Faculty and Researchers   Faculty and Researchers' Publications

2017-04-08

Mitigating the curse of dimensionality: sparse grid characteristics method for optimal feedback control and HJB equations

Kang, Wei; Wilcox, Lucas C.

Springer

CrossMark

Mitigating the curse of dimensionality: sparse grid characteristics method for optimal feedback control and HJB equations

Wei Kang[1] · **Lucas C. Wilcox**[1]

Abstract We address finding the semi-global solutions to optimal feedback control and the Hamilton–Jacobi–Bellman (HJB) equation. Using the solution of an HJB equation, a feedback optimal control law can be implemented in real-time with minimum computational load. However, except for systems with two or three state variables, using traditional techniques for numerically finding a semi-global solution to an HJB equation for general nonlinear systems is infeasible due to the curse of dimensionality. Here we present a new computational method for finding feedback optimal control and solving HJB equations which is able to mitigate the curse of dimensionality. We do not discretize the HJB equation directly, instead we introduce a sparse grid in the state space and use the Pontryagin's maximum principle to derive a set of necessary conditions in the form of a boundary value problem, also known as the characteristic equations, for each grid point. Using this approach, the method is spatially causality free, which enjoys the advantage of perfect parallelism on a sparse grid. Compared with dense grids, a sparse grid has a significantly reduced size which is feasible for systems with relatively high dimensions, such as the 6-D system shown in the examples. Once the solution obtained at each grid point, high-order accurate polynomial interpolation is used to approximate the feedback control at arbitrary points. We prove an upper bound for the approximation error and approximate it numerically. This sparse grid characteristics method is demonstrated with three examples of rigid body attitude control using momentum wheels.

✉ Wei Kang
   wkang@nps.edu

   Lucas C. Wilcox
   lwilcox@nps.edu

[1] Department of Applied Mathematics, Naval Postgraduate School, Monterey, CA, USA

🖄 Springer

1 Introduction

The optimal feedback control of nonlinear systems is a challenging problem. Using dynamic programming, the feedback optimal control law is constructed based on the solution of a partial differential equation (PDE) that is called the Hamilton–Jacobi–Bellman (HJB) equation. This theoretically elegant approach suffers some difficulties in computation due to the curse of dimensionality, a term that was coined by Richard E. Bellman when considering problems in dynamic optimization, which relates to the fact that the size of the discretized problem in solving HJB equations increases exponentially with the dimension. Finding an approximate solution to HJB-type of equations in a local neighborhood of a trajectory has been extensively studied, see for example Al'brekht [1], Cacace et al. [8], Kang et al. [19], Lukes [25], Navasca and Krener [26], Falcone and Ferretti [12], Kang and Wilcox [20], and more recently Assellaou et al. [2]. Some of the previously proposed methods can be applied to systems with high dimensions. However, finding semi-global solutions to HJB equations, i.e., solutions satisfying a required accuracy in a given domain, faces the curse of dimensionality.

We present a new computational approach to finding semi-global solutions of optimal feedback control and HJB equations for nonlinear systems. The method is designed to mitigate the curse of dimensionality. Our approach does not discretize the HJB equation directly, instead we introduce a sparse grid in the state space and use Pontryagin's maximum principle (PMP) to derive a set of necessary conditions in the form of a boundary value problem (BVP), also known as the characteristic equations, for each grid point independently. For details on PMP and its relationship with HJB equations, the reader is referred to Pontryagin [27], Barron and Jensen [4], and Bardi and Capuzzo-Dolcetta [3]. The curse of dimensionality is mitigated by using sparse grids. The idea of sparse grids is based on Smolyak's work [28]. The reader is also referred to Bungartz and Griebel [7], Garcke [14], and Zenger [33] for more details and different perspectives. Relative to a dense grid, the size of the sparse grid is significantly reduced. In the propose computational method, the solution at each grid point is found using a Lobatto IIIa method that solves a two-point BVP. Different from many existing algorithms of solving PDEs, this approach is not based on spatial causality. A significant advantage of this causality free method lies in its perfect parallelism, a desirable property for modern computational equipment with many-core clusters. Some results on the rate of convergence and the upper bound of approximation error are proved.

The method is exemplified by three numerical examples of rigid body attitude control using momentum wheels. In this case, the HJB equation is six dimensional. The second example in itself is interesting, in which the system has two pairs of momentum wheels. The system is uncontrollable, which makes optimal feedback control difficult. To the best of our knowledge, no solutions have been found for the HJB equation of this problem.

2 Problem formulation

An optimal feedback control law is a function

$$u^*(t, x)$$

that minimizes the cost functional

$$\int_t^T L(t, \boldsymbol{x}(s), \boldsymbol{u}(s)) \, ds + h(\boldsymbol{x}(T))$$

subjecting to the control system

$$\dot{\boldsymbol{x}}(s) = \boldsymbol{f}(s, \boldsymbol{x}(s), \boldsymbol{u}(s)) \qquad (t < s < T), \tag{1a}$$

$$\boldsymbol{x}(t) = x. \tag{1b}$$

In this formulation, $\boldsymbol{x} : (t, T) \to \mathbb{R}^n$ is the state, $\boldsymbol{u} : (t, T) \to \mathbb{R}^m$ is the control, $L : (t, T) \times \mathbb{R}^n \times A \to \mathbb{R}$ is running cost, $h : (t, T) \times \mathbb{R}^n \to \mathbb{R}$ is the final cost, $\boldsymbol{f} : (t, T) \times \mathbb{R}^n \times A \to \mathbb{R}^n$ is a bounded, Lipschitz continuous function, and $A \subset \mathbb{R}^m$ is a compact set. Here $t \in \mathbb{R}$ is the initial time and $x \in \mathbb{R}^n$ is the initial condition. For the simplicity of discussion, we assume that the final time, $T \in \mathbb{R}$, is fixed. Following the standard approach in optimal control, we define the Hamiltonian

$$H(t, x, \lambda, u) = L(t, x, u) + \lambda^{\mathsf{T}} f(t, x, u)$$

where $\lambda \in \mathbb{R}^n$ is the costate and $u \in A$. The function

$$u^*(t, x, \lambda) = \arg\min_u H(t, x, \lambda, u)$$

minimizes the Hamiltonian. The value function is defined as

$$V(t, x) = \inf_{\boldsymbol{u}} \int_t^T L(s, \boldsymbol{x}(s), \boldsymbol{u}(s)) \, ds + h(\boldsymbol{x}(T)) \qquad (0 \le t < T),$$

where $\boldsymbol{x}(s)$ satisfies the identity (1). Further V satisfies the HJB equation, a PDE, with a final time condition given as

$$V_t(t, x) + H^* \left(t, x, V_x^{\mathsf{T}}(t, x) \right) = 0 \qquad (x \in \mathbb{R}^n, 0 \le t < T), \tag{2a}$$

$$V(T, x) = h(x) \qquad (x \in \mathbb{R}^n), \tag{2b}$$

where $H^*(t, x, \lambda) = H(t, x, \lambda, u^*)$, $V_t = \frac{\partial V}{\partial t}$, and $V_x = \frac{\partial V}{\partial x}$. If Eq. (2) can be solved, the feedback control law is a function defined as

$$u^*(t, x) = u^* \left(t, x, V_x^{\mathsf{T}}(t, x) \right). \tag{3}$$

The design and control of engineering systems involve both on-line and off-line computations. In the method proposed here, the HJB equation is solved in the off-line design phase. Once the solution of the HJB equation is found on a sparse grid, the on-line computation for real-time feedback control is carried out using interpolation, a numerical process that is simple and reliable. This approach offsets the main computational load from on-line computation to an off-line design phase. In addition, off-line computations allow one to use more powerful computers than those onboard systems such as satellites or unmanned vehicles. Because of the causality-free property, which is explained later, using parallel computers significantly reduces the required computational time when solving a HJB equation. In this paper, we make the following assumption.

Assumption 1 The optimal control is uniquely determined by the PMP at each point in the state space.

If $H(t, x, \lambda, u)$ is a strictly convex function of (x, u) over an open convex set containing all the admissible values of (x, u) at fixed t and λ, Assumption 1 holds true. In general, Assumption 1 can be guaranteed based on necessary and sufficient conditions of optimal control, which has a vast literature of publications, including both classic and viscosity solutions [3,13]. For problems with a proved existence and uniqueness of solutions, finding the optimal feedback control for practical real-time applications is still a difficult problem. If a system has four or more state variables, finding a desecrate solution and implementing real-time interpolation for feedback control suffer the curse of dimensionality. The main contribution of this paper is that, for a well-defined problem of optimal control with a moderate dimension, it is possible to achieve optimal feedback control using interpolation on a discrete approximate solution.

3 The sparse grid characteristics method

In many numerical methods for HJB equations, which are typically solved backward in time, the discretization is based on spatial causality and the computation is explicit in time. The value of the solution function $V(t, x)$ at a grid point is computed at an earlier time using the known value of the function at neighboring grid points at a later time. This coupling usually comes from the discretization of the spatial derivatives. For HJB equations of high dimensions, in our examples the dimension $d = 6$, solving the equation using traditional algorithms based on dense grids is computationally challenging. For instance, in a six dimensional space, if $2^5 = 32$ grid points are used to approximate a single variable, which is quite small, the total number of grid points for a 6-D problem is over 10^9. If 100 points are used for a single variable, then the size of the dense grid is 10^{12}. The curse of dimensionality is a bottleneck problem in solving HJB equations for practical applications.

To mitigate the curse of dimensionality, we introduce a causality free computational method. It consists of two components: (1) A solver that can find the optimal control and the value of $V(t, x)$ at any grid point; the computation is independent of the value of $V(t, x)$ at other points in the state space, i.e., causality free. (2) A set of grid points, such as a sparse grid, with a reduced size to make the problem tractable. The causality

free method introduced in this section is based on BVP solvers and sparse grids. The goal is to solve the problem of optimal control and its associated HJB equation with a moderate dimension. In this paper, we use a 6-D example to illustrate the algorithm.

