymbol{\eta}_j^T \rho_j(\tau_k) \frac{\partial \boldsymbol{q}_k}{\partial \boldsymbol{x}^R} + \frac{2}{t_f - t_0} \sum_{j=1}^{N} \boldsymbol{\xi}_j^T \dot{\rho}_j(\tau_k) \right\} \rho_m(\tau_k) = 0 \end{aligned} \tag{4-26}$$

$$\sum_{k=1}^{N} w_k \left\{ \frac{\partial L_k}{\partial \boldsymbol{u}^R} + \sum_{j=1}^{N} \boldsymbol{\xi}_j^T \rho_j(\tau_k) \frac{\partial \boldsymbol{f}_k}{\partial \boldsymbol{u}^R} + \sum_{j=1}^{N} \boldsymbol{\mu}_j^T \rho_j(\tau_k) \frac{\partial \boldsymbol{q}_k}{\partial \boldsymbol{u}^R} \right\} \rho_m(\tau_k) = 0 \tag{4-27}$$

$$
\begin{gathered}
\boldsymbol{\gamma}^T\left(\boldsymbol{\alpha}_i,\rho_i(-1),\rho_i(1),t_0,t_f\right)=\mathbf{0}\\
\frac{\partial\Gamma}{\partial\boldsymbol{x}_1^R}+\boldsymbol{\upsilon}^T\frac{\partial\boldsymbol{\gamma}}{\partial\boldsymbol{x}_1^R}=-\sum_{j=1}^{N}\boldsymbol{\xi}_j^T\rho_j(-1)\\
\frac{\partial\Gamma}{\partial\boldsymbol{x}_N^R}+\boldsymbol{\upsilon}^T\frac{\partial\boldsymbol{\gamma}}{\partial\boldsymbol{x}_N^R}=\sum_{j=1}^{N}\boldsymbol{\xi}_j^T\rho_j(1)\\
\sum_{k=1}^{N}w_k\left(\boldsymbol{q}_k^T+\sum_{i=1}^{N}\boldsymbol{\kappa}_i^T\,\rho_i(\tau_k)\right)\rho_m(\tau_k)=\mathbf{0}\\
\sum_{k=1}^{N}w_k\sum_{j=1}^{N}\boldsymbol{\eta}_j^T\rho_j(\tau_k)\,\rho_m(\tau_k)=\mathbf{0}\\
\bar{H}(t_0)=\frac{\partial\Gamma}{\partial t_0}+\boldsymbol{\upsilon}^T\frac{\partial\boldsymbol{\gamma}}{\partial t_0}\,,\;\bar{H}(t_f)=-\frac{\partial\Gamma}{\partial t_f}-\boldsymbol{\upsilon}^T\frac{\partial\boldsymbol{\gamma}}{\partial t_f}
\end{gathered}
\tag{4-28}
$$

4.2.4 Costate Mapping Theorem

A set of conditions is required to be added to the KKT optimality conditions of Eqs. (4-18)—(4-20) to provide an exact equivalency between KKT conditions and discretized form of first-order necessary conditions of Eqs. (4-25)—(4-28). These conditions are

$$
\begin{gathered}
\frac{\partial\Gamma}{\partial\boldsymbol{x}_1^R}+\tilde{\boldsymbol{\upsilon}}^T\frac{\partial\boldsymbol{\gamma}}{\partial\boldsymbol{x}_1^R}=-\sum_{j=1}^{N}\tilde{\boldsymbol{\xi}}_j^T\rho_j(-1)\\
\frac{\partial\Gamma}{\partial\boldsymbol{x}_N^R}+\tilde{\boldsymbol{\upsilon}}^T\frac{\partial\boldsymbol{\gamma}}{\partial\boldsymbol{x}_N^R}=\sum_{j=1}^{N}\tilde{\boldsymbol{\xi}}_j^T\rho_j(1).
\end{gathered}
\tag{4-29}
$$

Also, comparing Eqs. (4-18)—(4-20) with Eqs. (4-25)—(4-28) implies that

$$
\bar{H}(t_0)=\bar{H}(t_f)=\frac{1}{2}\sum_{k=1}^{N}w_k\left(L_k+\sum_{j=1}^{N}\tilde{\boldsymbol{\xi}}_j^T\rho_j(\tau_k)\boldsymbol{f}_k\right)
\tag{4-30}
$$

Discrete conditions of Eqs. (4-29), (4-30) are applied to the costate boundaries (transversality conditions) to guarantee that first-order necessary conditions of the NLP (KKT conditions) are equivalent to discretized form of the first-order necessary conditions of the optimal control problem. In other words, by adding Eqs. (4-29), (4-30) to the KKT conditions, the solution of the direct method would be the same as the solution of the indirect method for the optimal control problem of Eqs. (3-1)—(3-4).

RBF-Galerkin Costate Mapping Theorem: There exists an exact equivalency between KKT multipliers of the NLP derived from the RBF-Galerkin method and Lagrange multipliers of the optimal control problem discretized by the RBF-Galerkin method. Lagrange multipliers (costates) of the optimal control problem can be estimated from KKT multipliers of the NLP at discretization points using the following equations:

$$\boldsymbol{\xi}_j = \tilde{\boldsymbol{\xi}}_j \quad , \quad \boldsymbol{\eta}_j = \frac{2}{t_f - t_0}\tilde{\boldsymbol{\eta}}_j \quad , \quad \boldsymbol{\upsilon} = \tilde{\boldsymbol{\upsilon}} \quad , \quad j = 1, 2, ..., N \tag{4-31}$$

Proof: substitution of Eq. (4-31) at Eqs. (4-26), (4-27), and (4-28) proves that Eqs. (4-18)—(4-20) and Eqs. (4-25)—(4-28) are the same, and hence the equivalency condition holds.

4.3 Numerical Examples

In this section, two numerical examples are solved using the RBF-Galerkin method. The first example is a bang-bang optimal control problem [80] with an analytical solution, so the optimal trajectories found by the proposed method are evaluated against the exact solutions. Also, the costates estimated by the RBF-Galerkin costate mapping theorem are compared with the exact costates obtained from an analytical method. The second example is a UAV navigation problem in 2D space with nonlinear path constraints. Performance of the proposed method is compared against a Local polynomial method (OPTRAGEN) [81] and a global polynomial approach (DIDO) [68] in terms of accuracy and computation time for the UAV navigation example.

4.3.1 Bang-Bang Control

A bang-bang optimal control problem with a quadratic cost [80] is considered here as to minimize the cost functional

$$J = \frac{1}{2}\int_0^5 \left(x_1^2(t) + x_2^2(t)\right) dt \tag{4-32}$$

subject to:

$$\begin{aligned} &\dot{x}_1(t) = x_2(t) \\ &\dot{x}_2(t) = -x_1(t) + x_2(t) + u(t) \\ &x_1(0) = 0.231, \quad x_2(0) = 1.126, \quad |u(t)| \le 0.8. \end{aligned} \tag{4-33}$$

According to [80], the optimal control can be calculated from an analytical approach as

$$u^*(t) = \begin{cases} -0.8 & 0 \le t \le 1.275 \\ \ \ 0.8 & 1.275 \le t \le 5 \end{cases} \tag{4-34}$$

and the corresponding value of the cost functional would be 5.660.

It has been shown in [82] that a classic pseudospectral method would not be able to accurately find the optimal control for this example due to the nonsmoothness/discontinuity in the solution. For instance, the numerical solution for the control function obtained from a Chebyshev pseudospectral method [83] has shown undesired fluctuations at the boundaries, and the switching time estimated was not accurate enough even with the modest number of discretization points (i.e. $N = 32$, see page 4, Figure 1 in [82] for details). It should be noted that increasing the number of points would not help to remedy this issue and only cause ill-conditioning of the discretized problem.

The efficiency of the RBF-Galerkin method for solving this example is investigated. To evaluate the arbitrary discretization capability, a set of pseudorandom points along with the trapezoidal quadrature is chosen for discretization. 40 randomly distributed nodes are selected in the interval $[0 \;\; 5]$ from which at least 5 nodes should be located between $[1.2 \;\; 1.3]$. Increasing the density of points around the switching time $t = 1.275$, which is not possible in a pseudospectral method without using mesh refinement techniques, helps to capture the discontinuity of the control function and also can be used as a measure of accuracy of the proposed method for the points located around the discontinuity. IMQ RBFs are used for the parameterization, and with aforementioned pseudorandom points, the optimal control problem is transcribed into an NLP problem, which is then solved by SNOPT with default feasibility and optimality tolerances $(\approx 10^{-6})$.

Figure 4.1 shows states and control trajectories obtained from the RBF-Galerkin method for 40 random nodes along with their exact solutions. Also, costates estimated from the RBF-Galerkin costate mapping theorem are illustrated along with the exact costates in Figure 4.2. The accuracy of the RBF-Galerkin method is clearly demonstrated in graphs even for those nodes located near the control discontinuity $(t = 1.275)$. The cost value calculated from the RBF-Galerkin method is 5.663 (error≈ 0.003), and the switching time of the optimal control found as 1.279 (error≈ 0.004). The maximum absolute errors in states $x_1(t)$ and $x_2(t)$ (over all 40 random nodes) are approximately 2.5×10^{-6} and 7.9×10^{-6}, respectively.

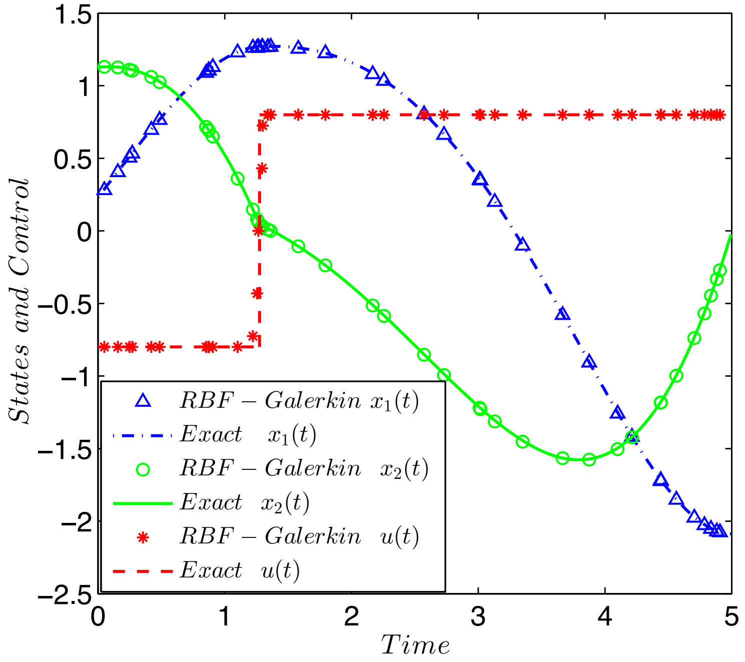


Figure 4.1 States and control trajectories obtained from the RBF-Galerkin method for 40 random nodes along with their exact solutions

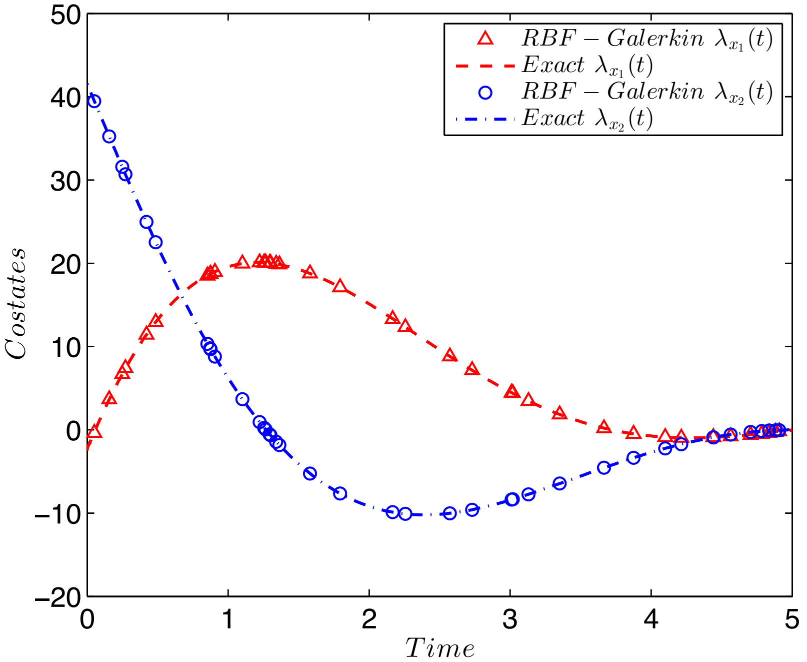


Figure 4.2 Costates estimated from the RBF-Galerkin costate mapping theorem along with the exact costates for 40 random nodes

Also, the maximum absolute error of the optimal control $u(t)$ is 0.63, which occurs during the switching time (maximum error in the optimal control is 2.3×10^{-6} without considering nodes located between $[1.2 \;\; 1.3]$). The maximum absolute errors for costates $\lambda_{x_1}(t)$ and $\lambda_{x_2}(t)$ (over 40 nodes) are 6.6×10^{-5} and 3.6×10^{-5}, respectively. This numerically verifies accuracy of the RBF-Galerkin costate mapping theorem as in Eq. (4-31). Even higher accuracy would be obtained by increasing the number of nodes. For example, the maximum absoulte errors of $\lambda_{x_1}(t)$ and $\lambda_{x_2}(t)$ will be decreased to 3.4×10^{-6} and 3.0×10^{-6} (close to the level of feasibility and optimality tolerances set in the NLP solver) by using 80 dicreatization nodes.

