JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 40, No. 2, February 2017

Sparse Approximation–Based Collocation Scheme for Nonlinear Optimal Feedback Control Design

Nagavenkat Adurthi* and Puneet Singla†
University at Buffalo, The State University of New York, Buffalo, New York 14260-4400
and
Manoranjan Majji‡
University at Buffalo, The State University of New York, Amherst, New York 14260-4400

DOI: 10.2514/1.G001755

This paper discusses the development of a computationally efficient approach to generate optimal feedback control laws for infinite time problems by solving the corresponding Hamilton–Jacobi–Bellman (HJB) equation. The solution process consists of iteratively solving the linear generalized HJB (GHJB) equation starting with an admissible stable controller. The collocation methods are exploited to solve the GHJB equation in the specified domain of interest. Recently developed nonproduct quadrature method known as Conjugate Unscented Transformation is used to manage the curse of dimensionality associated with the growth of collocation points with increase in the state dimension. Furthermore, recent advances in sparse approximation are leveraged to automatically generate the appropriate polynomial basis function set for the collocation-based approximation from an overcomplete dictionary of basis functions. It is demonstrated that the solution process uses the basis function selection process to automatically identify a form for the feedback control law, which is frequently unknown. Several numerical examples demonstrate the efficacy of the proposed approach in accurately generating feedback control policies for nonlinear systems.

I. Introduction

OPTIMAL control theory occupies a central role in the generation of guidance and control laws for aerospace vehicles [1]. Extensive research has been carried out to develop semianalytical and numerical solutions to optimal open-loop and feedback control problems [2,3]. Optimal open-loop problems are used in trajectory optimization and path planning to provide optimal trajectories of dynamical systems that extremalize a given cost function of interest [1]. In guidance and control problems, energy, time, and fuel are some performance indices that are minimized [4] to find optimal trajectories for guidance of aerospace vehicles. Application of the variational principles [5], in conjunction with the Pontryagin's principle [6], typically yields a two-point boundary value problem for optimal state and the control law. Direct and indirect methods are used to solve the resulting dynamic optimization problem to determine the optimal control and state trajectories. Indirect methods to solve optimal open-loop problems rely on the use of continuation and other multiple shooting-type methods to solve the nonlinear two-point boundary value problems (the state-costate equations for necessary conditions of optimality [1,7,8]). Quasi linearization has also been used to solve the two-point boundary value problems [9]. Direct methods involve the transcription of the dynamic optimization problems by various discretization schemes to convert the dynamic optimization problem into a system of nonlinear equations that need to be solved for the control and state values at specified time instances. Depending on the quadrature scheme used to approximate the performance index and the differential equations, a variety of methods have been developed in the literature [10]. Recently, the application of pseudo-spectral collocation methods have led to the development of various direct collocation methods [11–15] and certain combined direct and indirect methods for open-loop solution of the optimal control problems [16,17]. The main shortcoming of the open-loop solutions to optimal control problems is their sensitivity to the initial conditions of the dynamical system and the unstructured perturbations, including modeling errors and exogenous disturbance inputs. With errors in initial conditions, model parameters, or even forcing functions, the optimal control problem has to be re-computed as the problem specifications change. Repeated solution of the two-point boundary value problems resulting from the Pontryagin's minimum principle provides useful means of generating optimal control laws within a local domain. Although for linear systems a rapid solution of the open-loop optimal control problem provides a feasible methodology for practical applications, for general nonlinear systems repeated solution of optimal open-loop problems using both the direct and indirect methods constitutes a formidable challenge.

Feedback solutions provide an attractive alternative to the optimal open-loop solutions in that they are fundamentally conceived from Bellman's principle of optimality [18], which forms a basis for dynamic programming. In contrast to the open-loop solutions, feedback solutions depend upon the current state and provide an optimal control policy to minimize the value function and provide a means for the state of the dynamic system to reach a terminal manifold that may be specified by the optimal control problem [1,19]. The optimal control policy can be shown to be related to the value function and it turns out that the value function satisfies a nonlinear hyperbolic partial differential equation (PDE) called the Hamilton–Jacobi–Bellman (HJB) equation. The feedback control solutions can be determined by solving this PDE over the domain of interest, with specified boundary conditions for some or all initial states. Local solutions obtained by solving open-loop problems satisfy the HJB equation and define the characteristic curves of this equation, sometimes known as the field of extremals [2,18,19]. Theoretically, the power of the feedback solutions is derived from the fact that the sufficient conditions for optimality [1] can be folded back into the solution process by using the positivity and convexity of the value functional.

Although the quest for a unified solution approach to the time-dependent and asymptotic HJB equation remains a holy-grail for a general dynamical system with arbitrary functions as performance indices and terminal manifolds, researchers have worked on various

Presented at the AAS/AIAA Astrodynamics Specialist Conference, Vail, CO, 9 –13 August 2015; received 2 October 2015; revision received 5 July 2016; accepted for publication 6 July 2016; published online 15 September 2016.

*Graduate Student, Department of Mechanical and Aerospace Engineering; nagavenk@buffalo.edu. Member AIAA.
†Associate Professor, Department of Mechanical and Aerospace Engineering; psingla@buffalo.edu. Associate Fellow AIAA.
‡Assistant Professor, Department of Mechanical and Aerospace Engineering. Senior Member AIAA.

methods to provide local solutions to this important problem. Following Kalman's celebrated linear quadratic regulator (LQR), Bryson devised the sweep method [1] to derive the linear state feedback solution for a problem with quadratic cost. Subsequently, a variety of solutions have been developed to derive optimal control solutions for smooth problems (where performance index, dynamics, and manifold are smooth functions of the state and control variables). State-dependent Riccati equation approach provides a mechanism to absorb the nonlinear functions of the state variable within the control gain function and provides a systematic approach to derive optimal feedback control solutions [20]. However, the resulting solution is seminumerical in nature and is shown to demonstrate suboptimal performance in certain problems. An alternative approach is the so-called $\theta - D$ method [21], which introduces an embedding parameter (θ) to serve as an ordering parameter and solves the resulting sequence of semilinear problems. This quasi-linearization style strategy, in which the value function is updated recursively, as dictated by the ordering parameter, has been shown to work in a class of missile guidance problems. Park and Scheeres employ a similar strategy, while simultaneously making use of the Hamiltonian dynamics and obtain feedback solutions via a sequence of generating functions [22]. Each generating function is aimed at solving a particular term in the HJB equation that is expanded in terms of an embedding parameter. The resulting process also mimes the Lie-series method in averaging of dynamic systems [23]. An alternative solution methodology emerges by developing a series expansion of the value functional and writing the resulting optimal control law as a function of the high-order nonlinear feedback gains, following a process originally developed by Albrekht [24] and later applied by Carrington and Junkins [25] to derive nonlinear state feedback control laws for attitude control of rigid spacecraft. This process was recently generalized to specialize the feedback control process to reach terminal manifolds at a specified terminal time by Vadali and Sharma [26]. Although the series solution methodology was shown to be useful, the curse of dimensionality renders the method to only provide local solutions. This shortcoming is shared by all other solutions to the problem. Other strategies include the development of inverse optimal solutions for nonlinear systems [27]; however, the inverse optimal solutions assume a structure for the optimal feedback control, which is unknown in the general problem. Approximate solutions involving the time-varying approximations of HJB have also been recently developed by leveraging the immersion and invariance concepts [28]. Although this solution appears to be promising, it is unclear how the process can be extended to fixed time problems with the terminal constraints. In discrete time problems, the differential dynamic programming methods [29–33] use successive quadratic approximations of the dynamics and the cost functions about a nominal trajectory. These methods provide localized optimal control trajectories.

An alternative approach for the solution of the HJB equation is to solve the PDE directly. In [34], an adaptive finite difference method is proposed to solve the HJB equation. In addition to the finite difference approach, the method of characteristics is used to compute a discretized solution to the HJB. Further, the finite element approach can also be used to approximate the solution to HJB equation [35]. The high computational cost involved in these methods for higher dimensions limits the applicability of these methods for many practical engineering problems. Collocation [36–38] and Galerkin methods [39,40] can be used to carry out this solution process by the careful selection of basis functions. Beard et al. [39–41] investigated the development of a successive Galerkin approximation techniques for the development of nonlinear optimal and robust control laws. Building upon Kleinman's [42] observation that the solution of the Riccati equation can be obtained by successively solving a sequence of Lyapunov equations (which are linear in nature), policy iteration is used in this process to successively improve upon a stabilizing control solution to the generalized HJB (GHJB). However, the number of basis functions used in the Galerkin projection and the collocation operations also suffer from the curse of dimensionality. Level set methods [43,44] have also been developed recently and applied to generation of optimal trajectories for UAV path planning [45]. The solution process in the level set methods involves the computation of the field value over the set of grid points in the domain of interest. This becomes computationally expensive when the dimensions grow beyond 2.

The main challenge of formulating discretization strategies to solve multivariate PDEs like the HJB equation lies in the *curse of dimensionality*. The principal contribution of this paper is to address this computational challenge. Recently developed Conjugate Unscented Transformation (CUT) method is leveraged to alleviate this *curse of dimensionality* and provide a computationally efficient algorithm to solve the HJB equation in moderate to high dimension spaces. The CUT methodology provides nonproduct quadrature points directly in the multidimensional space that can be used as collocation points. Furthermore, the recent advances in sparse approximation are exploited to formulate the interpolation polynomials directly in the multidimensional space for the chosen collocation points. In particular, sequential l_1-norm minimization is carried out to select appropriate basis from a dictionary of over complete basis functions. An iterative procedure for the solution of the GHJB is presented to avoid solving the nonlinear set of equations that generally arise in the solution process of the HJB equation. Finally, numerical experiments involving some benchmark problems like the tumbling rigid body in space and attitude control of satellite are performed to show the efficacy of the proposed approach.

II. Problem Formulation

The objective of the paper is to solve the optimal regulation problem:

$$P_1 : \begin{cases} \min\limits_{u(t)} V(\boldsymbol{x}(t_0)) = \int_{t_0}^{\infty} \{l(\boldsymbol{x}(t)) + \boldsymbol{u}(t)^T R \boldsymbol{u}(t)\}\, \mathrm{d}t \\ \text{subject to: } \dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})\boldsymbol{u}, \qquad \boldsymbol{x}(t_0) \in \Omega \end{cases} \tag{1}$$

where $\boldsymbol{x} \in \mathbb{R}^n$, $\boldsymbol{u} \in \mathbb{R}^r$, and $\Omega \subset \mathbb{R}^n$ with the origin $0 \in \Omega$. R is a symmetric positive definite matrix and is used to penalize the energy expended by each of the control inputs. We note that the dynamical system constraints considered in the problem P_1 are of a special structure, where the control input $\boldsymbol{u}(t)$ influences the evolution of the state vector in an affine manner. Without loss of generality, we assume that the origin is an equilibrium point; that is, $\boldsymbol{f}(0) = 0$, contained in the domain of interest. The vector functions $\boldsymbol{f}(\cdot)$ and $\boldsymbol{g}(\cdot)$ are Lipschitz continuous on Ω [46]. The cost $l(\cdot)$ is positive definite and monotonically increasing in $\|\boldsymbol{x}\|$, where $\|\cdot\|$ denotes the norm of the state vector. A calculus of variations approach provides a solution to this problem by imposing the stationarity of the augmented cost functional with respect to possible control functions. The dynamical system constraints are augmented to the integrand with the aid of Lagrange multipliers called the costates. Costates measure the sensitivity of the cost with respect to the state vector (denoted by $\lambda(t) \in \mathbb{R}^n$, $\lambda(t) = (\partial V/\partial \boldsymbol{x})$). Pontryagin's principle [6] provides a means of generating the optimal control laws as a function of these costates, by minimizing the Hamiltonian function defined as $H(\boldsymbol{x}(t), \boldsymbol{u}(t)) \coloneqq l(\boldsymbol{x}(t)) + \boldsymbol{u}(t)^T R \boldsymbol{u}(t) + \lambda^T[\boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})\boldsymbol{u}(t)]$. For the problem P_1, where the cost associated with the control authority appears quadratically, the control function that minimizes the Hamiltonian can be written explicitly as $\boldsymbol{u}(t) = -(1/2)R^{-1}\boldsymbol{g}(\boldsymbol{x})^T\lambda$. The explicit dependence of the control on the costate function mandates the integration of the adjoint equations with appropriate initial conditions. Therefore, the determination of optimal control leads to a search for the specific initial costate values, $\lambda(t_0)$, that also depend implicitly on the initial state $\boldsymbol{x}(t_0)$. Because the control profile is distinctly specified for each initial condition, the set of solutions obtained by looking at the necessary conditions of optimality is known as the open-loop solutions. Existence of local minima and the lack of proper choice for initial costates make the determination of open-loop solutions quite challenging in practical applications.

An alternative approach to open-loop solution attempts to develop an explicit functional relationship between the state and the costate functions, by recognizing the fact that the costates measure the sensitivity of the value function being optimized with respect to the dynamics of the system. By making use of the relationship, in

conjunction with the necessary conditions obtained from the variational principle $H_u = 0$, for the performance index in P_1, the control can be written explicity as $\boldsymbol{u}(t) = -(1/2)R^{-1}\boldsymbol{g}(\boldsymbol{x})^T(\partial V/\partial \boldsymbol{x})$. Thus, following the developments of Caratheodory [3], one finds that the problem of finding an optimal control law gets transformed into a problem of determining a value function to regulate the dynamical system to the origin from all initial conditions in the domain of interest. This cost to go function is obtained by solving the nonlinear PDE called the HJB equation [18] written as

$$l(\boldsymbol{x}(t)) + \frac{\partial V^T}{\partial \boldsymbol{x}} f(\boldsymbol{x}) - \frac{1}{4}\frac{\partial V^T}{\partial \boldsymbol{x}} g(\boldsymbol{x})R^{-1}g(\boldsymbol{x})^T \frac{\partial V}{\partial \boldsymbol{x}} = 0 \tag{2}$$

To ensure the stability of the feedback control law, the boundary condition $V(0) = 0, \forall\, \boldsymbol{x} \in \Omega$ needs to be enforced. Intuitively, this PDE represents the fact that, for the autonomous dynamical system, the characteristics of the optimal value functional flow in a way so as to attain a minimum value of the Hamiltonian, uniformly in the state and time dimensions. In general, the solution of this PDE is difficult to compute in a closed form; that is, the form of the value function $V(\boldsymbol{x}(t))$ cannot be determined as an explicit function of the state vector so as to minimize a general cost functional, subject to the constraints of a nonlinear dynamical system [1,47]. Kalman [48] used calculus of variations to demonstrate that the solution of the HJB equation with a quadratic cost function subject to the constraints of a linear system is a quadratic form of the state vector itself. This is shown to provide a linear state feedback controller. Bryson and Ho extended this approach and devised a sweep method [1] that provides different feedback forms for tracking time-varying reference signals and reaching a terminal manifold, for the fixed and free terminal time problems.

