2015-01

A Causality Free Computational Method for HJB Equations with Application to Rigid Body Satellites

Wilcox, Lucas

AIAA SciTech 5-9 January 2015, Kissimmee, Florida AIAA Guidance, Navigation, and Control Conference
http://hdl.handle.net/10945/48654

AIAA SciTech
5-9 January 2015, Kissimmee, Florida
AIAA Guidance, Navigation, and Control Conference

AIAA 2015-2009

A Causality Free Computational Method for HJB Equations with Application to Rigid Body Satellites*

Wei Kang† and Lucas Wilcox†
Naval Postgraduate School, Monterey, CA 93943, USA

Solving Hamilton-Jacobi-Bellman (HJB) equations is essential in feedback optimal control. Using the solution of HJB equations, feedback optimal control laws can be implemented in real-time with minimum computational load. However, except for systems with two or three state variables, numerically solving HJB equations for general nonlinear systems is unfeasible due to the curse of dimensionality. In this paper, we develop a new computational method of solving HJB equations. The method is 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 6-D HJB equations for the attitude control of rigid bodies. The method is applied to the optimal attitude control of a satellite system using momentum wheels. The accuracy of the numerical solution is verified at a set of randomly selected sample points.

I. Introduction

The attitude control of rigid satellite systems with momentum wheels has been an active topic of research for many years. The nonlinear nature of satellite models makes the attitude control problem attractive and challenging. The huge literature on this topic includes almost all popular feedback design approaches such as Lyapunov functions, linear and nonlinear H^∞ control, fuzzy-neuro control, linear and nonlinear output regulations, and adaptive control.

Optimal feedback control is used to stabilize the attitude while minimizing a cost function. Using dynamic programming, the feedback 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 Al'brecht,[1] Cacace et al.,[5] Kang et al.,[8] Lukes,[11] Navasca-Krener[12] and references therein. Some of the 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. In fact, numerically solving the HJB equation for a nonlinear system with six state variables is extremely challenging, if not impossible. To the best of our knowledge, no semi-global solutions have been found for the HJB equations for rigid body systems controlled by momentum wheels, either the controllable case or the underactuated case of two wheels.

In this paper, we develop a new computational method to solve the HJB equation for the optimal attitude control of rigid satellite with three or two momentum wheels. The curse of dimensionality is mitigated by using a sparse grid that employs Smolyak's construction.[13] The solution at each of the gridpoints is found using the Lobatto IIIa method to solve a two-point boundary problem. Different from many algorithms of solving PDEs, this approach is not based on spatial causality. A significant advantage of this causality free method lies in its perfectly parallelism, a desirable property for modern computation equipment with manycore clusters.

*This work was supported in part by AFOSR and NRL.
†Professor, Department of Applied Mathematics, Naval Postgraduate School.

In Section II, the problem of optimal attitude control is formulated using a quadratic cost function. The causality free computational method is introduced in Section III. Two examples are given in Section IV, one is a fully controllable system and the other is underactuated.

II. Problem formulation

Various models of rigid satellites using momentum wheels have been widely studied in the literature, for instance Byrnes-Isidori,[4] Crouch,[6] and Krishnan.[10] 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[7])

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

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,

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

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

$$\begin{aligned} \dot{v} &= E(v)\omega \\ J\dot{\omega} &= S(\omega)R(v)H + Bu \end{aligned} \tag{1}$$

where $B \in \Re^{3\times m}$ is a constant matrix, m is the number of momentum wheels, u is the control torque, $J \in \Re^{3\times 3}$ is a combination of inertia matrices of the rigid body without wheels and the momentum wheels, $H \in \Re^3$ is the total and constant angular momentum of the system, and $E(v)$, $S(\omega)$, $R(v)$ are the following matrices

$$E(v) = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi/\cos\theta & \cos\phi/\cos\theta \end{bmatrix}, \quad S(\omega) = \begin{bmatrix} 0 & \omega_3 & -\omega_2 \\ -\omega_3 & 0 & \omega_1 \\ \omega_2 & -\omega_1 & 0 \end{bmatrix}$$

$$R(v) = \begin{bmatrix} \cos\theta\cos\psi & \cos\theta\sin\psi & -\sin\theta \\ \sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi & \sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi & \cos\theta\sin\phi \\ \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi & \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi & \cos\theta\cos\phi \end{bmatrix}$$