4.3.2 UAV Navigation Problem

The RBF-Galerkin method is applied to a UAV navigation problem with obstacle avoidance in a 2D space. Two UAVs set off from $[5m, 10m]$ and $[10m, 5m]$ to the final destination of $[80m \sim 85m \;,\; 60m \sim 65m]$ in the time span of 20 seconds using the minimum kinetic energy and avoiding three circular obstacles in the path. The obstacles are located at $[30m, 20m]$, $[40m, 45m]$, and $[65m, 45m]$, all with the radius of $r = 10\,m$. In addition, there would be other limitations that UAVs need to fly near each other within a distance of $[5m, 8m]$, and their maximum horizontal and vertical speeds must not exceed $15\,m/s$. The optimal control problem is formulated as to minimize

$$J = \int_0^{20} (\dot{x}_1{}^2 + \dot{y}_1{}^2 + \dot{x}_2{}^2 + \dot{y}_2{}^2)\, dt \qquad (4\text{-}35)$$

subject to linear constraints

$$\begin{aligned}
&\left|\dot{x}_1(t)\right| \le 15 \ , \ \left|\dot{y}_1(t)\right| \le 15 \\
&\left|\dot{x}_2(t)\right| \le 15 \ , \ \left|\dot{y}_2(t)\right| \le 15 \\
&0 \le x_1(t) \le 90 \ , \ 0 \le y_1(t) \le 90 \\
&0 \le x_2(t) \le 90 \ , \ 0 \le y_2(t) \le 90 \\
&x_1(0) = 5 \ , \ y_1(0) = 10 \\
&x_2(0) = 10 \ , \ y_2(0) = 5 \\
&80 \le x_1(20) \le 85 \ , \ 60 \le y_1(20) \le 65 \\
&80 \le x_2(20) \le 85 \ , \ 60 \le y_2(20) \le 65
\end{aligned} \tag{4-36}$$

and nonlinear constraints

$$\begin{aligned}
&5^2 \le (x_1(t) - x_2(t))^2 + (y_1(\mathrm{t}) - y_2(\mathrm{t}))^2 \le 8^2 \\
&10^2 \le (x_1(t) - x_{ob_i})^2 + (y_1(\mathrm{t}) - y_{ob_i})^2 \le 90^2 \quad , \quad i = 1,2,3 \\
&10^2 \le (x_2(t) - x_{ob_i})^2 + (y_2(\mathrm{t}) - y_{ob_i})^2 \le 90^2 \quad , \quad i = 1,2,3
\end{aligned} \tag{4-37}$$

where (x_{ob_i}, y_{ob_i}) are the locations of the obstacles. To find the optimal trajectories, $x_1(t)$, $\mathrm{y}_1(t)$, $x_2(t)$, $\mathrm{y}_2(t)$ each are parameterized with IMQ RBFs and equally-spaced nodes employed for the collocation points as well as the RBF centers. In addition, the integral cost of Eq. (4-35) is approximated with a trapezoidal quadrature using the equally-spaced nodes selected. The cost and computation time of the RBF-Galerkin method are shown in Table 4.1 along with those obtained from a Legendre pseudospectral method (DIDO) [68] and a B-spline local polynomial method (OPTRAGEN) [81] for *N*= [10, 20, 30]. Also, the optimal trajectories calculated by all three methods have been demonstrated in Figure 4.3. Although the amount of the cost for all three methods in Table 4.1 gets higher by increasing the number of nodes from 10 to 30, the calculated trajectories by all three methods become more accurate by increasing the number of nodes, as expected (see Figure 4.3).

Table 4.1 Comparison of cost and computation time of the RBF-Galerkin method with a Legendre pseudospectral method and a B-spline method for *N*=[10, 20, 30]

Method	*N*	Cost	Time (s)
RBF-Galerkin Method	10	2179.7	3.22
Legendre Pseudospectral (DIDO)	10	2254.3	23.23
B-Spline Method (OPTRAGEN)	10	2159.2	5.75
RBF-Galerkin Method	20	2268.3	4.98
Legendre Pseudospectral (DIDO)	20	2302.7	55.39
B-Spline Method (OPTRAGEN)	20	2299.4	8.73
RBF-Galerkin Method	30	2245.6	9.16
Legendre Pseudospectral (DIDO)	30	2356.4	546.09
B-Spline Method (OPTRAGEN)	30	2269.9	10.18

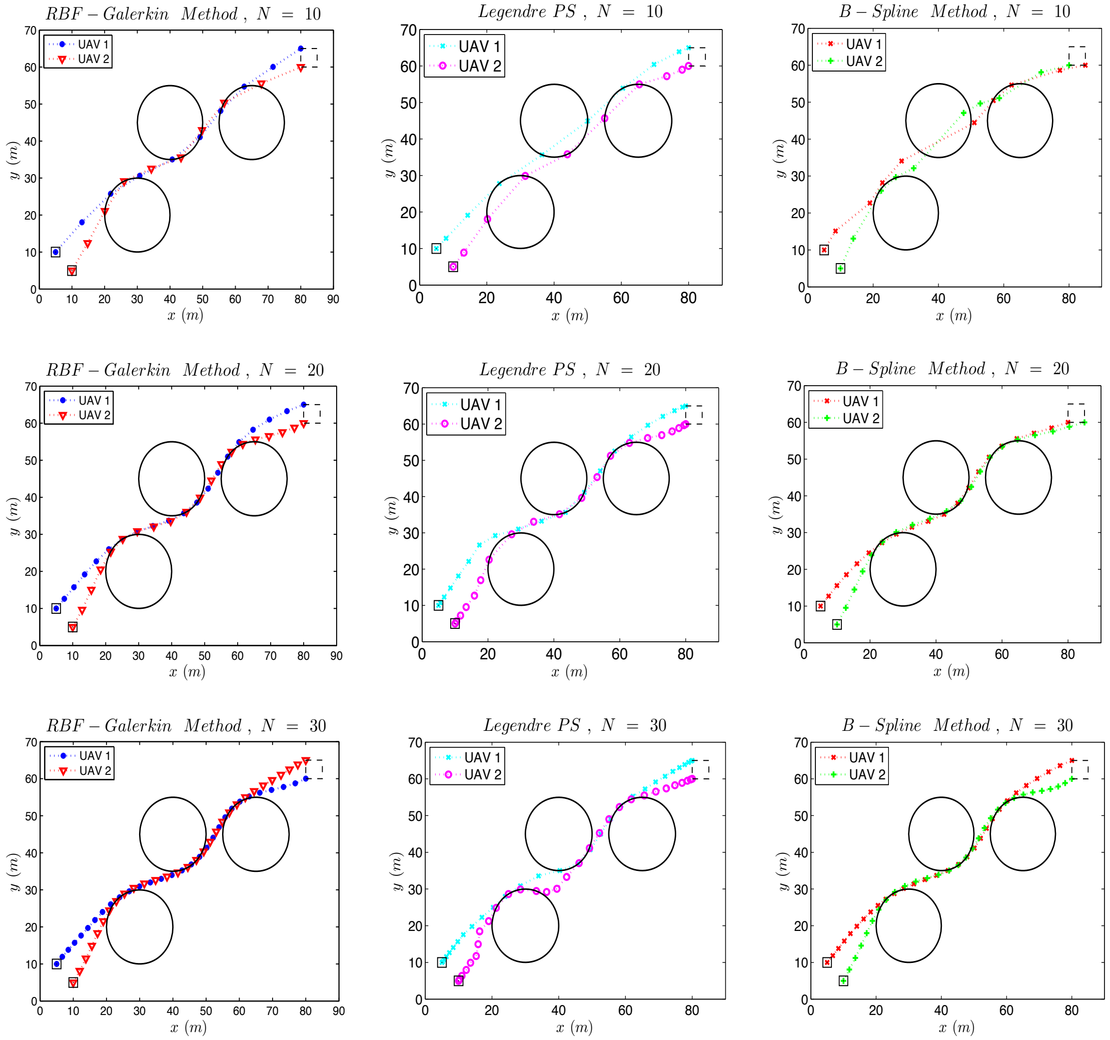


Figure 4.3 Comparison of optimal trajectories developed by RBF-Galerkin Method, Legendre Pseudospectral method, and B-spline approach for UAV navigation problem for *N*=[10, 20, 30]

Checking for path constraint violations reveals that solutions found for smaller number of nodes, N=[10, 20], have some errors in calculated trajectories, as seen in Figure 4.3, and so the calculated cost is less than what it should be for each method because of those violations. Comparison of cost values for N=[20, 30] discloses that the accuracy of the RBF-Galerkin method is higher than either of the other methods for this example. This is clearly verified by comparing the trajectories developed by the proposed method with those calculated by the other two methods in Figure 4.3. The computation time of the proposed method is also less than two other methods for the same number of nodes for this example, as shown in Table 4.1. For instance, the RBF-Galerkin method is about 10% faster than OPTRAGEN and 98.3% faster than DIDO for N=30.

CHAPTER V

INDIVIDUALIZED DRUG DOSING USING RBF-GALERKIN METHOD: CASE OF ANEMIA MANAGEMENT IN CHRONIC KIDNEY DISEASE

In this Chapter, anemia management is formulated as a constrained optimal control problem and successfully solved by the RBF-Galerkin method. Then, a multiple receding horizon control (MRHC) approach based on the RBF-Galerkin method is presented for individualized anemia management in Chronic Kidney Disease (CKD). As a common complication of CKD, anemia is associated with decreased quality of life (QoL) as well as increased risk of hospitalizations and even mortality among patients [84]. It has been strongly recommended by FDA regulations [85] and national guidelines [84] to administer the Erythropoietin (EPO) therapy -the universal treatment of anemia for CKD patients- in an individual-based fashion with more precise control on the hemoglobin (Hb) level. In accordance with those recommendations, the proposed solution is an individualized approach based on the RBF-Galerkin method aiming to provide an accurate, reliable solution to the anemia management problem. Simulation results are compared with a population-oriented clinical protocol as well as a state-of-the-art individual-based method for anemia management to investigate the efficacy of the proposed method.

5.1 Introduction

Erythropoietin (EPO) is a glycoprotein hormone produced by kidney promoting the formation of red blood cells (RBCs) in bone marrow. Anemia, a condition resulted from lower than normal number of RBCs, is frequently developed in CKD because of EPO deficiency. Anemia can adversely affect the QoL and also increase the risk of cardiovascular disease, hospitalizations, and even mortality [84]. Discovery of exogenous recombinant human EPO in the late 1980s has revolutionized the treatment of renal anemia [86], which was previously only controlled by repeated blood transfusion – a procedure associated with several complications, including increased risk of infections, allergic reactions, and sensitization impeding kidney transplantation [87]-[89]. Since then, renal anemia has been effectively treated by the administration of exogenous EPO as well as other erythropoiesis stimulating agents (ESAs) [90]-[92]. The ESA dose adjustments are determined in such a way to promote patient's hemoglobin (Hb) level and maintain it within recommended ranges, according to national guidelines [84]. Lower doses of ESAs are ineffective for the anemia management, whereas higher doses may cause higher-than-necessary Hb concentration increasing the risk of serious cardiovascular problems and blood clots [93], [94]. As a result, optimal control of the ESA dose would be of great importance during the whole treatment which justifies the use of advanced control techniques [95]-[101] for the anemia management problem. Most of these techniques [95]-[99] are population-based approaches considering a one-size-fits-all model for the whole patient groups, and thereby may not achieve the desired outcomes because of the variability of dose-response profile among patients [102]. Variations of responses among patient groups make the ESA dose adjustment rather challenging and impose the need for more precise *individual-based* control approaches for the anemia management problem [103]. Recent studies have shown that individualization of ESA dosing can stabilize the Hb concentration and hence reduce the risk of blood transfusion [85], [101]. In [100], a support vector regression approach along with a multilayer perception neural network was utilized to develop a personalized model for the anemia management problem. Despite their successful model in improving the QoL for CKD patients, their work was primarily focused on predicting the EPO dosage rather than the Hb level. In a clinical setting, the prediction of Hb level would be more useful and practical than the prediction of EPO drug dose, since the former has well-established protocols and recommended guidelines [104]. A model predictive control (MPC) approach was designed and successfully applied to the anemia management problem as a population-oriented method in [97], [98] and later as an individualized approach [101]. While their recent work for individualized anemia management has shown promising results in declining Hb variability among CKD patients [101], the method is not sufficiently fast for achieving the desired Hb level, especially for poor responder patients. Also, as shown later in Section 5.5 , around 5% random error (±0.5 g/dL), which is not unlikely in practice considering measurement errors as well as other factors that are not included in a patient model, can make the ESA dose recommended by the approach of [101] very fluctuating and quite unreliable. Moreover, the weekly dose of the ESA drug can be greatly improved using a more accurate optimal control solution.

In this Chapter, an RBF-Galerkin optimization approach is proposed for the ESA drug dosing problem and a multiple receding horizon controller designed based on the proposed method for individualized anemia management. The method provides a highly accurate numerical solution to constrained optimal control problems, so it is applied as a means to find the individualized weekly dose of the anemia drug for CKD patients. Performance of the proposed approach are compared in silico with a population-oriented anemia management protocol (AMP) [84] as well as an MPC-based individualized method for anemia management [101] for two case scenarios: Hb measurement without observational error and in presence of error/measurement noise. The outcome of this work is twofold: first, by finding the individualized optimum dosages necessary for achieving a desired Hb level, both the side effects of drug overdose and the risk of blood transfusion [84] are minimized. In addition, considering the high cost of medication, individualization of the ESA drug dose can reduce its usage, and hence result in potential saving for health care costs, which is another benefit of the current work.