As outlined in the introduction, several subsequent investigations have been carried out to establish feedback solutions $\boldsymbol{u}(t) = \boldsymbol{k}(\boldsymbol{x}, t)$ in a systematic manner with limited success. In the series solution methods [24–26], researchers have parameterized this general function $\boldsymbol{k}(\boldsymbol{x}, t)$ as a polynomial series, with time varying coefficients in an attempt to generalize Kalman's canonical developments involving the linear quadratic regulation problem. This leads to the development of nonlinear state feedback control laws, written as $u_i(t) = S^1_{i,j}x_j + S^2_{i,j_1j_2}x_{j_1}x_{j_2} + \ldots$, where Einstein's summation convention [49] was employed to enable compact representation of the series involved. Equations for the high-order feedback control gains $S^k_{i,j_1,j_2,\ldots,j_k}$ are derived using coupled sets of nonlinear algebraic equations that may be difficult to solve.

The difficulties imposed by the solution of the HJB equation in Eq. (2) arise from the fact that it is nonlinear in the first spatial derivate of the value function, $\partial V(\boldsymbol{x})/\partial \boldsymbol{x}$. The nonlinearity manifests itself in various forms in all the solution methods (i.e., coupled algebraic equations in the series solution methods being a case in point). These issues are compounded by the high dimensionality of the control problems. Stabilization and reachability constraints also impose additional conditions on the feedback control laws so that the challenge of finding the gains is rendered intractable.

With the aim of alleviating the problems associated with the nonlinearity of the equation, Saridis and Lee [41] propose a sequential updating algorithm to generate a sequence of stable control laws that improve the performance function with each iteration. Starting from a known admissible controller $\boldsymbol{u}_a(\boldsymbol{x})$, the iterative process then solves for the value function produced by this stabilizing control by solving the associated generalized HJB equation given by

$$\frac{\partial V^T}{\partial \boldsymbol{x}}[f(\boldsymbol{x}) + g(\boldsymbol{x})\boldsymbol{u}_a(\boldsymbol{x})] + l(\boldsymbol{x}) + \boldsymbol{u}_a^T(\boldsymbol{x})R\boldsymbol{u}_a(\boldsymbol{x}) = 0 \tag{3}$$

where $\boldsymbol{u}_a(\boldsymbol{x})$ is a known admissible control [41,46]. Note that this PDE, known as the GHJB, is now a linear equation in the first derivative of the value function $V(\boldsymbol{x})$ and becomes more amenable to standard solution methods. Upon solving the GHJB equation, the control law for the next iteration is recomputed using the relationship $\boldsymbol{u}_a(\boldsymbol{x}) = -(1/2)R^{-1}g(x)^T(\partial V/\partial \boldsymbol{x})$. The control cost in Eq. (1) can in general be assumed to be any positive definite function of the control, but the quadratic form of the control cost makes it simple to express the controller as a linear function of the value function. This process is repeated until the value function converges uniformly in the domain of interest. This approach has come to be known as the policy iteration, and the limiting value function $V(\boldsymbol{x})$ is shown to satisfy the HJB Eq. (2). The policy iteration process is akin to quasi linearization [9], originally devised by Bellman, and a cousin of the perturbation methods well known to classical mechanicians [23]. Starting from a known stable controller $\boldsymbol{u}_{(0)}(\boldsymbol{x})$, the two-step process involved in the GHJB value iteration process can be written as

$$\text{Step 1: } \frac{\partial V^{(k)T}}{\partial \boldsymbol{x}}[f(\boldsymbol{x}) + g(\boldsymbol{x})\boldsymbol{u}_{(k)}(\boldsymbol{x})] + l(\boldsymbol{x}) + \boldsymbol{u}^T_{(k)}(\boldsymbol{x})R\boldsymbol{u}_{(k)}(\boldsymbol{x}) = 0$$

$$\text{Step 2: } \boldsymbol{u}_{(k+1)}(\boldsymbol{x}) = -\frac{1}{2}R^{-1}g(x)^T\frac{\partial V^{(k)}}{\partial \boldsymbol{x}} \tag{4}$$

Hence, the problem of solving the nonlinear HJB equation has been converted into the problem of iteratively solving the linear GHJB equation. The initial stabilizing controller to start the policy iteration process may be chosen by exploiting the passivity results of mechanical systems. Passivity results of mechanical systems ensure that a stabilizing state feedback controller always exists and forms a basis for chosen initializing controller. Because optimality always follows the necessary condition of stabilizability, this assumption is not unreasonable. A continuous functional approximation of the value function can be considered to solve the GHJB equation during this iteration process. Beard et al. [46] makes use of the Galerkin method to solve the GHJB equation to provide an iterative approach to solve for the value function. The main shortcoming of discretizing the solution of multivariate PDEs using functional approximation is the *curse of dimensionality* of the resulting algebraic equations that result from the transcription process. As discussed in [46], the Galerkin procedure suffers from explosive growth of basis functions as the state dimension is increased, which makes the resulting approach intractable for computational purposes for many practical engineering problems. In the next section, a collocation-based discretization of the GHJB is discussed that leverage recent advances in nonproduct quadrature methods and sparse approximation to alleviate the *curse of dimensionality.*

III. Approximate Value Function

As discussed in the previous section, optimal feedback control laws can be derived by iteratively solving the GHJB equation. In this section, the conventional collocation methods to solve the GHJB are now discussed. Assuming the value function $V(\boldsymbol{x})$ to be a smooth function with at least continuous first-order derivative, $V(\boldsymbol{x})$ is written as a linear sum of known basis functions:

$$V(\boldsymbol{x}) = \sum_{i=0}^{m} c_i\phi_i(\boldsymbol{x}) = \boldsymbol{c}^T\boldsymbol{\phi}(\boldsymbol{x}) \tag{5}$$

The basis functions $\boldsymbol{\phi}(\boldsymbol{x}) = [\phi_1(\boldsymbol{x}), \phi_2(\boldsymbol{x}), \ldots, \phi_m(\boldsymbol{x})]^T$ are assumed to have at least continuous first-order derivatives and satisfy the boundary condition $\phi_i(\boldsymbol{0}) = 0$. There are infinitely many choices for basis functions such as polynomials, wavelets, B-splines, radial basis functions, and so on. The problem of choosing an appropriate basis function is difficult because one usually cannot specify the characteristics of the unknown value function. However, polynomial functions are generally used because of the remarkable result in the approximation theory. According to the Stone–Weierstrass theorem, a sequence of prescribed continuous function always exists on a compact interval [50–52]. As a virtue of this theorem, any continuous function on a compact interval can be approximated well with a polynomial variables having sufficient number of terms.

By using the approximate value function of Eq. (5), the problem of solving the GHJB is transformed into the corresponding problem of solving for the m coefficients from m independent equations. The

method of weighted residuals is used to derive the set of equations governing each of the m coefficients [53]. The residual is formed by substituting the approximate value function of Eq. (5) into the GHJB PDE of Eq. (2):

$$e(\boldsymbol{x}) = \boldsymbol{c}^T J(\boldsymbol{x})[f(\boldsymbol{x}) + g(\boldsymbol{x})\boldsymbol{u}_{(k)}(\boldsymbol{x})] + l(\boldsymbol{x}) + \boldsymbol{u}_{(k)}^T R \boldsymbol{u}_{(k)} \tag{6}$$

where the partial derivative of the approximate value function with respect to $\boldsymbol{x}$ is given as

$$\frac{\partial V^T}{\partial \boldsymbol{x}} = \boldsymbol{c}^T \begin{bmatrix} \frac{\partial \phi_1}{\partial x_1} & \frac{\partial \phi_1}{\partial x_2} & \frac{\partial \phi_1}{\partial x_3} & \cdots & \frac{\partial \phi_1}{\partial x_n} \\ \frac{\partial \phi_2}{\partial x_1} & \frac{\partial \phi_2}{\partial x_2} & \frac{\partial \phi_2}{\partial x_3} & \cdots & \frac{\partial \phi_2}{\partial x_n} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ \frac{\partial \phi_m}{\partial x_1} & \frac{\partial \phi_m}{\partial x_2} & \frac{\partial \phi_m}{\partial x_3} & \cdots & \frac{\partial \phi_m}{\partial x_n} \end{bmatrix} = \boldsymbol{c}_{1\times m}^T J(\boldsymbol{x})_{m\times n}$$
$$= \boldsymbol{c}^T J(\boldsymbol{x}) \tag{7}$$

In the method of weighted residuals, the m independent equations are constructed by requiring the residual $e(\boldsymbol{x})$ to vanish in a weighted integral sense:

$$\int_\Omega \Psi_j(\boldsymbol{x}) e(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x} = 0 \qquad j = 1, 2, \ldots, m \tag{8}$$

where the set $\boldsymbol{\Psi} = \{\Psi_j(\boldsymbol{x})\}$ is a set of linearly independent weight functions also known as *test functions*. Some conventional methods to generate the m independent equations are Galerkin, least squares, and collocation. In Galerkin method, the weight functions $\boldsymbol{\Psi}$ are taken to be the same as the basis functions $\boldsymbol{\phi}$:

$$\int_\Omega \phi_j(\boldsymbol{x}) e(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x} = 0 \qquad j = 1, 2, \ldots, m \tag{9}$$

Minimizing the integral of the square of the residual error leads to the least squares formulation given by

$$\frac{\partial}{\partial c_j} \int_\Omega e(\boldsymbol{x})^2\, \mathrm{d}\boldsymbol{x} = \int_\Omega \frac{\partial e(\boldsymbol{x})}{\partial c_j} e(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x} = 0 \quad j = 1, 2, \ldots, m \tag{10}$$

where it can be observed that the weight functions are $\Psi_j(\boldsymbol{x}) = (\partial e(\boldsymbol{x})/\partial c_j)$. In collocation, the weight functions are taken to dirac-delta functions, $\Psi_j(\boldsymbol{x}) = \delta(\boldsymbol{x} - \boldsymbol{x}_j)$, at specially chosen points $\boldsymbol{x}_j$.

$$\int_\Omega \delta(\boldsymbol{x} - \boldsymbol{x}_j) e(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x} = e(\boldsymbol{x}_j) = 0 \qquad j = 1, 2, \ldots, m \tag{11}$$

In other words, the residual is required to be identically zero at the m chosen collocation points. In all methods of weighted residuals, the solution of original GHJB is transcribed into a problem of solving a linear set of equations. Both the Galerkin and least square methods require the computation of multidimension projection integrals. The computation of these integrals become increasingly expensive with the increase in state-space dimension or order of basis functions [46]. On the other hand, the collocation method avoids the integration process altogether by requiring the GHJB to be exactly satisfied at specially chosen collocation points within the domain Ω. The selection of the collocation points is crucial in obtaining a well-conditioned resulting system of equations for the unknown coefficients. In this paper, a new choice of collocation points is presented that efficiently capture the structure of the value function, leading to a well-conditioned system of equations as a solution for the HJB equation. In addition to the collocation points, the choice of interpolating functions is of consequence in ensuring the implementation of an effective collocation approach for solution of multidimensional PDEs like the HJB equation. In one dimension, Lagrange interpolation polynomials constitute an ideal choice for collocation, owing to their minimal order. However, this choice ceases to be a minimal order set of interpolation functions in higher dimensions. Constructing the least-degree polynomials in the general multidimensional space is an active area of research. Recent advances in sparse approximation provide a means of constructing minimal order interpolation functions. A novel collocation scheme that builds upon two key features is now detailed. The features are 1) the use of nonproduct cubature points to curtail the growth of collocation points with increase in the state dimension, and 2) sparse approximation tools to facilitate an automated construction of least-degree interpolating polynomials.

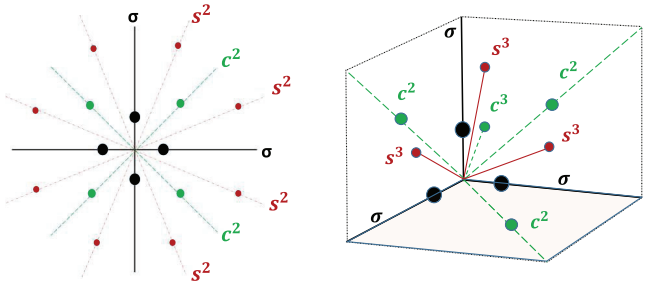
A. Collocation Points

In the one-dimensional system, the Gaussian quadrature points, along with Lagrange interpolation polynomials, provide the optimal choice for collocation points along with minimal order basis functions. In [11–15], the Gauss–Legendre (GLgn) quadrature points are used to discretize the time domain to satisfy the necessary conditions. However, the Gaussian quadrature methods suffer from *curse of dimensionality* because the number of quadrature points in general n-dimensional space is constructed from the tensor product of one-dimensional quadrature points. Even for a moderate-dimension system involving, for example, six state variables, the number of points is $5^6 = 15{,}625$, with only five points along each direction.

The sparse grid quadratures, and in particular Gauss–Legendre–Smolyak (GLgnSM) quadrature method, take the sparse product of one-dimensional quadrature rules and thus have fewer points than the equivalent Gaussian quadrature rules, but at the cost of introducing negative weights [54], which can further lead to numerical instability [55]. Note that the Gaussian quadrature rule is not minimal for $m \geq 2$, and there exists quadrature rules requiring fewer points in high dimensions [56]. For instance, the well-known unscented transformation (UT) [57] provides third-order quadrature points, with the number of sigma points exhibiting a linear growth as a function of the dimensionality of the problem at hand. Recently developed CUT methodology provides an extension of the UT to generate nonproduct quadrature points of order greater than three in a multidimensional space. In this work, recently developed CUT methodology is leveraged to alleviate this *curse of dimensionality* and have a computationally efficient collocation scheme to solve the GHJB equation in a multidimensional space. As evident from our recent work on uncertainty propagation, the CUT methodology provides much fewer quadrature points than its conventional counterparts without compromising on the accuracy [58,59]. In the following section, a succinct discussion pertaining to the CUT algorithm is provided.

1. *Conjugate Unscented Transformation*

Rather than using tensor products as in Gauss quadrature, the CUT approach judiciously selects special structures to extract symmetric quadrature points constrained to lie on specially defined axes as shown in Fig. 1. These specially defined axes include conjugate and scaled conjugate axes in addition to the conventional principal axes used in the UT algorithm. In $n - D$ Cartesian space, the *principal axes* are defined as the n orthogonal axes centered at the origin. The points on the principal axes are enumerated as



a) 2D axis b) 3D Octant-perspective view

Fig. 1 Symmetric set of points and axis 2D and 3D space.

$$\sigma_i \in \{\pm(I)_k | k \in \mathcal{D}\}, \qquad i = 1, 2, 3, \ldots, 2n \tag{12}$$

where $\mathcal{D} = \{1, 2, 3, \ldots, n\}$ and $(I)_k$ represent the kth row or column of the $n \times n$-dimensional identity matrix I. Each point on the principal axes is at a unit distance from the origin.

