An Example of Solving HJB Equations Using Sparse Grid for Feedback Control

Wei Kang and Lucas Wilcox

Abstract—It is well known that solving the Hamilton-Jacobi-Bellman (HJB) equation in moderate and high dimensions 
\( \left( d > 3 \right) \) suffers the curse of dimensionality. In this paper, we introduce and demonstrate an example of solving the 6-D HJB equation for the optimal attitude control of a rigid body equipped with two pairs of momentum wheels. The system is uncontrollable. To mitigate the curse-of-dimensionality, a computational method based on sparse grids is introduced. The method is causality free, which enjoys the advantage of perfect parallelism. The problem is solved using several hundred CPU cores in parallel. In the simulations, the solution of the HJB equation is integrated into a model predictive control for optimal attitude stabilization.

I. INTRODUCTION

It is well known that a feedback control law of nonlinear systems can be constructed based on the solution of the Hamilton-Jacobi-Bellman (HJB) equation, which is a partial differential equation (PDE). 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 the HJB-type of equations in a local neighborhood of a trajectory has been extensively studied [1], [6], [14], [20], [21]. Some of them can be applied to systems in 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 more than four state variables is already challenging, if not impossible. For instance, the HJB equations for rigid body systems controlled by momentum wheels have six state variables. To the best of our knowledge, no semi-global solutions have been found for this problem.

In [15], a computational method of solving HJB equations for nonlinear systems is developed based on sparse grids. The algorithm is causality free, i.e. the value of the solution at a gridpoint in the state space is computed independently from the value at other gridpoints. A causality free method has a perfect parallelism, which is a significant advantage for high dimensional problems. This method has the potential to mitigate the curse of dimensionality for system with a moderate dimension. The idea of sparse grids is based on Smolyak's work [22]. The reader is also referred to [4], [10], [25] for more details in different perspectives. Relative to a dense grid, the size of the sparse grid is significantly reduce. In the computation, the solution at each gridpoint is found using the Lobatto IIIa method that solves a two-point boundary problem (TPBVP). Different from some popular 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. In the example, all computations are carried out in a parallel computer and 512 CPU cores are used.

In this paper, we exemplify the computational method by solving a problem of optimal attitude control of rigid body. The example in itself is interesting. Consider a rigid body controlled by two pairs of momentum wheels. Although satellite systems are equipped with at least three pairs of control momentum wheels, malfunction may occur to some wheels. It has been proved in [7] that this system is uncontrollable. How to stabilize the satellite around a desired attitude is a challenging problem. Related work can be found in [24], [11], [17], [19], [23], [12] 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 [15] where the stabilization does not guarantee an optimal attitude and the control is open-loop. 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.

II. A CAUSALITY FREE ALGORITHM

Consider a control system

\[
\dot{x}(t) = f(t, x(t), u(t)) \quad (t_0 < t < t_f) \\
-x(t_0) = x_0
\]

(1a)

(1b)

The cost functional to be minimized is

\[
\min_u \int_{t_0}^{t_f} L(t, x, u) \, dt + h(t_f, x(t_f))
\]

(2)

Define the Hamiltonian

\[
H(t, x, \lambda, u) = L(t, x, u) + \lambda^T f(t, x, u)
\]
Suppose we can minimize it

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

Define

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

The HJB equation with boundary condition is defined as follows

$$V_t(t, x) + H^*(t, x, V_x(t, x)) = 0$$

(3)

It is a PDE in which

$$V_x(t, x) = \left( \frac{\partial V}{\partial x} \right)^T$$

is a column vector. The feedback control law is a function defined as follows

$$u_{\text{optimal}}(t, x) = u^*(t, x, V_x(t, x))$$

Partial differential equations like (3) are numerically solved using discretization methods that result in a finite dimensional problem. The dimension of the discretized space increases exponentially with the dimension of the PDE. If \(d\) is the dimension of a PDE, then the size of a dense grid is \(N^d\), where \(N\) is the number of gridpoints used to approximate a single variable in the PDE. For example, if \(2^5 = 32\) gridpoints are used to approximate a single variable, which is quite small, the total number of gridpoints for a 6-D problem is over \(10^9\). If 100 points are used for a single variable, then \(N^d > 10^{12}\). The required computational time and memory size by such dense grids are simply too high for practical applications.

