International Journal of
Applied & Experimental Mathematics
Research Article Open Access

Derivative Enhanced Optimal Feedback Control Using Computational Differentiation

Ahmad Bani Younes*, **James D. Turner**
Aerospace Engineering Department, Khalifa University, PO Box 127788, Abu Dhabi, United Arab Emirates

Abstract

Feedback control is a powerful methodology for handling model and parameter uncertainty in real-world applications. Given a useful nominal plant model for developing the control approach, it is well-known that optimal solutions only perform well for a limited range of model and parameter uncertainty. A higher-order optimal nonlinear feedback control strategy is presented where the feedback control is augmented with feedback gain sensitivity partial derivatives for handling model uncertainties. A computational differentiation (CD) toolbox is used for automatically generating high-order partial derivatives for the feedback gain differential equations. An estimator is assumed to be available for predicting the model parameter changes. The optimal gain is computed as a Taylor series expansion, where the feedback gains are expanded as a function of the system model parameters. Derivative enhanced optimal feedback control is shown to be robust to large changes in the model parameters. Numerical examples are presented that demonstrate the effectiveness of the proposed methodology.

Publication History:

Received: September 22, 2016
Accepted: December 27, 2016
Published: December 29, 2016

Keywords:

Feedback control,Computational differentiation, OCEA

Introduction

Feedback control is a powerful methodology for handling model and parameter uncertainty in real-world applications. The calculations required for developing optimal solutions, however, demand significant analyst time and computational resources. Given a useful nominal plant model for developing the control approach, it is well-known that optimal solutions only perform well for a limited range of model and parameter uncertainty before the system response degrades and the resulting control objectives are not met. The classical approach is to re-compute the optimal gains each time large changes arise in either the model or parameter values, or develop a gain-scheduling strategy. This paper addresses this limitation by augmenting the development of feedback control solution with sensitivity calculations that allow the feedback control to be generalized by replacing the classical feedback gain with a truncated Taylor series that accounts for the model and parameter changes.

In science and engineering, the system equations of motion are described by the following first order vector differential equation:

$$\dot{x} = f\left(x, p; t\right); \quad givenx\left(t_0\right) = x_0$$

Where $x \in R^N$ denotes the state vector, t denotes the state vector, t denotes time, x_0 denotes the initial condition, and $\dot{x} = \frac{dx}{dt}$. The parameter variations are assumed to be time-varying (i.e. $P= P$ (t). For a given initial condition and well-defined parameters, each trajectory is unique. As a result, in the real-world, one must address the problem that model parameters may only be approximately known. Classically, this problem is handled by re-computing the solution each time large changes arise in either the model or parameter values. We seek more globally robust methods. After formulating our Taylor series -based feedback control strategy, several examples are presented that demonstrate the significantly expanded domain convergence obtained for the proposed methodology. Only a single control calculation is required. Potential sources of model uncertainty include, but are not limited to: expendable fuels, articulated moving parts (i.e., pointing subsystems, rotating machinery, scanning systems, etc.), time-varying stiffness and damping behaviors, changing moments of inertia, robotic manipulations of external objects, environmental effects, etc. To this end, a key step in the control design process is concerned with establishing the expected range in the model and parameter uncertainties. To handle the expected range of model and parameter variation, a sensitivity based control design is developed where high-order partial derivatives are computed for the feedback control gains. By combining the classical gains and the sensitivity partial derivatives a new high-order optimal nonlinear feedback control strategy is presented where the feedback control is augmented with feedback gain sensitivity partial derivatives. A generalized gain matrix and disturbance rejection control formulation is presented. Derivative enhanced optimal feedback control laws are shown to be robust to large changes in the model parameters. The computational differentiation (CD) toolbox automatically generates the higher-order partial derivatives for the feedback gain differential equations. An estimator is assumed to be available for predicting the model parameter changes. The optimal gain is computed as a Taylor series expansion in the gains, where the feedback gains are expanded as a function of the system model parameters. The pre-calculation of the sensitivity gains eliminates the need for gain scheduling for handling model parameter changes. Higher-Order feedback gain sensitivity calculations are applied on the full nonlinear model using computational differentiation tools.