In $n - D$ Cartesian space, the "pth-conjugate axes with $p \leq n$" is defined as the directions that are constructed from all the combinations, including the sign permutations, of the set of principal axes taken p at a time. The set of pth conjugate axes labeled as c^p, where the points are listed as c_i^p, are

$$c_i^p \in \left\{ \left(\pm\sigma_{N_1} \pm \sigma_{N_2} \ldots \pm \sigma_{N_p} \right) \middle| \{N_1, N_2, \ldots, N_p\} \subset \mathcal{D} \right\}$$
$$i = 1, 2, \ldots, 2^p \binom{n}{p}$$

In $n - D$ Cartesian space, the "pth-scaled conjugate axes" is defined as the set of directions constructed from all combinations, including sign permutations, of the set of principal axes such that in every combination exactly one principal axis is scaled by a parameter $h \in \mathbb{R}$. The set of pth-scaled conjugate axes is labeled as s^p, and the points are listed as s_i^p:

$$s_i^p \in \left\{ \left(\pm h\sigma_{N_1} \pm \sigma_{N_2} \ldots \pm \sigma_{N_p} \right) \middle| \{N_1, N_2, \ldots, N_p\} \subset \mathcal{D} \right\}$$
$$i = 1, 2, \ldots, n2^p \binom{n}{p}$$

The various axes for the $2\mathbb{D}$ case are shown in Fig. 1a, whereas Fig. 1b shows a different perspective views of the first octant for a $3 - D$ case. It should be mentioned that all the eight octants in the $3 - D$ case are symmetrical. As the Gaussian and uniform probability density function (pdf) are symmetric, choosing points on the symmetric axes inherently satisfy all the moment constraint equations involving odd exponents. Given a single point, the fully symmetric set for each type can be easily compiled by taking all possible permutations of coordinate position and sign as shown in Table 1 [55]. For each quadrature point, two unknown variables, a weight w_i and a scaling parameter r_i, are assigned. The equations representing the integrals of monomials of desired order, known as moment constraint equations, are derived in terms of unknown variables r_i and w_i. Because of the spatial symmetries of quadrature points, the odd-order moment constraint equations are automatically satisfied, and so w_i and r_i are found by solving the even-order equations. Notice that different sets of cubature points can be found depending on p and the order of the moment constraint equations. The conjugate axes are chosen sequentially to keep the total number of quadrature points to be minimum. These new sets of so-called quadrature points are guaranteed to exactly evaluate integrals involving polynomial functions with significantly fewer points. Table 2 enumerates CUT points that satisfy seventh-order moment equations (designated as CUT6) up to dimension 6. The set of moment constraint equations using points in Table 2 is given as

Table 1 Fully symmetric set of points

Type	Sample point	No. of points
σ	$(1, 0, 0, \ldots, 0)$	$2n$
c^p	$(\underbrace{1, 1, \ldots, 1}_{p}, \underbrace{0, 0, \cdots, 0}_{(n-p)})$	$2^p \binom{n}{p}$
$s^n(h)$	$(h, 1, 1, \ldots, 1)$	$n2^n$

Table 2 Points for CUT6 ($n \leq 6$)

	Position	Weights
$1 \leq i \leq 2n$	$\boldsymbol{x}_i = r_1\sigma_i$	$\boldsymbol{w}_i = w_1$
$1 \leq i \leq 2^n$	$\boldsymbol{x}_{i+2n} = r_2 c_i^n$	$\boldsymbol{w}_{i+2n} = w_2$
$1 \leq i \leq 2n(n-1)$	$\boldsymbol{x}_{i+2n+2^n} = r_3 c_i^2$	$\boldsymbol{w}_{i+2n+2^n} = w_3$
Central weight	$\boldsymbol{x}_0 = 0$	$\boldsymbol{w}_0 = w_0$
	$N = 2n^2 + 2^n + 1$	

$$2r_1^2 w_1 + 2^n r_2^2 w_2 + 4(n-1) r_3^2 w_3 = 1 \tag{13}$$

$$2r_1^4 w_1 + 2^n r_2^4 w_2 + 4(n-1) r_3^4 w_3 = 3 \tag{14}$$

$$2^n r_2^4 w_2 + 4 r_3^4 w_3 = 1 \tag{15}$$

$$2r_1^6 w_1 + 2^n r_2^6 w_2 + 4(n-1) r_3^6 w_3 = 15 \tag{16}$$

$$2^n r_2^6 w_2 + 4 r_3^6 w_3 = 3 \tag{17}$$

$$2^n r_2^6 w_2 = 1 \tag{18}$$

$$1 - 2n w_1 - 2^n w_2 - 2n(n-1) w_3 = w_0 \tag{19}$$

The quadrature points on the principal axes have been assigned a weight of w_1 and are constrained to lie symmetrically at a distance r_1 from the origin. Points on the nth-conjugate axes (c_i^n) have been chosen with a weight w_2 and are constrained to lie symmetrically at a distance scaled by r_2. Finally, the third set of points is selected with weight w_3 and is constrained to lie symmetrically at a distance scaled by r_3 along the 2nd-conjugate axes (c_i^2). Solving Eqs. (16–18) for unknown weights leads to following analytical expressions:

$$w_1 = \frac{8 - n}{r_1^6}, \qquad w_2 = \frac{1}{2^n r_2^6}, \qquad w_3 = \frac{1}{2r_3^6} \tag{20}$$

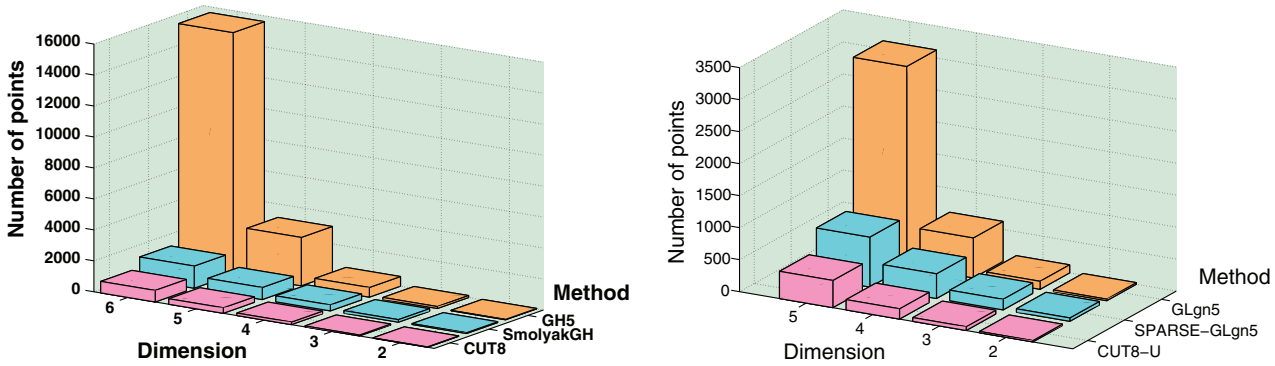
The substitution of the analytical expression Eq. (20) for weight variables in Eqs. (13–15) leads to following system of three polynomial equations:

$$\begin{aligned} 2(8-n)a_1^2 + a_2^2 + 2a_3^2(n-1) &= 1 \\ 2(8-n)a_1 + a_2 + 2a_3(n-1) &= 3 \\ a_2 + 2a_3 &= 1 \end{aligned} \tag{21}$$

where $r_1 = (1/\sqrt{a_1})$, $r_2 = (1/\sqrt{a_2})$, and $r_3 = (1/\sqrt{a_3})$. This reduced system of equations in Eq. (21) is simpler to solve than the original system of equations given by Eqs. (13–19). More details about the CUT methodology and its comparison with conventional quadrature rules to compute statistical moments can be found in [60–63]. The CUT points compared with the equivalent GLgn quadratures are shown in Fig. 2. Clearly, CUT points have much smaller growth as compared with both the GLgn quadrature points and the Smolyak scheme. In the following section, the CUT points are used to derive a unique collocation process to solve the GHJB equation.

2. *Collocation Process*

Assuming that there are total N collocation points, the collocation process leads to following system of N equations in m unknowns to exactly solve the GHJB at prescribed points, $\boldsymbol{x}_i$:



a) 9^{th}order CUT gaussian points **b) 9^{th}order CUT uniform points**

Fig. 2 Comparison of number of points.

$$S_1: \begin{cases} \boldsymbol{c}^T J(\boldsymbol{x}_1)[f(\boldsymbol{x}_1) + g(\boldsymbol{x}_1)\boldsymbol{u}_{(k)}(\boldsymbol{x}_1)] + l(\boldsymbol{x}_1) + \boldsymbol{u}_{(k)}^T R\boldsymbol{u}_{(k)} = 0 \\ \boldsymbol{c}^T J(\boldsymbol{x}_2)[f(\boldsymbol{x}_2) + g(\boldsymbol{x}_2)\boldsymbol{u}_{(k)}(\boldsymbol{x}_2)] + l(\boldsymbol{x}_2) + \boldsymbol{u}_{(k)}^T R\boldsymbol{u}_{(k)} = 0 \\ \vdots \\ \boldsymbol{c}^T J(\boldsymbol{x}_N)[f(\boldsymbol{x}_N) + g(\boldsymbol{x}_N)\boldsymbol{u}_{(k)}(\boldsymbol{x}_N)] + l(\boldsymbol{x}_N) + \boldsymbol{u}_{(k)}^T R\boldsymbol{u}_{(k)} = 0 \end{cases} \tag{22}$$

This can also be written as

$$S_2: \begin{cases} \boldsymbol{D}\boldsymbol{c} = \boldsymbol{F} \\ \boldsymbol{D} = [\,\boldsymbol{d}_1^T \quad \boldsymbol{d}_2^T \quad \cdots \quad \boldsymbol{d}_N^T\,]^T \\ \boldsymbol{d}_i^T = J(\boldsymbol{x}_i)[f(\boldsymbol{x}_i) + g(\boldsymbol{x}_i)\boldsymbol{u}_{(k)}(\boldsymbol{x}_i)], \quad i = 1, 2, \ldots, N \\ F_i = -l(\boldsymbol{x}_i) - \boldsymbol{u}_{(k)}^T R\boldsymbol{u}_{(k)} \end{cases} \tag{23}$$

Regardless of the quadrature method used, the key challenge lies in selecting the appropriate order for the polynomial basis set. This is because the number of collocation points and polynomial basis functions would not be the same for a general n-dimensional system. The growth of polynomial basis functions up to a fixed degree, d, is combinatorial in nature with the increase in state-space dimension. For a set of polynomial basis functions up to total degree d, the total number of basis functions is given as $m = \binom{n+d}{d}$, which is different from the total number of quadrature points (denoted by N) given by any of the methods. When $m < N$, that is, the number of collocation points are greater than the number of basis functions, the collocation process leads to an *overdetermined* system of equations. The *overdetermined* system typically does not possess sufficient degrees of freedom to accommodate the physics of the value function. This manifests itself as lack of an exact solution to the system of equations and hence should be avoided. An alternative approach would be to have $m > N$, that is, fewer collocation points than the number of basis function. The collocation process in this case leads to an *underdetermined* system of equations. Given the fact that the growth of CUT points is slower than the equivalent Gaussian quadrature points, they constitute a judicious choice for collocation points. For discretizing the GHJB equation, the collocation process in conjunction with the CUT methodology leads to an *underdetermined* system of equations. This additional design freedom offered by the redundant basis functions manifests itself as a lack of uniqueness in choosing the appropriate polynomial basis function set. In the next subsection, a convex optimization method is described to automatically select the appropriate basis function set from a dictionary of basis functions.

B. Basis Function Selection

As discussed earlier, the Lagrange interpolation polynomials provide the minimal order $(N-1)$ polynomial basis function set to interpolate a smooth function of one variable through N given points. This result does not hold for a general smooth function in n variables [64]. This is because there are multiple monomials with the same total degree. For example, in two dimensions, the third-degree monomials would be $x_1^3, x_1^2x_2, x_1x_2^2$, and x_2^3. In general, the number of monomials in a polynomial of n variables of total degree d would be $m = \binom{n+d}{d}$. If $N < m$, many interpolation polynomials exist. If one considers the construction of Lagrange interpolation polynomials from the tensor product of one-dimensional Lagrange interpolation polynomials, it can be observed that the total degree of the resultant interpolation polynomial grows quickly even with few number of points. The higher-order polynomial basis functions are not desirable because of Gibbs phenomenon [65].

In general, the appropriate set of basis or polynomial degree cannot be determined just from the number of points. The actual location of the points also must be considered as it affects the condition number of matrix $\boldsymbol{D}$ [64]. Hence, it is desired to construct the interpolation polynomial directly for the given set of points and dimension. In [64], an algorithm is proposed to construct a minimal degree interpolation polynomial for the given set of points in general multidimensional space. The least-degree interpolation polynomial is generated by first constructing the Vandermonde matrix [64] from the given points, with each row corresponding to one point. The columns are formed from the monomials of increasing order. Gauss elimination with partial pivoting is applied to the Vandermonde matrix, where the partial pivoting process follows special rules as outlined in [64]. This algorithm produces the least-degree interpolation polynomial for the given set of points and function values, which can in turn be used for collocation. However, this process can become computationally expensive with dimension and as the number of points increase.

To overcome this difficulty, a novel sparse optimization–based basis selection process is incorporated to select the basis that is best for the given set of points and dynamics of the system. To illustrate the utility and role of the sparse optimization tools, let us consider a cubic polynomial approximation problem in one dimension. Four samples are sufficient to compute the coefficients of the basis functions if the basis consists of monomials of degree ≤ 3. On the other hand, when the basis consists of monomials of degree ≤ 9 (total of 10 basis functions), the approximation process presents itself as an underdetermined problem, with an infinite choice of basis functions that fit the given four samples with the same tolerance. The sparse approximation process adds conditioning to this underdetermined problem by minimizing the number of participating monomials.

In particular, the system S_2 is solved by minimizing the l_1 norm of the coefficients. Ideally, l_0 norm of the coefficient vector, which corresponds to the cardinality of the coefficient vector $\boldsymbol{c}$ needs to be minimized. However, the l_0-norm optimization leads to a nonconvex optimization problem. On the other hand, l_1 norm is convex and provides a close approximation to l_0-norm cost function, by making the coefficients close to zero. Hence, to make the number of basis functions (or coefficients) equal to the number of points, the excess $m - N$ coefficients (and thereby basis functions) are sequentially made zero. The following optimization problem is proposed to select the coefficients that satisfy S_2:

Algorithm 1 Iterative l_1 optimization: $c^* = Opt_{(m,N)}(D,F,A)$

Data: A, D, F, ϕ, m, N, s
Result: $\boldsymbol{W}\boldsymbol{c}^*$ with $m-N$ zeros
$\mathcal{C} = \emptyset$
for $i = 0, i \le m-N, i = i+1$ **do**
  $\boldsymbol{c}^* = arg\min_{\boldsymbol{c}} \|\boldsymbol{W}\boldsymbol{c}\|_1$
    $D\boldsymbol{c} = F$
    $A\boldsymbol{c} \ge 0$
    $\boldsymbol{c}^T\boldsymbol{\phi}(\boldsymbol{0}) = 0$
    $\boldsymbol{c}_{\mathcal{C}} = 0$
  $\mathcal{C} = \mathcal{C} \cup \{arg\min \boldsymbol{c}^*_{\mathcal{C}}\}$

$$P_3: \begin{cases} \min_{\boldsymbol{c}} \|\boldsymbol{W}\boldsymbol{c}\|_1 \\ D\boldsymbol{c} = F \\ A\boldsymbol{c} \ge 0 \\ \boldsymbol{c}^T\boldsymbol{\phi}(\boldsymbol{0}) = 0 \end{cases}$$

The cost function in P_3 is a vector with the ith coordinate multiplied by a known weights w_i; that is, $\boldsymbol{W}\boldsymbol{c} = [w_1c_1, w_2c_2, \ldots, w_mc_m]^T$. The known weights $\boldsymbol{W} = \{w_1, w_2, \ldots, w_m\}$ can be used to relatively weight the coefficients. Given the N collocation points $\boldsymbol{x}_i$, the corresponding cost-to-go values V_i at these points is given as

$$\bar{\boldsymbol{V}} = \begin{bmatrix} V_1 \\ V_2 \\ \vdots \\ V_N \end{bmatrix} = \begin{bmatrix} \phi_1(\boldsymbol{x}_1) & \phi_2(\boldsymbol{x}_1) & \ldots & \phi_m(\boldsymbol{x}_1) \\ \phi_1(\boldsymbol{x}_2) & \phi_2(\boldsymbol{x}_2) & \ldots & \phi_m(\boldsymbol{x}_2) \\ \vdots & \vdots & \ldots & \vdots \\ \phi_1(\boldsymbol{x}_N) & \phi_2(\boldsymbol{x}_N) & \ldots & \phi_m(\boldsymbol{x}_N) \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_m \end{bmatrix} = A_{N\times m}\boldsymbol{c}_{m\times 1} \tag{24}$$

Hence, the constraint $A\boldsymbol{c} \ge 0$ corresponds to positivity of the value function. The constraint that $\boldsymbol{c}^T\boldsymbol{\phi}(0) = 0$ guarantees the generation of stable feedback controller by making the cost-to-go function to be zero at the origin. Optimization problem P_3 is convex and can be solved very efficiently by convex optimization solvers such as CVX [66].