Why causality free? From a conventional viewpoint of computation, solving a two-point BVP is not an efficient approach for PDEs. However, the causality free method is perfectly parallel. In fact, each grid point can be assigned a CPU core. The computation at a grid point requires no communication with the computation process at other grid points. Although not preferred in a serial computational environment, causality free algorithms can easily be implemented in massively parallel computers. In addition, causality free algorithms are ideal for sparse grids in which the space between adjacent grid points varies significantly. The combination of sparse grids, BVP solvers, and parallel computation makes it possible to mitigate the curse of dimensionality effectively for problems in which d is not too large.

3.1 A causality-free method using the necessary condition of optimal control

In contrast to traditional PDE discretizations, our proposed technique does not discretize the HJB equations directly but instead uses the PMP to derive a set of necessary conditions in the form of a BVP for each grid point, also known as the method of characteristics. As a result, the computation of the solution at an initial point in space is independent of other points.

The optimal trajectory for the optimal feedback control law described in Sect. 2 is a solution of the two-point BVP

$$\dot{x}(s) = \left(\frac{\partial H}{\partial \lambda}(s, x(s), \lambda(s), u^*(s, x(s), \lambda(s))) \right)^{\mathsf{T}}, \tag{4a}$$

$$\dot{\lambda}(s) = -\left(\frac{\partial H}{\partial x}(s, x(s), \lambda(s), u^*(s, x(s), \lambda(s))) \right)^{\mathsf{T}}, \tag{4b}$$

$$\dot{z}(s) = L(s, x(s), u^*(s, x(s), \lambda(s))), \tag{4c}$$

where $t \leq s \leq T$ with the boundary conditions

$$x(t) = x, \tag{4d}$$

$$\lambda(T) = \left(\frac{dh}{dx}(x(T)) \right)^{\mathsf{T}}, \tag{4e}$$

$$z(t) = 0. \tag{4f}$$

The optimal control and the minimum costs are

$$u^*(t, x) = u^*(t, x, \lambda(t)), \qquad V(t, x) = z(T) + h(x(T)). \tag{5}$$

Given any grid point, x, we can solve the BVP (4) and use the identities (5) to find the optimal control and the corresponding minimum cost without using the value of $V(t, x)$ in any nearby points, i.e., the computation is causality free. Numerical

algorithms for solving BVPs similar to (4) have been studied by many authors. In the examples, we adopt an algorithm based on a four-point Lobatto IIIa formula for our BVP solver. This is a collocation formula and the collocation polynomial provides a solution that is fifth-order accurate (see Kierzenka and Shampine [21] for more detail). We would like to point out that the computation is not limited to the Lobatto IIIa BVP solver. The problem of optimal control at each grid point can be solved using any computational method.

3.2 Sparse grids

In the approximation of multivariable functions, sparse grid interpolation is an approach in which the approximation is build on a subset of a dense grid with a significantly reduced size. Sparse grids are derived from the Smolyak's construction [28]. Although the original idea was invented more than fifty years ago, some recent work reveals potentials of its applications [5–7,20,23,32]. It is a known fact that the size of sparse grids increases with the dimension, d, on the order of

$$O\left(N(\log N)^{d-1}\right),$$

which is in sharp contrast to the size of the corresponding dense grid

$$O(N^d).$$

Here $N = 2^{q-d}$ where q is a measurement of the level of refinement of the sparse grid. Obviously, the significantly reduced number of grid points has its impact on the accuracy. For example, an upper bound of interpolation error using a classic sparse grid satisfies

$$\|e\|_{L^2} = O\left(N^{-2}(\log N)^{d-1}\right)$$

for all functions with bounded mixed derivatives up to the second order. Compared with the error bound using a dense grid, which is $O(N^{-2})$, we pay a small price in terms of accuracy for problems with a moderate dimension and we achieve a significantly reduced size of the grid.

Sparse grids have a hierarchical structure. For each variable, the set of grid points contains several layers of subsets, denoted by X^i. Let N_i be the number of points in X^i. These subsets are assumed to have a telescope structure, $X^{i-1} \subset X^i$ for $i > 1$. For illustration purposes, we exemplify the definition of the classic sparse grid using equally spaced nodes in $[0, 1]^d$ as

$$N_i = 2^{i-1} + 1, \quad X^i = \left\{\frac{k-1}{2^{i-1}} \,\middle|\, k = 1, 2, \ldots, N_i\right\}, \tag{6}$$

for $i \geq 1$.

The set of points in X^i but not in X^{i-1} is denoted by ΔX^i; and the number of points in the set is $_\Delta N_i$, i.e.,

$$
\begin{aligned}
\Delta X^1 &= X^1, \\
\Delta X^i &= X^i - X^{i-1} \quad (i \geq 2), \\
_\Delta N_i &= \left| \Delta X^i \right| \quad (i \geq 1).
\end{aligned}
$$

In this paper, the points in X^i are represented by x_j^i, $1 \leq j \leq N_i$, and the points in ΔX^i are represented by $_\Delta x_j^i$, $1 \leq j \leq {}_\Delta N_i$, i.e.,

$$
\begin{aligned}
X^i &= \left\{ x_1^i, x_2^i, \ldots, x_{N_i}^i \right\}, \\
\Delta X^i &= \left\{ _\Delta x_1^i, {}_\Delta x_2^i, \ldots, {}_\Delta x_{_\Delta N_i}^i \right\},
\end{aligned}
$$

for $i \geq 1$. In the sequel, we adopt the multi-index notations

$$
\begin{aligned}
\mathbf{i} &= \begin{bmatrix} i_1 & i_2 & \cdots & i_d \end{bmatrix}, \\
|\mathbf{i}| &= i_1 + i_2 + \cdots + i_d, \\
x_{\mathbf{j}}^{\mathbf{i}} &= (x_{j_1}^{i_1}, x_{j_2}^{i_2}, \ldots, x_{j_d}^{i_d}), \\
\Delta X^{\mathbf{i}} &= \Delta X^{i_1} \times \Delta X^{i_2} \times \cdots \times \Delta X^{i_d}.
\end{aligned}
$$

The dense grid build on X^q for an integer $q > 0$, denoted as G_{dense}^q, is

$$
G_{\text{dense}}^q = X^q \times \cdots \times X^q = \bigcup_{1 \leq i_1, \ldots, i_d \leq q} \Delta X^{\mathbf{i}}.
$$

Following Smolyak's approximation algorithm [5,28], the sparse grid, denoted by G_{sparse}^q, is defined as

$$
G_{\text{sparse}}^q = \bigcup_{|\mathbf{i}| \leq q} \Delta X^{\mathbf{i}}.
$$

Two plots of 2-D sparse grids are shown in Fig. 1 for $q = 6$ and 8. If $q = 8$, G_{sparse}^q has a total of 385 grid points whereas the corresponding dense grid, G_{dense}^q, has $(2^6 + 1)^2 = 4225$ points. The difference becomes increasingly significant for higher dimensions.

Sparse grids can be build using other nested sequences of grids X^i, $i \geq 1$. For instance, a modified sparse grid is defined using

$$
\begin{cases}
N_1 = 1, & X^1 = \left\{ \dfrac{1}{2} \right\}; \\[2ex]
N_i = 2^{i-1} + 1, & X^i = \left\{ \dfrac{k-1}{2^{i-1}} \;\middle|\; k = 1, 2, \ldots, N_i \right\};
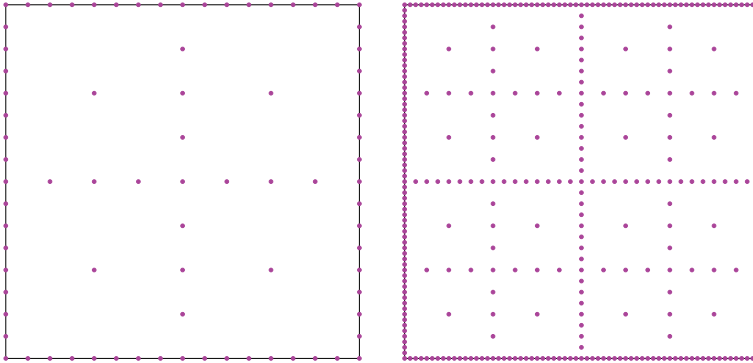\end{cases}
\tag{7}
$$

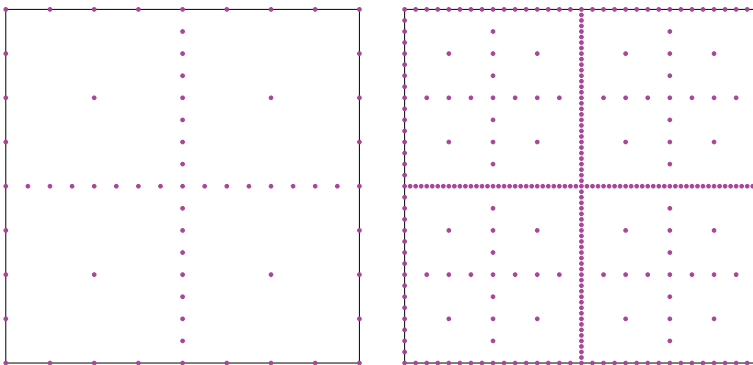**Fig. 1** The classic sparse grid in $[0, 1]^2$, $q = 6$ and $q = 8$

Fig. 2 Modified sparse grids in $[0, 1]^2$, $q = 6$ and $q = 8$

for $i > 1$. Except for $i = 1$, it is identical to the classic sparse grid (6). The size of G_{sparse} is further reduced. The modified sparse grids for $q = 6$ and $q = 8$ are shown in Fig. 2. When $q = 8$, the total number of points is 321.

The Chebyshev Gauss–Lobatto (CGL) sparse grid is defined in a similar way. The the grid points are defined as

$$
\begin{cases}
N_1 = 1, & X^1 = \left\{ \dfrac{1}{2} \right\}; \\
N_i = 2^{i-1} + 1, & X^i = \left\{ \dfrac{1}{2} \left(1 - \cos \dfrac{(k-1)\pi}{2^{i-1}} \right) \,\middle|\, k = 1, 2, \ldots, N_i \right\};
\end{cases}
\tag{8}
$$

for $i > 1$. Two examples of this grid are shown in Fig. 3.