In Crouch,[6] it is proved that the system is controllable if $m = 3$ and uncontrollable if $m < 3$ (underactuated). The problem of optimal control to be solved is to find a feedback $u_{\text{optimal}}(t, v, \omega)$ that minimizes the following cost

$$\begin{aligned} &\int_0^{t_f} L(v,\omega,u)dt + h(v(t_f),\omega(t_f)) \\ &L(v,\omega,u) = \frac{W_1}{2}||v||^2 + \frac{W_2}{2}||\omega||^2 + \frac{W_3}{2}||u||^2 \\ &h(v(t_f),\omega(t_f)) = \frac{W_4}{2}||v(t_f)||^2 + \frac{W_5}{2}||\omega(t_f)||^2 \end{aligned} \tag{2}$$

where W_i, $i = 1, 2, 3, 4$, are weight constants. Following the standard approach of optimal control, we define the Hamiltonian

$$H(v,\omega,\lambda_v,\lambda_\omega,u) = L(v,\omega,u) + \lambda_v^T E(v)\omega + \lambda_\omega^T J^{-1}(S(\omega)R(v)H + Bu)$$

The function

$$u^*(v,\omega,\lambda_v,\lambda_\omega) = -\frac{1}{W_3}B^T J^{-1}\lambda_\omega \tag{3}$$

minimizes the Hamiltonian. Define

$$H^*(v,\omega,\lambda_v,\lambda_\omega) = H(v,\omega,\lambda_v,\lambda_\omega,u^*) \tag{4}$$

Consider the following HJB equation with a final time condition

$$\begin{aligned} &V_t(t,v,\omega) + H^*(v,\omega,V_v^T(t,v,\omega),V_w^T(t,v,\omega)) = 0 \\ &V(t_f,v,\omega) = h(v,\omega) \end{aligned} \tag{5}$$

It is a PDE in which

$$V_v^T(t,v,\omega) = \left(\frac{\partial V}{\partial v}\right)^T, \quad V_\omega^T(t,v,\omega) = \left(\frac{\partial V}{\partial \omega}\right)^T$$

are column vectors. If one can solve (5), the feedback control law is a function defined as follows

$$u_{\text{optimal}}(t,v,\omega) = u^*(v,\omega,V_v^T(t,v,\omega),V_\omega^T(t,v,\omega))$$

III. A causality free algorithm

Typically, partial differential equations are numerically solved based on a discretization that results in a finite dimensional problem. However, the dimension of the discretization space increases exponentially with the dimension of the PDE. Taking dense grids as an example, if d is the dimension of a PDE, then the size of a grid is N^d, where N is the number of grid points used to approximate a single dimension in the PDE. The size of the grid is proportional to the dimension of the discretization space. There are six state variables, (v,ω), in (5). If $2^5 = 32$ gridpoints are used to approximate a single variable, which is quite small, the total number of gridpoints for the 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}. For conventional methods of computational PDEs, the required computational time and memory size are simply too high for practical applications.

In this paper, a causality free computational method consists of two components: (1) A solver that can find the value of $V(t,x)$ at any grid point; and the computation is independent of the approximation of $V(t,x)$ at other points. (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 boundary value problem solvers and sparse grids. The goal is to solve HJB equations for $d < 10$. In this paper, we use (5), in which $d = 6$, to test the method.

A. Sparse grids

In this paper, we adopt the Chebyshev-Gauss-Lobatto (CGL) sparse grid. It is a known fact that the size of sparse grids increases with the dimension, d, in the order of

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

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

$$O(N^d).$$

Obviously, the significantly reduced number of gridpoints has its impact to accuracy. An upper bound of interpolation error using a CGL sparse grid is derived in Barthelmann et al.[2] For a given function $f : [-1,1]^d \to \Re$, let k be an integer such that all derivatives $D^\beta f$, where $\beta \in \mathbb{N}^d$ and $\beta_i \le k$, are continuous. Then the error of interpolation on a sparse grid is

$$||e||_\infty = O(N^{-k}|\log N|^{(k+2)(d+1)+1})$$

Compared with the error bound using a dense grid,

$$O(N^{-k})$$

we pay a small price in terms of accuracy to achieve a significantly reduced size of the grid.