Literature Review

Several Authors have attempted generalizations of feedback control where the co-state variable is expanded as power series in the state variable. Recently, Majji et al. [1,2] have explored the developments of this theory to high-order. Malanowski and Maurer [3,4] introduced a first-order sensitivity method to investigative the parametric variation of solutions to constrained optimal control. Later work considers theoretical issues such as regularity. Pinho and Rosenblueth [5] utilize

***Corresponding Author:** Dr. Ahmad Bani Younes, Aerospace Engineering Department, Khalifa University, PO Box 127788, Abu Dhabi, United Arab Emirates, E-mail: ahmad.younes@kustar.ac.ae

Citation: Younes AB, TurnerJD (2016) Derivative Enhanced Optimal Feedback Control Using Computational Differentiation. Int J Appl Exp Math 1: 112. doi: http://dx.doi.org/10.15344/ijaem/2016/112

the implicit function theorem to transform a constrained optimal control problem into an unconstrained form and propose a solution approach. Recent work by McCrate [6] utilized higher-order differentiations of the Hamilton-Jacobi-Bellman (HJB) equation to solve nonlinear optimal control problems and their sensitivities. The success of these earlier approaches motivates our investigation into developing high order sensitivity models. Carrington and Junkins [7, 8] expanded the co-state variable as a power series in the state variable. This work builds on the foundational earlier works of Bani Younes et al. [9-11] by formulating a nonlinear tracking problem, where the co-state variable is expanded as a power series in the tracking error relative to a reference trajectory. A further generalization has been considered where the control strategy has been further enhanced by augmenting the control gains with sensitivity partial derivatives for the system parameters and model errors [12-14]. This approach, however, proved to be very cumbersome to implement because the gain sensitivity calculations require very complicated array algebra [15-17] calculations for high-order gains.

The key contribution of the paper is the demonstration that the limitations of these earlier efforts are shown to be easily overcome by replacing the gain calculations with a straightforward Taylor series, where computational differentiation tools handle all of the complicated sensitivity calculations in a hidden way, where no array algebra calculations are required. Numerical experiments are conducted to establish the required number of control partial derivatives. In real world applications, CD generates all numerical partial derivatives required for the feedback control design process. This approach trades computer memory for storing the pre-computed gain gradients for on-the-fly re-computation of the control gains. It is anticipated that by pre-computing sensitivity gains that many real-time applications can be handled that are experiencing large time-varying plant changes. All parameters being used in the feedback control formulation are included in the state estimation routines. In a very general setting, it is anticipated that even articulated sub-system motions are handled by accounting for rotational motion and the associated mass, stiffness, and damping property impacts. The utility of these ideas are topics of continuing future research, where trade-offs are performed between gain-scheduling approaches versus the sensitivity-based approach considered here.