3.3 Interpolation on sparse grids

Given a problem of optimal control, its solution on a sparse grid can be computed off-line. Then in real-time feedback control, the value of minimum cost, $V(t, x)$, and the
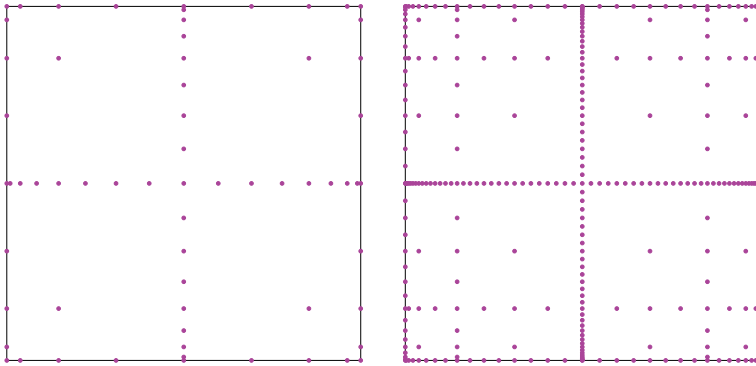
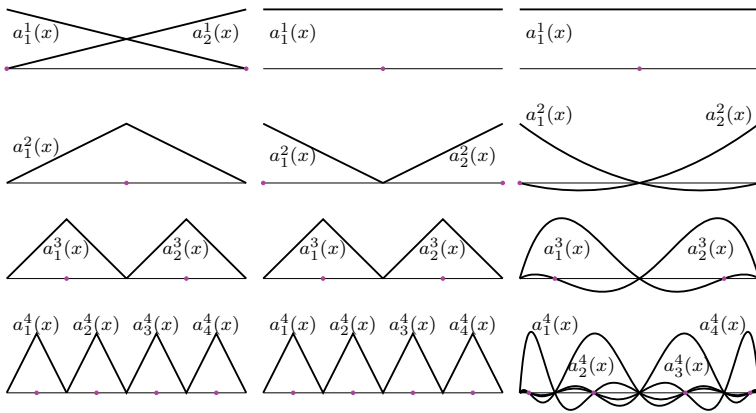**Fig. 3** CGL sparse grid in $[0, 1]^2$, $q = 6$ and $q = 8$

Fig. 4 Basis functions for the classic, modified, and CGL sparse grids

costate, $\lambda(t)$, can be computed using interpolation on the sparse grid. For simplicity, this section will focus on interpolating $V(t, x)$. Consider $X^i \subseteq [0, 1]$, $i \geq 1$. A basis function, $a_{\tilde{x}}^i(x)$, for a point $\tilde{x} \in X^i$ is defined on $[0, 1]$ satisfying

$$a_{\tilde{x}}^i(x) = \begin{cases} 1, & x = \tilde{x}; \\ 0, & x \in X^i, x \neq \tilde{x}. \end{cases}$$

For $_{\Delta}x_j^i \in \Delta X^i \subseteq X^j$, we define a simplified notation

$$a_j^i(x) = a_{\Delta x_j^i}^i(x).$$

Figure 4 shows a few basis functions for the various sparse grids.

The classic sparse grid uses piecewise linear basis functions

$$a_1^1(x) = x, \tag{9a}$$

$$a_2^1(x) = 1 - x, \tag{9b}$$

$$a_j^i(x) = \begin{cases} 1 - |2^{i-1}x - (2j-1)|, & x \in \left[\dfrac{2(j-1)}{2^{i-1}}, \dfrac{2j}{2^{i-1}}\right]; \\ 0, & \text{otherwise}; \end{cases} \tag{9c}$$

for $i \geq 2$ and $1 \leq j \leq 2^{i-2}$. For the modified sparse grid, $a_j^i(x)$ is the same as in identity (9c) if $i \geq 3$ and otherwise

$$a_1^1(x) = 1, \tag{10a}$$

$$a_1^2(x) = \begin{cases} 1 - 2x, & x \in \left[0, \dfrac{1}{2}\right]; \\ 0, & x \in \left(\dfrac{1}{2}, 1\right]; \end{cases} \tag{10b}$$

$$a_2^2(x) = \begin{cases} 0, & x \in \left[0, \dfrac{1}{2}\right]; \\ 2x - 1, & x \in \left(\dfrac{1}{2}, 1\right]. \end{cases} \tag{10c}$$

For the CGL sparse grid, the Lagrange polynomials form the basis functions

$$a_j^i(x) = \prod_{\substack{x^i \in X^i \\ x^i \neq x_j^i}} \frac{x - x^i}{x_j^i - x^i},$$

for $i > 1$ and $1 \leq j \leq 2^{i-2}$.

In the following, the inequality $\mathbf{j} \leq {}_\Delta N_{\mathbf{i}}$ implies

$$j_1 \leq {}_\Delta N_{i_1}, \quad j_2 \leq {}_\Delta N_{i_2}, \quad \ldots, \quad j_d \leq {}_\Delta N_{i_d}.$$

The inequality $\mathbf{k} \leq N_{\mathbf{i}}$ is similarly defined. The interpolation on a sparse grid does not need every basis function. In fact, for each $i > 1$, an interpolation function uses only those $a_{\tilde{x}}^i$ for which $\tilde{x} \in \Delta X^i$. Let $\mathcal{I}^q(f)$ be the interpolation of f at grid points of G_{sparse}^q. It is defined recursively on $[0, 1]^d$ as

$$\mathcal{I}^{d-1}(f) = 0, \tag{11a}$$

$$\mathcal{I}^q(f) = \mathcal{I}^{q-1} + \Delta\mathcal{I}^q(f), q \geq d, \tag{11b}$$

$$\Delta\mathcal{I}^q(f) = \sum_{|\mathbf{i}|=q} \sum_{1 \leq \mathbf{j} \leq {}_\Delta N_{\mathbf{i}}} w_{\mathbf{j}}^{\mathbf{i}} a_{j_1}^{i_1} \otimes \cdots \otimes a_{j_d}^{i_d}, q \geq d, \tag{11c}$$

$$w_{\mathbf{j}}^{\mathbf{i}} = f(x_{\mathbf{j}}^{\mathbf{i}}) - \mathcal{I}^{q-1}(f)(x_{\mathbf{j}}^{\mathbf{i}}), \tag{11d}$$

where

$$a_{j_1}^{i_1} \otimes \cdots \otimes a_{j_d}^{i_d}(x_1, \ldots, x_d) = a_{j_1}^{i_1}(x_1) \cdots a_{j_d}^{i_d}(x_d).$$

The weights, $w_{\mathbf{j}}^{\mathbf{i}}$, are scalars called *hierarchical surpluses*.

An alternative formulation uses basis functions $a_{\tilde{x}}^i(x)$ for $\tilde{x} \in X^i$. The corresponding basis functions are denoted by

$$u_k^i(x) = a_{x_k^i}^i(x),$$

for $x_k^i \in X^i$. Note that the subindex in this notation, k, represents the index in X^i, not in ΔX^i. Again, we use piecewise linear basis functions for the classic and modified sparse grids as in the definitions (9) and (10). Similarly, we use Lagrange polynomials as the basis functions for the CGL sparse grid. For the X^i defined in the Eqs. (6)–(8), we have the relation

$$a_j^i(x) = u_{2j-1}^i(x),$$

for $1 \leq j \leq {}_\Delta N_{i_d}$. The interpolations in [0, 1] and $[0, 1]^d$ are

$$\mathcal{U}^i(f) = \sum_{k=1}^{N_i} f(x_k^i) u_k^i,$$

$$\mathcal{U}^{i_1} \otimes \mathcal{U}^{i_2} \otimes \cdots \otimes \mathcal{U}^{i_d}(f) = \sum_{1 \leq \mathbf{k} \leq N_{\mathbf{i}}} f(x_{\mathbf{k}}^{\mathbf{i}}) u_{k_1}^{i_1} \otimes u_{k_2}^{i_2} \otimes \cdots \otimes u_{k_d}^{i_d}.$$

Let $\mathcal{I}^q(f)$ be the interpolation on sparse grids. Following the formulation in Barthelmann et al. [5], Delvos [10], and Wasilkowski and Wozniakowski [31], the interpolation on the sparse grid, $\mathcal{I}^q(f)$, in terms of the new basis is

$$\mathcal{I}^q(f) = \sum_{q-d+1 \leq |\mathbf{i}| \leq q} (-1)^{q-|\mathbf{i}|} \binom{d-1}{q-|\mathbf{i}|} \mathcal{U}^{i_1} \otimes \mathcal{U}^{i_2} \otimes \cdots \otimes \mathcal{U}^{i_d}(f)$$

$$= \sum_{q-d+1 \leq |\mathbf{i}| \leq q} \sum_{1 \leq \mathbf{k} \leq N_{\mathbf{i}}} (-1)^{q-|\mathbf{i}|} \binom{d-1}{q-|\mathbf{i}|} f(x_{\mathbf{k}}^{\mathbf{i}}) u_{k_1}^{i_1} \otimes u_{k_2}^{i_2} \otimes \cdots \otimes u_{k_d}^{i_d}.$$

$$(12)$$

Before the next section on error analysis, the algorithm is summarized as follows. The off-line computation has three steps.

Step I Generating a sparse grid, G_{sparse}^q, and its basis functions $\left\{ a_j^i \right\}_{i=1,j=1}^{q-d+1, {}_\Delta N_i}$.

Step II Solve the two-point BVP defined in Eq. (4) at each grid point.

Step III Generating the hierarchical surpluses, $\{w_j^i\}$, using (11) or other equivalent formulae.