Sparse grids have a hierarchical structure. For each variable, the set of gridpoints contains several layers of subsets, denoted by X^i. The number of points in each subset satisfies

$$\begin{cases} m_1 = 1, \\ m_i = 2^{i-1} + 1, \end{cases} \tag{6}$$

The gridpoints are defined as follows

$$\begin{cases} X^1 = \left\{ \dfrac{1}{2} \right\} \\ X^i = \left\{ \dfrac{1}{2}\left(1 - \cos\dfrac{(k-1)\pi}{2^{i-1}}\right), \quad k = 1, 2, \cdots m_i \right\} \end{cases} \tag{7}$$

Note that $X^{i-1} \subset X^i$. The set of points in X^i but not in X^{i-1} is denoted by ΔX^i. In $[-1,1]^d$, the dense grid build on X^q for an integer $q > 0$ is

$$X^q \times \cdots \times X^q = \bigcup_{1 \le |\mathbf{i}| \le q} \Delta X^{\mathbf{i}}$$

where

$$\begin{aligned} &\mathbf{i} = \left[\begin{array}{cccc} i_1 & i_2 & \cdots & i_d \end{array}\right] \\ &|\mathbf{i}| = i_1 + i_2 + \cdots + i_d \\ &\Delta X^{\mathbf{i}} = \Delta X^{i_1} \times \Delta X^{i_2} \times \cdots \times \Delta X^{i_d} \end{aligned}$$

Following Smolyak's approximation algorithm,[2,13] the sparse grid, denoted by G^q_{sparse}, is defined as follows,

$$G^q_{\text{sparse}} = \bigcup_{|\mathbf{i}| \le q} \Delta X^{\mathbf{i}}$$

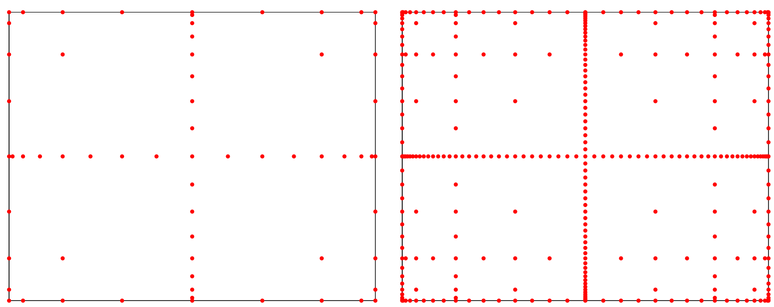

Figure 1. CGL sparse grid in $[0,1]^2$, $q = 6$ and $q = 8$

Figure 1 shows two examples of G^q_{sparse}. For $q = 8$, G^q_{sparse} has 385 gridpoints whereas the corresponding dense grid has $(2^6 + 1)^2 = 4225$ points. The difference of grid size is increasingly significant for higher dimensions. In fact, the size of G^q_{sparse} increases with d in the order of

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

in sharp contrast to $O(N^d)$, the size of the corresponding dense grid. The significantly reduced number of gridpoints makes it possible to discretize a PDE into a tractable numerical problem.

B. Achieving causality-free using necessary conditions of optimal control

In most numerical methods for the HJB equation (which is typically solved backwards in time), the discretization is based on spatial causality and explicit in time, i.e., the value of the solution function $V(t, v, \omega)$ at a gridpoint is computed at an earlier time using the known value of the function at neighboring gridpoints at a later time. This coupling usually comes from the discretization of the spatial derivatives. For HJB equations of high dimensions, in this case $d = 6$, solving the equation using traditional algorithms based on dense grids is very difficult, if not impossible. Developing algorithms based on sparse grids is a promising approach to mitigate the curse of dimensionality. In our opinion, applying spatial causality in algorithms on a sparse grids is challenging. The distance between adjacent points in a sparse grid varies in a large range due to the hierarchical structure. In contrast to this, our proposed discretization technique does not discretize the HJB equations directly but instead uses Pontryagin's minimum principle to derive a set of necessary conditions in the form of a boundary value problem for each grid point. As a result, the computation of the solution at an initial point in space is independent of other points. This approach is also different from the semi-Lagrangian method on sparse grid in Bokanowski et al.[3] in which the HJB is integrated backward in time while the gridpoints are adaptively adjusted based on the value of the computed solution at neighboring points at a later time.