5.2 Anemia Management Problem

5.2.1 Anemia Introduction and Definition

Healthy kidneys produce EPO prompting bone morrow to make RBCs. As a consequence of kidney failure, patients with end stage renal disease (ESRD) develop anemia primarily because of reduced EPO production. Anemia is usually associated with extreme weakness, tiredness, dizziness, and inability to think clearly, so it may reduce the QoL for CKD patients significantly. The efficient treatment for renal anemia is the administration of exogenous EPO, commonly called EPO drug or ESA drug. Weekly dose of the ESA drug is currently determined based on the patient's Hb level and its variation, which should be maintained within specified ranges recommended by national guidelines [84]. Dialysis facilities in North America often use their own nonvalidated therapeutic approach, called anemia management protocols (AMP), for the treatment of renal anemia. A typical AMP is a rule-based algorithmic system that acts as a standardized protocol for anemia management developed based upon national guidelines [84] and FDA regulations [85]. However, the AMP rules are made for the treatment of population as a whole and would not consider dose-response profile for individual patients and therefore may not attain the desired outcome for each individual patient in practice [101], [102]. On the other hand, modern control techniques would allow more sophisticated individualized anemia management that eventually reduce the risk of blood transfusion, according to recent studies [85], [101]. Moreover, automatic control techniques provide more precise control on the Hb level, and so their performance exceeds that of AMPs [97], [98], [101], [105], [107].

5.2.2 Individualized ESA Drug Dosing as an Optimal Control Problem

Hb measurement and ESA drug dose for 56 ESRD patients were collected from the University of Louisville Kidney Disease Program. Patient's data contain Hb measurement (weekly) and ESA drug dose (0-3 times given per week), each for 52 weeks. Six patients had data for only 15 weeks because of either missing appointment or kidney transplantation, so they were eliminated in this study. First-order to third-order models were developed for each patient using the system identification toolbox in MATLAB. Since there were no meaningful change from the second to third order models, the second-order model was chosen for each patient, which is consistent with the other pharmacodynamic models previously developed for the *erythropoietic process* [95], [104]. To individualize the anemia management problem, a second-order dose-response model depending on unknown parameters k and τ that include a wide range of patients (good, average, and poor responders) are considered. The transfer function is written as

$$G(s) = Y(s)/U(s) = k/(\tau s+1)^2 \tag{5.1}$$

where $Y(s)$ and $U(s)$ are Hb level (output) and ESA dose (input), respectively. The parameter k describes patient's responsiveness to the medication and could vary in the range of 0.2–1 g/dL/1,000 U. High values of k imply the good response to the medication, while low values imply the poor response (insensitivity) to the ESA drug. The parameter τ is the time constant, related to the RBC lifespan, and assumed to vary in the range of 60–120 days, which is consistent with the published clinical data [108], [109]. The following assumptions are also being made: Baseline hemoglobin (Hb_0), the patient's Hb level before starting the ESA treatment, is assumed to be in the range of 7–9 g/dL. The desired hemoglobin (Hb_T) range is considered 10–12 g/dL, recommended by national guidelines [84]. The maximum permissible ESA dose is set to 20,000 U and the ESA dose variation limited to less than 50% of its steady-state value. Also, the Hb variation is limited to ± 0.05 g/dL to provide a more stabilized Hb level. Using the state space model for the transfer function of Eq. (5.1) and considering the aforementioned assumptions, the anemia management problem can be represented in the format of an optimal control problem as to minimize the performance index

$$J = \int_0^{t_f} \left(y_1(t) - Hb_T\right)^2 + (u(t) - u_{ess})^2 dt \ \rightarrow \textit{general form} \ J = \int_0^{t_f} L\left(\boldsymbol{x}(t), u(t)\right) dt \tag{5.2}$$

subject to state dynamics,

$$\dot{\boldsymbol{x}}(t)=\begin{bmatrix}\dot{x}_1(t)\\ \dot{x}_2(t)\end{bmatrix}=\begin{bmatrix}-(2/\tau)x_1(t)-(1/\tau^2)x_2(t)+(k/\tau^2)u(t)\\ x_1(t)\end{bmatrix} \qquad (5.3)$$

$$\rightarrow \textit{general form} \quad \dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t),u(t))$$

mixed state-control path constraints,

$$\left.\begin{array}{l} 0\le u(t)\le 20,000\\ |\dot{u}(t)|\le 0.5u_{ess}\\ |x_1(t)|\le 0.05 \end{array}\right\} \rightarrow \textit{general form}:\ \boldsymbol{q}\left(\boldsymbol{x}(t),u(t)\right)\le \boldsymbol{0} \qquad (5.4)$$

and boundary conditions,

$$\boldsymbol{x}(0)=\begin{bmatrix}x_1(0)\\ x_2(0)\end{bmatrix}=0 \ \rightarrow \textit{general form}:\ \boldsymbol{\gamma}(\boldsymbol{x}(0),\ \boldsymbol{x}(t_f))=\boldsymbol{0} \qquad (5.5)$$

where the outputs $\boldsymbol{y}(t)=[y_1(t)\ y_2(t)]^T=[x_2(t)+Hb_0\ \ x_1(t)]^T$, and $u(t)$, $\dot{u}(t)$, $y_1(t)$, and $y_2(t)$ denote ESA dose, ESA dose derivative, Hb level, and Hb level derivative, respectively. Also, $x_1(t)$ and $x_2(t)$ are states of the system, and u_{ess} is the steady-state value of the ESA drug calculated from the Eq. (5.1) for each individual, i.e. $u_{ess}=(\mathrm{Hb_T}-\mathrm{Hb_0})/k$. To have a more flexible problem, a general form has also been provided for the ESA drug dosing optimal control problem that may even include nonlinear equations for state dynamics, path constraints and boundary conditions, if those equations need to get updated based on the physician's assessment or updated patient model.

5.3 RBF-Galerkin Solution to Anemia Management

To solve the ESA dosing problem with the RBF-Galerkin method, GA RBFs with LGL nodes were used for $N = 40$. The simulation time was set to a year, $(t_f = 365\ days)$, and baseline Hb and Hb target were assumed to be 9 g/dL and 11 g/dL, respectively. Achieved Hb level and weekly dose of the ESA drug produced by the RBF-Galerkin method for three types of patients, including good $(k = 1)$, average $(k = 0.6)$, and poor $(k = 0.2)$ responders, are illustrated in Figure 5.1 and Figure 5.2, respectively. According to Figure 5.1, the RBF-Galerkin method is capable of achieving the Hb target in a reasonable time for all patient groups. For example, a good responder can hit the Hb target within 79 days of starting the treatment.

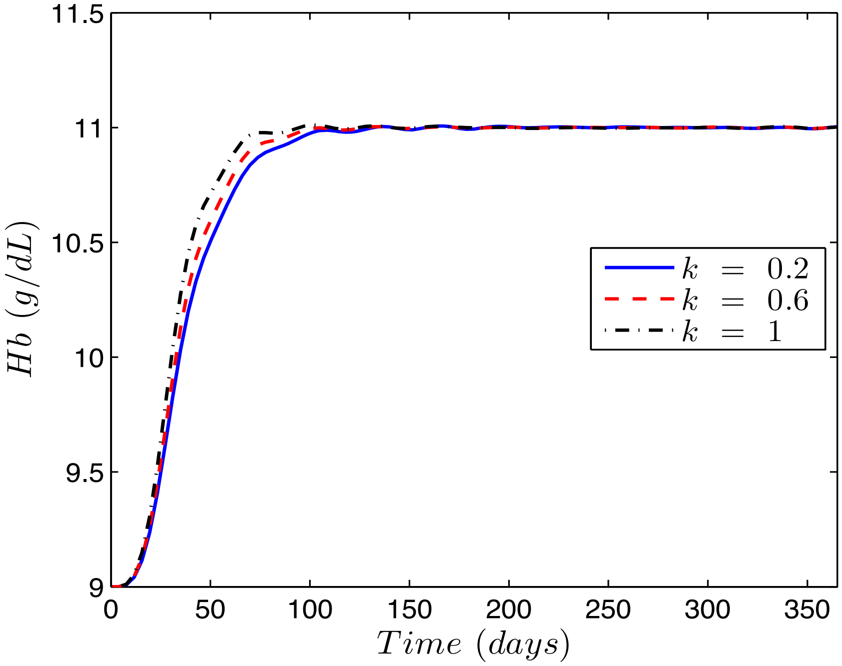


Figure 5.1 Achieved hemoglobin level using the RBF-Galerkin method for three types of simulated patients: poor ($k = 0.2$), average ($k = 0.6$), and good ($k = 1$) responders

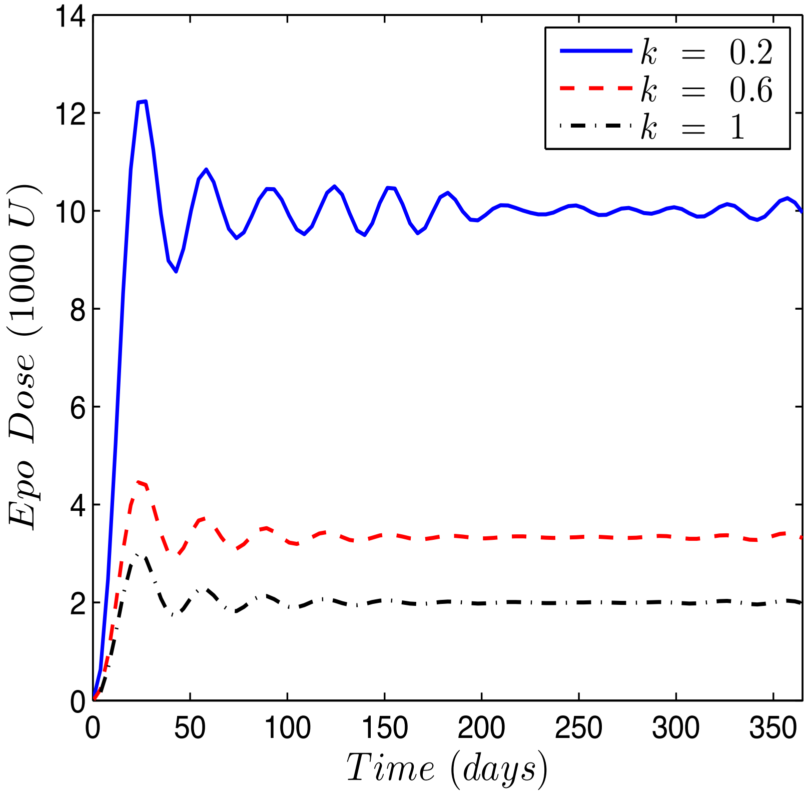


Figure 5.2 ESA drug (EPO) dose recommendations produced by the RBF-Galerkin method for three types of simulated patients: poor ($k = 0.2$), average ($k = 0.6$), and good ($k = 1$) responders

This time frame would increase to 90 days for an average responder and about 106 days for a poor responder patients. As seen in Figure 5.2, the weekly dose recommendations for a poor responder is the highest dose, and it decreases corresponding to patient's response to the medication. The steady-state weekly doses of the ESA drug for good, average and poor responders are 2,000 U, 3,333 U, 10,000 U, and 3,500 U, respectively. Figure 5.3 and Figure 5.4 illustrate the achieved Hb level and ESA dose recommendations for an average responder with different red cell turnover τ, respectively. As expected, by increasing the τ, the time constant of the system is increased, and therefore, the patient's dose-response profile gets slower.

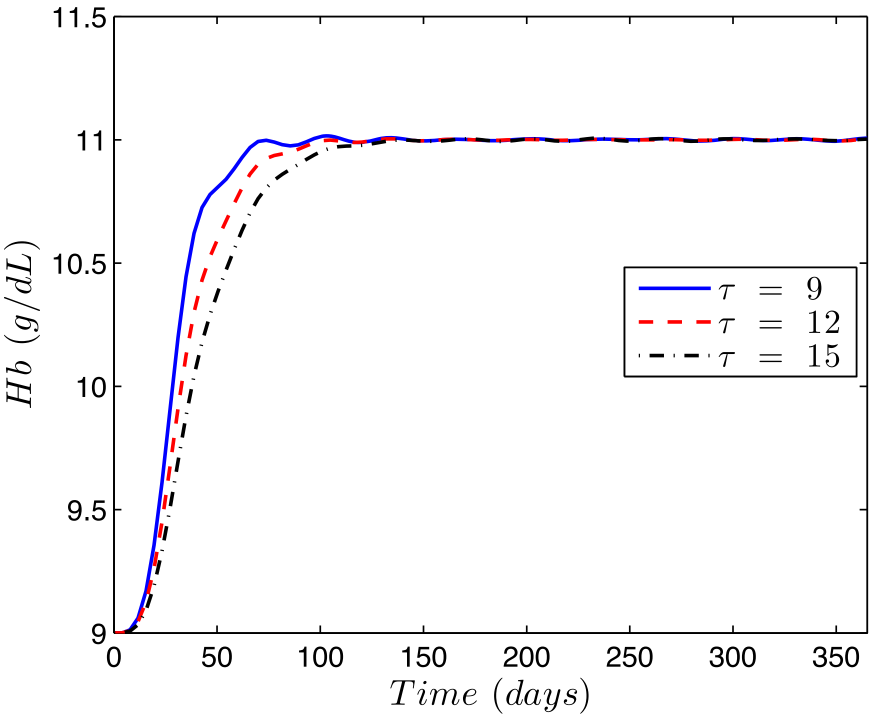


Figure 5.3 Achieved hemoglobin concentration obtained from the RBF-Galerkin method for average responder patients with different red cell turnover, $\tau = 9, 12, 15$