The output of the process consists of the values of $\lambda(t)$ and $V(t, x)$ at grid points in G_{sparse}^q. For on-line optimal feedback control, the value of these functions at an arbitrary x is approximated using interpolation on the sparse grid. Then the control input (5) is computed for real-time control and operation. For a feedback using model predictive control, like the examples in Sect. 5, the feedback requires the value of $\lambda(t)$

at $t = 0$ in each time interval. It is not necessary to solve the problem for $t \neq 0$. In the case of interpolation at $t \geq 0$, which is not addressed in this paper, the time variable can be included in the sparse grid as another dimension.

4 Error analysis

Let $V_{\mathbf{j}}^{\mathbf{i}}$ represent the value of $V(t, x)$ evaluated at $x = x_{\mathbf{j}}^{\mathbf{i}}$, i.e.,

$$V_{\mathbf{j}}^{\mathbf{i}} = V(t, x_{\mathbf{j}}^{\mathbf{i}}).$$

A causality free algorithm, such as the numerical solution of the characteristic Eqs. (4) and (5), approximates $V_{\mathbf{j}}^{\mathbf{i}}$ with an error

$$\bar{V}_{\mathbf{j}}^{\mathbf{i}} = V_{\mathbf{j}}^{\mathbf{i}} + \epsilon_{\mathbf{j}}^{\mathbf{i}}. \tag{13}$$

At an arbitrary point x, the approximation based on sparse grid interpolation is

$$\mathcal{I}^q(\bar{V})(x) = \mathcal{I}^q(V)(x) + e_{\text{BVP}},$$

where e_{BVP} is the error due to $\epsilon_{\mathbf{j}}^{\mathbf{i}}$, the numerical error of the solution of the BVP (4) and (5). More specifically,

$$e_{\text{BVP}} = \sum_{q-d+1 \leq |\mathbf{i}| \leq q} \sum_{1 \leq \mathbf{k} \leq N_{\mathbf{i}}} (-1)^{q-|\mathbf{i}|} \binom{d-1}{q-|\mathbf{i}|} \epsilon_{\mathbf{k}}^{\mathbf{i}} u_{k_1}^{i_1} \otimes u_{k_2}^{i_2} \otimes \cdots \otimes u_{k_d}^{i_d}. \tag{14}$$

Relative to the true value, the interpolation process has an error

$$e_{\text{interp}} = \mathcal{I}^q(V)(x) - V(t, x).$$

Therefore,

$$\mathcal{I}^q(\bar{V})(x) = V(t, x) + e_{\text{interp}} + e_{\text{BVP}}. \tag{15}$$

4.1 Error upper bounds

For a survey of sparse grids and error estimation, the reader is referred to Bungartz and Griebel [7] and Garcke [14]. In the case of a classic sparse grid, applying a piecewise linear interpolation to functions with continuous second order partial derivatives, e_{interp} is at the order of

$$\|e_{\text{interp}}\| = O\left(\frac{(\log N)^{d-1}}{N^2}\right) \tag{16}$$

which holds both for the L^2- and L^∞-norm.

The estimate in identity (16) holds for both $\|e_{\text{interp}}\|_{L^2}$ and $\|e_{\text{interp}}\|_{L^\infty}$. In the case of a CGL sparse grid, an error upper bound of e_{interp} is proved by Barthelmann et

al. [5]. Suppose a function f has kth order continuous partial derivatives. Define the norm

$$\|f\|_{W^{k,\infty}} = \max\left\{\left\|\frac{\partial^{|\mathbf{i}|}}{\partial x_1^{i_1}\cdots\partial x_d^{i_d}}f\right\|_{L^\infty} \Bigg| 1 \le i_1, \ldots, i_d \le k\right\},$$

then the interpolation of f on a CGL sparse grid satisfies

$$\|e_{\text{interp}}\|_{W^{k,\infty}} = O\left(\frac{(\log M)^{(k+2)(d-1)+1}}{M^k}\right),$$

where M is the number of sparse grid points.

In this section, we prove an upper bound for e_{BVP}. Its value is small if the dimension, d, is one or two. However, the error becomes larger when solving PDEs with a higher dimension. In our examples of $d = 6$ and e_{BVP} is not negligible. Let's define

$$\Lambda_i = \max_x \sum_{k=1}^{N_i} \left|u_k^i(x)\right|$$

for $i \ge 1$. For polynomial interpolations, this number is the Lebesgue constant. For any integer $l \ge d$, we define

$$S_l = \sum_{|\mathbf{i}|=l} \Lambda_{i_1}\Lambda_{i_2}\cdots\Lambda_{i_d}.$$

Theorem 1 (i) *Suppose $\epsilon > 0$ is an upper bound of the numerical error at each grid point, i.e., $\left|\epsilon_{\mathbf{j}}^{\mathbf{i}}\right|$ in Eq. (13) are smaller than ϵ. Then*

$$\|e_{BVP}\|_{L^\infty} < \epsilon \sum_{l=q-d+1}^{q} \binom{d-1}{q-l} S_l \tag{17}$$

(ii) *Suppose $\tilde{\Lambda}_q > 0$ is a constant such that*

$$\Lambda_i \le \tilde{\Lambda}_q \quad (1 \le i \le q - d + 1).$$

Then

$$\frac{\|e_{BVP}\|_{L^\infty}}{\epsilon} = O\left((\log N)^{d-1}\tilde{\Lambda}_q^d\right) \tag{18}$$

where $N + 1 = 2^{q-d} + 1$ is the number of grid points in each dimension.

Proof   (i) It is easy to check that

$$\sum_{|\mathbf{i}|=l} \sum_{1\leq\mathbf{k}\leq N_{\mathbf{i}}} \left| u_{k_1}^{i_1} \otimes u_{k_2}^{i_2} \otimes \cdots \otimes u_{k_d}^{i_d} \right| = \sum_{|\mathbf{i}|=l} \sum_{k_1=1}^{N_1} \cdots \sum_{k_d=1}^{N_d} \left| u_{k_1}^{i_1} \otimes u_{k_2}^{i_2} \otimes \cdots \otimes u_{k_d}^{i_d} \right|$$

$$= \sum_{|\mathbf{i}|=l} \left(\sum_{k_1=1}^{N_1} \left| u_{k_1}^{i_1} \right| \right) \otimes \cdots \otimes \left(\sum_{k_d=1}^{N_d} \left| u_{k_d}^{i_d} \right| \right)$$

$$\leq \sum_{|\mathbf{i}|=l} \Lambda_{i_1} \Lambda_{i_2} \cdots \Lambda_{i_d}$$

for all $x \in [0, 1]^d$. Therefore, given any integer $l \geq d$ we have

$$\sum_{|\mathbf{i}|=l} \sum_{1\leq\mathbf{k}\leq N_{\mathbf{i}}} \left| u_{k_1}^{i_1} \otimes u_{k_2}^{i_2} \otimes \cdots \otimes u_{k_d}^{i_d}(x) \right| \leq S_l. \tag{19}$$

Then the upper bound (17) is a corollary of Eqs. (12), (14), and (19).

(ii) To find the order of $\|e_{\mathrm{BVP}}\|/\epsilon$, we first note

$$S_l = \sum_{|\mathbf{i}|=l} \Lambda_{i_1} \Lambda_{i_2} \cdots \Lambda_{i_d}$$

$$\leq \tilde{\Lambda}_q^d \sum_{|\mathbf{i}|=l} 1$$

$$= \binom{l-1}{l-d} \tilde{\Lambda}_q^d$$

$$= \frac{(l-1)(l-2)\cdots(l-d)\cdots d}{(l-d)!} \tilde{\Lambda}_q^d$$

$$\leq (l-1)^{d-1} \tilde{\Lambda}_q^d$$

$$\leq (q-1)^{d-1} \tilde{\Lambda}_q^d.$$

From identity (17),

$$\frac{\|e_{\mathrm{BVP}}\|_{L^\infty}}{\epsilon} < \sum_{l=q-d+1}^{q} \binom{d-1}{q-l} S_l$$

$$\leq (q-1)^{d-1} \tilde{\Lambda}_q^d \sum_{l=q-d+1}^{q} \binom{d-1}{q-l}$$

$$= 2^{d-1}(q-1)^{d-1} \tilde{\Lambda}_q^d.$$

Then the relation (18) follows from the fact

$$q = d + \log_2 N = O(\log N).$$

$\square$

Corollary 1 *For the classic or the modified sparse grid and piecewise linear interpolation,*

$$\frac{\|e_{BVP}\|_{L^\infty}}{\epsilon} = O\left((\log N)^{d-1}\right). \tag{20}$$

For the CGL sparse grid and polynomial interpolation,

$$\frac{\|e_{BVP}\|_{L^\infty}}{\epsilon} = O\left((\log N)^{2d-1}\right). \tag{21}$$

Proof The basis functions of the piecewise linear interpolation have the following property

$$\tilde{\Lambda}_q = 1.$$

The Lebesgue constant for the CGL grid points is bounded by (see for example Hesthaven et al. [16])

$$\tilde{\Lambda}_q \le \frac{2}{\pi} \log N + \frac{2}{\pi}\left(\gamma + \log \frac{4}{\pi}\right) + \frac{2}{\pi} \log 2 = O\left(\log N\right) \tag{22}$$

where γ is Euler's constant. Substitute $\tilde{\Lambda}_q$ into Eq. (18) to yield Eqs. (20) and (21). □

4.2 Numerically estimate $e_{\mathbf{BVP}}$

Some numerical examples show that the upper bound in inequality (17) tends to be conservative. Using relations (17) and (22), we can find an error upper bound for the CGL sparse grid G_{sparse}^q. For example, if $q = 13$ we have

$$\frac{\|e_{\text{BVP}}\|_{L^\infty}}{\epsilon} < \sum_{l=q-d+1}^{q} \binom{d-1}{q-l} S_l \le 3.66 \times 10^4. \tag{23}$$

In this estimation, $e_{\text{BVP}} = \epsilon_{\mathbf{j}}^{\mathbf{i}}$ at each grid point is assumed to be the maximum value, ϵ. It is a conservative assumption. As an alternative, we can assume $\epsilon_{\mathbf{j}}^{\mathbf{i}}$ is a random variable. (This approach is commonly used in uncertainty quantification, rightly or wrongly, to get a handle on the model error.) We will do this here to get another estimate of e_{BVP}.