In general, consider a control system

$$\dot{x}(t) = f(t, x(t), u(t)) \qquad (t_0 < t < t_f) \tag{8a}$$

$$x(t_0) = x_0 \tag{8b}$$

The cost functional to be minimized is

$$\min_{u(t)} \int_{t_0}^{t_f} L(t, x, u)\, dt + h(t_f, x(t_f)) \tag{9}$$

For the rigid satellite system, x consists of $\begin{bmatrix} \phi & \theta & \psi \end{bmatrix}^T$ and ω. Define the Hamiltonian

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

Suppose we can minimize it

$$u^*(t, x, \lambda) = \arg\min_{u(t)} H(t, x, \lambda, u).$$

Define

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

Then the optimal trajectory with an initial condition x_0 satisfies

$$\dot{x} = \left(\frac{\partial H^*}{\partial \lambda}(t, x, \lambda)\right)^T \tag{10a}$$

$$\dot{\lambda} = -\left(\frac{\partial H^*}{\partial x}(t, x, \lambda)\right)^T \tag{10b}$$

$$\dot{z} = L(t, x, u^*(t, x, \lambda)) \tag{10c}$$

with the boundary conditions

$$x(t_0) = x_0 \tag{10d}$$

$$\lambda(t_f) = h_x^T(t_f, x(t_f)) \tag{10e}$$

$$z(t_0) = 0 \tag{10f}$$

(10g)

The optimal control and the minimum costs are

$$u^*(t) = u^*(t, x(t), \lambda(t)), \quad V(t_0, x_0) = z(t_f) + h(t_f, x(t_f)) \tag{11}$$

Given any gridpoint, x_0, in G^q_{sparse}, we can solve the boundary value problem (10) 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.

C. Some remarks on numerical computations

Numerical algorithms for boundary value problems similar to (10) have been studied by many authors. In the examples, we adopt an algorithm based on the four-stage Lobatto IIIa formula. This is a collocation formula and the collocation polynomial provides a solution that is fifth-order accurate (see Kierzenka-Shampine[9]).

In the computation, the numerical solution is able to achieve accurate solutions at any given point with estimated errors smaller than 10^{-12}. For the reason of CPU time, we set error tolerance at 10^{-6}.

From a conventional viewpoint, solving two-point boundary value problems is not an efficient approach for PDEs. On the other hand, the causality free method is perfectly parallel. In fact, the computation of $V(t,x)$ at each gridpoint can be carried out without the need of any information from other points in its neighborhood. Although not a preferred algorithm in conventional serial computation, causality free algorithms can easily be implemented in massively parallel computational equipment. The combination of sparse grids, boundary value problem solvers, and parallel computation is the key to mitigate the curse of dimensionality effectively for problems in which d is not too large.

The interpolation on a sparse grid is used to compute $V(t,x)$ if x is not a gridpoint. Consider $X^i \subseteq [0,1]$, $i \geq 1$. A basis function, $a_{x^i}(x)$, for a point $x^i \in X^i$ is defined on $[0,1]$ satisfying

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

or a simplified notation for $x^i_j \in \Delta X^i$

$$a^i_j(x) = a_{x^i_j}(x)$$

A few basis functions at CGL gridpoints are shown in Figure 2. They are defined using a polynomial interpolation

$$a^i_j(x) = \prod_{x^i \in X^i} \frac{x - x^i_j}{x^i - x^i_j}$$

The interpolation on a sparse grid does not need every basis function. In fact, for each $i \geq 1$, an interpolation

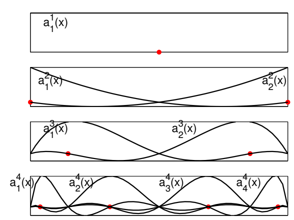

Figure 2. Basis functions for CGL grid

function uses only those a_{x^i} for which $x^i \in \Delta X^i$. Let $I^q(f)$ be the interpolation of f at gridpoints of G^q_{sparse}. It is defined iteratively on $[0,1]^d$

$$\begin{aligned} & I^{d-1}(f) = 0 \\ & I^q(f) = I^{q-1} + \Delta I^q(f), \qquad\qquad q \geq d \\ & \Delta I^q(f) = \sum_{|\mathbf{i}|=q} \sum_{1 \leq \mathbf{j} \leq \Delta m^{\mathbf{i}}} w^{\mathbf{i}}_{\mathbf{j}} a^{i_1}_{j_1} \otimes \cdots \otimes a^{i_d}_{j_d} \\ & w^{\mathbf{i}}_{\mathbf{j}} = f(x^{\mathbf{i}}_{\mathbf{j}}) - I^{q-1}(f)(x^{\mathbf{i}}_{\mathbf{j}}) \end{aligned} \tag{12}$$

where $\Delta m^{\mathbf{i}}$ is the size of $\Delta X^{\mathbf{i}} = \Delta X^{i_1} \otimes \cdots \otimes \Delta X^{i_d}$ and

$$a^{i_1}_{j_1} \otimes \cdots \otimes a^{i_d}_{j_d}(x_1, \cdots, x_d) = a^{i_1}_{j_1}(x_1) \cdots a^{i_d}_{j_d}(x_d)$$

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

IV. Examples

Two examples are used to test the algorithm, one is controllable and the other is underactuated.