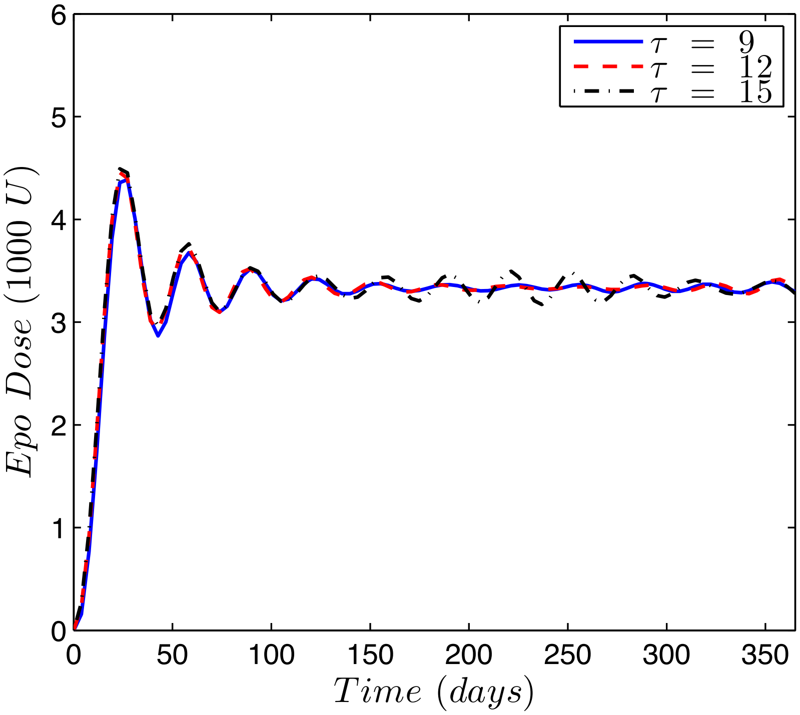


Figure 5.4 ESA drug (EPO) dose recommendations produced by the RBF-Galerkin method for average responder patients with different red cell turnover, $\tau = 9, 12, 15$

5.4 Control Approach

After finding a solution to the ESA administration problem using the RBF-Galerkin method (open-loop control), a receding horizon control (RHC) approach is developed based on the proposed method for individualized anemia management (closed-loop control). RHC, sometimes called model predictive control (MPC), is an advanced control method that has been in use in various applications including chemical and oil industries since the 1980s [110]. RHC is an efficient approach to design an optimization-based controller for constrained multivariable control problems [111]. The RHC approach to anemia management is as follows: The optimal ESA dose sequence $(u_{n+1}^{*}\ u_{n+2}^{*}\ ...\ u_{n+N}^{*})$, where n is the current time instance, is computed by the RBF-Galerkin method from the current state to the desired state over a finite time horizon t_f. However, only the first dose of the ESA sequence produced (i.e. u_{n+1}^{*}) is given to the patient, and the state is updated by measuring the patient's current Hb level (Hb_m). The finite horizon optimization problem will be repeated using the updated state $\boldsymbol{x}_{n+1}$, and the recent control u_{n+1}^{*}, as the initial values for the optimal control problem. The resulting control approach is called the RBF-Galerkin-based RHC method as illustrated in Figure 5.5.

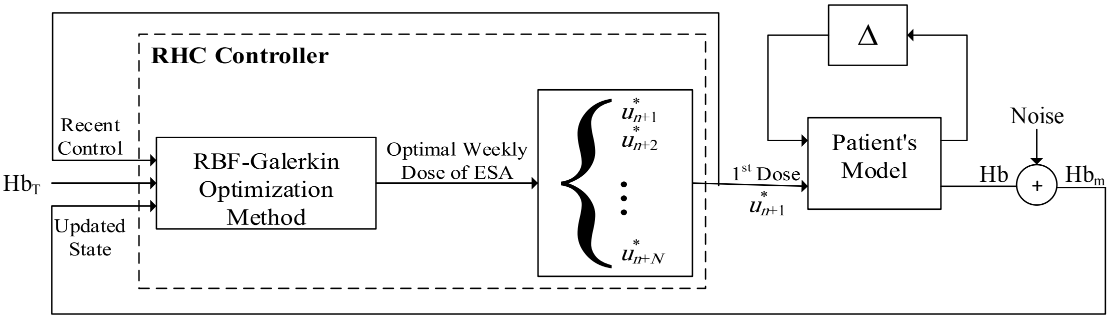


Figure 5.5. RHC controller designed based on the RBF-Galerkin method for anemia management

Multiple Receding Horizon Control (MRHC) Approach: If the patient model is known, the RHC controller shown in Figure 5.5 can properly update the ESA dose adjustments by measuring patient's Hb level regularly. However, the individual-based model developed in Eq. (5.1) is indeed dependent on parameters k and τ, which are unknown for new patients. Therefore, an MRHC approach is proposed to find the weekly dose of the ESA for each individual patient. Considering the responsiveness of each patient to the ESA drug, there would exist three types of patient groups: poor, average, and good responders. MRHC uses three RHC controllers, one for each patient group, to provide a weighted linear combination of each controller output as the recommended ESA dose for each individual patient. The weight of each controller, ω^l, is inversely proportional to the absolute difference between the previous calculated dose of ESA, u_n^*, and the steady-state value of the ESA for each patient group, u_{ess}^l, i.e.,

$$\omega^l = \frac{1}{1+\left|u_{ess}^l - u_n^*\right|} \quad , \quad for \quad l = 1,2,3 \tag{5-6}$$

and the control law (weekly dose of ESA drug, u_{n+1}^*), is calculated as the weighted mean of each controller output,

$$u_{n+1}^* = \frac{\sum_{l=1}^{3} \omega^l u_{n+1}^l}{\sum_{l=1}^{3} \omega^l} \tag{5-7}$$

Where u_{n+1}^l, $l = 1,2,3$, is the current output of each RHC controller. The MRHC control approach for individualized anemia management is illustrated in Figure 5.6. In this approach, each controller represents a different aspect of the dose-response profile, in which RHC 1, RHC 2, and RHC 3 are controllers designed for poor, average, and good responder patients, respectively. In contrast to switching strategy that chooses a single controller output [112], the control action in the proposed approach uses a weighted mean of controller outputs (blending of outputs).

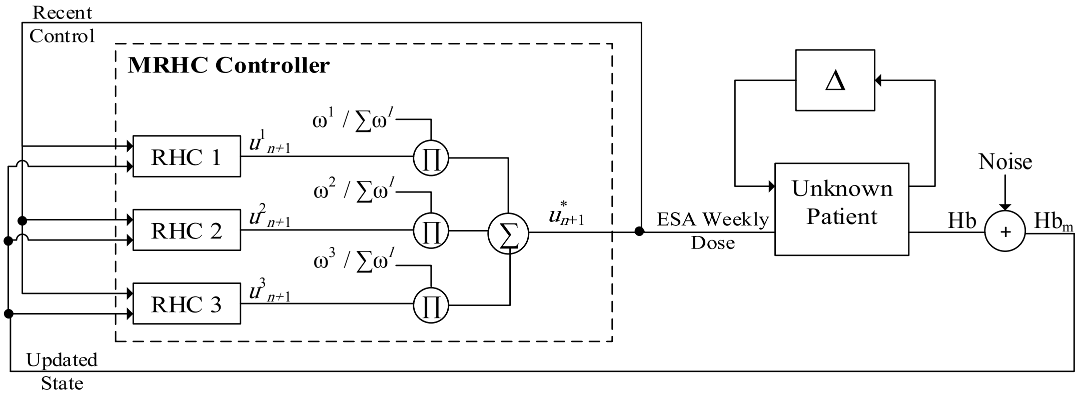


Figure 5.6. Multiple receding horizon control (MRHC) approach for anemia management

5.5 Results

In this Section, the MRHC approach based on the RBF-Galerkin optimization method is applied for individualized drug dosing in the anemia management problem. The simulation results of the proposed method are compared with those obtained from a population-oriented approach (AMP) [84] as well as an individual-based method (Smart Anemia Manager or SAM) [101]. The AMP used for this comparison is a clinical protocol for anemia management that has been in use at the University of Louisville Kidney Disease Program (dialysis facility) from 2011 to 2012. SAM, on the other hand, is an MPC-based individualized method developed by Gaweda et al. [101] using the MATLAB MPC toolbox, which is currently in use at the same dialysis center.

5.5.1 Hb Measurement without Observational Error

The simulation time is set to 15 months (65 weeks) from starting the treatment and the sampling rate fixed at 7 days. Hb_0 (baseline) is considered to be 8 g/dL, and Hb_T

(desired or target) set to 11 g/dL (the midpoint for the Hb recommended range 10-12 g/dL). In addition, it is assumed that the Hb_m (measured) is error free, so there would be no noise in the output. In dialysis facilities, Hb is usually measured weekly, but the ESA doses only be adjusted once in a month or every four weeks. To make the simulation results similar to the real case scenarios, the same regulations have been used here (i.e. dose adjustment of every four weeks and weekly measurement of Hb). Achieved Hb levels and ESA dose adjustments computed from MRHC, SAM, and AMP are shown in Figure 5.7 and Figure 5.8, respectively. Also, three different responders including good $(k = 0.94)$, average $(k = 0.5)$, and poor $(k = 0.3)$ responders are considered for the comparison.

5.5.2 Hb Measurement with Observational Error

Similar to Part A, $t_f = 65\ \text{weeks}$, $\text{Hb}_0 = 8\ \text{g/dL}$, $\text{Hb}_\text{T} = 11\ \text{g/dL}$, and sampling rate of 7 days are considered for the simulation. Also, it is assumed that the Hb concentration is measured weekly and the ESA dose adjusted every four weeks. However, Hb_m is assumed to be contaminated with the measurement error. A white Gaussian noise with the maximum amplitude of 0.5 g/dL is added to the output (i.e., -0.5 g/dL ≤ Hb error ≤ +0.5 g/dL), which is a realistic assumption for the Hb error measured weekly as part of a routine blood test. Achieved Hb level and ESA dose recommendations obtained from MRHC, SAM, and AMP in the presence of Hb measurement errors for good, average, and poor responders are illustrated in Figure 5.9 and Figure 5.10, respectively.

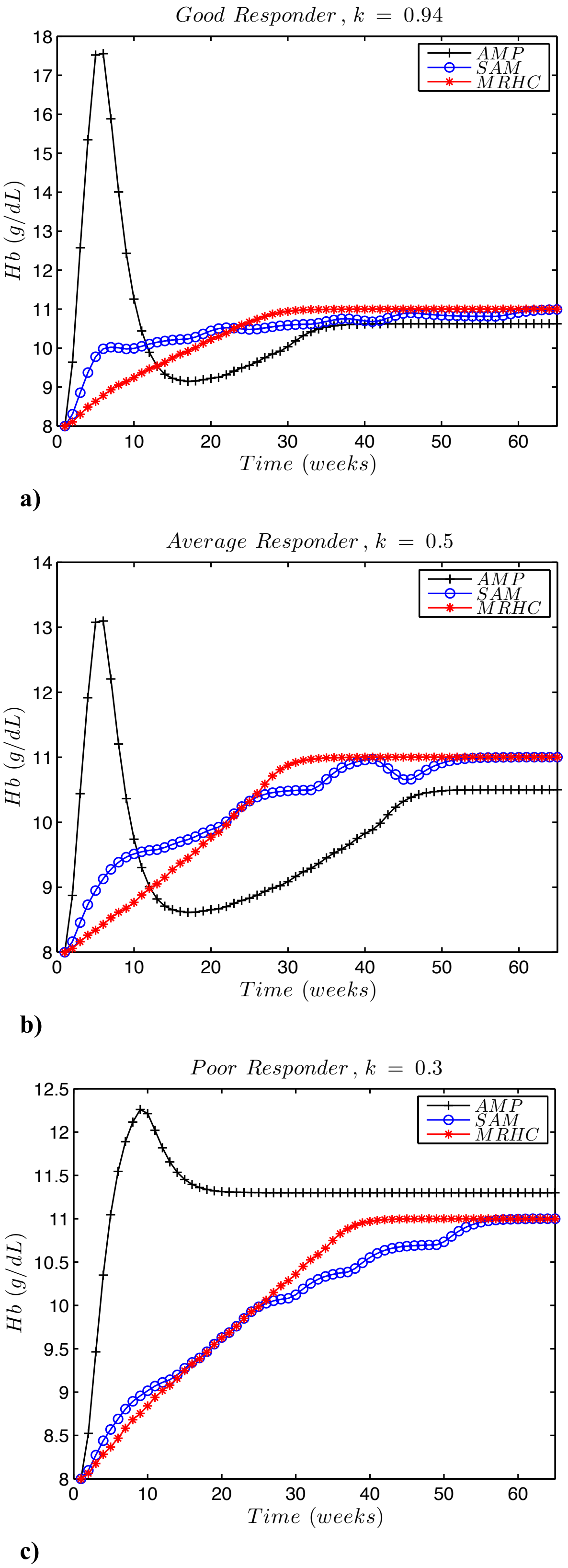
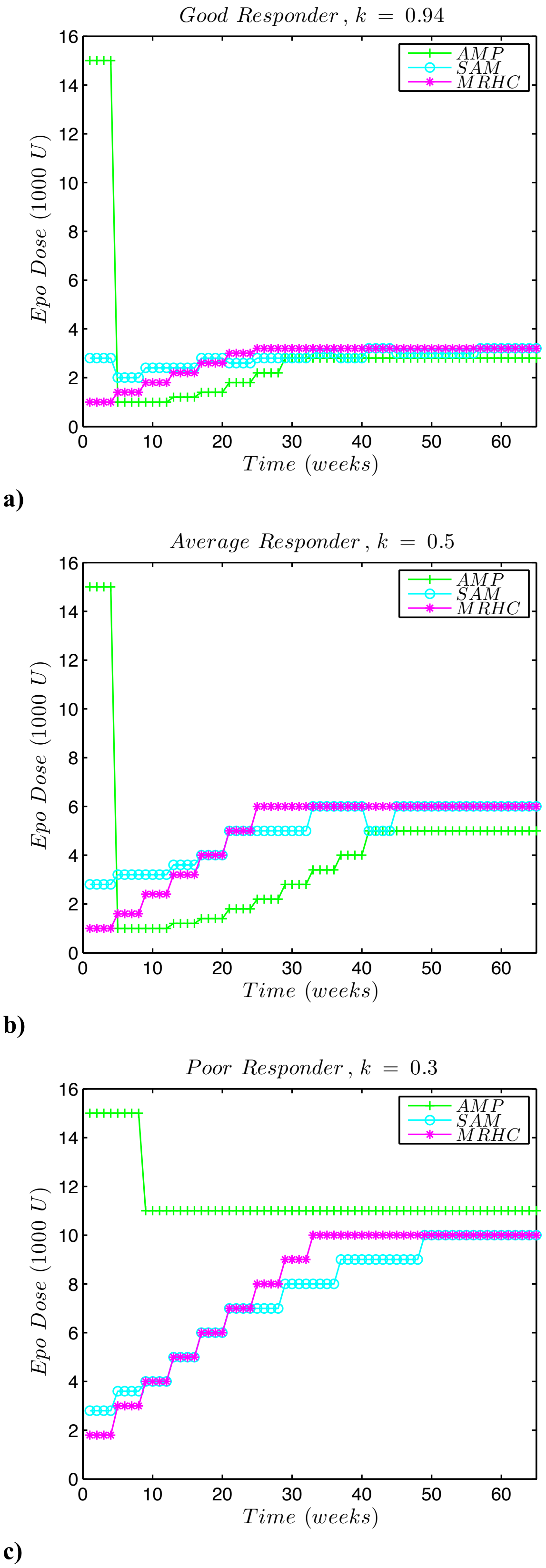