From identity (14), e_{BVP} has two special properties. Firstly, the value of e_{BVP} is not directly dependent on $f(x)$. The error is solely based on the type of grid, the interpolation basis functions, and the distribution of $\epsilon_{\mathbf{j}}^{\mathbf{i}}$. Therefore, the estimation of e_{BVP} can be done off-line for a given grid. The result is then applicable to a family of problems. Secondly, e_{BVP} is a linear function of $\epsilon_{\mathbf{j}}^{\mathbf{i}}$. The estimate of e_{BVP} for a given distribution of $\epsilon_{\mathbf{j}}^{\mathbf{i}}$ is applicable to a different distribution of the same type through a simple rescaling.

Given a sample set of $\epsilon_{\mathbf{j}}^{\mathbf{i}}$, e_{BVP} can be computed using Eq. (14) at any x in $[0, 1]^d$. Finding the probability distribution of $\epsilon_{\mathbf{j}}^{\mathbf{i}}$ is not a problem to be addressed in this paper. As an example, we assume that $\epsilon_{\mathbf{j}}^{\mathbf{i}}$ are independent random variables with uniform distribution in $[-\epsilon, \epsilon]$. Let $\left\{\bar{\epsilon}_{\mathbf{j}}^{\mathbf{i}}\right\}$ be a sample data set with a uniform distribution in $[0, 1]$. After rescaling, Eq. (14) implies

$$\frac{e_{\mathrm{BVP}}}{\epsilon} = \sum_{q-d+1 \le |\mathbf{i}| \le q} \sum_{1 \le \mathbf{k} \le N_{\mathbf{i}}} (-1)^{q-|\mathbf{i}|} \binom{d-1}{q-|\mathbf{i}|} \bar{\epsilon}_{\mathbf{k}}^{\mathbf{i}} u_{k_1}^{i_1} \otimes u_{k_2}^{i_2} \otimes \cdots \otimes u_{k_d}^{i_d}.$$

In an numerical example, a set of $\left|G_{\mathrm{sparse}}^q\right| = 44{,}689$ random numbers are generated as the sample value, $\bar{\epsilon}_{\mathbf{j}}^{\mathbf{i}}$, in $[0, 1]$. At $N = 2000$ random points in $[0, 1]^d$, we found

$$\frac{e_{\mathrm{BVP}}}{\epsilon} \le 66.60. \tag{24}$$

The bound in (24) is much smaller than the upper bound in the inequality (23) derived from Theorem 1. We would like to emphasize that this practical way of estimating e_{BVP} is independent of the function to be approximated. The overall error $e_{\mathrm{interp}} + e_{\mathrm{BVP}}$ can be approximated in a similar way based on the assumption that ϵ, the error upper bound of BVP solver, is known. This point is discussed in the next section and exemplified in Example I. A thorough numerical analysis of errors is outside the scope of this paper. By no means can the examples in this paper lead to a general conclusion about the error upper bound. However, the examples in this paper present a practical approach of using Monte Carlo simulations to analyze e_{BVP} and $e_{\mathrm{interp}} + e_{\mathrm{BVP}}$.

4.3 Numerically estimate $e_{\mathrm{interp}} + e_{\mathrm{BVP}}$

What we ultimately care about is the total error in an approximate of $V(t, x)$, which is $e_{\mathrm{interp}} + e_{\mathrm{BVP}}$ in Eq. (15). For causality free algorithms, such as solving the BVP (4) and (5), numerical errors do not propagate in space, i.e., the value of $V(t, x)$ can be computed independently from the approximation error at other points. This special property of causality free algorithms has an important implication. A BVP solver with accurate error control can be used to approximate the error of a numerical solution of the PDE. There is a sizable literature of numerical BVP algorithms. Some approaches are able to control the error within a given tolerance. For the examples in this paper, we use a four-point Lobatto IIIa formula which can be implemented with a controlled true error [21]. Given a numerical solution on G_{sparse}^q,

$$\bar{V}_{\mathbf{j}}^{\mathbf{i}} \approx V(t, x_{\mathbf{j}}^{\mathbf{i}}).$$

The approximate of $V(t, x)$ at any point x is obtained by interpolation

$$V(t, x) \approx \mathcal{I}^q(\bar{V}).$$

Meanwhile, a BVP solver with error control can be applied to solve the Eqs. (4) and (5) at the same point. Suppose the solution is $\tilde{V}(t, x)$. The error tolerance is set so that its true error is much smaller than $e_{\text{interp}} + e_{\text{BVP}}$. To find the distribution of $e_{\text{interp}} + e_{\text{BVP}}$, a set of sample points is generated in the state space, organized or random. The numerical error at these points can be approximated using

$$\left| e_{\text{interp}} + e_{\text{BVP}} \right| = \left| \mathcal{I}^q(\bar{V})(x) - V(t, x) \right| \approx \left| \mathcal{I}^q(\bar{V})(x) - \tilde{V}(t, x) \right|.$$

Although the computation of $\tilde{V}(t, x)$ can be slow, the entire process is perfectly parallel because of its nature of being causality free. In the examples below, 128 CPU cores are used to approximate the error at 1280 random points. The error tolerance of $\tilde{V}(t, x)$ is set at 10^{-7} or 10^{-9}. The numerical results show that this tolerance is smaller than $\left| e_{\text{interp}} + e_{\text{BVP}} \right|$ by at least three orders of magnitude.

5 Examples

In this section, two examples of optimal attitude control are presented. The system model represents a rigid body controlled by momentum wheels. The first example is a system with three pairs of controllable momentum wheels. The second example is an uncontrollable system with two pairs of momentum wheels. The uncontrollable case in itself is interesting. The numerical result is fundamentally different from existing controllers in the literature.

Consider a rigid body system. Let $\{e_1, e_2, e_3\}$ be an inertial frame of orthonormal vectors and let $\{e'_1, e'_2, e'_3\}$ be a body-fixed frame, or body frame. In this paper, the attitude of a satellite is represented by Euler angles (see Diebel [11] for a good introduction to representing attitude)

$$v = \begin{bmatrix} \phi \ \theta \ \psi \end{bmatrix}^{\mathsf{T}}$$

in which ϕ, θ, and ψ are the angles of rotation around e'_1, e'_2, and e'_3, respectively, in the order of (3, 2, 1). The angular velocity is a vector in the body frame,

$$\omega = \begin{bmatrix} \omega_1 \ \omega_2 \ \omega_3 \end{bmatrix}^{\mathsf{T}}.$$

The control system using momentum wheels is defined by a set of differential equations [9]

$$\dot{v} = E(v)\omega, \tag{25a}$$

$$J\dot{\omega} = S(\omega)R(v)H + Bu, \tag{25b}$$

where $B \in \mathbb{R}^{3 \times m}$ is a constant matrix in which m is the number of control variables, $u \in \mathbb{R}^m$ is the control torque, $J \in \mathbb{R}^{3 \times 3}$ is a combination of inertia matrices of the rigid body without wheels and the momentum wheels, $H \in \mathbb{R}^3$ is the total and constant angular momentum of the system, and $E(v)$, $S(\omega)$, $R(v) \in \mathbb{R}^3$ are matrices. Details can be found in Kang and Wilcox [20].

5.1 Example I

The system has three pairs of control momentum wheels, $m = 3$. In the model (25), the following parameter values are used

$$B = \begin{bmatrix} 1 & \frac{1}{20} & \frac{1}{10} \\ \frac{1}{15} & 1 & \frac{1}{10} \\ \frac{1}{10} & \frac{1}{15} & 1 \end{bmatrix}, \qquad J = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 4 \end{bmatrix}, \qquad H = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}.$$

The optimal control is

$$\arg \min_u \int_t^T L(v, \omega, u) \, ds + W_4 \|v(T)\|^2 + W_5 \|\omega(T)\|^2$$

where

$$L(v, \omega, u) = \frac{W_1}{2} \|v\|^2 + \frac{W_2}{2} \|\omega\|^2 + \frac{W_3}{2} \|u\|^2,$$
$$W_1 = 1, \quad W_2 = 1, \quad W_3 = 1/2, \quad W_4 = 1, \quad W_5 = 1, \quad t = 0, \quad T = 20.$$

Since L is a convex function it can be proved that a unique solution exists in a neighborhood of the target state. The solution $V(t, v, \omega)$ is computed at $t = 0$ for initial states $v(0)$ and $\omega(0)$ in two domains, D_1 and D_2, of different size,

$$D_1 = \left\{ v, \omega \in \mathbb{R}^3 \,\middle|\, -\frac{\pi}{6} \leq \phi, \theta, \psi \leq \frac{\pi}{6} \text{ and } -\frac{\pi}{8} \leq \omega_1, \omega_2, \omega_3 \leq \frac{\pi}{8} \right\},$$
$$D_2 = \left\{ v, \omega \in \mathbb{R}^3 \,\middle|\, -\frac{\pi}{3} \leq \phi, \theta, \psi \leq \frac{\pi}{3} \text{ and } -\frac{\pi}{4} \leq \omega_1, \omega_2, \omega_3 \leq \frac{\pi}{4} \right\}.$$

The computation is based on the CGL sparse grid with $q = 13$. The number of grid points for each dimension is $2^{q-6} + 1 = 129$. The total number of grid points in the 6-D domain is

$$\left| G_{\text{sparse}}^q \right| = 44{,}689,$$

which is small in comparison with the size of a dense grid,

$$\left| G_{\text{dense}}^q \right| = 129^6 > 4.6 \times 10^{12}.$$

In the computation, the two-pont BVP (4) is solved at each grid point in G_{sparse}^q using a four-stage Lobatto IIIa method [21]. The error tolerance is 10^{-12}. The computation is carried out in Hamming, a parallel computer of Naval Postgraduate School. Although as many as $\left| G_{\text{sparse}}^q \right|$ can be used, we limit the computation to 512 CPU cores. To check the accuracy of the overall solution, the upper bound of $e_{\text{interp}} + e_{\text{BVP}}$ is numerically approximated using the method in Sect. 4.3. More specifically, 1280 points are randomly generated in D_1 and D_2. At each sample point, the value of $\bar{V}(0, v, \omega)$ is computed using interpolation. The true value at the same point is approximated