Computational Differentiation

Computational differentiation is a specialized topic in applied mathematics and computer science for developing and fielding software tools for numerically evaluating partial derivative models [18,19] Computational differentiation has existed since the 1960s starting with the seminal works of Wengert [20] and Wilkins [21] and developed by many others [18-23]. Early approaches used an existing coded math model as a template for applying the rules of calculus to write a new code that implements the partial derivatives for the math model. This approach has proven to be very effective for generating first-order sensitivity models. Alternatively, this paper presents higher-order derivative models that are generated by using operator overloading methods, where the transformations are implicitly handled by the programming language, when the compiler detects a derivative enhanced data type. Turner's Object-Oriented Coordinate Embedding Algorithm (OCEA) [1, 9, 12-14, 24- 26] CD software tool builds all of the partial derivative models. OCEA acts as a Language extension for FORTAN 95/2003 when it is linked to application codes. This paper makes use of Turner's (OCEA) program for computing 1st-4th-order mixed partial derivative models [1, 12, 27, 28]. OCEA is a CD tool for computing arbitrarily complex partial derivative models. At compile time, OCEA uses the programmer's math model as a template for deriving, coding, and generating an executable for simultaneously compiling the simulation and sensitivity models. No symbolic or finite difference tools are used for any of the gradient tensor calculations; all results are immediately processed to produce numerical results for all partial derivative orders. Each time a program is compiled OCEA derives, assemblies, and codes the partial derivative model in the program executable, yielding numerical results that are accurate to the working level of precision for the machine. This approach benefits the user in four ways: (1) only the basic math model is programmed and checked out; (2) no analyst effort is required to either derive or code sensitivity models; (3) the user recaptures the development time normally required for developing sensitivity math models, coding, and validating nonlinear and high-order models; and (4) most importantly, math model changes are automatically handled each time the code is complied. A CD-based approach is very attractive because hand derived models are very vulnerable to model changes that can potentially force a new derivation and coding or the sensitivity model. OCEA transforms all math and intrinsic functions to embed multiple levels of the chain rule of calculus for building partial derivative models. OCEA consists of a suite of programs/modules that process the user supplied software math models as a template for generating high-order sensitivity models. Minimal modifications of the user existing software models are required for enabling derivative-enhanced calculations. Users identify independent variables for the derivative calculations and define the variables for which partial derivatives are required (e.g., typically one is required to convert real variable(s) to an OCEA variable TYPE(EB), which instructs the complier that partial derivatives are computed for the TYPE(EB) variable(s). The software accepts user math models coded in standard FORTRAN 95/2003 language constructs. At compile time the compiler itself assembles the partial derivative solution by identifying variable data types, intrinsic/lib functions, and inserting Function/Subroutine calls in the compiler generated executable for computing the associated partial derivatives. From the user perspective, OCEA behaves as a language extension for FORTRAN 95/2003: its hidden operations exploit operator overloading and user-defined data types for handling all memory management details and derivative enhanced operations. It should not be surprising that OCEA introduces a computational overhead when compared to highly optimized hand coded partial derivative models. The tradeoff is this: if the basic math model requires X man-months to develop, then the derivation, coding, and validation of the sensitivity model can add 5X-10X man-months to the project development effort. Since the computer time required for deriving and compiling the sensitivity solution is measured in seconds vs. 5X-10X man-months for an analyst, OCEA's impact is both clear and unambiguous for developing and solving real-world projects subject to man-power resource constraints. OCEA is particularly valuable in the normally fluid engineering design environment, where frequent design changes and -what if- experiments must be carried out to fully explore the opportunities available in the notional system design space. The analyst always has the most up to date model. This is in stark contrast the case of a hand-derived model where even seemingly simple model changes can devastate all previous derivation efforts; thereby, forcing a restart for the derivation and coding effort from scratch for each new sensitivity model is a daunting unwelcome task. OCEA's derivative enhanced variables are defined as abstract compound data objects, where objects such as the variable F, defined below, consist of the following list of concatenated data in computer

Citation: Younes AB, TurnerJD (2016) Derivative Enhanced Optimal Feedback Control Using Computational Differentiation. Int J Appl Exp Math 1: 112. doi: http://dx.doi.org/10.15344/ijaem/2016/112

memory: $F := \{F, \nabla F, \nabla^2 F, ..., \nabla^m F\}$ where $\nabla^m F$ denotes the *m*-th order tensor gradient operation. The only variable visible to the analyst is F. Numerical values for the sub-object component values of the tensor gradient operators are obtained by using structure constructor designators $(e.g., \nabla^m F = F\%T_m, donates\ the\ tensor\ order)$. A detailed description of all of OCEA's capabilities is found in the software user manual [29].

Mathematical Model and Applications

Optimal control methodologies are presented for supporting a robust sensitivity-enhanced feedback control approach. The optimal trajectory is obtained by minimizing the following quadratic performance index [16].

$$\Im = \frac{1}{2}\int_{t_0}^{t_f}\left(x^T Q x + u^T R u\right)dt$$

Subject to $\dot{x} = Ax + Bu$

where *Q* and *R* are weight matrices, *A* is the plant matrix, *B* is the control matrix, $x \in R^N$ is the state vector and is the control input. Invoking the standard necessary condition for optimality [7, 8, 16] after some algebraic manipulation, the stabilizing control is given by

$$u(t) = -R^{-1}B^T S(t)x(t)$$

where the feedback gain matrix *S*(*t*) is obtained by numerically integrating the Riccati matrix differential equation:

By assuming the S, A, B, R, Q are derivative enhanced, one simply computes Eq. (5), where OCEA builds

$$S := \{S, \nabla S, \nabla^2 S, ..., \nabla^n S\}$$

This equation is numerically integrated to provide the desired gain solutions. The sensitivity enhanced optimal control is implemented as feedback control

$$u(t) = -R^{-1}B^T\left\{S + \sum_{n=1}\frac{1}{n!}\nabla^n S \bullet (\delta p)^n\right\}x(t)$$

Where δp denotes an *n*-th order tensor-based product for the parameters variations.

Results and Discussion

Consider the optimal control problem of minimizing the following performance index:

$$\Im = \frac{1}{2}\int_{t_0}^{t_f}\left(x^2 + u^2\right)dt$$

subject to the linear model $\dot{x} = ax + u$

Let us assume that the model parameter α is poorly approximated with huge uncertainty δα~±5%.