Figure 5.7. Achieved Hb level obtained from MRHC (proposed method), SAM, and AMP for a) good responder, b) average responder, and c) poor responder patients


Good Responder, $k = 0.94$

a)

Average Responder, $k = 0.5$

b)

Poor Responder, $k = 0.3$

c)

Figure 5.8. ESA drug (EPO) dose adjustments computed by MRHC (proposed method), SAM, and AMP for a) good responder, b) average responder , and c) poor responder patients

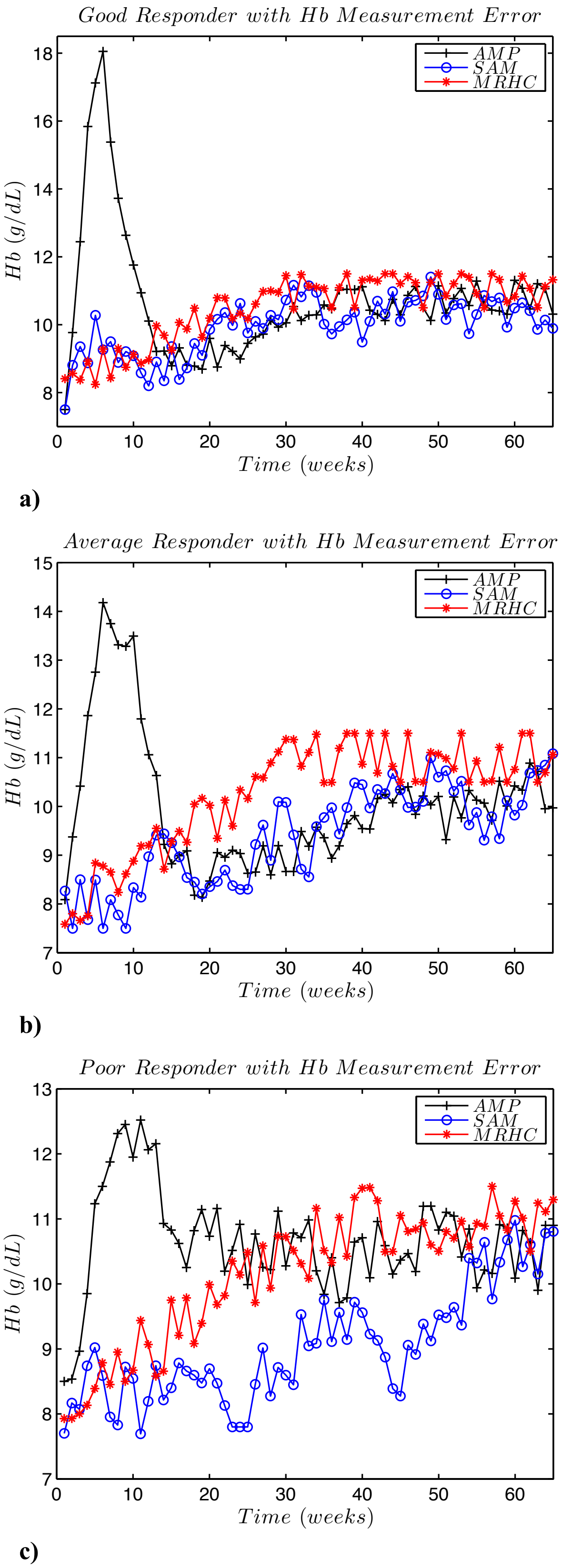
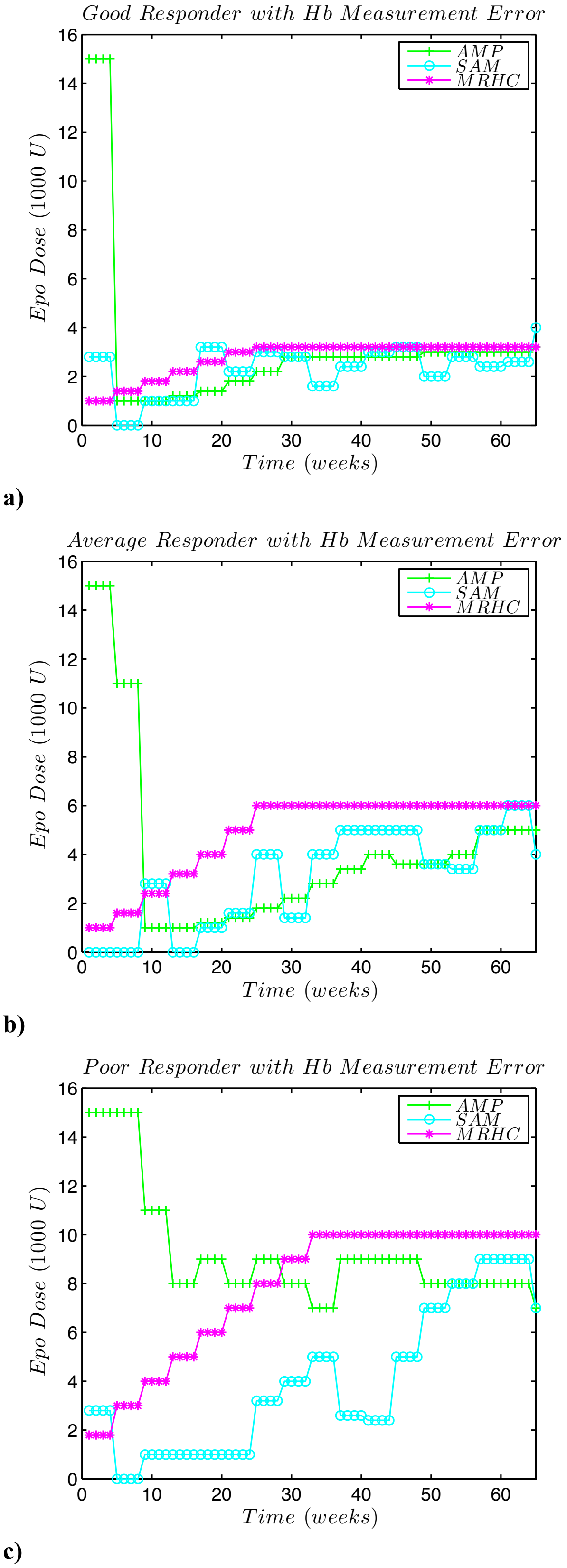


Figure 5.9. Achieved Hb level with Hb measurement error obtained from MRHC, SAM, and AMP for a) good responder, b) average responder, and c) poor responder patients



a)

b)

c)

Figure 5.10. ESA drug (EPO) dose adjustments in presence of Hb error computed by MRHC, SAM, and AMP for a) good responder, b) average responder, and c) poor responder patients

5.6 Discussions

5.6.1 Hb Measurement without Observational Error

According to Figure 5.7, while the achieved Hb level from the AMP is not that impressive, both MRHC and SAM approaches can successfully attain the Hb target of 11 g/dL. There would also be a considerable difference between MRHC and SAM in terms of the time required to achieve the desired Hb level. While for good, average, and, poor responder patients, it took around 44, 51, and 56 weeks, respectively, to hit the Hb target by using the SAM, this timeframe would improve to 30, 33, and 40 weeks for the same patient groups using the MRHC. This faster response clearly demonstrates the higher efficiency of the proposed method for achieving the desired Hb level, as compared to SAM. In addition, the Hb levels obtained from SAM exhibits small variations for all patient groups, whereas those obtained from MRHC are more monotone and uniformly increasing to the desired level. For all three patient groups (good, average, and poor responders), the performance of AMP is not comparable with two individual-based methods in terms of precise control of Hb concentration. Hb levels achieved by AMP show wide undesirable fluctuations around the Hb_T, especially for the initial weeks of treatment. Another drawback of AMP is that it would not be able to exactly achieve the desired Hb level (11 g/dL) and only maintain the Hb concentration within a range, compared to MRHC and SAM hitting the Hb target.

Figure 5.8 illustrates ESA dose adjustments recommended by MRHC, SAM, and AMP. According to Figure 5.8, both initial and steady-state doses recommended by the AMP are quite inaccurate. In particular, those unnecessary high doses recommended for the initial weeks of treatment can be associated with the increased health care costs as well as the higher risk of cardiovascular problems for CKD patients. In contrast, both MRHC and SAM have less aggressive dose recommendations throughout the treatment starting from the low dose and achieving the steady-state dose of the anemia drug for all three patient groups. For good and average responders, while ESA dose recommendations computed by SAM exhibit undesirable fluctuations around the steady-state level, those produced by MRHC tend to be more consistent. For all three patient groups, doses recommended by MRHC achieve the steady-state level much faster than those generated by the other two methods. In addition, ESA doses produced by MRHC are more stable and uniform than those produced by either SAM or AMP, which is more desirable for the EPO therapy, starting from the lower doses and uniformly increasing to the steady-state level.

5.6.2 Hb Measurement with Observational Error

According to Figure 5.9.a, all three methods have quite acceptable Hb concentrations achieved for a good responder patient, among which MRHC provides the fastest response with the least fluctuations, and AMP produces the lowest response with the most oscillations. For an average responder, MRHC is significantly faster than the other two methods for attaining the Hb level fairly close to the target. Also, the Hb steady-state level achieved by the MRHC is more accurate than that of the either two methods for an average responder (see Figure 5.9.b). AMP and SAM seem to have similar performances for achieving the Hb target for an average responder, with the exception that AMP has unnecessary fluctuations for the initial weeks of treatment. For poor responders, MRHC still acts better than the other two methods for rejecting the noise and achieving the desired Hb level within a reasonable time, according to Figure 5.9.c. While the Hb concentration obtained from SAM cannot reach the steady-state level within the simulation time (15 months), AMP would be able to keep the Hb in range after around 25 weeks, for a poor responder (Figure 5.9.c). However, considering those unnecessary high Hb concentrations for the initial weeks, performance of the AMP is still less efficient than MRHC for the poor responder patients.

Figure 5.10 demonstrates ESA dose adjustments recommended by MRHC, SAM, and AMP for different responders with the maximum Hb error of ±0.5 g/dL per measurement. As expected, initial doses recommended by AMP are unnecessarily high for all patient groups.

For good responders (Figure 5.10.a), AMP acts slightly better than SAM in rejecting the noise and finding the appropriate weekly dose, but for average (Figure 5.10.b) and poor responders (Figure 5.10.c), doses computed by both methods have undesirable fluctuations and need a relatively long time to achieve the steady-state level. On the other hand, ESA dose adjustments by the MRHC tend to be more stable and accurate in the presence of measurement errors and also reach the steady-state value notably faster than those generated by the other two methods. More interestingly, comparing ESA doses recommended by MRHC for good, average, and poor responders for the Hb measurement with error (Figure 5.10) and without error (Figure 5.8) reveals that weekly doses are exactly the same for similar patient groups. Therefore, ESA dose adjustments using the MRHC approach are quite robust for the anemia management problem.

5.7 Statistical Comparison

In this section, 40 hypothetical CKD patients with different responsiveness to medication, k, and red cell turnover, τ, are considered for the simulation. Also, three different methods including MRHC, SAM, and AMP are applied to control the Hb level of patients. It is assumed that all patients have the baseline Hb of 9 g/dL and the goal is to attain the target Hb of 11.5 g/dL. The simulation is divided into two parts. The first 12 months are called the transient cycle (even though it usually takes less time for all methods to attain the Hb target, transient cycle is assumed 12 months to ensure the Hb steady-state level has been achieved), followed by the steady-state period for the next 6 months. A random error of ±5% is added to the simulation to account for the measurement errors as well as factors that have not been included in the model such as infections or hospitalizations.

Table 5.1 compares the mean values for average achieved Hb level (per patient), standard deviation of achieved Hb level (per patient), and absolute difference between the average achieved Hb level and Hb target (11.5 g/dL) obtained from MRHC, SAM, and AMP for the steady-state period of treatment. Among mean values of the average achieved Hb level, MRHC achieves the closest value to the target, while AMP results in the lowest mean, which is still in the range but far from the target.

Table 5.1 Statistical comparison of means of MRHC, SAM, and AMP for anemia management

	MRHC	SAM	AMP
Average Achieved Hb Level (g/dL)	11.432±0.043	11.237±0.094	10.558±0.104
SD for Achieved Hb Level (g/dL)	0.358±0.010	0.468±0.021	0.391±0.015
Absolute diff. Achieved Hb & Target Hb (g/dL)	0.117±0.031	0.276±0.090	0.942±0.104

Comparison of mean values for the standard deviation of the Hb level reveals that MRHC and AMP have less variation of the achieved Hb level than SAM, which is expected considering the fluctuating output of SAM in presence of measurement error, especially for poor responders (see also Figure 5.9.c). Comparing the absolute difference between the achieved Hb level and Hb target, which is a measure of accuracy for methods, shows that the proposed method achieves the lowest value of 0.117 ± 0.031 g/dL, and hence is very successful in achieving the desired Hb level compared to SAM and AMP. It simply means that by choosing the RBF-Galerkin-based MRHC approach among these three methods:

1- The average Hb level achieved for each simulated patient is relatively close to 11.5 g/dL (efficacy of proposed method).