Table 1 Summary of results for Example I

Domain	q	$\left|G^q_{\text{dense}}\right|$	$\left|G^q_{\text{sparse}}\right|$	MAE	Relative MAE
D_1	13	$>10^{12}$	44,698	4.9×10^{-7}	4.0×10^{-7}
D_2	13	$>10^{12}$	44,698	3.6×10^{-3}	7.3×10^{-4}
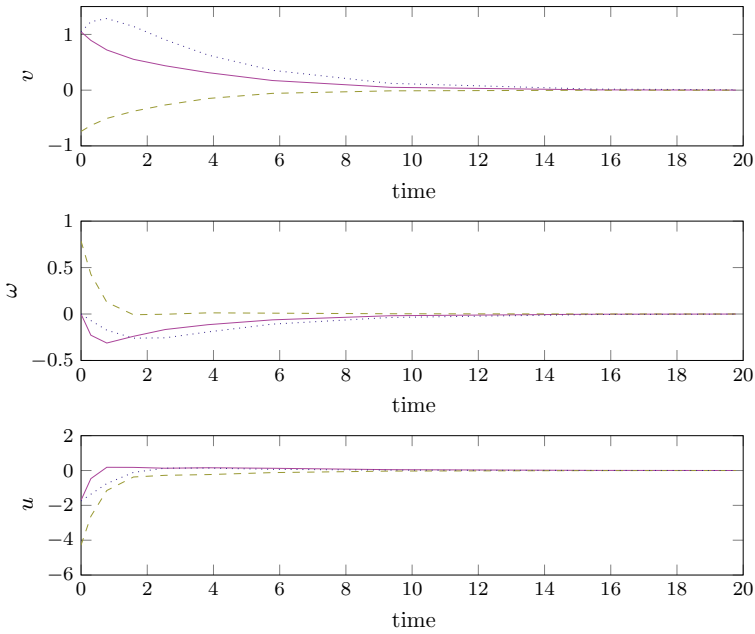
Fig. 5 Trajectories (*solid line ϕ, ω_1, u_1; dashed line θ, ω_2, u_2; dotted line ψ, ω_3, u_3*)

by $\tilde{V}(0, v, \omega)$, which is computed by applying the BVP solver to the Eq. (4) with an error tolerance 10^{-9} in D_1 and 10^{-7} in D_2. This tolerance is much smaller than $e_{\text{interp}} + e_{\text{BVP}}$ so that the error of the BVP solver can be ignored. The difference, $\bar{V}(0, v, \omega) - \tilde{V}(0, v, \omega)$, is an approximate of $e_{\text{interp}} + e_{\text{BVP}}$ at the sample point. In D_1, the mean absolute error (MAE) is 4.9×10^{-7}. The MAE for the relative error is 4.0×10^{-7}. In the larger domain D_2, the MAE equals 3.6×10^{-3} and the relative error is 7.3×10^{-4}. The results are summarized in Table 1. Figure 5 shows a typical trajectory in which (v, ω) converges to zero, i.e., the rigid body is stabilized. The accuracy of the solution, especially in D_2, is not as high as PDE solvers in some 2-D or 3-D cases. This is not surprising because we sacrifice the accuracy in exchange for a sparse grid that is tractable in 6-D. For a fixed dimension d, the accuracy of the solution at an arbitrary point depends on q and the size of the domain. For a fixed q, the accuracy is increase when the size of the domain is decreased. Shown in Table 1, decreasing the linear dimension of D_2 by 50% increases the accuracy by three orders of magnitude.

5.2 Example II

In the following, we consider a rigid body controlled by two pairs of momentum wheels. Although satellite systems are quipped with at least three pairs of control momentum wheels, malfunction may occur in some wheels. It is proved by Crouch [9] that this system is uncontrollable. How to stabilize the satellite around a desired attitude is a challenging problem. Related work can be found in Gui et al. [15], Horri and Hodgart [17], Kim and Kim [22], Krishnan et al. [24], Terui et al. [29], Tsiotras and Luo [30] and references therein. Different from exiting results, we do no assume zero angular momentum. In addition, the controller based on the solution of the associated HJB equation is smooth. The optimal control is able to smoothly stabilize the rigid body at an attitude that is closest to the desired orientation. The result in this paper is different from those in Kang and Wilcox [20] where the stabilization does not guarantee an optimal attitude. In the present paper, we optimize a cost function that automatically stabilizes the system at an optimal attitude. In addition, the result is integrated with a model predictive control (MPC) to achieve feedback stabilization in the presence of noise.

In this section, the following values are assigned to the parameters

$$
B = \begin{bmatrix} 1 & \frac{1}{10} \\ 0 & 1 \\ \frac{1}{12} & 0 \end{bmatrix}, \qquad J = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 4 \end{bmatrix}, \qquad H = \begin{bmatrix} 12 \\ 12 \\ 6 \end{bmatrix}.
$$

The optimal control is

$$
\arg \min_{u} \int_{t}^{T} L(v, \omega, u) \, ds \tag{26}
$$

where

$$
L(v, \omega, u) = \frac{W_1}{2} \|v - v_e(v, \omega)\|^2 + \frac{W_2}{2} \|\omega\|^2 + \frac{W_3}{2} \|u\|^2,
$$
$$
W_1 = 1, \quad W_2 = 2, \quad W_3 = 0.5, \quad t_0 = 0, \quad T = 30.
$$

The function $v_e(v, \omega)$ represents the optimal attitude reachable from (v, ω). It is a known fact that this system is uncontrollable. The desired attitude, in our example $v = 0$, may not be reachable. In the case of $H = 0$, nonsmooth controllers can be derived to stabilize the system [15,17,22,24,29,30]. In the case of $H \neq 0$, a manifold of reachable states (v, ω) satisfies [9]

$$
C^{\mathsf{T}}(J\omega - R(v)H) = \text{constant},
$$
$$
\text{for some } C \in \mathbb{R}^3 \text{ such that } C^{\mathsf{T}}B = 0.
$$

The attitude $v_e(v, \omega)$ is a target attitude in this reachable manifold. A satellite system may have to meet multiple requirements of orientation, such as pointing sensors to the desired direction and at the same time keeping its solar panel facing the Sun. But the desired attitude, for instance $v = 0$, may not lie on the manifold of reachable states.
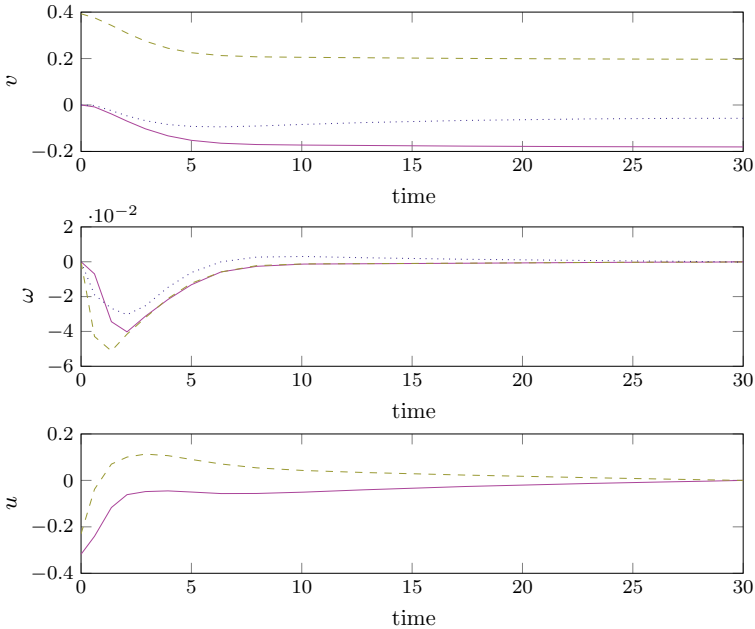
Fig. 6 Trajectories (*solid line* ϕ, ω_1, u_1; *dashed line* θ, ω_2, u_2; *dotted line* ψ, ω_3)

We define the following optimal $v_e(v, \omega)$ as the target state for stabilization. For this purpose, we minimize the Frobenius distance between $R(v)$ and $I = R(0)$. Because both matrices are orthogonal, it is equivalent to

$$v_e(v, \omega) = \arg \max_{\tilde{v}} \mathrm{tr}(R(\tilde{v})) \tag{27a}$$

$$\text{subject to } -C^\mathsf{T} R(\tilde{v}) H = C^\mathsf{T}(J\omega - R(v)H). \tag{27b}$$

It can be proved that $v_e(v(t), \omega(t))$, without noise and uncertainty, is a constant along any controlled trajectory. Therefore, it can be treated as a constant in the derivation of the BVP (4). The solution of the maximization problem (27) can be found by algorithms of numerical nonlinear programming.

The HJB equation is solved at $t = 0$ on a sparse grid in the domain

$$D = \left\{ v, \omega \in \mathbb{R}^3 \,\middle|\, -\frac{\pi}{6} \leq \phi, \theta, \psi \leq \frac{\pi}{6} \quad \text{and} \quad -\frac{\pi}{8} \leq \omega_1, \omega_2, \omega_3 \leq \frac{\pi}{8} \right\}.$$

The CGL sparse grid of $q = 13$ is used. Similar to the approach in Example I, we solve the HJB equation at the grid points using 512 CPU cores in parallel. Then the accuracy is checked at a random of 1280 point in D. The MAE of the relative error is 8.5×10^{-3}. A typical trajectory is shown in Fig. 6.