The challenge lies in forcing $m-N$ components of $\boldsymbol{c}$ to be zero. For this purpose, the weight w_i for ith basis function is chosen to be proportional to the total degree of corresponding basis function. As the order of the basis function increases, the penalty for corresponding coefficient is appropriately increased. In this manner, the higher-degree polynomials have larger participation penalty in the collocation process. Furthermore, an iterative algorithm involving sequentially solving the optimization problem of P_3 is outlined in Algorithm 1, to ensure the cardinality of $\boldsymbol{c}$ to be N. The main idea of this algorithm is to sequentially force the least polynomial coefficient to be zero in the successive iteration, and hence this method requires exactly $m-N+1$ iterations to force the cardinality of $\boldsymbol{c}$ to be N. In Algorithm 1, the set C is the set of all indices of the vector $\boldsymbol{c}$ that are constraint to zero. Hence, the constraint $\boldsymbol{c}_C = 0$ is just a set of linear constraints requiring the corresponding components of $\boldsymbol{c}$ to be zero. Alternatively, the optimization problem can just be solved over the remaining components of $\boldsymbol{c}$.

C. Minimal Degree Interpolation Versus Sparse Interpolation

1. *Example 1: Polynomial Function*

Let us consider a simple case of polynomial interpolation in two dimensions to compare the relative merits of both the least-degree interpolation as described in [64] and the proposed approach of iterative l_1-norm optimization. The approximation domain is assumed to be a square region with opposite corners $[-5,-5]$ and $[5,5]$. GLgn quadrature with 4 points in each dimension are used, and hence there are in total 16 points in 2D. The true and known function that is being interpolated is assumed to be the following sixth-order polynomial:

$$p(x) = \frac{(3x_1^3 + 2x_1^2x_2 + 2x_1x_2^2 + 5x_2^3 + 8x_1^6 + 16x_1^3x_2^3 + 8x_2^6)}{10^3} \tag{25}$$

The function values for interpolation are computed by evaluating $p(x)$ at the 16 GLgn points; that is, $V_i = p(x_i)$ for $i = 1, 2, \ldots, 16$. The l_1-norm optimization is performed over a dictionary of polynomial basis up to total degree 6 and is given as

$$\min_{\boldsymbol{c}}: \|\boldsymbol{c}\|_1 \quad \text{subject to: } A\boldsymbol{c} = V \tag{26}$$

The total number of monomials up to sixth order is 28. As there are only 16 points, the remaining 12 basis have to be deleted from the dictionary. This is achieved by making their corresponding coefficients identically to be zero. The least-degree polynomial interpolation is computed from the algorithm as described in [64]. Figure 3 shows the plots of the true function and the interpolation polynomials computed by both the methods over uniform grid of 500 points over the domain of approximation. It can be observed that the interpolation polynomials computed by the optimization problem in Eq. (26) is accurate and in fact overlaps the given true function $p(x)$, whereas the least-degree interpolation polynomial has error in interpolation. This is made evident by the max error in interpolation in the square domain: 1.03×10^{-8} for l_1-norm optimal interpolation, while 106.1224 for the least polynomial interpolation algorithm. Further, the optimization in Eq. (26) exactly reproduces the coefficients of the monomials that appear in $p(x)$, whereas the others are made very close to zero and does so with only the 16 chosen points. Figures 3c and 3d show the absolute error surfaces for the least-degree interpolation and the l_1-norm-optimized polynomial interpolation, respectively. It can be observed that the l_1-norm-optimized polynomial interpolation achieves lower error over the entire domain.

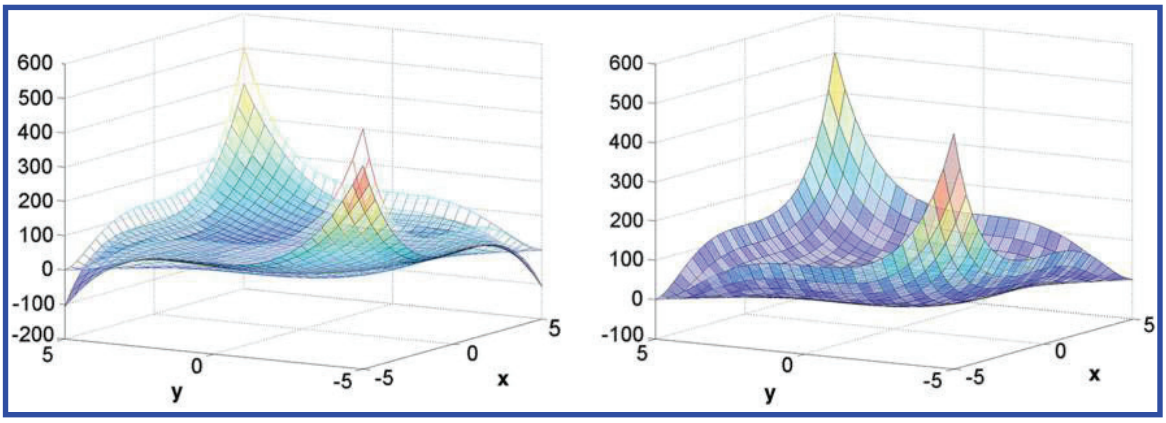
2. *Example 2: Nonpolynomial Function*

In this example, a nonpolynomial function is interpolated using the sparse optimization–based interpolation and the least-degree interpolation. The unknown true function being interpolated within the region $\Omega = [-4,4] \times [-4,4]$ is

$$p(x) = \frac{\left(x_1^4 \sin(x_1) + 10x_2^4 \cos(x_2)^2 + x_1^8x_2 + 100x_2^6\right)}{100} \tag{27}$$

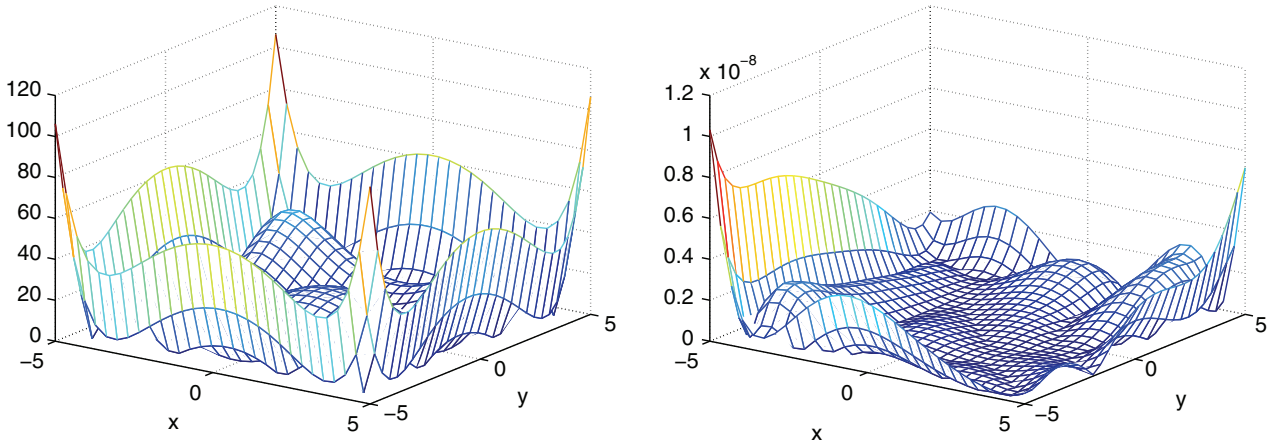
The GLgn points are used as the interpolation points for this 2D example. It can be observed that the true function has a ninth-degree monomial $x_1^8x_2$ and a sixth-degree monomial x_2^6. For the sparse optimization–based interpolation, a dictionary of basis polynomial functions up to the sixth order are used. It can be observed that the ninth-degree monomial in the true function is not present in the dictionary of basis functions. When the GLgn method is used with three points in each direction, which amounts to nine points in total, the sparse optimization–based method results in an root mean square error (RMSE) of 98.99, whereas the least-degree interpolation method results in an RMSE of 350.08. The RMSE is computed over a grid of spacing 0.25 in the domain Ω. When 4 points are used in each direction, or 16 points in total, the RMSE error for the sparse optimization method and the least-degree interpolation method is 50.84 and 330.86, respectively. The true functions and the interpolated approximation are shown in Figs. 4a and 4b. Figures 4c and 4d show the absolute error surfaces for the least-degree interpolation and the l_1-norm-optimized polynomial interpolation, respectively. It can be seen that the l_1-norm-optimized polynomial interpolation achieves lower error than the least-degree polynomial interpolation.

Finally, the complete algorithm of obtaining an approximate solution to the HJB through policy iteration is described in Algorithm 2. The constraint $A\boldsymbol{c} \ge 0$ is used to make the V_i, computed from Eq. (24), positive as the cost-to-go function has to be positive. It



a) Least degree interpoaltion vs true function

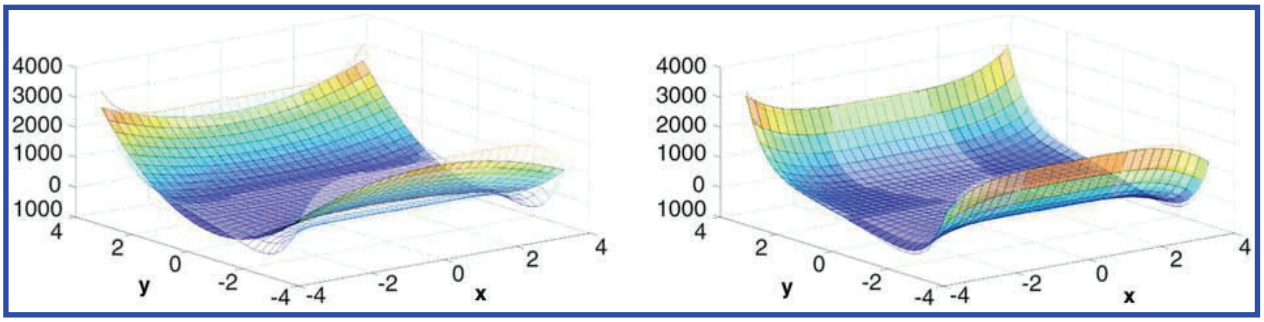
b) l_1-norm optimal interpolation vs true function



c) Least degree interpolation absolute error

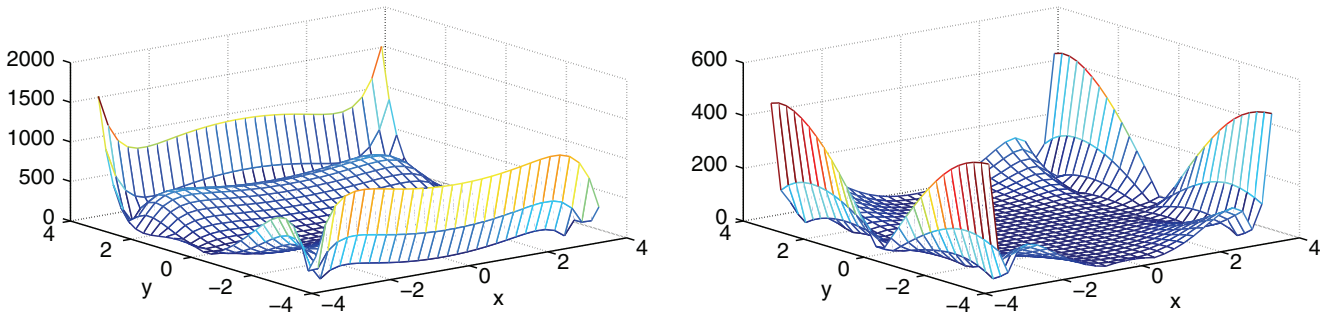
d) l_1-norm optimal interpolation absolute error

Fig. 3 Example 1: comparison of least-degree interpolation and sparsity-based interpolation.



a) Least degree interpolation vs true function

b) l_1-norm optimal interpolation vs true function



c) Least degree interpolation absolute error

d) l_1-norm optimal interpolation absolute error

Fig. 4 Example 2: comparison of least-degree interpolation and sparsity-based interpolation.

should be emphasized that the main advantage of the proposed approach is that one does not need to prescribe the structure of the feedback control law. The basis function selection process automatically selects the optimal feedback control structure and the form of the value function.

IV. Numerical Results

In this section, numerical examples are presented in support of the utility of the proposed ideas. In the first example, a two-dimensional system such as the Vander Pol oscillator is considered. In the second

Algorithm 2 Collocation-based Policy Iteration for Stationary Hamilton–Jacobi–Bellman Equation

Data: $f(\boldsymbol{x})$, $g(\boldsymbol{x})$, m basis $\boldsymbol{\phi}(\boldsymbol{x})$, initial admissible control $u^{(0)}(\boldsymbol{x})$, and collocation points $\boldsymbol{X}_i$ $i = 1, 2, \ldots, N$
Result: $\bar{V}(\boldsymbol{x})$, $\bar{u}(\boldsymbol{x})$
Compute matrix A from Eq. (24)
for $k = 0$, $k \leq K$, $k = k + 1$ **do**
  Compute D and F using $\boldsymbol{u}_{(k)}(\boldsymbol{x})$ as in S_2
  $\boldsymbol{c} = Opt_{(m,N)}(D, F, A)$
  $V_{(k)}(\boldsymbol{x}) = \boldsymbol{c}^T\boldsymbol{\phi}(\boldsymbol{x})$
  $\boldsymbol{u}_{(k+1)}(\boldsymbol{x}) = -(1/2)R^{-1}g(\boldsymbol{x})^T J(\boldsymbol{x})^T\boldsymbol{c}$

example, the three-dimensional system of satellite spin-stabilization is considered. Finally, a six-dimensional system of optimal attitude control is considered that clearly illustrates the computational advantage of the proposed optimization problem and CUT points. In all the examples, the initial stabilizing controller structure is derived by appealing to the passivity of the closed loop. Although the domain Ω of the solution process may be adapted to correspond to the level set of an initial value function, in the examples provided in the paper, a square domain in the state space is employed for convenience. Simple regions such as hypercubes make it easy to generate standard collocation points such as Gaussian quadratures and CUT, which are well behaved in terms of polynomial approximation. The problem of trajectories escaping the domain Ω is overcome by considering a large-enough domain.