2- Standard deviation of the achieved Hb level for each simulated patient would be relatively close to zero, i.e. the lowest standard variation among three methods, (reliability of proposed method).

The F-test from analysis of variance (ANOVA) is applied to test the hypothesis about the equality of mean values of average achieved Hb level from MRHC, SAM, and AMP. The ANOVA results in F-Value=112.94 and P-value=0.000, meaning that mean values are significantly different. Post-ANOVA pairwise comparison of means using the Tukey test with 99% confidence interval (CI) is demonstrated in Figure 5.11. Please note that if an interval does not contain zero, the corresponding means are significantly different. Figure 5.11 clearly indicates that the mean of the average achieved Hb level from each method is significantly different than that of the other two methods.

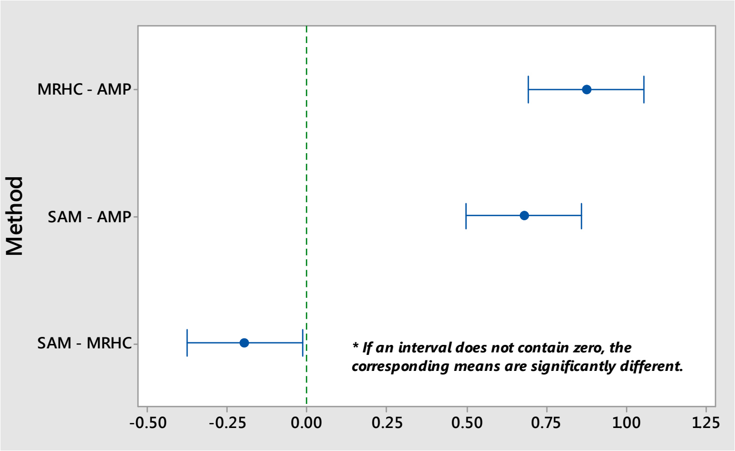


Figure 5.11. Differences of means for the average achieved Hb level (Tukey Test with 99% CIs)

It is also tested the hypothesis about the equality of mean values of the absolute difference between the average achieved Hb level and Hb target for MRHC, SAM, and AMP. ANOVA results reveal that mean values are notably different (F-Value=110.98 and P-value=0.000). Also, Tukey method with 95% CIs indicates that there are considerable differences between the means of MRHC and AMP, also between the means of MRHC and SAM as well as means of SAM and AMP, for the absolute difference between the achieved Hb level and Hb target (see Figure 5.12). This statistical comparison verifies the efficiency of the proposed method for achieving the desired Hb level for the anemia management problem.

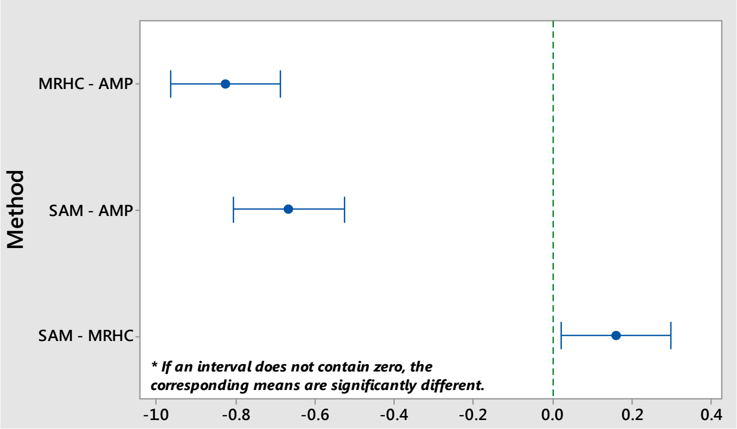


Figure 5.12. Differences of means for the absolute difference between the achieved Hb level and Hb target level (Tukey Test with 95% CIs)

CHAPTER VI

CONCLUSION AND FUTURE WORK

6.1 Conclusion

In this dissertation, two direct transcription methods based on the RBF interpolation have been proposed for solving optimal control problems numerically: RBF Collocation Method and RBF-Galerkin Method. Also, a multiple receding horizon control (MRHC) approach based on the RBF-Galerkin method was proposed for solving an important drug dosing problem: individualized anemia management in chronic kidney disease. Key features and important results of each method along with the results of the individualized anemia management are summarized in the following.

6.1.1 RBF Collocation Method

The first method proposed for solving optimal control problems was the RBF collocation method, or simply the RBF method. The proposed approach satisfies defect constraints at discretized nodes by combining the RBF parameterization of states and controls with arbitrary discretization at collocation points to eventually transcribe the continuous-time optimal control problem into a discretized NLP problem. The resulted NLP can then be solved by efficient NLP solvers available. For this work, SNPOT, a sparse solver, was used for solving the NLP.

Performance of the RBF method was tested and verified through both fixed-final-time and free-final-time numerical examples, including temperature control problem, brachistochrone example, Vanderpol example, and robot motion planning problem. For the temperature control problem and brachistochrone example, the exact solutions were available from analytical approaches, so the accuracy of the RBF method was confirmed by comparing the cost values of the proposed method with those obtained from exact solutions. In the Vanderpol oscillator example, the accuracy of the RBF method has been superior to a local polynomial method (NTG). In addition, the proposed method was more computationally efficient than a global polynomial method (DIDO) for the same accuracy. The RBF method was then applied to a robot motion planning problem with obstacle avoidance to evaluate the efficiency of the method for solving a trajectory optimization problem with nonlinear path constraints. The results were compared with those obtained from PROPT and DIDO, two MATLAB-based commercial off-the-shelf software packages using pseudospectral methods for trajectory optimization. The accuracy of the RBF method was higher than either of the other two methods for different number of optimization parameters. Also, the computation time of the proposed method was competitive with two other methods (much less than DIDO, but slightly higher than PROPT).

6.1.2 RBF-Galerkin Method

The second proposed method was the RBF-Galerkin method, in which residuals were handled differently. Similar to the RBF method, global RBFs have been used for parameterization and arbitrary nodes utilized for discretization. However, by applying the

Galerkin error projection to residuals (i.e., defect constraints as well as residuals of path constraints), they were set to be orthogonal to every member of the RBF basis functions. In fact, by applying the Galerkin error projection, residuals are minimized in L^2-norm sense (in the limit $N \to \infty$, they converge to zero in the mean). It was also shown that there would be an exact equivalency between the KKT optimality conditions of the NLP resulted from the RBF-Galerkin method and discretized form of the first-order necessary conditions of the optimal control problem, if a set of transversality conditions were added to the KKT conditions. RBF-Galerkin Costate Mapping Theorem has been developed stating that KKT multipliers of the NLP would be exactly equivalent to the Lagrange multipliers of the optimal control problem at discretization points, if a set of conditions hold. It means that the solution of the direct transcription by the RBF-Galerkin method is the same as the solution of an indirect method for solving optimal control problems (verifying the optimality of the solution).

Two numerical examples were solved by the RBF-Galerkin method. The first example was a bang-bang optimal control problem with an analytical solution. The optimal control for this example had a discontinuity point making it difficult to solve using the classic pseudospectral methods. The RBF-Galerkin solution with a set of arbitrary random points has been shown to be very accurate for this example, even for those points located near the control discontinuity. Also, the costates estimated by the RBF-Galerkin costate mapping theorem was accurate enough to numerically verify the theorem developed (costate maximum errors over all nodes $\approx 10^{-5}$ for N=40). The proposed method was then applied to a 2D UAV navigation problem with obstacle avoidance and the solution compared with those obtained from a Legendre pseudospectral method and a B-spline approach. The RBF-Galerkin method was more accurate and also computationally more efficient than the other two methods for the UAV navigation problem.

6.1.3 Individualized Anemia Management

After those successful examples solved by the RBF-Galerkin method, the proposed approach was applied to solve an optimal control drug delivery problem. A multiple receding horizon control (MRHC) approach based on the RBF-Galerkin optimization method has been proposed for individualized drug dosing in the anemia management problem. Anemia management has been formulated as a constrained optimal control problem solved by the RBF-Galerkin method. Then a multiple receding horizon controller was built based upon the optimization algorithm to precisely control and achieve the desired Hb concentration for individual patients. Simulation results have been compared with those obtained from a population-oriented approach (AMP) as well as an individual-based method (SAM) for anemia management to verify the efficiency of the proposed method. In silico comparison between the proposed method and two other approaches has indicated that the performance of the RBF-Galerkin based MRHC has been superior to either of the other two methods for both precise control of the Hb level and accurate adjustments of the ESA dose. Hb steady-state level achieved by the MRHC approach in presence of the measurement error has been more accurate than that of the either two methods, especially for average and poor responder patients. Also, ESA dose recommendations by the proposed method were more consistent, uniform and accurate than those provided by SAM and AMP, and also achieved the steady-state value notably faster than those generated by the other two methods.

Statistical comparison between three different methods has also revealed that mean of the average achieved Hb level from the MRHC approach has been significantly closer to the Hb target than that of the other two methods. Also, the results of Tukey test with 95% CI indicated that the absolute difference between the achieved and target Hb for the proposed method was notably lower than those for SAM and AMP, confirming the efficacy of the proposed method for the anemia management problem.

6.2 Future Work

6.2.1 Tuning the Shape Parameter in RBF Based Methods

Both RBF method and RBF-Galerkin approach use global RBFs for parameterization of states and controls. Global RBFs, including GA RBFS, MQ, and IMQ, contain a free shape parameter ε that plays an important role for the accuracy of an RBF interpolation. In the RBF and RBF-Galerkin methods proposed, the shape parameter is often tuned by trial and error or sometimes by using a simple “for” loop in the code. However, it would have been more convenient and more efficient to select this parameter by using a systematic approach. A few strategies were proposed in the literature for choosing the appropriate shape parameter, for instance, Leave One Out Cross Validation (LOOCV) algorithm [113] and its extensions [114] or Maximum Likelihood Estimator (MLE) [115]. These strategies try to minimize a predictor function (type of function is different in each approach) that mimics the RBF interpolation error. Finding a proper strategy to automate fine-tuning of the shape parameter in the RBF-Galerkin method could be a possible extension of this work.

6.2.2 Suggestions for the Anemia Management Problem

A. Combining System Identification Techniques with RBF-Galerkin based RHC

The second-order model, dependent on unknown parameters k and τ, used for patients was developed based on the erythropoietic process and dose-response profile of each patient. The patient model could vary during the treatment, and therefore, system identification methods may be investigated as systematic tools for adjusting those parameters or even for developing a more accurate nonlinear model for each individual patient. Unlike a regular MPC that can only use a linear model for its design, the RHC controller proposed in this work can apply any linear as well as nonlinear models for the control purpose (nonlinear MPC). It's because the RBF-Galerkin method can solve general optimal control problems with nonlinear dynamics. Applying the RHC controller proposed in this dissertation in combination with a more accurate nonlinear patient model could be interesting to investigate.

B. Optimal Control of Drug Administration for CKD Patients with Comorbidities

Anemia is a common complication of CKD. Besides that, many CKD patients may suffer from other health problems or comorbidities, one or more health conditions or diseases existing alongside another disease. For example, diabetes and cardiovascular disease were reported as common comorbidities of CKD in surveys [116]. Those patients with comorbidities may need to follow different dosage recommendations as a result of multiple health conditions. In addition, they may take multiple medicines for their treatment, which makes the EPO therapy for anemia management even more challenging. It could be a topic of future research to find the optimal dosages of the ESA drug for CKD patients with multiple health conditions or comorbidities.

REFERENCES

[1] T. A. Weber, *Optimal control theory with applications in economics*. MIT Press, Cambridge, MA, 2011.

[2] Y. Batmani and H. Khaloozadeh, “Optimal drug regimens in cancer chemotherapy: a multi-objective approach,” *Comput. Biology Medicine*, vol. 43, no. 12, pp. 2089–2095, 2013.

[3] J. T. Betts, “Survey of numerical methods for trajectory optimization,” *J. Guid. Control Dynam.,* vol. 21, pp. 193–207, Mar.–Apr. 1998.

[4] A. V. Rao, “A survey of numerical methods for optimal control,” *Advanc. Astronaut. Sci.,* vol. 135, no. 1, pp. 497–528, 2010.

[5] U. M. Ascher, R. M. Mattheij, and R. D. Russell, *Numerical solution of boundary-value problems in ordinary differential equations.* Philadelphia: SIAM Press, 1995.

[6] M. Majji, J. D. Turner, and J. L. Junkins, “Solution of two-point boundary value problems using Lagrange implicit function theorem,” *J. Guid. Control Dynam.,* vol. 32, no. 5, pp. 1684–1687, Sept.–Oct. 2009.

[7] L. Kawecki and T. Niewierowicz, “Hybrid genetic algorithm to solve the two point boundary value problem in the optimal control of induction motors,” *IEEE Latin America Trans.,* vol. 12, no. 2, pp. 176–181, 2014.

[8] P. J. Enright and B. A. Conway, “Discrete approximations to optimal trajectories using direct transcription and nonlinear programming,” *J. Guid. Control Dynam.*, vol. 19, pp. 994–1002, Jul.–Aug. 1996.

[9] R. Dai, “B-splines based optimal control solution,” *AIAA Guid., Navigation, Control Conf.*, August 2010, Toronto, Ontario, CA.

[10] G. N. Elnagar, M. A. Kazemi, and M. Razzaghi, “The pseudospectral Legendre method for discretizing optimal control problems,” *IEEE Trans. Autom. Control,* vol. 40, no. 10, pp. 1793–1796, 1995.

[11] F. Fahroo and I. M. Ross, “Direct trajectory optimization by a Chebyshev pseudospectral method,” *J. Guid. Control Dynam.,* vol. 25, no. 1, pp. 160–166, Jan.–Feb. 2002.