A. A satellite system with three control momentum wheels

Consider a rigid body controlled by three momentum wheels. In the model (1), B is a 3×3 matrix. The following parameter values are used in this example

$$\begin{aligned} B &= \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1/2 & 1/2 \\ 1/2 & 0 & 1/3 \end{bmatrix} \\ J &= \begin{bmatrix} 2 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 4 \end{bmatrix} \\ H &= \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}^T \\ W_1 &= 1, W_2 = 1, W_3 = 1/2, W_4 = 1, W_5 = 1 \\ t_0 &= 0,\ t_f = 20 \end{aligned} \tag{13}$$

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,

$$D_1 : \quad \begin{aligned} -\frac{\pi}{12} \leq \phi, \theta, \psi \leq \frac{\pi}{12} \\ -0.1 \leq \omega_1, \omega_2, \omega_3 \leq 0.1 \end{aligned} \tag{14}$$

$$D_2 : \quad \begin{aligned} -\frac{\pi}{3} \leq \phi, \theta, \psi \leq \frac{\pi}{3} \\ -0.2 \leq \omega_1, \omega_2, \omega_3 \leq 0.2 \end{aligned} \tag{15}$$

In D_1, the sparse grid of $q = 11$ is used. The number of gridpoints for each dimension is $2^{q-6} + 1 = 33$. The total number of gridpoints in the 6-D domain is

$$|G^q_{\text{sparse}}| = 4,865$$

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

$$|G_{\text{dense}}| = 33^6 > 10^9$$

In the larger domain, D_2, we increase the size of G^q_{sparse} to the level of $q = 13$. In this case,

$$|G^q_{\text{sparse}}| = 44,698$$

The size of the corresponding dense grid has more than 4×10^{12} points.

In both D_1 and D_2, the TPBVP (10) is solved at each gridpoint in G^q_{sparse} using a method based on four-stage Lobatto IIIa formula in Kierzenka-Shampine.[9] The hierarchical surpluses for interpolation are computed using (12). Then we check the accuracy of $V(0, v, \omega)$. More specifically, 500 points are randomly generated in D_1 and D_2. The value of $V(0, v, \omega)$ is computed at these points using interpolation on G^q_{sparse}. The true value at the same point is approximated by solving (10). The difference between these two numbers is an approximation of error in interpolation. In the small domain D_1, the root-mean-square error (RMSE) is 1.2×10^{-5}. In the large domain D_2, the result is less accurate. The RMSE equals 4.3×10^{-3}. The results are summarized in Table 1.

Domain	q	$\lvert G^q_{\text{sparse}}\rvert$	Dense grid size	RMSE
D_1	$q = 11$	$4,865$	$> 10^9$	1.2×10^{-5}
D_2	$q = 13$	$44,698$	$> 10^{12}$	4.3×10^{-3}

Table 1. Summary of results

At $\omega = 0$, the graph of $V(0, v, \omega)$ is shown in Figure 3. The optimal control is supposed to stabilize the system. Figure 4 has a typical trajectory in which (v, ω) converges to zero.