A. Example 1: Vander Pol Oscillator

The first example demonstrates the effectiveness of the proposed approach in deriving a feedback controller for the Vander Pol oscillator given as follows:

$$f(x) = \begin{bmatrix} x_2 \\ -x_1 + (1 - x_1^2)x_2 \end{bmatrix} \tag{28}$$

$$g(x) = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{29}$$

The objective is to develop an optimal feedback controller that asymptotically stabilizes the system to the origin while minimizing the quadratic cost function; that is, $l(\boldsymbol{x}) = \boldsymbol{x}^T Q\boldsymbol{x}$. The proposed approach is used to approximate the value function that satisfies the HJB equation. The corresponding controller is then used as the optimal feedback controller. A square domain Ω with opposite corners at $[-3, -3]$ and $[3, 3]$ is considered for simulation purposes.

For simulation purposes, the following Q and R matrices are considered for the quadratic cost function:

$$Q = \begin{bmatrix} 50 & 0 \\ 0 & 50 \end{bmatrix}, \qquad R = \begin{bmatrix} 5 & 0 \\ 0 & 5 \end{bmatrix} \tag{30}$$

An initial stable controller to start the policy iteration process is chosen to be

$$u(\boldsymbol{x}) = [-1.2, -1.2]\boldsymbol{x} \tag{31}$$

The flow of the closed-loop system with the initial stable controller is shown in Fig. 5a. The eighth-order CUT points (CUT8) in two dimensions are used as the collocation points in Algorithm 2. In addition to the CUT8 points, collocation points corresponding to the boundary of the region Ω ($[3, 3]$, $[3, -3]$, $[-3, 3]$, $[-3, -3]$, $[3, 0]$, $[0, 3]$, $[0, -3]$, $[-3, 0]$) are added. The points defined on the boundary help in minimizing the interpolation error in the neighborhood of the boundary. Therefore, the value function has been evaluated at a total of 29 collocation points. A set of polynomial basis functions with total degree not exceeding 7 is used to define the overcomplete dictionary comprising of a total of 36 basis functions. The redundant basis functions are selected by the iterative application of the sparse optimization process. The resultant controller converges well within 20 policy iterations and is used as the optimal feedback controller. The converged optimal value function thus obtained is given as follows:

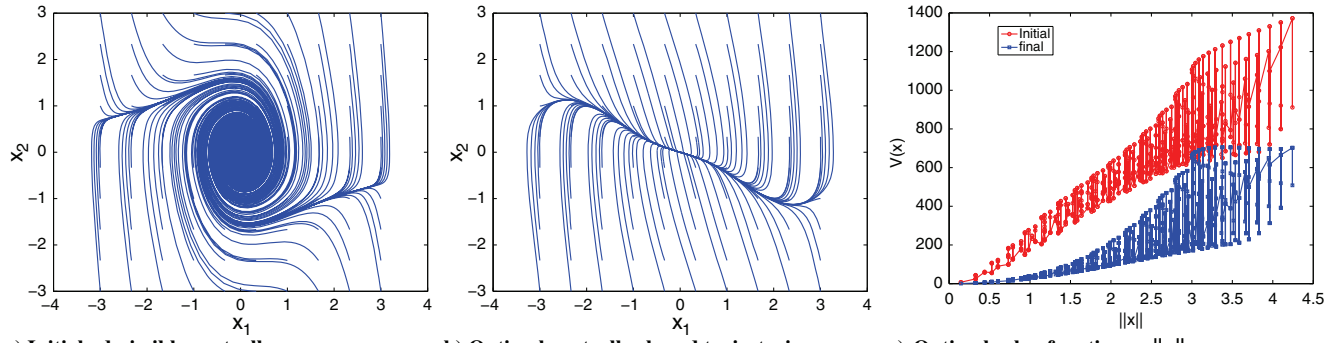
$$\begin{aligned} V(x) = {} & 3.047x_1x_2 + 27.881x_2^2 + 33.926x_1^2 - 1.735x_1^2x_2^2 \\ & + 8.09x_1^3x_2 - 1.216x_1x_2^3 - 1.265x_2^4 + 4.349x_1^4 - 0.043x_1^3x_2^3 \\ & - 0.208x_1^4x_2^2 + 0.049x_1^2x_2^4 - 0.731x_1^5x_2 + 0.106x_1x_2^5 + 0.076x_2^6 \end{aligned} \tag{32}$$

It can be observed that the proposed algorithm picks only certain monomials from overcomplete dictionary of basis functions. Figure 5b shows the closed-loop flow of the optimal trajectories in the domain of interest. The initial value function (plotted as a transparent mesh) and the converged optimal value function (plotted as a shaded mesh) are shown in Fig. 5d. Similarly, Fig. 5c shows the plot of the value function as a function of the distance of the point from origin in the state space. From these plots, it can be inferred that the optimal cost function is significantly lower than the initial value function used by the policy iteration process. Figure 5e shows the extremal field map contours of the converged optimal value function juxtaposed on top of the vector field of the unforced system. The optimal control law overpowers the natural limit cycle usually present in the unforced vector field to asymptotically drive the trajectories to the origin. Finally, Fig. 6 shows the initial and final trajectories of the state and control for the initial condition $[3, 3]$, where u_0 is the initial controller to initiate the GHJB policy iteration process and u_f is final controller. It can be seen that the large state penalty reduces the settling time of the optimal controller.

To evaluate the performance of the sparse approximation algorithm in automatically selecting appropriate basis functions, the least-degree polynomial interpolation algorithm [64] is also implemented with same set of collocation points. The basis functions selected by using the least-degree polynomial interpolation algorithm lead to poor convergence of the policy iteration process. Figure 7a shows the state flow of the closed-loop system with final computed controller after 20 iterations. The corresponding value function is shown in Fig. 7b. From these plots, it can be inferred that the state trajectories are slow in converging to the origin as compared with the results obtained by using the sparse approximation algorithm (as shown in Fig. 5). Furthermore, the controller becomes unstable toward the boundaries at the bottom and top of the region Ω as shown in Fig. 7a, in which the initial conditions for the unstable trajectories are shown with circular markers (red colored). In Fig. 7c, the level sets for the cost-to-go function do not remain closed toward the top and bottom boundaries of Ω, implying that $\dot{V}(\boldsymbol{x})$ does not remain negative in the entire region Ω. This poor performance in convergence can be attributed to the fact that the least-degree polynomial interpolation algorithm selects the appropriate basis functions solely to improve the condition number of the Vandermonde matrix. Thus, the sparse approximation algorithm clearly emerges as an appropriate choice for the basis function selection.

B. Example 2: Attitude Spin Stabilization

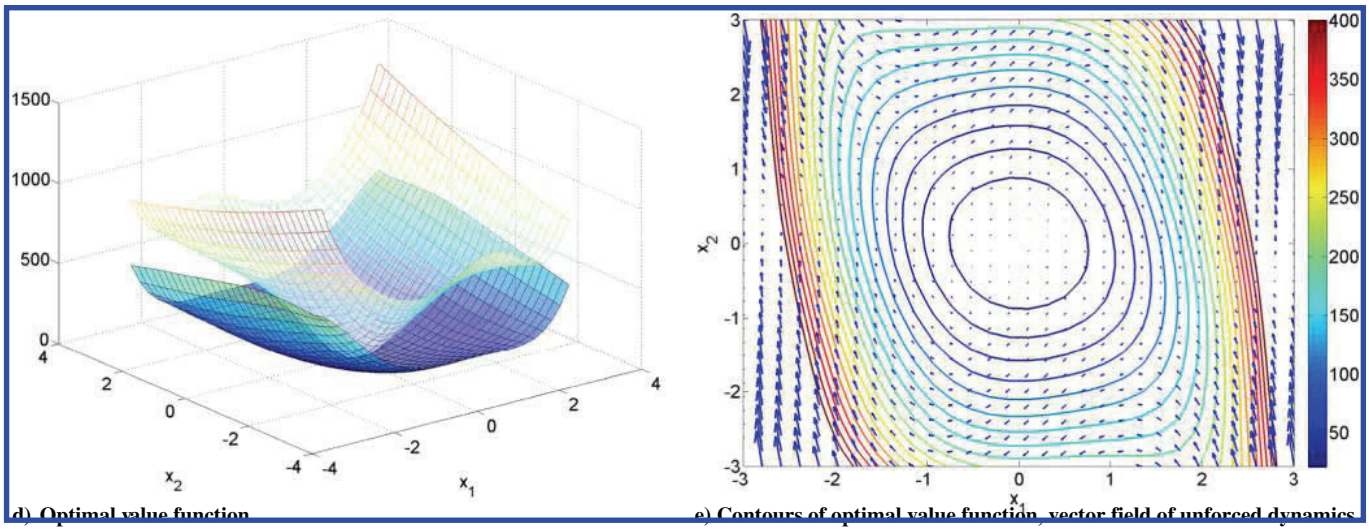
The second example corresponds to the spin stabilization of a rigid body. The spin stabilization involves the usage of appropriate feedback control laws to suppress the rotational motion of a tumbling rigid body. Accordingly, the Euler's equations for the rotation motion



a) Initial admissible controller based trajectories

b) Optimal controller based trajectories

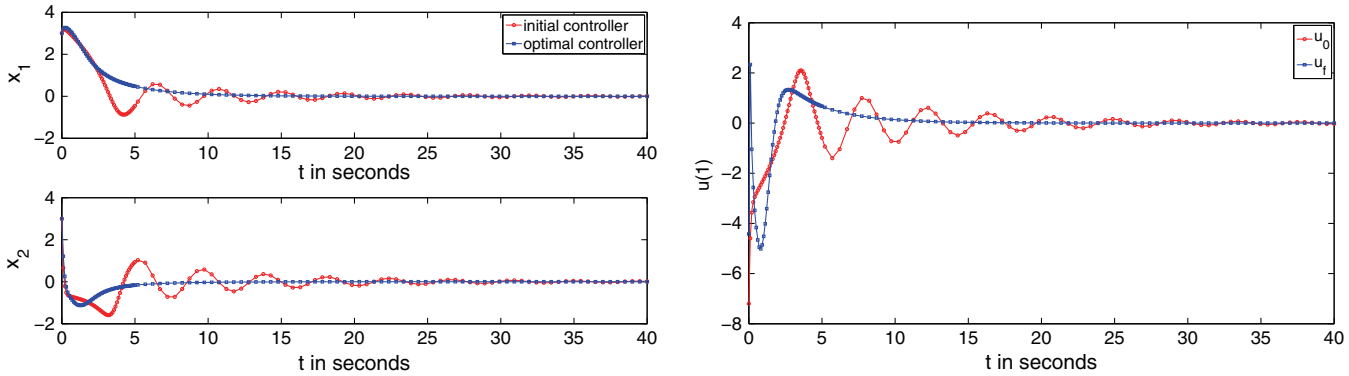
c) Optimal value function vs $\|x\|$



d) Optimal value function

e) Contours of optimal value function, vector field of unforced dynamics

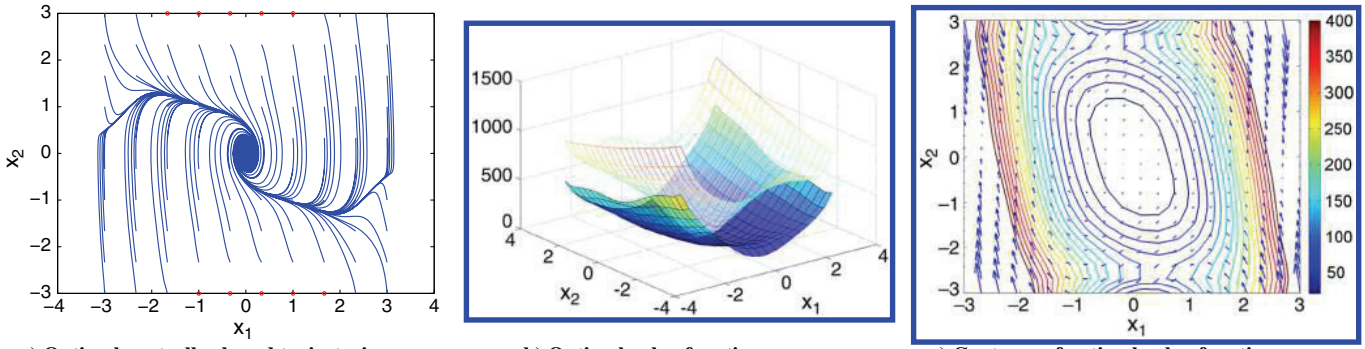
Fig. 5 Example 1: Vander Pol oscillator with low control cost (proposed approach).



a) Optimal state trajectories

b) Corresponding control trajectory

Fig. 6 Optimal state and control trajectories for initial condition [3, 3].



a) Optimal controller based trajectories

b) Optimal value function

c) Contours of optimal value function, vector field of unforced dynamics

Fig. 7 Example 1: Vander Pol oscillator with low control cost (least-degree polynomial interpolation).

of a rigid body are used as the dynamical system constraint. When expressed in body principal axis frame of reference, these equations of motion are written as

$$I_1\dot{\omega}_1 + (I_3 - I_2)\omega_2\omega_3 = u_1 \tag{33}$$

$$I_2\dot{\omega}_2 + (I_1 - I_3)\omega_1\omega_3 = u_2 \tag{34}$$

$$I_3\dot{\omega}_3 + (I_2 - I_1)\omega_2\omega_1 = u_3 \tag{35}$$

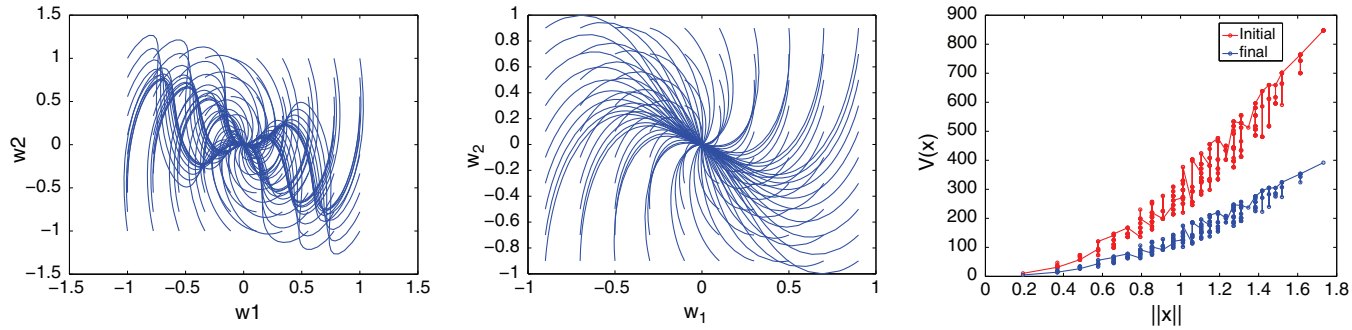
where $\omega = [\omega_1, \omega_2, \omega_3]^T$ constitutes the components of the angular velocity vector. I_1, I_2, and I_3 are the principal moments of inertia about the body axis. $\boldsymbol{u} = [u_1, u_2, u_3]^T$ is the three-dimensional control vector of torques about the body axis. A rigid body with the principal inertias $I_1 = 14\ \text{kg}\cdot\text{m}^2$, $I_2 = 10\ \text{kg}\cdot\text{m}^2$, and $I_3 = 8\ \text{kg}\cdot\text{m}^2$ is considered for evaluation of the control laws developed in this work. The control objective is to detumble the rigid body while minimizing the quadratic cost function of state ($l(\omega) = \omega^T Q\omega$) and control effort. The domain of interest is a hypercube defined by opposite corners at $[-1, -1, -1]$ and $[1, 1, 1]$. The set of polynomial basis functions not exceeding the total degree of six in three-dimensional space is used. The CUT8 points are chosen as the 59 collocation points in conjunction with 84 basis function to initialize the sparse approximation approach to solve the HJB equation. To evaluate the effectiveness of the sparse approximation approach, optimal feedback control laws are derived for two cases. The classical trade-off between the penalties associated with the state and control functions is investigated, with the control receiving higher penalty in one case and the state in the other. The state and control penalty factors, along with initial stable controller for both the cases, are given below:

$$Q = qI_{3\times3}, \quad R = rI_{3\times3}, \quad I_{3\times3} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{36}$$

$$\text{Case1: High State Cost: } q = 50, \ \ r = 3, \ \ \boldsymbol{u}_0 = -I_{3\times3}\omega \tag{37}$$