[12] P. Williams, “Jacobi pseudospectral method for solving optimal control problems,” *J. Guid. Control Dynam.*, vol. 27, no. 2, pp. 293–297, 2004.

[13] Q. Gong, W. Kang, and I. M. Ross, "A pseudospectral method for the optimal control of constrained feedback linearizable systems," *IEEE Trans. Autom. Control,* vol. 51, pp. 1115–1129, July 2006.

[14] F. Fahroo and I. M. Ross, "Pseudospectral methods for infinite-horizon nonlinear optimal control problems," *J. Guid. Control Dynam.*, vol. 31, no. 4, pp. 927–936, 2008.

[15] D. Garg, M. A. Patterson, W. W. Hager, A.V. Rao, D. A. Benson, and G. T. Huntington, "A unified framework for the numerical solution of optimal control problems using pseudospectral methods," *Automatica*, vol. 46, no. 11, pp. 1843–1851, Nov. 2010.

[16] C. G. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang, *Spectral methods: evolution to complex geometries and applications to fluid dynamics*. Heidelberg, Germany: Spinger-Verlag, 2007.

[17] W. Chen, Z. -J. Fu, and C. S. Chen, *Recent advances in radial basis function collocation methods.* Springer, 2014.

[18] M. D. Buhmann, *Radial basis functions: theory and implementations.* Cambridge University Press, 2003.

[19] J. A. Rad, S. Kazem, and K. Parand, "Radial basis functions approach on optimal control problems: a numerical investigation," *J. Vibration & Control,* vol. 20, no. 9, pp. 1394–1416, 2014.

[20] A. Heydari and S.N. Balakrishnan, "Fixed-final-time optimal tracking control of input-affine nonlinear systems," *Neurocomputing*, vol. 129, pp. 528–539, Apr. 2014.

[21] C. Y. Kaya, "Control in the plane and time-optimal bang-bang control," PhD Thesis, University of Western Australia, Nedlands, W.A. 6907, Australia, 1995.

[22] C. Y. Kaya and J. L. Noakes, "Computational method for time-optimal switching control," *J. Optimization Theory & Applications,* vol. 117, no. 1, pp. 69–92, April 2003.

[23] H. G. Bock and K. J. Plitt, "A multiple shooting algorithm for direct solution of optimal control problems," in *Proc. 9th IFAC World Congress*, Budapest, Hungary, 1984, pp. 243–247.

[24] C. R. Hargraves and S. W. Paris, "Direct trajectory optimization using nonlinear programming and collocation," *J. Guid. Control Dynam.,* vol. 10, no. 4, pp. 338–342, 1987.

[25] B. R. Geiger, J. F. Horn, G. L. Sinsley, J. A. Ross, L. N. Long, and A. F. Niessner, "Flight testing a real-time direct collocation path planner," *J. Guid. Control Dynam.,* vol. 31, no. 6, pp. 1575–1586, Nov.–Dec. 2008.

[26] M. B. Milam, K. Mushambi, and R. M. Murray, “A new computational approach to real-time trajectory generation for constrained mechanical systems,” in *Proc. IEEE Conf. Decision & Control,* Sydney, Australia, Dec. 2000, pp. 845–851.

[27] H. B. Keller, “Numerical solution of two point boundary value problems,” *CBMS-NSF Regional Conf. Series in Applied Mathematics,* SIAM, 1976.

[28] J. Stoer and R. Bulirsch, *Introduction to numerical analysis*. Springer-Verlag, 2002.

[29] W. Grimm and A. Markl, “Adjoint estimation from a multiple-shooting method,” *J. Optimization Theory & Applications*, vol. 92, pp. 263–283, Feb. 1997.

[30] E. D. Dickmanns, “Efficient convergence and mesh refinement strategies for solving general ordinary two-point boundary value problems by collocated Hermite approximation,” *2nd IFAC Workshop Optimisation*, Oberpfaffenhofen, Germany, Sept. 1980.

[31] W. H. Press, B. P. Flannery, S. Teukolsky, and W. T. Vetterling, *Numerical recipes: The art of scientific computing*. Cambridge University Press, 1990.

[32] E. D. Dickmanns and K. H. Well, “Approximate solution of optimal control problems using third-order Hermite polynomial functions,” in *Proc. 6th Technical Conf. Optimization Techniques*, vol. IFIP-TC7, Springer–Verlag, New York, 1975.

[33] G. L. Brauer, D. E. Cornick, and R. Stevenson, “Capabilities and applications of the program to optimize simulated trajectories (POST),” *NASA CR-2770,* Feb. 1977.

[34] J. C. Butcher, “Implicit Runge-Kutta processes,” *Mathematics of Computation,* vol. 18, no. 85, pp. 50–64, 1964.

[35] J. C. Butcher, *Numerical methods for ordinary differential equations.* New York: John Wiley and Sons, 2008.

[36] M. Fliess, J. Levine, P. Martin, and P. Rouchon, “Flatness and defect of nonlinear systems: introductory theory and examples,” *Intern. J. Control,* vol. 61, no. 6, pp. 1327–1361, 1995.

[37] M. Fliess, J. Levine, P. Martin, and P. Rouchon, “A Lie-Backlund approach to equivalence and flatness of nonlinear systems,” *IEEE Trans. Autom. Control,* vol. 44, no. 5, pp. 928–937, 1999.

[38] C. de Boor, *A practical guide to splines.* Springer-Verlag, 1978.

[39] P. E. Gill, W. Murray, M. A. Saunders, and M. Wright, “User's guide for NPSOL 5.0: a Fortran package for nonlinear programming,” Systems Optimization Laboratory, Stanford Univ., Stanford, CA, 1998.

[40] M. E. Flores and M. B. Milam, "Trajectory generation for differentially flat systems via NURBS basis functions with obstacle avoidance," *American Control Conf.*, Minneapolis, Minnesota, 2006.

[41] C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang, *Spectral methods in fluid dynamics*. Heidelberg, Germany: Spinger-Verlag, 1988.

[42] B. R. Geiger, "Unmanned aerial vehicle trajectory planning with direct methods," PhD dissertation, Penn. State Univ., 2009.

[43] A. L. Herman and B. A. Conway, "Direct optimization using collocation based on high-order Gauss-Lobatto quadrature rules," *J. Guid. Control Dynam.,* vol. 19, no. 3, pp. 592–599, 1996.

[44] P Williams, "Hermite-Legendre-Gauss-Lobatto direct transcription in trajectory optimization," *J. Guid. Control Dynam.*, vol. 32, no. 4, pp. 1392–1395, 2009.

[45] C. L. Darby, W. W. Hager, and A. V. Rao, "Direct trajectory optimization using a variable low-order adaptive pseudospectral method," *J. Spacecraft Rockets,* vol. 48, no. 3, pp. 433–445, 2011.

[46] B. Fornberg, *A practical guide to pseudospectral methods.* Cambridge Univ. Press, Cambridge, 1998.

[47] J. B. Dennis, "Mathematical programming and electrical networks," Massachusetts Inst. Techn. (MIT), Ph.D. thesis, 1959.

[48] L. O. Chua and G. N. Lin, "Nonlinear programming without computation," *IEEE Trans. Circuits Syst.*, vol. CAS-31, no. 2, pp. 182–188, Feb. 1984.

[49] D. W. Tank and J. J. Hopfield, "Simple neural optimization networks: an A/D converter, signal decision circuit, and a linear programming circuit," *IEEE Trans. Circuits Syst.*, vol. CAS-33, no. 5, pp. 533–541, May 1986.

[50] M. P. Kennedy and L. O. Chua, "Neural networks for nonlinear programming," *IEEE Trans. Circuits Syst.*, vol. 35, no. 5, pp. 554–562, 1988.

[51] W. Lillo, S. Hui, and S. Zak, "Neural networks for constrained optimization problems," *Intern. J. Circuit Theory and Applications*, vol. 21, pp. 385–399, 1993.

[52] D. G. Luenberger, *Linear and nonlinear programming*, Reading, MA, Addison-Wesley, 1984.

[53] S. Effati and M. Baymani, "A new nonlinear neural network for solving quadratic programming problems," *Applied Math. Computation*, vol. 165, no. 3, pp. 719–729, 2005.

[54] J. Reifman and E. E. Feldman, "Nonlinear programming with feedforward neural networks," in *Proc. Intern. Joint Conf. Neural Networks*, Washington, DC, July 1999, vol. 1, pp. 594–598.

[55] I. C. Yeh, “Design of high performance concrete mixture using neural networks and nonlinear programming,” *J. Computing Civil Eng.*, vol. 13, no. 1, pp. 36–42, 1999.

[56] M. Niestroy, “Optimal feedback controller approximation using neural networks and nonlinear programming techniques,” in *Neural Networks Proc., IEEE World Congress on Computational Intelligence*, Anchorage, AK, May 1998, vol. 3, pp. 2110–2115.

[57] C. Y. Seong and B. Widrow, “Neural dynamic optimization for control systems-part iii: Applications,” *IEEE Trans. Syst. Man Cyber.- Part B: Cyber.,* vol. 31, no. 4, pp. 502–513, Aug. 2001.

[58] X. Peng, A. Verma, and R. J. Mayer, “Neural dynamic optimization for autonomous aerial vehicle trajectory design,” in *Proc. SPIE 6576*, *Independent Component Analyses, Wavelets, Unsupervised Nano-Biomimetic Sensors, Neural Networks V,* Orlando, Florida, Apr. 2007.

[59] T. Inanc, M. K. Muezzinoglu, K. Misovec, and R. M. Murray, “Framework for low-observable trajectory generation in presence of multiple radars,” *J. Guid. Control Dynam.*, vol. 31, no. 6, pp. 1740–1749, 2008.

[60] B. Geiger and J. Horn, “Neural network based trajectory optimization for unmanned aerial vehicles,” *47th AIAA Aerospace Sciences Meeting*, Orlando, Florida, 2009.

[61] H Jack, “Application of neural networks to motion planning and control for an articulated robot arm,” Master of Engineering thesis, Univ. of Western Ontario, London, Ontario, 1991.

[62] R. L. Hardy, “Multiquadric equations of topography and other irregular surfaces,” *J. Geophysical Res.,* vol. 76, no. 8, pp. 1905–1915, 1971.

[63] E. J. Kansa, “Multiquadrics– a scattered data approximation scheme with applications to computational fluid dynamics–I surface approximations and partial derivative estimates,” *Computers Math. Applications,* vol. 19, pp. 127–145, 1990.

[64] E. J. Kansa, “Multiquadrics - a scattered data approximation scheme with applications to computational fluid dynamics - II solutions to parabolic, hyperbolic and elliptic partial differential equations,” *Computers Math. Applications,* vol. 19, pp. 147–161, 1990.

[65] L. N. Trefethen, *Spectral Methods in MATLAB*. Philadelphia: Society for Industrial and Applied Mathematics (SIAM) Press, 2000.

[66] P. E. Gill, W. Murray, and M. A. Saunders, “SNOPT: an SQP algorithm for large-scale constrained optimization,” *SIAM Review,* vol. 47, no. 1, pp. 99–131, 2005.

[67] F. L. Lewis, D. Vrabie, V. L. Syrmos, *Optimal control.* J. Wiley & Sons, 3rd ed., 2012, pp. 118–119.

[68] I. M. Ross and F. Fahroo, “User’s manual for DIDO 2002: a MATLAB application package for dynamic optimization,” Dept. Aeronautics and Astronautics, Naval Postgraduate School, AA-02-002, Monterey, CA, 2002.

[69] P. E. Rutquist and M. M. Edvall, “PROPT: MATLAB optimal control software,” Tomlab Optimization, Inc, Pullman, Washington, Apr. 2010.

[70] I. M. Ross and F. Fahroo, “Legendre pseudospectral approximations of optimal control problems,” *Lecture Notes Control Inform. Sci.,* vol. 295, pp. 327–342, 2003.

[71] W. W. Hager, “Runge-Kutta method in optimal control and the transformed adjoint system,” *Numerische Mathematik*, vol. 87, no. 2, pp.247–282, 2000.

[72] Y-C. Chen, “Solving robot trajectory planning problems with uniform cubic B-splines,” *Optimal Control Appl. Meth.*, vol. 12, no. 4, pp. 247–262, Oct.-Dec. 1991.

[73] O. Von Stryk, “Numerical solution of optimal control problems by direct collocation,” *Intern. Series Numeric. Math.,* vol. 111, pp. 129–143, 1993.

[74] F. Fahroo and I. M. Ross, “A spectral patching method for direct trajectory optimization,” *J. Astronaut. Sci.,* vol. 48, pp. 269–286, Apr.-Sept. 2000.

[75] I. M. Ross and F. Fahroo, “A direct method for solving nonsmooth optimal control problems,” in *Proc. 2002 IFAC World Conf.*, Barcelona, Spain, July 2002.

[76] A. V. Rao, “Extension of a pseudospectral Legendre method to non-sequential multiple-phase optimal control problems,” *AIAA Guid., Navigation, Control Conf. & Exhibit,* Aug. 2003, Austin, Texas.

[77] D. A. Benson, G. T. Huntington, T. P. Thorvaldsen, and A. V. Rao, “Direct trajectory optimization and costate estimation via an orthogonal collocation method,” *J. Guid. Control Dynam.,* vol. 29, no. 6, pp. 1435–1440, Nov.–Dec. 2006.

[78] B. Singh, R. Bhattacharya, and S. R. Vadali, “Verification of optimality and costate estimation using Hilbert space projection,” *J. Guid. Control Dynam.,* vol. 32, no. 4, pp. 1345–1355, 2009.

[79] C. A. J. Fletcher, “Computational Galerkin Methods,” in *Springer Series Computational Phys.* Springer-Verlag, New York, 1984.

[80] G. T. Leondes and C. A. Wu, “Initial condition sensitivity functions and their applications,” *ASME J. Dynamical Syst. Measurm. Control,* vol. 93, pp. 116–123, 1971.