In [15], a causality free computational method consists of two components: (1) A solver that can find the value of \(V(t, x)\) at any gridpoint; and the computation is independent of \(V(t, x)\) at other points. (2) A set of gridpoints, such as a sparse grid, with a reduced size to make the problem tractable. 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 on the accuracy. For a given function \(f : [-1, 1]^d \rightarrow \mathbb{R}\), the error of interpolation on a sparse grid is

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

if \(f\) has bounded mixed derivatives up to the second order. Compared with the error bound using a dense grid,

$$O(N^{-2})$$

we pay a small price in terms of accuracy to achieve a significantly reduced size of the grid. For more details, the reader is referred to [4], [10] and references therein.

In this paper, we adopt the Chebyshev-Gauss-Lobatto (CGL) sparse grid [2]. 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

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

(4)

The gridpoints are defined as follows

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

(5)

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

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

where

$$|i| = i_1 + i_2 + \cdots + i_d,$$

$$\Delta X^i = \Delta X^{i_1} \times \Delta X^{i_2} \times \cdots \times \Delta X^{i_d}.$$

Following Smolyak’s approximation algorithm [22], [2], the sparse grid, denoted by \(G^q\text{_{sparse}}\), is defined as follows,

$$G^q\text{_{sparse}} = \bigcup_{|i| \leq q} \Delta X^i$$

Fig. 1. CGL sparse grid in \([0,1]^2\), \(q = 6\) and \(q = 8\)

Figure 1 shows two examples of \(G^q\text{_{sparse}}\). For \(q = 8\), \(G^8\text{_{sparse}}\) 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. The significantly reduced number of gridpoints makes it possible to discretize a PDE into a tractable numerical problem.

Some discretization methods, such as finite difference and central difference, are based on spatial causality. The value of \(V(t, x)\) at a gridpoint is approximated using the value of \(V(t, x)\) at some other gridpoints in its neighborhood. It is difficult to apply this type of methods on a sparse grid because the distance between adjacent points in a sparse grid varies in a large range due to the hierarchical structure. In
contrast to this, the discretization technique in [15] does not discretize a HJB equations directly. Instead, it uses Pontryagin’s minimum principle (PMP) to derive a set of necessary conditions in the form of a boundary value problem for each gridpoint. As a result, the computation of the solution at a point in space is independent of other points. This approach is also different from the semi-Lagrangian method on sparse grid in [3] in which the equation 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 the following, the value \( V(t_0, x_0) \) for any point \( x_0 \) in the sparse grid is computed by solving the following two point boundary value problem (TPBVP) derived from PMP,

\[
\begin{align}
\dot{x} &= \left( \frac{\partial H^*}{\partial x}(t, x, \lambda) \right)^T \tag{6a} \\
\dot{\lambda} &= - \left( \frac{\partial H^*}{\partial x}(t, x, \lambda) \right)^T \tag{6b} \\
\dot{z} &= L(t, x, u^*(t, x, \lambda)) \tag{6c}
\end{align}
\]

with the boundary conditions

\[
\begin{align}
x(t_0) &= x_0, & z(t_0) &= 0 \tag{6d} \\
\lambda(t_f) &= h_x^T(t_f, x(t_f)) \tag{6e}
\end{align}
\]

The optimal control and the minimum costs are

\[
u^*(t) = u^*(t, x(t), \lambda(t)), \quad V(t_0, x_0) = z(t_f) + h(t_f, x(t_f))
\]

Numerical algorithms for boundary value problems similar to (6) have a sizable literature. In this paper, 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 [16]. A nice feature of this algorithm is that the error of estimation can be numerically approximated. In the computation, the numerical solver is able to achieve approximate solutions with an error tolerance \( \epsilon = 10^{-12} \). For the reason of computational time, we set error tolerance at \( \epsilon = 10^{-6} \).

From a conventional viewpoint of computation, solving a TPBVP is not an efficient approach for PDEs. However, this method has perfect parallelism. Although not a preferred algorithm for serial computers, causality free algorithms can easily be implemented in massively parallel computational equipment. In this paper, all computations are carried out in parallel using 512 CPU cores. 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 a moderate or high dimensional space. In this paper, the examples have \( d = 6 \).