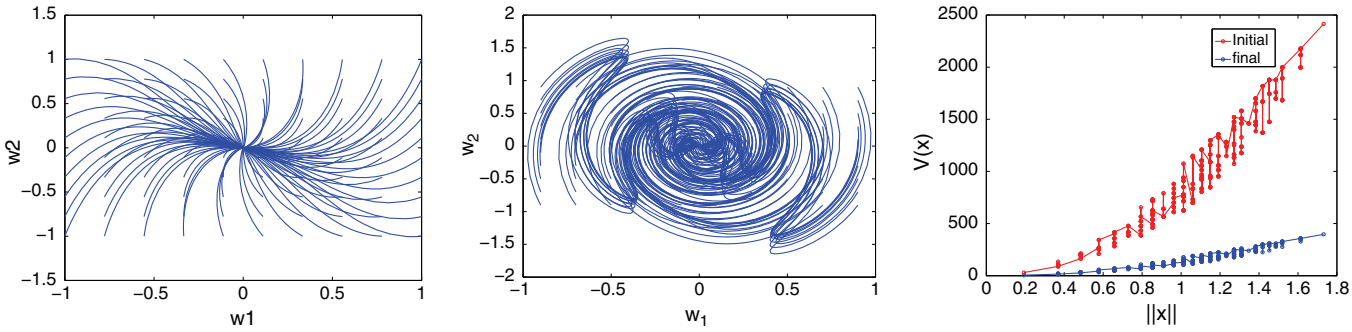
$$\text{Case2: High Control Cost: } q = 5, \ \ r = 30, \ \ \boldsymbol{u}_0 = -5I_{3\times3}\omega \tag{38}$$

Figures 8a and 9a show the state flow of the closed-loop dynamical system with initial stabilizing feedback control $\boldsymbol{u}_0$ for case 1 and case 2, respectively. The optimal closed-loop state flow fields associated with the converged optimal feedback laws for both cases are shown in Figs. 8b and 9b. The cursory examination of optimal state flow fields reveals that the optimal trajectories in case 1 yield direct path to the origin because of higher state penalty. This is in contrast to the optimal trajectories obtained in case 2, in which higher penalty on control effort ensures the proper use of the flow of the unforced dynamics to reach the origin. Figures 8c and 9c show the initial and the converged optimal value function as a function of the norm of the state vector, that is, $\|\omega\|$. It is evident from these plots that the converged value function is much smaller than the cost associated with the initial admissible policy for both the cases. The fact that multiple ordinates exist for each abscissa is representative of the fact that the cost functional is asymmetric along different cardinal directions. To develop the more intuitive understanding of the converged optimal value function, the surface and contour plots corresponding to initial and converged value function evaluated at three different values of ω_3 are shown in Figs. 10 and 11. In these plots, the initial value function is shown by the transparent surface plot, whereas the converged optimal value function is shown by the opaque surface plot. Finally, the initial and converged optimal state



a) Initial admissible controller based trajectories **b) Optimal controller based trajectories** **c) Optimal value function vs $\|x\|$**

Fig. 8 Optimal state trajectories and value function for case 1.



a) Initial admissible controller based trajectories **b) Optimal controller based trajectories** **c) Optimal value function vs $\|x\|$**

Fig. 9 Optimal state trajectories and value function for case 2.

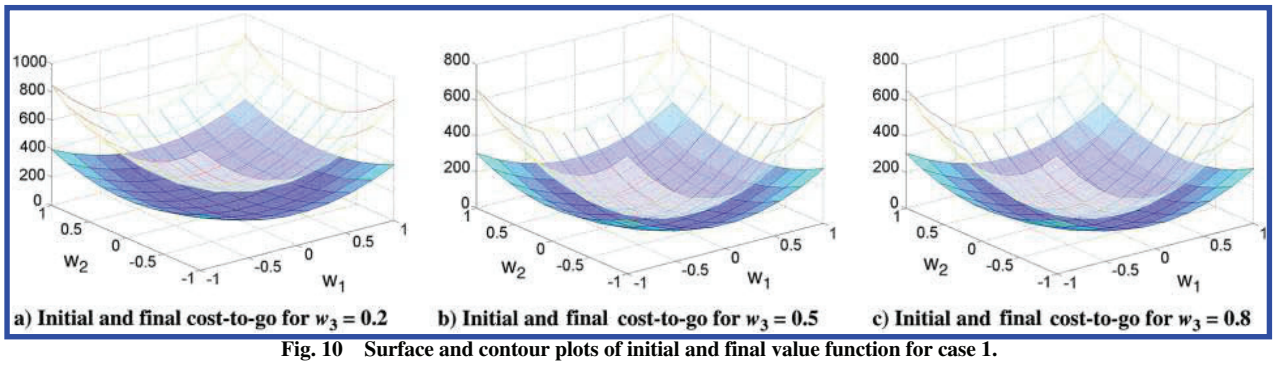


Fig. 10 Surface and contour plots of initial and final value function for case 1.

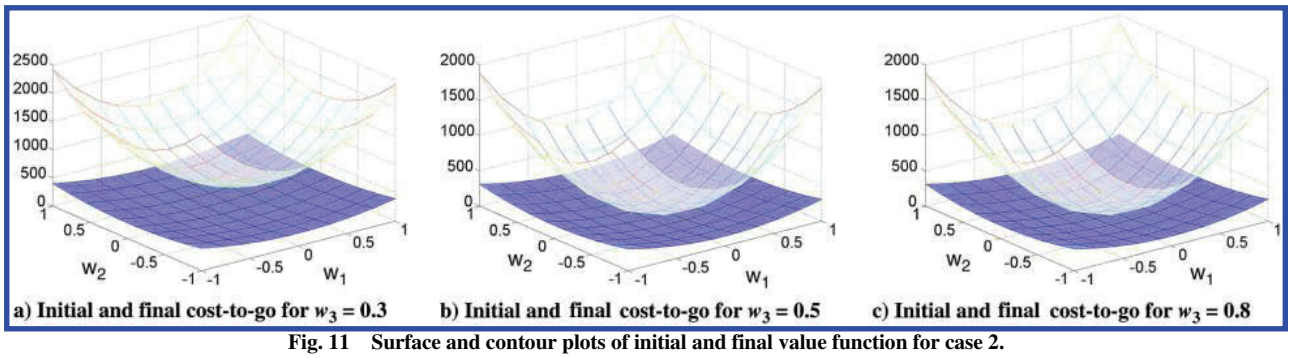


Fig. 11 Surface and contour plots of initial and final value function for case 2.

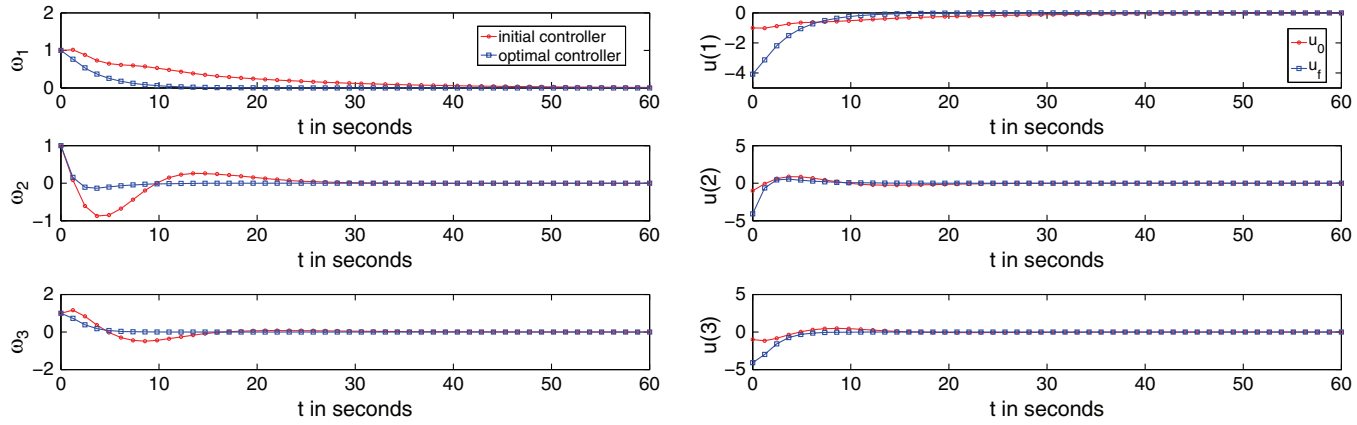


Fig. 12 Optimal state and control trajectories for initial condition [1, 1, 1] for case 1.

and control time histories for representative initial condition ([1, 1, 1]) are shown in Figs. 12 and 13 for case 1 and case 2, respectively. As expected, the high state cost leads to low settling time, whereas the high control cost leads to lower control torques.

It should be noted that the contour plots of the converged optimal value functions (as shown in Figs. 10 and 11) for both cases appear to have the same shape characteristics. This is because the converged value function in both cases is the same function as given below:

$$V(\omega) = 171.464\omega_1^2 + 122.474\omega_2^2 + 97.979\omega_3^2 \tag{39}$$

Intuitively, one expects the converged value function to explicitly reflect the penalties associated with the state and control effort. However, the converged value function in both cases, in which disparate penalties have been used in our study, is seemingly contradictory. To get more insight into this result, let us rewrite the value function as

$$V(\omega) = \frac{171.464}{I_1}\left(I_1\omega_1^2 + \frac{122.474 I_1}{171.464}\omega_2^2 + \frac{97.979 I_1}{171.464}\omega_3^2\right) \tag{40}$$

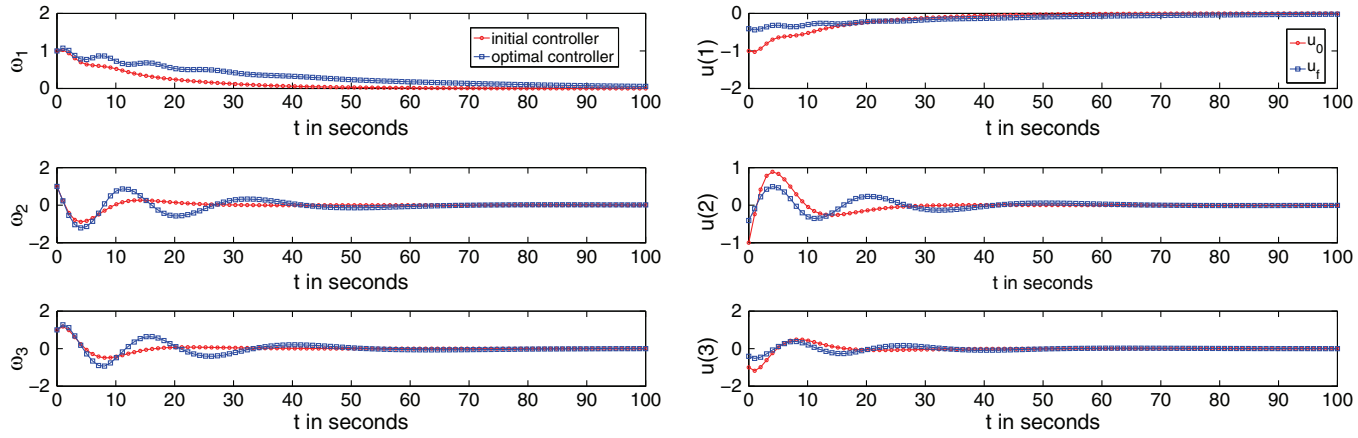
Further examination of the coefficients of the value function reveals the following structure:

$$\frac{I_1}{I_2} = \frac{14}{10} = \frac{171.464}{122.474}, \qquad \frac{I_2}{I_3} = \frac{10}{8} = \frac{122.474}{97.979}, \qquad \frac{I_1}{I_3} = \frac{14}{8} = \frac{171.464}{97.979} \tag{41}$$

It is interesting to observe that the coefficients computed from the value function enable us to establish the fact that the converged value function is proportional to the kinetic energy of the rigid body:

$$V(\omega) = \alpha(I_1\omega_1^2 + I_2\omega_2^2 + I_3\omega_3^2), \qquad \alpha = \frac{171.464}{I_1} = 12.247 \approx \sqrt{150} \tag{42}$$

The substitution of the value function in the HJB equation leads to the following expression for the equation error:



a) Optimal state trajectories b) Corresponding control trajectories

Fig. 13 Optimal state and control trajectories for initial condition [1, 1, 1] for case 2.

$$e = \frac{(qr - \alpha^2)}{r}(\omega_1^2 + \omega_2^2 + \omega_3^2) \tag{43}$$

It should be noted that our selected values of q and r lead to the coefficient of equation error, $(qr - \alpha^2)$, vanishing identically for both the cases. It is truly remarkable that the proposed computational framework automatically identifies the purely quadratic form without the cross terms from the dictionary of the sixth-polynomial basis function. This leads to the identification of a linear feedback control law as the optimal control law for the spin stabilization problem. This result agrees with the classical solution established by Debs and Athans [67]. This unique closure with an established analytical solution provides strong evidence in support of the utility of the proposed approach.