In an optimal control design based on HJB equations, the most computationally intensive part, i.e., solving the HJB equation, is done off-line. Once the equation is solved, the real-time closed-loop control can be computed using interpolation with a
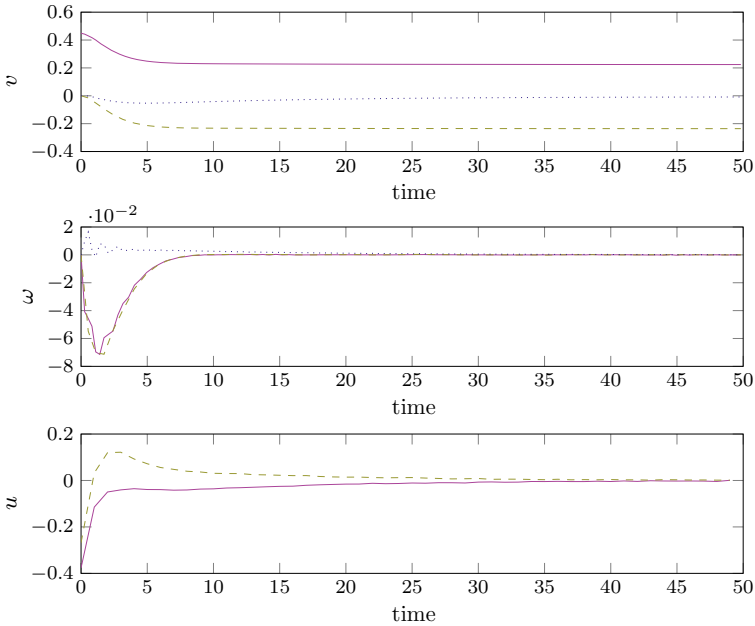
Fig. 7 Closed-loop trajectories (*solid line ϕ, ω_1, u_1; dashed line θ, ω_2, u_2; dotted line ψ, ω_3*)

minimum computational load. In the following simulations of closed-loop control, the solution of the HJB equation in Example II is integrated with a MPC to stabilize the system at a desired optimal attitude. We adopt a basic zero-order hold MPC controller. The sampling rate is 10 Hz. In each time interval the optimal control u^* is computed using interpolation on the sparse grid. It is assumed that the measurement of v and ω is noise-corrupted. The noise has a uniform distribution with a magnitude about 0.5% of the maximum state value. Several trajectories under the closed-loop control are shown in Figs. 7, 8, and 9.

5.3 Example III

In previous examples, the closed-loop control is based on a fixed horizon, $[0, T]$. The MPC algorithm uses the value of $u(0, x)$ in feedback. In the following, we exemplify the method of computing $u(t, x)$ for t in a given interval $[0, t_1]$, where $t_1 \leq T$. Because of the causality free property, it is straightforward to include the time variable as an additional dimension in the sparse grid. Consider the following optimal control problem

$$\dot{x}_1 = -x_1 + x_2,$$
$$\dot{x}_2 = -x_2 + \frac{x_3}{1 + x_1^2 + x_2^2},$$
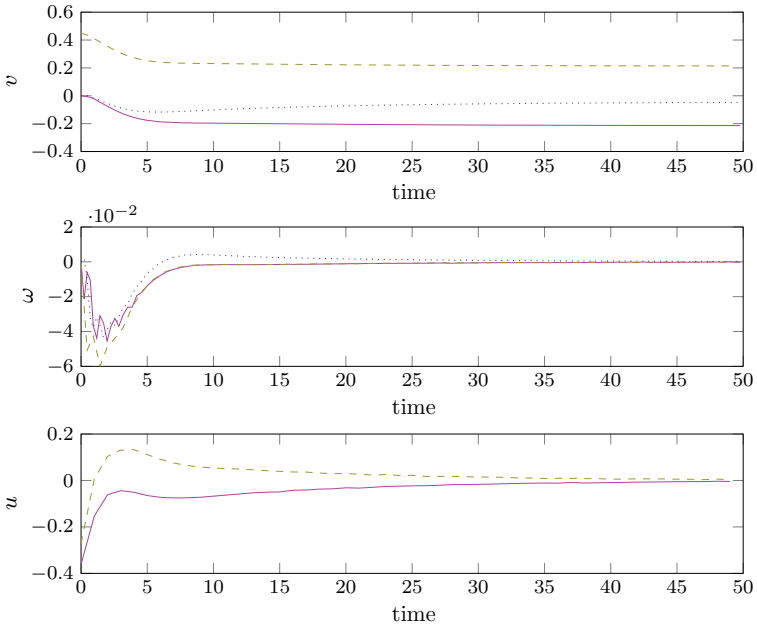
Fig. 8 Closed-loop trajectories (*solid line ϕ, ω_1, u_1; dashed line θ, ω_2, u_2; dotted line ψ, ω_3*)
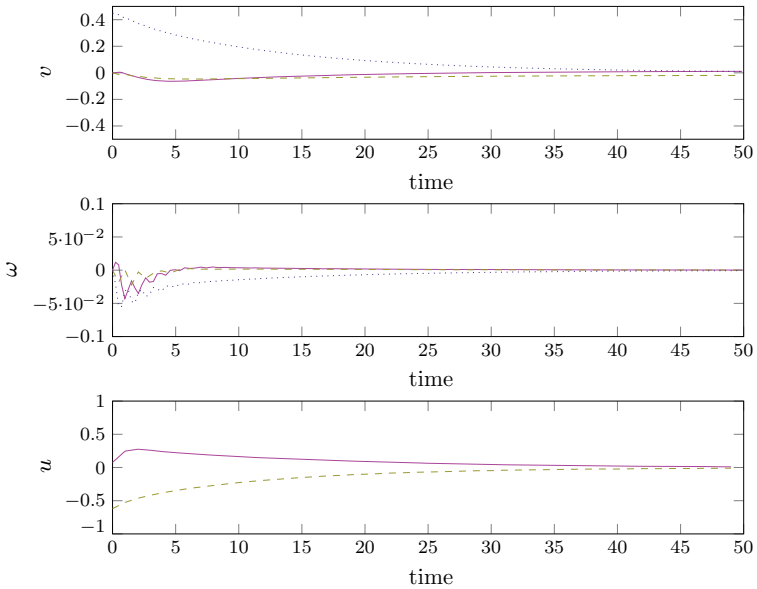
Fig. 9 Closed-loop trajectories (*solid line ϕ, ω_1, u_1; dashed line θ, ω_2, u_2; dotted line ψ, ω_3*)

$$\dot{x}_3 = \left(-2x_1^2 + 2x_1x_2 - 2x_2^2 + \frac{2x_2x_3}{1 + x_1^2 + x_2^2} \right) \frac{x_3}{1 + x_1^2 + x_2^2} + \left(1 + x_1^2 + x_2^2 \right) u,$$

where the cost functional is

$$\frac{1}{2} \int_{t_0}^{T} \frac{x_3^2}{\left(1 + x_1^2 + x_2^2 \right)^2} + u^2 \, dt.$$

For any initial condition, $x(t_0) = x_0$, an optimal control exists that achieves a minimum value, $V(t_0, x_0)$. Let $z = (t, x)$ represents a point in the time-state space. The sparse grid is generated in the following region in $\mathbb{R} \times \mathbb{R}^3$

$$D = \left\{ z = (t, x) \in \mathbb{R} \times \mathbb{R}^3 : t \in [0, 5] \quad \text{and} \quad x_i \in [-2, 2], \quad \text{for } i = 1, 2, 3 \right\}.$$

For each grid point $z_{\mathbf{j}}^{\mathbf{i}}$, the minimum value, V, and the associated costate, λ, at $z_{\mathbf{j}}^{\mathbf{i}}$ are computed by solving (4). Different from previous examples, the sparse grid is generated in (t, x)-space so that one can approximate $V(t, x)$ using an interpolation for $t \geq 0$ in a finite interval. As a result, the MPC feedback can be applied using a variable time horizon.

The problem has a known solution

$$V(t, x) = \frac{1}{2} \frac{x_3^2}{(1 + x_1^2 + x_2^2)^2} \tanh(T - t), \quad u^*(t, x,) = -\frac{x_3}{(1 + x_1^2 + x_2^2)} \tanh(T - t).$$

It is used to check the accuracy of the numerical result. In the computation, we adopted a CGL sparse grid with $q = 12$. The number of grid points for each dimension is $2^{q-4} + 1 = 257$. The total number of grid points in the 4-D domain is

$$\left| G_{\text{sparse}}^q \right| = 18{,}945,$$

in comparison to dense grids, a finite difference discretization using 33 points in each dimension results in a grid size greater than 1.18×10^6, which is impractical for personal computers.

At all points in the sparse grid, we set the error tolerance to be 10^{-7}. The upper bound of $e_{\text{interp}} + e_{\text{BVP}}$ is numerically approximated using the method in Sect. 4.3. More specifically, 1200 points are randomly generated in D. At each sample point, the value of $V(t, x)$ is computed using interpolation. The MAE is 8.5×10^{-4}. Figure 10 is the histogram of the approximation error, which shows a distribution concentrated around the MAE with a small variance 2.8×10^{-6}.

In the MPC feedback, we use a variable time window. Figure 11 is a trajectory under the closed-loop control. The initial time is $t_0 = 0$. Then the control input is updated using a fixed time step, $\Delta t = 1/7$. At each $t_k < 5$, the time window for the feedback is $T - t_k$. An interpolation on the sparse grid is used to compute $\lambda(t_k, x_k)$. Then control input $u(t_k, x_k)$ is applied to the system. At $t = 5$, the initial time is set to

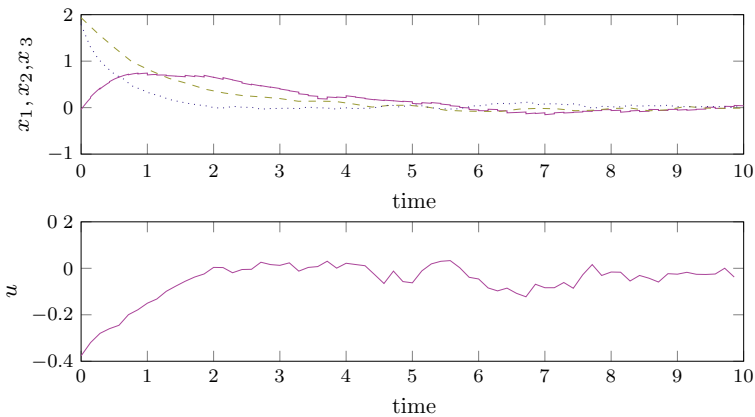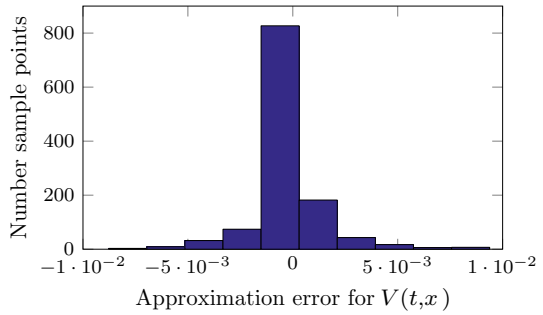**Fig. 10** The histogram of approximation errors for $V(t, x)$ in D

Fig. 11 Closed-loop trajectories (*solid line x_1 and u; dashed line x_2; dotted line x_3*)

zero and the process repeats. In the simulations, random noise of uniform distribution in $[-0.05, 0.05]$ is added to the state variables.