Solving the HJB equation can be done off-line using massive computation. The result is then uploaded onto the system for real-time feedback control. The on-line computation must be simple and fast. For this purpose, an interpolations 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 \( x^j \in \Delta X^i \)

\[
a^j_i(x) = a_{x^i}(x)
\]

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

\[
a^j_i(x) = \prod_{x^i \in X^i} \frac{x - x^j}{x^i - x^j}
\]

The interpolation on a sparse grid does not need every basis

![Fig. 2. Basis functions for CGL grid](image)

function. In fact, for each \( i \geq 1 \), an interpolation 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 \( G^q_{\text{sparse}} \). It is defined iteratively on \( [0, 1]^d \)

\[
\begin{align}
I^0(f)(x) &= f(x), \\
I^q(f)(x) &= I^{q-1}(f) + \Delta I^q(f), \\
\Delta I^q(f) &= \sum_{i=q} I^q - I^{q-1}(f)(x_i^q)
\end{align}
\]

where \( \Delta m^q \) is the size of \( \Delta X^1 \) and \( \Delta X^{i+1} \) and

\[
a^{j_1}_{i_1} \cdots a^{j_d}_{i_d}(x_1, \ldots, x_d) = a^{j_1}_{i_1}(x_1) \cdots a^{j_d}_{i_d}(x_d)
\]

The weights, \( w^j_i \), are called hierarchical surpluses.

III. THE OPTIMAL CONTROL OF RIGID BODY - AN UNCONTROLLABLE CASE

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 [8])

\[
v = \begin{bmatrix} \phi & \theta & \psi \end{bmatrix}^T
\]
in which \( \phi, \theta, \) and \( \psi \) 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 [7]

\[
\begin{align*}
\dot{v} &= E(v)\omega \\
J\dot{\omega} &= S(\omega)R(v)H + Bu
\end{align*}
\]

where \( B \in \mathbb{R}^{3 \times 2} \) is a constant matrix, \( u \) 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 \( E(v), S(\omega), R(v) \) are matrices. Details can be found in [15]. The system has only two control inputs. In [7], it is proved that (9) is uncontrollable. In this section, the following values are assigned to the parameters

\[
B = \begin{bmatrix} 1 & 1 \\
0 & 1 \\
1 & 2 \\
\end{bmatrix}, \quad J = \begin{bmatrix} 2 & 0 & 0 \\
0 & 3 & 0 \\
0 & 0 & 4 \\
\end{bmatrix}, \quad H = \begin{bmatrix} 12 & 12 & 6 \\
\end{bmatrix}^T
\]

The optimal control is to minimize the following functional

\[
\min_u \int_0^{t_f} L(v, \omega, u) dt
\]

\[
L(v, \omega, u) = \frac{W_1}{2}||v - v_e(v, \omega)||^2 + \frac{W_2}{2}||\omega||^2 + \frac{W_3}{2}||u||^2
\]

where

\[
W_1 = 1, \quad W_2 = 2, \quad W_3 = 0.5
\]

\[
t_0 = 0, \quad t_f = 30
\]

The function \( v_e(v, \omega) \) represents the optimal attitude reachable from \((v, \omega)\). 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 are derived to stabilize the system [24], [11], [17], [19], [23], [12]. In the case of \( H \neq 0 \), a manifold of reachable states \((v, \omega)\) satisfies [7]

\[
C^T(J\omega - R(v)H) = \text{constant},
\]

for some \( C \in \mathbb{R}^3, C^TB = 0 \)

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. 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

\[
v_e(v, \omega) = \arg\max_v \text{tr}(R(\tilde{v}))
\]

subject to

\[
-C^T R(\tilde{v})H = C^T(J\omega - R(v)H)
\]

It can be proved that \( v_e(v(t), \omega(t)) \) is a constant along any controlled trajectory. Therefore, it can be treated as a constant in the derivation of (6). The solution of (13) can be found by means of numerical nonlinear programming.