C. Example 3: Attitude Regulation

To demonstrate the computational efficiency of the sparse approximation method in high-dimensional regulation problems, the optimal attitude regulation is now considered. The six-dimensional state vector in the attitude regulation problem consists of the three elements of the Gibbs vector (also known as classical Rodrigues parameters, CRPs) to parameterize the attitude and three body angular rates along principal body axes.

$$\dot{\rho} = H(\rho)\omega \tag{44}$$

$$I_1\dot{\omega}_1 + (I_3 - I_2)\omega_2\omega_3 = u_1 \tag{45}$$

$$I_2\dot{\omega}_2 + (I_1 - I_3)\omega_1\omega_3 = u_2 \tag{46}$$

$$I_3\dot{\omega}_3 + (I_2 - I_1)\omega_2\omega_1 = u_3 \tag{47}$$

where

$$H(\rho) = \frac{1}{2}(I - S(\rho) + \rho\rho^T) \tag{48}$$

$$S(\rho) = \begin{bmatrix} 0 & \rho_3 & -\rho_2 \\ -\rho_3 & 0 & \rho_1 \\ \rho_2 & -\rho_1 & 0 \end{bmatrix} \tag{49}$$

The objective is to optimally regulate the system to the origin while minimizing the quadratic cost function, with $l(\rho, \omega) = q(\rho^T\rho + \omega^T\omega)$. The initial admissible controller to initialize the policy iteration process is assumed to be given by

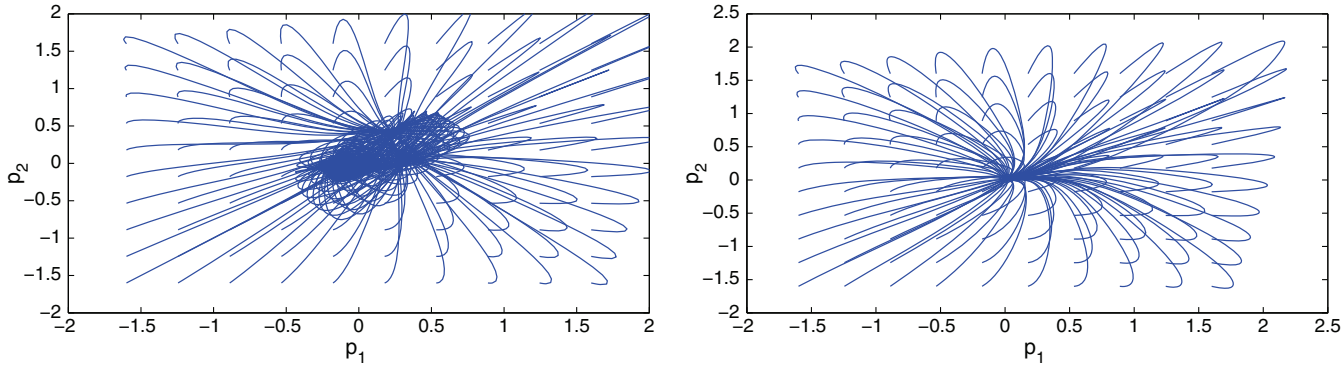
$$\boldsymbol{u} = -\begin{bmatrix} 15 & 0 & 0 & 5 & 0 & 0 \\ 0 & 15 & 0 & 0 & 5 & 0 \\ 0 & 0 & 15 & 0 & 0 & 5 \end{bmatrix}\begin{bmatrix} \rho \\ \omega \end{bmatrix} \tag{50}$$

This problem is also considered in [68], in which the Galerkin method is used to solve the GHJB equation. Because of the high computationally complexity involved in the integration process and the large number of basis functions considered, the authors heuristically add basis functions to have as few polynomial bases as possible. The sparse approximation–based collocation approach, described in Algorithm 2, aids in eliminating this heuristic process, by optimally selecting the suitable basis functions.

The ninth-order CUT points corresponding to the Gaussian function, 745 in number, are used as the collocation points. This is in contrast to using ninth-order Gauss Hermite quadrature points, with 5 points in each direction, leading to a total number of 15,625 points to achieve the same level of interpolation accuracy in the domain of interest. In addition to the CUT points within the domain, points are also added at the boundary of the domain Ω to improve the interpolation accuracy near the boundaries of the hyper cube. A total of 284 boundary points are added at the coordinates corresponding to the following discrete sets: $[\pm 1, \pm 1, \pm 1, \pm 1, \pm 1, \pm 1]$, $[\pm 1, \pm 1, 0, 0, 0, 0]$, and $[\pm 1, \pm 1, \pm 1, 0, 0, 0]$. A set of multivariate polynomial basis functions not exceeding a total degree of 8 is considered to compile the overcomplete dictionary of a total of 3003 basis functions. It can be further inferred that the ninth-order Gauss Hermite points lead to an overdetermined system of equations with no exact solution, and one has to resort to fifth-order Gauss Hermite quadrature points, with three points along each direction to guarantee a solution. Clearly, this leads to larger interpolation error and loss in solution accuracy.

The closed-loop state flow fields corresponding to the initial admissible and the converged optimal feedback laws in $\rho_1 - \rho_2$ subspace at a given value of other state vectors are shown in Fig. 14. Figure 15 shows the initial and converged optimal value functions as a function of the norm of the state vector. As anticipated, the converged value function is observed to be lower in cost than the cost associated with the initial admissible policy for both the cases. The fact that multiple ordinates exist for each abscissa is representative of the fact that the cost functional is asymmetric along different cardinal directions and is once again prominently shown in Fig. 15. Representative state and control time histories corresponding to initial admissible and converged optimal feedback control laws associated with the initial state vector $[1.5, 1.5, 1.5, 0.8, 0.8, 0.8]^T$ are reported in Figs. 16 and 17. These plots clearly show that the optimal trajectories asymptotically converge to origin.

Recall that the largest-order Gauss Hermite quadrature points one can use for the collocation process along with eighth-order multivariate basis functions is 5 to realize an underdetermined system of equations. To emphasize the efficacy of the ninth-order CUT



a) Trajectories for initial control b) Trajectories for final control

Fig. 14 Example 3: optimal attitude regulation.

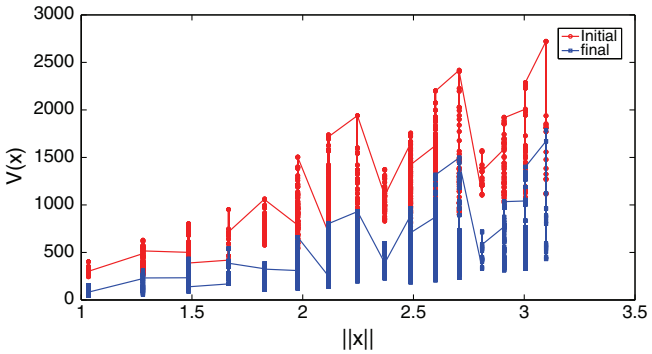


Fig. 15 Example 3: initial and final cost.

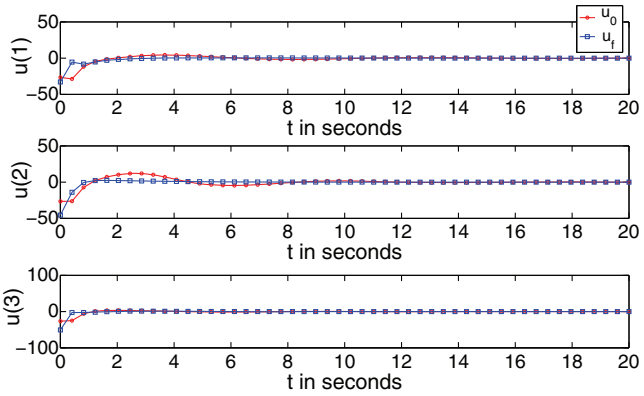
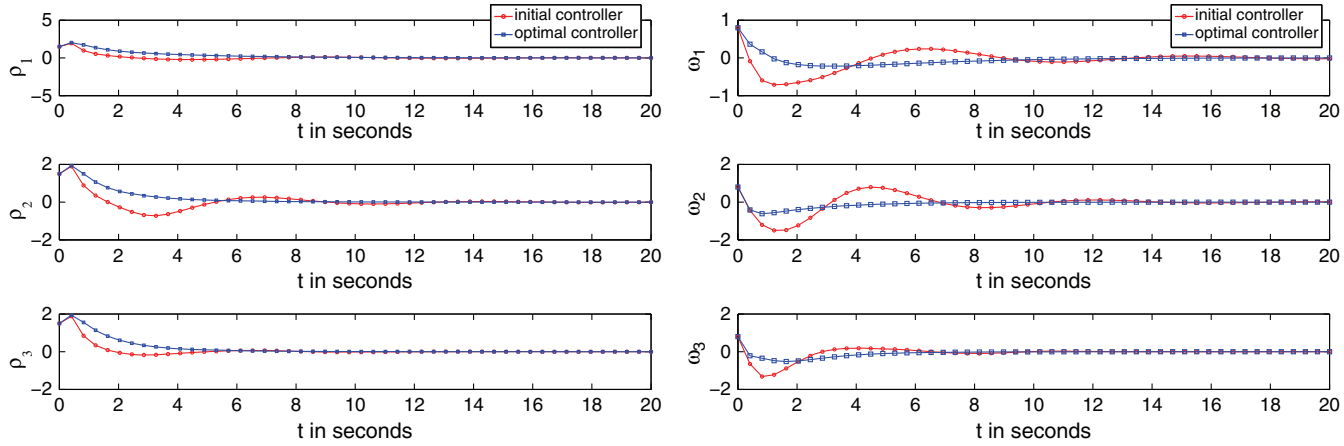


Fig. 17 Example 3: final control trajectories computed by CUT points for a specific initial condition.

points, the fifth Gauss Hermite points (729 in number), along with the 284 boundary points, are used to solve the GHJB equation for each policy iteration. The closed-loop state flow fields in the ρ_1, ρ_2 subspace for the initial stabilized and the converged optimal feedback laws are shown in Fig. 18a. Figure 18b shows the initial admissible and the converged optimal value functions as a function of the norm of the state vector. For ease in comparison, the number of iterations required for convergence of the optimal value function in the policy iteration process is held fixed. The negative value of final converged value function in these plots can be attributed to the use of lower-order Gauss Hermite quadrature points. This fact is also evident by the observation that some trajectories in Fig. 18a corresponding to initial condition represented by red circles become unbounded and hence are not shown. These results clearly illustrate the effectiveness of the proposed methodology in solving the HJB equation accurately for high-dimension problems.

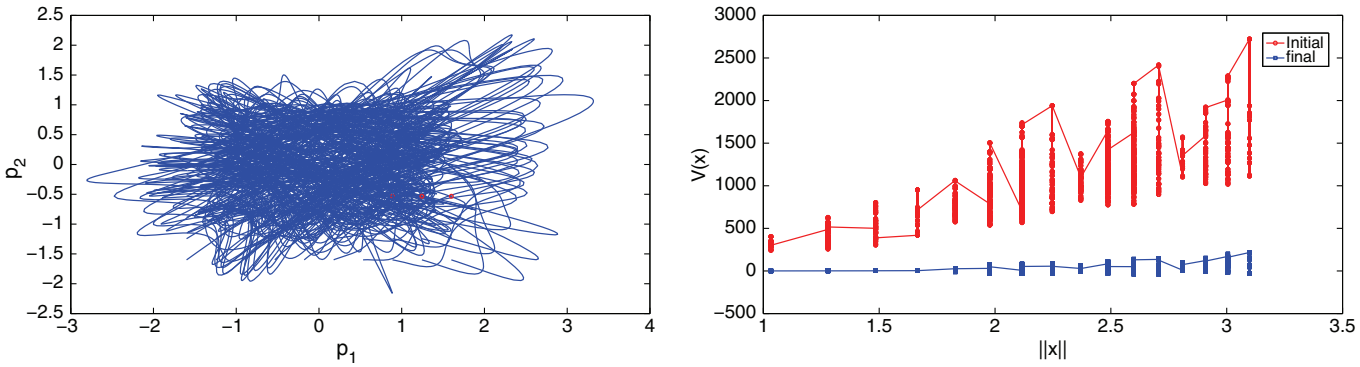
D. Computational Details

The curse of dimensionality is pervasive in solving high-dimensioned multivariate PDEs and represents a formidable challenge. The proposed approach is also affected by this challenge. This is evident from the increase in the size of the dictionary along with the number of the collocation points as a function of the state dimension. In Example 3, solution of the HJB equation necessitated the use of 745 collocation points in conjunction with 3003 eighth-order basis functions in six dimensions, whereas the solution of Example 1 is produced using 36 seventh-order polynomial basis functions with 29 collocation points. Notice that the use of nonproduct quadrature methods like CUT limits the growth of collocation points to be much smaller than the combinatorial growth of the basis functions. The proposed approach exploits the sparsity



a) Optimal state trajectories for ρ_1, ρ_2 and ρ_3 b) Optimal State trajectories for ω_1, ω_2 and ω_3

Fig. 16 Example 3: final state trajectories computed by CUT points for a specific initial condition.



a) Final control trajectories using Gauss Hermite points b) Initial and final cost using Gauss Hermite points

Fig. 18 Example 3: attitude control problem.

that results from the disparate distinction between the growth patterns of the collocation points and the interpolation functions with the state dimension. Exploiting advances in sparse approximation tools, a systematic approach is thus realized to mitigate the curse of dimension.

Building upon the fact that the sparse approximation problem is a convex optimization problem, a variety of recently developed tools can be used to aid in implementing the proposed solution approach. In this paper, an open-source convex optimization toolbox from CMSoft (called CVX) is employed to solve the sparse approximation problem. Just to provide an idea, the processor time to compute the feedback controller for Example 3 in a Matlab framework was about 20 min. Finally, it is of consequence to emphasize that the feedback control synthesis approach realized in this paper subsumes the realization of the feedback control structure along with the optimal state feedback gains, and can be carried out entirely offline. The computational time incurred in the synthesis process can therefore be modest.

V. Conclusions

A computationally efficient algorithm is discussed to derive the optimal feedback control laws for infinite time problems while solving the infinite time HJB equation. The numerical approaches to solve the Hamilton–Jacobi–Bellman (HJB) equation suffer from *curse of dimensionality* with an increase in the state dimension, making conventional methods computationally intractable for deducing feedback laws for optimal control of a variety of dynamical systems of interest in engineering practice. The solution process consists of iteratively solving the linear generalized HJB (GHJB) equation starting with an admissible stable controller. Recent advances in nonproduct quadrature methods and sparse approximation results are used to tackle the challenges associated with the growth of the state dimension. The Conjugate Unscented Transformation (CUT) method is used to define collocation points in an n-dimensional space. While maintaining the same level of interpolation accuracy as the conventional Gauss quadrature points for a given order, the CUT collocation process generates much fewer points, aiding in an effective solution of the GHJB equation in moderate- to large-dimensional problems. To overcome the challenges associated with the combinatorial nature of the growth of the multivariate polynomial basis function that exceeds the growth rate of the number of collocation points, a sparse approximation–based approach is used to select appropriate basis functions from an overcomplete dictionary. The basis function selection approach solves an l_1-norm convex optimization problem to carry out the selection process while simultaneously identifying a form of the optimal feedback control laws that is typically unknown in a variety of problems. Several numerical examples are presented to provide evidence in support of the efficacy of the proposed approach in solving the HJB equation in a computationally efficient manner. In particular, the unique agreement of the computed solution with the analytical solution for the spin stabilization of a rigid body problem provides a strong basis for optimism in demonstrating the utility of the approach in solving the HJB equation. Although the CUT points have been used in this work as a potential collocation points, the developed methodology is fairly generic and can make use of any approach to generate collocation points in general n-dimensional space.

Acknowledgments

This paper is based on the work jointly supported by the National Science Foundation under Award No. CMMI-1054759, CMMI-1634590, and AFOSR Grant FA9550-15-1-0313.

References

[1] Bryson, A. E., and Ho, Y. C., *Applied Optimal Control: Optimization, Estimation and Control*, Taylor and Francis, Connecticut, 1975, Chap. 2.

[2] Vinter, R., *Dynamic Programming*, Birkh äuser Boston, Boston, 2010, Chap. 6.

[3] Caratheodory, C., *Calculus of Variations and Partial Differential Equations of the First Order*, English ed., American Mathematical Soc., Providence, Rhode Island, 1999, Chap. 1.

[4] Athans, M., and Falb, P. L., *Optimal Control: An Introduction to the Theory and Its Applications*, Dover Publ., New York, 2014, Chap. 8 (Reprint).