[81] R Bhattacharya, “OPTRAGEN 2.0, a MATLAB toolbox for optimal trajectory generation,” Aerospace Engineering Dept., Texas A&M University, College Station, TX, 2007.

[82] E. Tohidi and S. L. Noghabi, "An efficient Legendre pseudospectral method for solving nonlinear quasi bang-bang optimal control problems," *J. Applied Math. Statistics Inform.*, vol. 8, no. 2, pp. 73–85, 2012.

[83] H. Ma, T. Qin, and W. Zhang, "An efficient Chebyshev algorithm for the solution of optimal control problems," *IEEE Trans. Autom. Control,* vol. 56, no. 3, pp. 675–680, Mar. 2011.

[84] National Kidney Foundation. KDOQI clinical practice guidelines and clinical practice recommendations for anemia in chronic kidney disease in adults. *Am. J. Kidney Dis.,* 2006; vol. 47, no. 5, (suppl. 3):S16-S85, 2006.

[85] US Food and Drug Administration (FDA), "Information on Erythropoiesis-Stimulating Agents (ESA) Epoetin alfa (marketed as Procrit, Epogen), Darbepoetin alfa (marketed as Aranesp)," Postmarket Drug Safety Information for Patients and Providers, June 2011.

[86] D. M. Spiegel, "Anemia management in chronic kidney disease: what have we learned after 17 years?" *Semin. Dial.,* vol. 19, no. 4, pp. 269–272, July-Aug. 2006.

[87] T. Fjornes, "Response and prediction of response to recombinant human erythropoietin in patients with solid tumors and platinum-associated anemia," *J. Oncology Pharmacy Practice,* vol. 5, no. 1, pp. 22–31, 1999.

[88] J. W. Eschbach, "The anemia of chronic renal failure: Pathophysiology and the effects of recombinant erythropoietin," *Kidney Int.,* vol. 35, no. 1, pp. 134–148, Jan. 1989.

[89] J. W. Eschbach, J. C. Egrie, M. R. Downing, J. K. Browne, and J. W. Adamson, "Correction of the anemia of end-stage renal disease with recombinant human erythropoietin. Results of a combined phase I and II clinical trial," *N. Engl. J. Med.,* vol. 316, no. 2, pp. 73–78, Jan. 1987.

[90] J. W. Eschbach, J. W. Adamson, "Anemia of end-stage renal disease (ESRD)," *Kidney Int.*, vol. 28, no. 1, pp. 1–5, 1985.

[91] F. Falderrabano, "Erythropoietin in chronic renal failure," *Kidney Int.*, vol. 50, no. 4, pp. 1373–1391, 1996.

[92] J. Cody et al., "Recombinant human erythropoietin for chronic renal failure anaemia in pre-dialysis patients" *Cochrane Database Syst. Rev.* vol. 4, CD003266, 2001.

[93] A.K. Singh, et al., "Correction of anemia with epoietin alfa in chronic kidney disease," *N. Engl. J. Med.*, vol. 355, pp. 2085–2098, 2006.

[94] T. Drueke, et al., "Normalization of hemoglobin level in patients with chronic kidney disease and anemia," *N. Engl. J. Med.*, vol. 355, pp. 2071–2084, 2006.

[95] D. E. Uehlinger, F. A. Gotch, and L. B. Sheiner, "A pharmacodynamic model of erythropoietin therapy for uremic anemia," *Clin. Phamacol. Ther.,* vol. 51, no. 1, pp. 76–89, Jan. 1992.

[96] R. Bellazzi, “Drug delivery optimization through Bayesian networks: an application to erythropoietin therapy in uremic anemia,” *Comput. Biomed. Res.,* vol. 26, no. 3, pp. 274–293, Jun. 1993.

[97] A. E. Gaweda, A. A. Jacobs, G. R. Aronoff, and M. E. Brier, “Model predictive control of erythropoietin administration in the anemia of ESRD,” *Am. J. Kidney Dis.,* vol. 51, no. 1, pp. 71–79, Jan. 2008.

[98] M. E. Brier, A. E. Gaweda, A. Dailey, G. R. Aronoff, and A. A. Jacobs, “Randomized trial of model predictive control for improved anemia management,” *Clin. J. Am. Soc. Nephrol.,* vol. 5, no. 5, pp. 814–820, May 2010.

[99] S. W. Lines, E. J. Lindley, J. E. Tattersall, and M. J. Wright, “A predictive algorithm for the management of anemia in hemodialysis patients based on ESA pharmacodynamics: Better results for less work,” *Nephrol. Dial. Transplant,* vol. 27, no. 6, pp. 2425–2429, Jun. 2012.

[100] J. D. Martin-Guerrero, et al., “Dosage individualization of erythropoietin using a profile-dependent support vector regression,” *IEEE Trans. Biomed. Eng.*, vol. 50, no. 10, pp. 1136–1142, Oct. 2003.

[101] A. E. Gaweda, G. R. Aronoff, A. A. Jacobs, S. N. Rai, and M. E. Brier, “Individualized anemia management reduces hemoglobin variability in hemodialysis patients,” *J. Am. Soc. Nephrol.,* vol. 25, pp. 159–166, 2014.

[102] E. F. Unger, A. M. Thompson, M. J. Blank, and R. Temple, “Erythropoiesis-stimulating agents—time for a reevaluation,” *N. Engl. J. Med.* vol. 362, pp. 189–192, Jan. 2010.

[103] A. S. Kliger, S. Fishbane, and F. O. Finkelstein, “Erythropoietic stimulating agents and quality of a patient’s life: Individualizing anemia treatment,” *Clin. J. Am. Soc. Nephrol.,* vol. 7, no. 2, pp. 354–357, Feb. 2012.

[104] E. Akabua, T. Inanc, A. E. Gaweda, M. E. Brier, S. Kim, and J. M. Zurada, “Individualized model discovery: The case of anemia patients,” *Comput. Methods Programs Biomed.,* vol. 118, no. 1, pp. 23–33, 2015.

[105] H. Mirinejad and T. Inanc, “RBF method for optimal control of drug administration in the anemia of hemodialysis patients,” in *Proc. Biomed. Eng. Conf. (NEBEC), 41st Ann. Northeast*, Troy, NY, Apr. 2015.

[106] H. Mirinejad, T. Inanc, M. E. Brier, and A. E. Gaweda, “RBF-based receding horizon control for anemia management,” in *Proc. Biomed. Eng. Conf. (NEBEC), 41st Ann. Northeast*, Troy, NY, Apr. 2015.

[107] H. Mirinejad and T. Inanc, “Individualized anemia management using a radial basis function method,” in *Proc. Biomed. Conf. (GLBC), IEEE Great Lakes,* Milwaukee, WI, May 2015.

[108] A.C. Allison, “Turnovers of erythrocytes and plasma proteins in mammals,” *Nature,* vol. 188, pp. 37–40, Oct. 1960.

[109] S. L. Schrier, (2015, Mar.). Red blood cell survival: Normal values and measurement. [Online]. Available: http://www.uptodate.com/contents/red-blood-cell-survival-normal-values-and-measurement

[110] J. H. Lee, "Model predictive control: Review of the three decades of development," *Intern. J. Control Autom. Sys.,* vol. 9, no. 3, pp. 415–424, 2011.

[111] M. Morari and J. H. Lee, "Model predictive control: past, present and future," *Comput. & Chemical Eng.,* vol. 23, pp. 667–682, 1999.

[112] M. Kuure-Kinsey and B. W. Bequette, "Multiple model predictive control of nonlinear systems," in *Nonlinear Model Predictive Control*, Lecture Notes in Control and Information Sciences, vol. 384, Springer, 2009, pp. 153–165.

[113] S. Rippa, "An algorithm for selecting a good value for the parameter *c* in radial basis function interpolation," *Adv. Comput. Math.*, vol. 11, pp. 193–210, 1999.

[114] G. E. Fasshauer, J. G. Zhang, "On choosing optimal shape parameters for RBF approximation," *Numer. Algor.* vol. 45, pp. 345–368, 2007.

[115] M. Mongillo, (2011, Oct.). Choosing basis functions and shape parameters for radial basis function methods. [Online]. Available: http://www.siam.org/students/siuro/vol4/S01084.pdf

[116] C. M. Gullion, D. S. Keith, G. A. Nichols, and D. H. Smith, "Impact of comorbidities on mortality in managed care patients with CKD," *Am. J. Kidney Dis*. Vol. 48, pp. 212–220, 2006.

CURRICULUM VITAE

Hossein Mirinejad

h.mirinejad@louisville.edu

ADDRESS

Lutz Hall, Room 17
University of Louisville
Louisville, KY 40292
(502) 852-0409

EDUCATION

Ph.D., Electrical Engineering
University of Louisville, Louisville, KY
Graduated May 2016

M.S. with Honors, Mechatronics Engineering
K. N. Toosi University of Technology, Tehran, IRAN
Graduated Sept. 2008

B.S., Electrical Engineering-Control
Iran University of Science & Technology, Tehran, IRAN
Graduated Mar. 2006

HONORS AND AWARDS

Graduate Dean Citation, Graduate School, University of Louisville	2016
Outstanding Graduate/Professional Student Award, University of Louisville	2016
ECE Outstanding Graduate Student Award, ECE Dept. University of Louisville	2016
Doctoral Dissertation Completion Fellowship, University of Louisville,	2016
Theobald Scholarship Award, Electrical Eng. Dept., University of Louisville	2015
ACC Student Travel Award, American Control Conference	2015
Graduate Student Council Research Scholarship, University of Louisville	2015
Graduate Student Council Travel Award, IEEE GLBC Conf., Univ. of Louisville	2015
University Fellowship for Two Consecutive Years, University of Louisville	2011-2013
Graduate Student Council Travel Award, EnergyTech Conf., Univ. of Louisville	2012
IEEE EnergyTech Conf. Student Funding for Travel and Registration	2012
Elite University Student, Iranian National Institute of Elites	2009
Ranked 1st among the students of Mechatronics, K.N.T. University of Technology	2008
Exceptionally Talented Student, awarded by NODET	2006
Ranked 5th among 961 applicants, nation-wide entrance exam for M.S. degree	2006
High-Ranked newcomer Student, awarded by President of Iran Univ. Sci. & Tech.	2001
Ranked 662nd among 450,000 applicants, nation-wide entrance exam for B.S.	2000

RESEARCH INTERESTS

Optimal Control Theory and Applications, Trajectory Optimization & Motion Planning
Modeling and Control of Autonomous Vehicles
Model Predictive Control
Intelligent Control based on Neural Networks, Fuzzy, and Genetic Algorithms

PUBLICATIONS

[1] **H. Mirinejad** and T. Inanc, "A Radial Basis Function Method for Direct Trajectory Optimization," *2015 American Control Conf. (ACC)*, Chicago, IL, July 2015.

[2] **H. Mirinejad** and T. Inanc, "Individualized Anemia Management using a Radial Basis Function Method" *2015 IEEE-USA Annual Meet. &* 36th *Great Lakes Biomed. Conf.*, Milwaukee, WI, May 2015.

[3] **H. Mirinejad,** T. Inanc, A. E. Gaweda, and M. E. Brier, "RBF-based Receding Horizon Control for Anemia Management," *Biomed. Eng. Conf. (NEBEC), 41st Annual Northeast,* Troy, NY, April 2015.

[4] **H. Mirinejad** and T. Inanc, "RBF Method for Optimal Control of Drug Administration in the Anemia of Hemodialysis Patients," *Biomed. Eng. Conf. (NEBEC), 41st Annual Northeast,* Troy, NY, April 2015.

[5] F. Yaghouby, B. Madahian, **H. Mirinejad**, S. Sunderam, "Feasibility of a Seizure Prediction Algorithm Using Electrocorticogram Measurements in Dogs," *Biomed. Eng. Conf. (NEBEC), 41st Annual Northeast,* Troy, NY, April 2015.

[6] **H. Mirinejad** and T. Inanc, "An RBF Collocation Method for Solving Optimal Control problems," under review, *Robotics and Autonomous Systems,* 2016.

[7] **H. Mirinejad** and T. Inanc, and J. M. Zurada, "Costate Estimation and Trajectory Optimization using Radial Basis Function Interpolation and Galerkin Error Projection," under review, *IEEE Trans. Automatic Control,* 2016.

[8] **H. Mirinejad**, T. Inanc, A. E. Gaweda, M. E. Brier, and J. M. Zurada, "Individualized Drug Dosing using RBF-Galerkin Method: Case of Anemia Management in Chronic Kidney Disease," under preparation, IEEE Trans. Biomed. Eng., 2016.

[9] **H. Mirinejad**, K. Welch, L. Spicer, "A Review of Intelligent Control Techniques in HVAC Systems," *2012 IEEE EnergyTech Conference,* Cleveland, OH, pp. 1–5, May 2012.

[10] M. R. Yousefi, S. Hasanzadeh, **H. Mirinejad,** and M. Ghasemian, "A Hybrid Neuro-Fuzzy Approach for Greenhouse Climate Modeling," *5th IEEE Intern. Conf. on Intelligent Syst. (IS),* London, UK, pp. 212–217, July 2010.

[11] **H. Mirinejad**, S. H. Sadati, M. Ghasemian, H. Torab, "Control Techniques in Heating, Ventilating and Air Conditioning (HVAC) Systems," *J. Computer Science,* Vol. 4, No. 9, pp.777–783, 2008.

[12] **H. Mirinejad**, S. H. Sadati, S. Hasanzadeh, A. M. Shahri, and M. Ghasemian, "Design and Simulation of an Automated System for Greenhouse using LabVIEW," *American-Eurasian J. Agricultural & Environmental Sciences,* Vol. 3, No. 2, pp. 279–284, 2008.

TEACHING CERTIFICATE

Certificate of Graduate Teaching Assistant Academy, University of Louisville 2013-2014