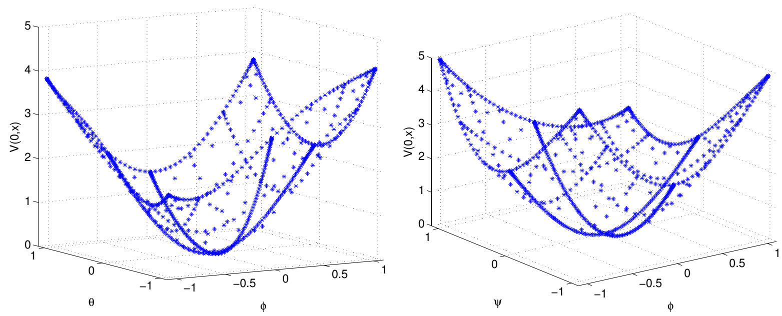

Figure 3. $V(0, x)$ **on** $\phi\theta$**- and** $\phi\psi$**-planes in** D_2

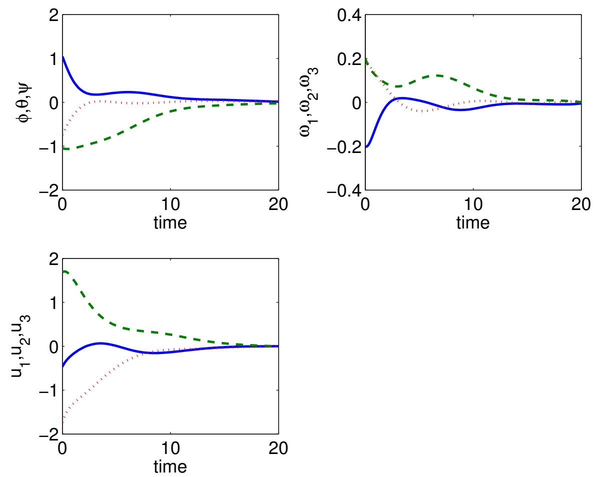

Figure 4. Trajectories (Line: ϕ, ω_1, u_1; **dash:** θ, ω_1, u_2; **dot:** ψ, ω_3, u_3 **)**

B. A satellite system with two actuators

Consider a rigid body controlled by two momentum wheels. The parameters in this example are assigned the same values as in (13) except that the input matrix is changed,

$$B = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}$$

All computations in this example are based on a sparse grid of $q = 11$ in the region D_1. A graph of $V(0, v, \omega)$ at $\omega = 0$ is shown in Figure 5. A typical trajectory is shown in Figure 6. It is proved in Crouch[6] that the system is uncontrollable. In general, $v(t)$ does not approach to zero even under an optimal control. It is also proved in the literature that, if appropriate control input is applied, $\omega(t)$ may converge to zero. In other words, one can stop the rotation using an optimal control but it is impossible to point the satellite to a given orientation. A typical trajectory in Figure 6 does converge. In fact, ω is almost stabilized at $t = 10$. When t approaches the end of the time interval, $\omega(t)$ is re-activated. This is due to the end point cost in the problem formulation (2).

The value of $V(0, v, \omega)$ at the gridponts are computed by solving the TPBVP (10). Then the hierarchical surpluses for the interpolation are computed using (12). Similar to the previous example, we check the accuracy at 500 random points in D_1. The value of $V(0, v, \omega)$ is computed at these points using interpolation on G^q_{sparse}. The true value at the same point is approximated by solving (10). The resulting RMSE equals 6.2×10^{-3}.

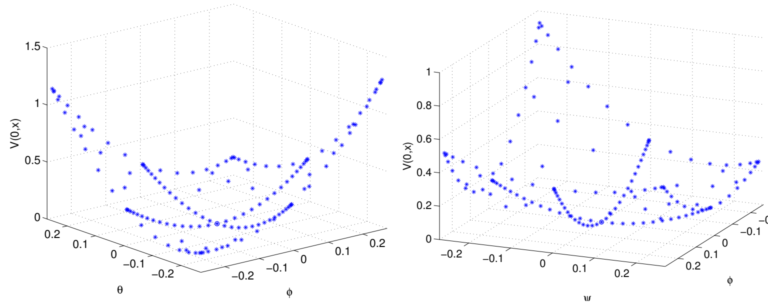

Figure 5. $V(0,x)$ on $\phi\theta$- and $\phi\psi$-planes in D_1

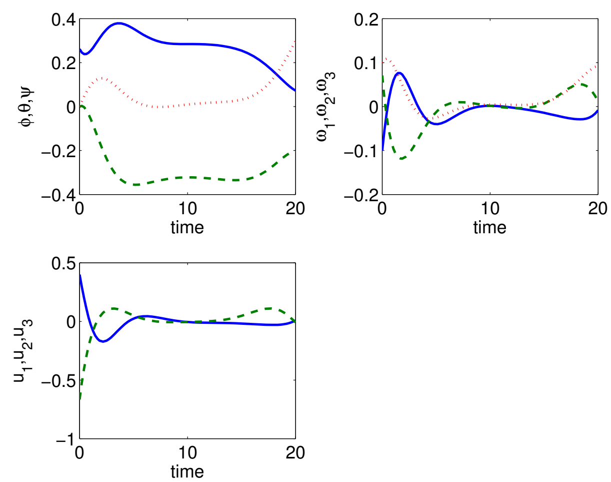

Figure 6. Trajectories (Line: ϕ, ω_1, u_1; dash: θ, ω_1, u_2; dot: ψ, ω_3, u_3)