[5] Gelfand, I. M., and Fomin, S. V., *Calculus of Variations*, Dover Publ., Mineola, New York, 1963, Chap. 3.

[6] Pontryagin, L. S., Boltyanskii, V. G., Gamkrelidze, R. V., and Mishchenko, E. F., *L. S. Pontryagin Selected Works, Volume 4: The Mathematical Theory of Optimal Processes*, Interscience, New York, 1962, Chap. 1.

[7] Vadali, S. R., and Junkins, J. L., "Optimal Openloop and Stable Feedback Control of Rigid Spacecraft Attitude Maneuvers," *Journal of Astronautical Sciences*, Vol. 32, No. 2, 1984, pp. 105–122.

[8] Junkins, J. L., and Turner, J. D., *Optimal Spacecraft Rotational Maneuvers*, Elsevier, New York, 1986, Chap. 7.

[9] Bellman, R. E., *Perturbation Techniques in Mathematics, Engineering and Physics*, Dover Publ., Mineola, NY, 2003, Chap. 1.

[10] Betts, J. T., "Survey of Numerical Methods for Trajectory Optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 21, No. 2, 1998, pp. 193–207. doi:10.2514/2.4231

[11] Fahroo, F., and Ross, I. M., "Direct Trajectory Optimization by a Chebyshev Pseudospectral Method," *Journal of Guidance, Control, and Dynamics*, Vol. 25, No. 1, 2002, pp. 160–166. doi:10.2514/2.4862

[12] Elnagar, J., Kazemi, M. A., and Razzaghi, M., "The Pseudospectral Legendre Method for Discretizing Optimal Control Problems," *IEEE Transactions on Automatic Control*, Vol. 40, No. 10, 1995, pp. 1793–1796. doi:10.1109/9.467672

[13] Elnagar, G. N., and Kazemi, M. A., "Pseudospectral Chebyshev Optimal Control of Constrained Nonlinear Dynamical Systems," *Computational Optimization and Applications*, Vol. 11, No. 2, 1998, pp. 195–217. doi:10.1023/A:1018694111831

[14] Williams, P., "Jacobi Pseudospectral Method for Solving Optimal Control Problems," *Journal of Guidance, Control, and Dynamics*,

Vol. 27, No. 2, 2004, pp. 293–297.
doi:10.2514/1.4063
[15] Darby, C. L., Hager, W. W., and Rao, A. V., "Direct Trajectory Optimization Using a Variable Low-Order Adaptive Pseudospectral Method," *Journal of Spacecraft and Rockets*, Vol. 48, No. 3, 2011, pp. 433–445.
doi:10.2514/1.52136
[16] Ross, I. M., and Fahroo, F., *Legendre Pseudospectral Approximations of Optimal Control Problems*, Vol. 295, Springer, Berlin, 2003, pp. 327–342.
[17] Darby, C. L., Garg, D., and Rao, A. V., "Costate Estimation Using Multiple-Interval Pseudospectral Methods," *Journal of Spacecraft and Rockets*, Vol. 48, No. 5, 2011, pp. 856–866.
doi:10.2514/1.A32040
[18] Dreyfus, S., *Dynamic Programming and Calculus of Variations*, Academic Press, New York, 1965, Chap. 2.
[19] Bertsekas, D. P., *Dynamic Programming and Optimal Control*, 2nd ed., Athena Scientific, Belmont, MA, 2000, Chap. 3.
[20] Cloutier, J. R., "State-Dependent Riccati Equation Techniques: An Overview," *American Control Conference*, Red Hook, NY, June 1997, pp. 932–936.
[21] Xin, M., Balakrishnan, S. N., Stansbery, D. T., and Ohlmeyer, E. J., "Nonlinear Missile Autopilot Design with θ-D Technique," *Journal of Guidance, Control, and Dynamics*, Vol. 27, No. 3, 2004, pp. 406–417.
doi:10.2514/1.1217
[22] Park, C., and Scheeres, D. J., "Solutions of Optimal Feedback Control Problems with General Boundary Conditions Using Hamiltonian Dynamics and Generating Functions," *American Control Conference*, Red Hook, NY, 2004, pp. 679–684.
[23] Deprit, A., "Canonical Transformations Depending on a Small Parameter," *Celestial Mechanics*, Vol. 1, No. 1, 1969, pp. 12–30.
doi:10.1007/BF01230629
[24] Albrekht, E. G., "On the Optimal Stabilization of Nonlinear Systems," *Journal of Applied Mathematics and Mechanics*, Vol. 25, No. 5, 1961, pp. 1254–1266.
doi:10.1016/0021-8928(61)90005-3
[25] Carrington, C. K., and Junkins, J. L., "Optimal Nonlinear Feedback Control for Spacecraft Attitude Maneuvers," *Journal of Guidance, Control, and Dynamics*, Vol. 9, No. 1, 1986, pp. 99–107.
doi:10.2514/3.20073
[26] Vadali, S. R., and Sharma, R., "Optimal Finite-Time Feedback Controllers for Nonlinear Systems with Terminal Constraints," *Journal of Guidance, Control, and Dynamics*, Vol. 29, No. 4, 2006, pp. 921–928.
doi:10.2514/1.16790
[27] Kristic, M., and Tsiotras, P., "Inverse Optimal Stabilization of a Rigid Spacecraft," *IEEE Transactions on Automatic Control*, Vol. 44, No. 5, 1999, pp. 1042–1049.
doi:10.1109/9.763225
[28] Sassano, M., and Astolfi, A., "Dynamic Approximate Solutions of the HJ Inequality and of the HJB Equation for Input Affine Nonlinear Systems," *IEEE Transactions on Automatic Control*, Vol. 57, No. 10, 2012, pp. 2490–2503.
doi:10.1109/TAC.2012.2186716
[29] Gershwin, S. B., and Jacobson, D. H., "A Discrete-Time Differential Dynamic Programming Algorithm with Application to Optimal Orbit Transfer," *AIAA Journal*, Vol. 8, No. 9, 1970, pp. 1616–1626.
doi:10.2514/3.5955
[30] Colombo, C., Vasile, M., and Radice, G., "Optimal Low-Thrust Trajectories to Asteroids Through an Algorithm Based on Differential Dynamic Programming," *Celestial Mechanics and Dynamical Astronomy*, Vol. 105, Nos. 1–3, 2009, pp. 75–112.
doi:10.1007/s10569-009-9224-3
[31] Lantoine, G., and Russell, R. P., "A Hybrid Differential Dynamic Programming Algorithm for Constrained Optimal Control Problems. Part 1: Theory," *Journal of Optimization Theory and Applications*, Vol. 154, No. 2, 2012, pp. 382–417.
doi:10.1007/s10957-012-0039-0
[32] Lantoine, G., and Russell, R. P., "A Hybrid Differential Dynamic Programming Algorithm for Constrained Optimal Control Problems. Part 2: Application," *Journal of Optimization Theory and Applications*, Vol. 154, No. 2, 2012, pp. 418–442.
doi:10.1007/s10957-012-0038-1
[33] Olympio, J. T., "A Continuous Implementation of a Second-Variation Optimal Control Method for Space Trajectory Problems," *Journal of Optimization Theory and Applications*, Vol. 158, No. 3, 2013, pp. 687–716.
doi:10.1007/s10957-013-0274-z
[34] GrÃijne, L., "An Adaptive Grid Scheme for the Discrete Hamilton-Jacobi-Bellman Equation," *Numerische Mathematik*, Vol. 75, No. 3, 1997, pp. 319–337.
doi:10.1007/s002110050241
[35] Boulbrachene, M., and Haiour, M., "The Finite Element Approximation of Hamilton-Jacobi-Bellman Equations," *Computers and Mathematics with Applications*, Vol. 41, Nos. 7–8, 2001, pp. 993–1007.
doi:10.1016/S0898-1221(00)00334-5
[36] Park, C., and Tsiotras, P., "Sub-Optimal Feedback Control Using a Successive Wavelet-Galerkin Algorithm," *Proceedings of the American Control Conference*, American Automatic Control Council, Dayton, OH, 2003, pp. 1926–1931.
[37] Abu-Khalaf, M., and Lewus, F. L., "Nearly Optimal Control Laws for Nonlinear Systems with Saturating Actuators Using a Neural Network HJB Approach," *Automatica*, Vol. 41, No. 5, 2005, pp. 779–791.
doi:10.1016/j.automatica.2004.11.034
[38] Abu-Khalaf, M., and Lewis, F. L., "Nearly Optimal State Feedback Control of Constrained Nonlinear Systems Using a Neural Networks HJB Approach," *IFAC Annual Reviews in Control*, Vol. 28, No. 2, 2004, pp. 239–251.
doi:10.1016/j.arcontrol.2004.07.002
[39] Beard, R., Saridis, G., and Wen, J., "Gelarkin Approximation of the Generalized Hamilton-Jacobi Bellman Equation," *Automatica*, Vol. 33, No. 12, 1997, pp. 2159–2177.
doi:10.1016/S0005-1098(97)00128-3
[40] Beard, R., and McLain, T., "Successive Galerkin Approximation Techniques for Nonlinear Optimcal and Robust Control," *International Journal of Control*, Vol. 71, No. 5, 1998, pp. 717–743.
doi:10.1080/002071798221542
[41] Saridis, G. N., and Lee, C. S., "An Approximation Theory of Optimal Control for Trainable Manipulators," *IEEE Transactions Systems, Man, and Cybernetics*, Vol. 9, No. 3, 1979, pp. 152–159.
doi:10.1109/TSMC.1979.4310171
[42] Klienman, D., "On an Iterative Technique for Riccati Equation Computations," *IEEE Transactions on Automatic Control*, Vol. 13, No. 1, 1968, pp. 114–115.
doi:10.1109/TAC.1968.1098829
[43] Osher, S., and Sethian, J. A., "Fronts Propagating with Curvature Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulations," *Journal of Computational Physics*, Vol. 79, No. 1, 1988, pp. 12–49.
doi:10.1016/0021-9991(88)90002-2
[44] Osher, S., and Fedkiw, R., *Level Set Methods and Dynamic Implict Surfaces*, Vol. 153, Applied Mathematical Sciences, Springer, New York, 2003, Chap. 5.
[45] Shaw-Cortez, W., and Frew, E. W., "Efficient Trajectory Development for Unmanned Aircraft System Dynamic Soaring," *Journal of Guidance, Control, and Dynamics*, Vol. 38, No. 3, Jan. 2015, pp. 519–523.
doi:10.2514/1.G000543
[46] Beard, R. W., Saridis, G. N., and Wen, J. T., "Galerkin Approximations of the Generalized Hamilton-Jacobi-Bellman Equation," *Automatica*, Vol. 33, No. 12, 1997, pp. 2159–2177.
doi:10.1016/S0005-1098(97)00128-3
[47] Bertsekas, D. P., *Dynamic Programming and Optimal Control*, Vol. 2, Athena Scientific, Belmont, MA, 1995.
[48] Kalman, R. E., "Contributions to the Theory of Optimal Control," *Boletín de la Sociedad Matemática Mexicana*, Vol. 5, No. 2, 1960, pp. 102–119.
[49] Lawden, D. F., *Optimal Trajectories for Space Navigation*, Butterworths Mathematical Texts, London, 1963, Chap. 1.
[50] Stone, F. P., "The 'Worried Well' Response to CBRN Events: Analysis and Solutions," Technical Rept. DTIC Document, USAF Counter Profileration Center, Maxwell Air Force Base, Alabama, 2007.
[51] Stone, M. H., "The Generalized Weierstrass Approximation Theorem," *Mathematics Magazine*, Vol. 21, No. 4, 1948, pp. 167–184.
doi:10.2307/3029750
[52] Pinkus, A., "Weierstrass and Approximation Theory," *Journal of Approximation Theory*, Vol. 107, No. 1, 2000, pp. 1–66.
[53] Reddy, J. N., *An Introduction to the Finite Element Method*, Vol. 2, McGraw–Hill, New York, 1993, Chap. 2.
[54] Gerstner, T., and Griebel, M., "Numerical Integration Using Sparse Grids," *Numerical Algorithms*, Vol. 18, No. 3/4, 1998, pp. 209–232.
doi:10.1023/A:1019129717644
[55] Stroud, A. H., *Approximate Calculation of Multiple Integrals*, Prentice Hall, Englewood Cliffs, NJ, 1971, Chap. 2.
[56] Stroud, A. H., and Secrest, D., *Gaussian Quadrature Formulas*, Prentice Hall, Englewood Cliffs, NJ, 1966.
[57] Julier, S., Uhlmann, J., and Durrant-Whyte, H., "A New Method for the Nonlinear Transformation of Means and Covariances in Filters and Estimators," *IEEE Transactions on Automatic Control*, Vol. 45, No. 3, March 2000, pp. 477–482.
doi:10.1109/9.847726

[58] Adurthi, N., Singla, P., and Singh, T., "The Conjugate Unscented Transform: An Approach to Evaluate Multi-Dimensional Expectation Integrals," *American Control Conference (ACC)*, Red Hook, NY, June 2012, pp. 5556–5561.

[59] Adurthi, N., and Singla, P., "Conjugate Unscented Transformation-Based Approach for Accurate Conjunction Analysis," *Journal of Guidance, Control, and Dynamics*, Vol. 38, No. 9, 2015, pp. 1642–1658.

[60] Adurthi, N., "The Conjugate Unscented Transform—A Method to Evaluate Multidimensional Expectation Integrals," Master's Thesis, State Univ. of New York, Buffalo NY, 2013.

[61] Adurthi, N., Singla, P., and Singh, T., "Conjugate Unscented Transform Rules for Uniform Probability Density Functions," *American Control Conference*, Red Hook, NY, June 2013, pp. 2454–2459.

[62] Adurthi, N., Singla, P., and Singh, T., "Conjugate Unscented Transform and Its Application to Filtering and Stochastic Integral Calculation," *AIAA Guidance, Navigation, and Control Conference*, Reston, VA, 2012, p. 4934.

[63] Nagavenkat, A., Singla, P., and Singh, T., "Conjugate Unscented Transform and Its Application to Filtering and Stochastic Integral Calculation," *AIAA Guidance, Navigation, and Control Conference*, Aug. 2012.

[64] De Boor, C., and Ron, A., "Computational Aspects of Polynomial Interpolation in Several Variables," *Mathematics of Computation*, Vol. 58, No. 198, 1992, pp. 705–727. doi:10.1090/S0025-5718-1992-1122061-0

[65] Golub, G. H., and Van Loan, C. F., *Matrix Computations*, 3rd ed., The Johns Hopkins University Press, Baltimore, MD, 1996, pp. 140–149, 601.

[66] Grant, M., and Boyd, S., "CVX: Matlab Software for Disciplined Convex Programming," Ver. 2.1, March 2014, http://cvxr.com/cvx [retrieved 16 Aug. 2015].

[67] Debs, A. S., and Athans, M., "On the Optimal Angular Velocity Control of Asymmetrical Space Vehicles," *IEEE Transactions on Automatic Control*, Vol. 14, No. 1, 1969, pp. 80–83. doi:10.1109/TAC.1969.1099098

[68] Lawton, J. R., Beard, R. W., and McLain, T. W., "Successive Galerkin Approximation of Nonlinear Optimal Attitude," *American Control Conference*, Vol. 6, Red Hook, NY, 1999, pp. 4373–4377.