IV. A NUMERICAL SOLUTION OF V(t₀, v, ω)

The HJB equation (3) is solved at \( t = 0 \) on a sparse grid in the following domain, \( D \),

\[
D : \quad \frac{\pi}{10} \leq \phi, \theta, \psi \leq \frac{\pi}{5}, \quad \frac{\pi}{8} \leq \omega_1, \omega_2, \omega_3 \leq \frac{\pi}{8}
\]

The sparse grid of \( q = 13 \) is used. The number of gridpoints for each dimension is \( 2^{q-2} + 1 = 129 \). The total number of gridpoints in the 6-D domain is

\[
|G_{\text{sparse}}^q| = 44,689
\]

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

\[
|G_{\text{dense}}| > 4 \times 10^{12}
\]

In \( D \), the TPBVP (6) is solved at each gridpoint in \( G_{\text{sparse}}^q \) using a method based on a four-stage Lobatto IIIa formula [16]. The computation is carried out in Hamming, a parallel computer in Naval Postgraduate School. Although as many as \( |G_{\text{sparse}}^q| \) can be used, we limit the computation to 512 CPU cores.

For \((v, \omega)\) not in \( G_{\text{sparse}}^q \), the value of \( V(0, v, \omega) \) is approximated through interpolation. The hierarchical surpluses for interpolation are computed using either (8) or other more efficient algorithms. An advantage of a causality free method based on a Lobatto IIIa algorithm is that the error of \( V(0, v, \omega) \) can be approximated at an arbitrary point in \( D \), without being contaminated by the computational error at the gridpoints. In fact, under some smoothness assumptions, the error in the solution of (6) can be numerically estimated [16].

To validate the accuracy of the numerical results, \( N = 1280 \) points are randomly generated in \( D \). The value of \( V(0, v, \omega) \) is computed at these points using interpolation on \( G_{\text{sparse}}^q \). The true value at the same point is approximated by solving (6). We use the difference between these two numbers as an approximate of the error. Because the error of the TPBVP solver is significantly smaller than the interpolation error, the later is ignored. The mean absolute error (MAE) is \( 8.5 \times 10^{-3} \) and the root-mean-square error (RMSE) is \( 1.7 \times 10^{-2} \). The results are summarized in Table I. The accuracy 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. A typical trajectory is shown in Figure 3.

| \( q \) | \( |G_{\text{sparse}}^q| \) | Dense grid size | MAE | RMSE |
|---|---|---|---|---|
| 13 | 44,689 | \( > 10^{12} \) | \( 8.5 \times 10^{-3} \) | \( 1.7 \times 10^{-2} \) |

TABLE I

SUMMARY OF RESULTS
V. FEEDBACK CONTROL

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 computation of real-time feedback control at \((v, \omega)\) is a straightforward interpolation with a minimum computational load. The solution of the HJB equation in the previous section is integrated with a MPC to stabilize the system at a desired optimal attitude. In the simulations we adopt a basic zero-order hold MPC controller. The sampling rate is 10 Hz. In each time interval, \(u_{\text{optimal}}\) is computed using interpolation on the sparse grid. It is assumed that the measurement of \(v\) and \(\omega\) is corrupted by a random noise of uniform distribution with a magnitude about 0.5% of the maximum state value. Several trajectories under the closed-loop control are shown in Figures 4 - 6.

VI. CONCLUSIONS AND FUTURE WORKS

A. Conclusions

The causality free method used in this paper has perfect parallelism. It can be easily integrated with any grid, specifically sparse grid in this paper to mitigate the curse of dimensionality. The example shows that the algorithm is tractable for a 6-D HJB equation. In contrast to dense grids, the size of sparse grids is significantly smaller. In the case of an uncontrollable rigid body with six state variables, a solution is achieved on a sparse grid with about \(4.5 \times 10^5\) points, whereas the corresponding dense grid has more than \(10^{12}\) points. The solution of the HJB equation is integrated into a MPC controller. Simulations show that the uncontrollable system is stabilized at an optimal attitude in the manifold of reachable states.

B. Future Works

In exchange for a tractable algorithm of solving high dimensional HJB equations, the accuracy of the solution is reduced by a factor of \((\log N)^{d-1}\). In the future work, error analysis will be carried out to analyze and estimate the error in causality free methods. Also included in the future research is to improve the accuracy and to test the method using various different problems.