V. Conclusions

The causality free method introduced in this paper is perfectly parallel. In contrast to dense grids, the size of sparse grids is significantly small. In the case of a controllable satellite with six state variables, an accurate solution is achieved on a sparse grid of less than five thousand points, whereas a similar accuracy on a dense grid would require more than 10^9 gridpoints. The accuracy decreases when the size of the region of initial states is increased. For the uncontrollable example, the error is large in the relatively small region. The accuracy is not satisfactory. For future work, the algorithm will be implemented in computers with manycore clusters so that a larger sparse grid and higher error tolerance can be used to increase the overall accuracy of the solution. A feedback control law will be constructed based on the solution of the HJB equation. It will be implemented in a closed-loop simulation to test the performance of the feedback law.

References

[1]Al'Brekht, E. G., "On the Optimal Stabilization of Nonlinear Systems," *Journal of Applied Mathematics and Mechanics*, Vol. 25, No. 5, 1961, pp. 1254–1266.

[2]Barthelmann, V., Novak, E., and Ritter, K., "High Dimensional Polynomial Interpolation on Sparse Grids," *Advances in Computational Mathematics*, Vol. 12, No. 4, 2000, pp. 273–288.

[3]Bokanowski, O., Garcke, J., Griebel, M., and Klompmaker, I., "An Adaptive Sparse Grid Semi-Lagrangian Scheme for First Order Hamilton-Jacobi Bellman Equations," *Journal of Scientific Computing*, Vol. 55, No. 3, 2013, pp. 575–605.

[4]Byrnes, C. I. and Isidori, A., "On the Attitude Stabilization of Rigid Spacecraft," *Automatica*, Vol. 27, No. 1, 1991, pp. 87–95.

[5]Cacace, S., Cristiani, E., Falcone, M., and Picarelli, A., "A Patchy Dynamic Programming Scheme for a Class of

Hamilton–Jacobi–Bellman Equations," *SIAM Journal on Scientific Computing*, Vol. 34, No. 5, 2012, pp. A2625–A2649.

[6]Crouch, P., "Spacecraft Attitude Control and Stabilization: Applications of Geometric Control Theory to Rigid Body Models," *IEEE Transactions on Automatic Control*, Vol. 29, No. 4, April 1984, pp. 321–331.

[7]Diebel, J., "Representing Attitude: Euler Angles, Unit Quaternions, and Rotation Vectors," `http://www.astro.rug.nl/software/kapteyn/_downloads/attitude.pdf`, 2006.

[8]Kang, W., De, P. K., and Isidori, A., "Flight Control in a Windshear via Nonlinear H^∞ Methods," *Proceedings of the 31st IEEE Conference on Decision and Control*, Vol. 1, 1992, pp. 1135–1142.

[9]Kierzenka, J. and Shampine, L. F., "A BVP Solver that Controls Residual and Error," *Journal of Numerical Analysis, Industrial and Applied Mathematics*, Vol. 3, No. 1–2, 2008, pp. 27–41.

[10]Krishnan, H., McClamroch, N. H., and Reyhanoglu, M., "Attitude Stabilization of a Rigid Spacecraft Using Two Momentum Wheel Actuators," *Journal of Guidance, Control, and Dynamics*, Vol. 18, No. 2, 1995, pp. 256–263.

[11]Lukes, D. L., "Optimal Regulation of Nonlinear Dynamical Systems," *SIAM Journal on Control*, Vol. 7, No. 1, 1969, pp. 75–100.

[12]Navasca, C. and Krener, A. J., "Patchy Solutions of Hamilton–Jacobi–Bellman Partial Differential Equations," *Modeling, Estimation and Control*, edited by A. Chiuso, S. Pinzoni, and A. Ferrante, Vol. 364 of *Lecture Notes in Control and Information Sciences*, Springer Berlin Heidelberg, 2007, pp. 251–270.

[13]Smolyak, S. A., "Quadrature and Interpolation Formulas for Tensor Products of Certain Classes of Functions," *Dokl. Akad. Nauk SSSR*, Vol. 4, 1963, pp. 240–243.