6 Conclusions

The characteristics method addressed in this paper is causality free and has perfect parallelism. It can be easily integrated in a parallel computer with any grid, such as sparse grids to mitigate the curse of dimensionality. The examples show that the algorithm is tractable for 6-D HJB equations. In the case of the attitude control of rigid body with six state variables, a solution is computed on a sparse grid with about 4.5×10^5 points, whereas the corresponding dense grid has more than 10^{12} points. In another example, the method is applied to a sparse grid in the time-state space so that the solution can be approximated within a given time interval. A theorem on the error upper bound is proved. In addition to the parallelism, another advantage of the causality free method is that the accuracy of the interpolated solution can be numerically checked pointwise. The effectiveness of the numerical solution for closed-loop control is demonstrated in Example II using a MPC controller.

For future research, an interesting question is how to generalize the idea to problems with an infinite time horizon. Another interesting research topic is to solve minimum-time problems. In the presence of multiple solutions, identifying the globally or locally optimal solution remains a challenge that needs to be addressed.

References

1. Al'brekht, E.G.: On the optimal stabilization of nonlinear systems. J. Appl. Math. Mech. **25**(5), 1254–1266 (1961). doi:10.1016/0021-8928(61)90005-3
2. Assellaou, M., Bokanowski, O., Desilles, A., Zidani, H.: A Hamilton–Jacobi–Bellman approach for the optimal control of an abort landing problem. In: Proceedings of the 55th IEEE Conference on Decision and Control (2016). doi:10.1109/CDC.2016.7798815
3. Bardi, M., Capuzzo-Dolcetta, I.: Optimal control and viscosity solutions of Hamilton–Jacobi–Bellman equations. Systems and Control: Incollection and Applications. Birkhäuser, Boston (1997). doi:10.1007/978-0-8176-4755-1_1
4. Barron, E.N., Jensen, R.: The Pontryagin maximum principle from dynamic programming and viscosity solutions to first-order partial differential equations. Trans. Am. Math. Soc. **298**(2), 635–635 (1986). doi:10.1090/s0002-9947-1986-0860384-4
5. Barthelmann, V., Novak, E., Ritter, K.: High dimensional polynomial interpolation on sparse grids. Adv. Comput. Math. **12**(4), 273–288 (2000). doi:10.1023/A:1018977404843
6. Bokanowski, O., Garcke, J., Griebel, M., Klompmaker, I.: An adaptive sparse grid semi-Lagrangian scheme for first order Hamilton–Jacobi–Bellman equations. J. Sci. Comput. **55**(3), 575–605 (2013). doi:10.1007/s10915-012-9648-x
7. Bungartz, H.J., Griebel, M.: Sparse grids. Acta Numer. **13**, 147–269 (2004). doi:10.1017/S0962492904000182
8. Cacace, S., Cristiani, E., Falcone, M., Picarelli, A.: A patchy dynamic programming scheme for a class of Hamilton–Jacobi–Bellman equations. SIAM J. Sci. Comput. **34**(5), A2625–A2649 (2012). doi:10.1137/110841576
9. Crouch, P.E.: Spacecraft attitude control and stabilization: applications of geometric control theory to rigid body models. IEEE Trans. Autom. Control **29**(4), 321–331 (1984). doi:10.1109/TAC.1984.1103519
10. Delvos, F.-J.: d-variate Boolean interpolation. J. Approx. Theory **34**(2), 99–114 (1982). doi:10.1016/0021-9045(82)90085-5
11. Diebel, J.: Representing attitude: Euler angles, unit quaternions, and rotation vectors (2006). http://www.astro.rug.nl/software/kapteyn//_downloads/attitude.pdf
12. Falcone, M., Ferretti, R.: Semi-Lagrangian Approximation Schemes for Linear and Hamilton–Jacobi Equations. Society for Industrial and Applied Mathematics, Philadelphia (2013). doi:10.1137/1.9781611973051
13. Fleming, W.H., Soner, H.M.: Controlled Markov Processes and Viscosity Solutions. Stochastic Modelling and Applied Probability, vol. 25. Springer, New York (2006). doi:10.1007/0-387-31071-1_1
14. Garcke, J.: Sparse grids in a nutshell. In: Garcke J., Griebel M. (eds) Sparse Grids and Applications, Lecture Notes in Computational Science and Engineering, vol 88. Springer, Berlin, Heidelberg (2012)
15. Gui, H., Jin, L., Xu, S.: Attitude maneuver control of a two-wheeled spacecraft with bounded wheel speeds. Acta Astronaut. **88**, 98–107 (2013). doi:10.1016/j.actaastro.2013.03.006
16. Hesthaven, J.S., Gottlieb, S., Gottlieb, D.: Spectral Methods for Time-Dependent Problems. Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, Cambridge (2007). doi:10.1017/CBO9780511618352
17. Horri, N.M., Hodgart, S.: Attitude stabilization of an underactuated satellite using two wheels. In: Aerospace Conference, 2003. Proceedings. 2003 IEEE, vol. 6, pp. 6_2629–6_2635 (2003). doi:10.1109/AERO.2003.1235188

18. Jahn, T.U.: A feasibility problem approach for reachable set approximation. Ph.D. thesis, University of Bayreuth, Bayreuth, Germany (2015). https://epub.uni-bayreuth.de/2087/

19. Kang, W., De, P.K., Isidori, A.: Flight control in a windshear via nonlinear h_∞ methods. In: Proceedings of the 31st IEEE Conference on Decision and Control, pp. 1135–1142 (1992). doi:10.1109/CDC.1992.371539

20. Kang, W., Wilcox, L.: A causality free computational method for HJB equations with application to rigid body satellites. In: AIAA Guidance, Navigation, and Control Conference, AIAA 2015-2009. American Institute of Aeronautics and Astronautics (2015). doi:10.2514/6.2015-2009

21. Kierzenka, J., Shampine, L.F.: A BVP solver that controls residual and error. J. Numer. Anal. Ind. Appl. Math. **3**(1–2), 27–41 (2008). http://jnaiam.org/index.php?/archives/57-A-BVP-Solver-that-Controls-Residual-and-Error.html

22. Kim, S., Kim, Y.: Spin-axis stabilization of a rigid spacecraft using two reaction wheels. J. Guid. Control Dyn. **24**(5), 1046–1049 (2001). doi:10.2514/2.4818

23. Klimke, A.: Uncertainty modeling using fuzzy arithmetic and sparse grids. Ph.D. thesis, Universität Stuttgart, Shaker Verlag, Aachen (2006)

24. Krishnan, H., McClamroch, N.H., Reyhanoglu, M.: Attitude stabilization of a rigid spacecraft using two momentum wheel actuators. J. Guid. Control Dyn. **18**(2), 256–263 (1995). doi:10.2514/3.21378

25. Lukes, D.L.: Optimal regulation of nonlinear dynamical systems. SIAM J. Control **7**(1), 75–100 (1969). doi:10.1137/0307007

26. Navasca, C., Krener, A.J.: Patchy solutions of Hamilton–Jacobi–Bellman partial differential equations. In: Chiuso, A., Pinzoni, S., Ferrante, A. (eds.) Modeling, Estimation and Control. Lecture Notes in Control and Information Sciences, vol. 364, pp. 251–270. Springer, Berlin (2007). doi:10.1007/978-3-540-73570-0_20

27. Pontryagin, L.S.: The Mathematical Theory of Optimal Processes. Interscience, New York (1962)

28. Smolyak, S.A.: Quadrature and interpolation formulas for tensor products of certain classes of functions. Dokl. Akad. Nauk SSSR **4**, 240–243 (1963)

29. Terui, F., Kawamoto, S., Fujiwara, T., Noda, A., Sako, N., Nakasuka, S.: Target tracking attitude maneuver of a bias momentum micro satellite using two wheels. In: AIAA Guidance, Navigation, and Control Conference and Exhibit, AIAA 2000-4114. American Institute of Aeronautics and Astronautics (2000). doi:10.2514/6.2000-4144

30. Tsiotras, P., Luo, J.: Control of underactuated spacecraft with bounded inputs. Automatica **36**(8), 1153–1169 (2000). doi:10.1016/S0005-1098(00)00025-X

31. Wasilkowski, G.W., Wozniakowski, H.: Explicit cost bounds of algorithms for multivariate tensor product problems. J. Complex. **11**(1), 1–56 (1995). doi:10.1006/jcom.1995.1001

32. Xiu, D., Hesthaven, J.S.: High-order collocation methods for differential equations with random inputs. SIAM J. Sci. Comput. **27**(3), 1118–1139 (2005). doi:10.1137/040615201

33. Zenger, C.: Sparse grids. In: Hackbusch, W., Gamm, G.f.A.M.u.M. (eds.) Parallel Algorithms for Partial Differential Equations—Proceedings of the Sixth GAMM-Seminar-Kiel, 19–21 Jan 1990, vol. 31. Vieweg, Braunschweig (1991)