REFERENCES

[1] Al’Brekhht, E. G., On the Optimal Stabilization of Nonlinear Systems, *Journal of Applied Mathematics and Mechanics*, Vol. 25, No. 5, 1961, pp. 1254–1266.
Fig. 6. Closed-loop trajectories (Line: $\phi, \omega_1, u_1$; dash: $\theta, \omega_1, u_2$; dot: $\psi, \omega_2$)

[2] V. Barthelmann, E. Novak, and K. Ritter, High dimensional polynomial interpolation on sparse grids, *Adv. Comput. Math.*, 12(4):273-288, 2000.

[3] O. Bokanowski, J. Garcke, M. Griebel, I. Klompmaker, An adaptive sparse grid semi-Lagrangian scheme for first order Hamilton-Jacobi Bellman equations, *Journal of Scientific Computing*, Volume 55, Issue 3, 2003, pp 575-605.

[4] H.-J. Bungartz and M. Griebel, Sparse grids, *Acta Numerica*, 2004, pp 1-123.

[5] C. I. Byrnes and A. Isidori, A., On the attitude stabilization of a rigid spacecraft, *Automatica*, 27(1), 1991, pp 87-95.

[6] 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.

[7] P. E. Crouch, Spacecraft attitude control and stabilization: applications of geometric control theory to rigid body models, *IEEE Transactions on Automatic Control*, 29(4), 1984, pp 321-331.

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

[9] L. C. Evans, *Partial Differential Equations*, Graduate Studies in Mathematics Vol. 19, American Mathematical Society, 1 edition, 1998.

[10] J. Garcke, Sparse grid tutorial, preprint, 2011.

[11] H. Gui, L. Jin, S. Xu, Attitude maneuver control of a two-wheeled spacecraft with bounded wheel speeds, *Acta Astronautica*, 88, 2013, pp 98-107.

[12] N. M. Horri, S. Hodgart, Attitude stabilization of an underactuated satellite using two wheels, *IEEEAC paper # 1005*, November, 2002.

[13] N. Horri and P. Palmer, Practical implementation of attitude-control algorithms for an underactuated satellite, *Journal of Guidance, Control, and Dynamics*, 35(1), 2012, 40-50.

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

[15] W. Kang and L. Wilcox, A Causality Free Computational Method for HJB Equations with Application to Rigid Body Satellites, *Proceedings of AIAA GNC*, January, 2004, AIAA 2015-2009.

[16] J. Kierzenka, L. F. Shampine, A BVP solver that controls residue and error, Journal of Numerical Analysis, *Industrial and Applied Mathematics*, Vol. 3, No. 1-2, 2008, pp 27-41.

[17] S. Kim and Y. Kim, Spin-axis stabilization of a rigid spacecraft using two reaction wheels, *Journal of Guidance, Control, and Dynamics*, 24(5), 2001, pp 1046-1049.

[18] W. A. Klimke, Uncertainty Modeling using Fuzzy Arithmetic and Sparse Grids, *Dissertation, University of Stuttgart*, 2005.

[19] H. Krishnan, N. H. McClamroch, M. Reyhanoglu, Attitude stabilization of a rigid spacecraft using two momentum wheel actuators, *Journal of Guidance, Control, and Dynamics*, 18(2), 1995, pp 256-263.

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

[21] 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.

[22] S. A. Smolyak, Quadrature and interpolation formulas for tensor products of certain classes of functions, *Soviet Math. Dokl.*, 4:240-243, 1963.

[23] F. Terui, S. Kawamoto, T. Fujiiwara, A. Noda, N. Sako, S. Nakasuka, Target tracking attitude maneuver of a bias momentum micro satellite using two wheels, *AIAA Guidance, Navigation, and Control Conference*, Denver, CO, August, 2000.

[24] P. Tsai and J. Luo, Control of underactuated spacecraft with bounded inputs, *Automatica*, 36, 2000, pp 1153-1169.

[25] C. Zenger, Sparse Grids, in *Parallel Algorithms for Partial Differential Equations*, W. Hackbusch eds., Notes on Num. Fluid Mech, Vol 31, 1991, pp. 241-251.