For nominal solution, the parameter is chosen to be equal $\alpha^* = 1$. The feedback control is designed for handling model and parameter uncertainty. It is well-known that the stabilizing control for this problem is given by $u(t) = -s(\alpha,t)x(t)$, where the time varying state feedback gain $s(\alpha,t)$ is calculated by solving the Riccati differential equation backwards in time.

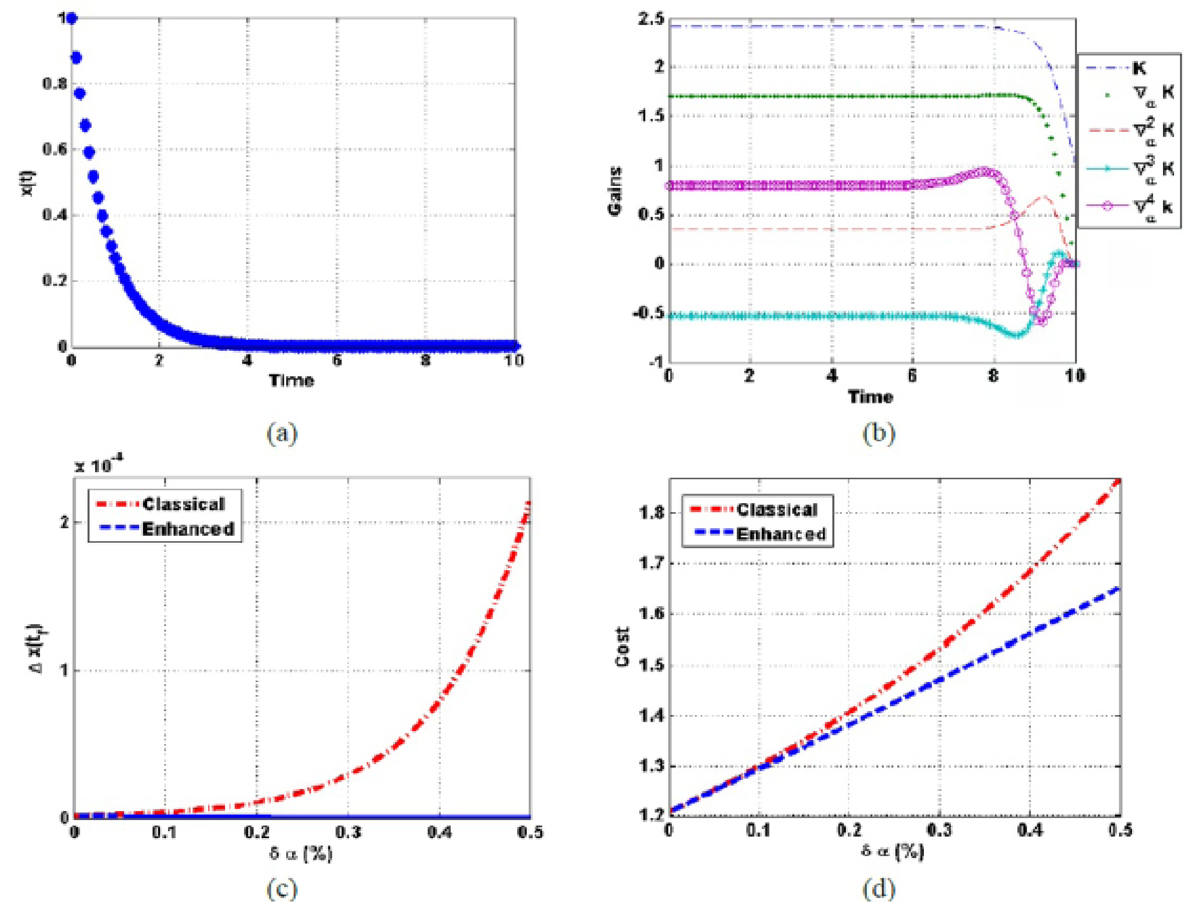

Figure 1: Enhanced Feedback Solutions for uncertain linear plant.

Citation: Younes AB, TurnerJD (2016) Derivative Enhanced Optimal Feedback Control Using Computational Differentiation. Int J Appl Exp Math 1: 112. doi: http://dx.doi.org/10.15344/ijaem/2016/112

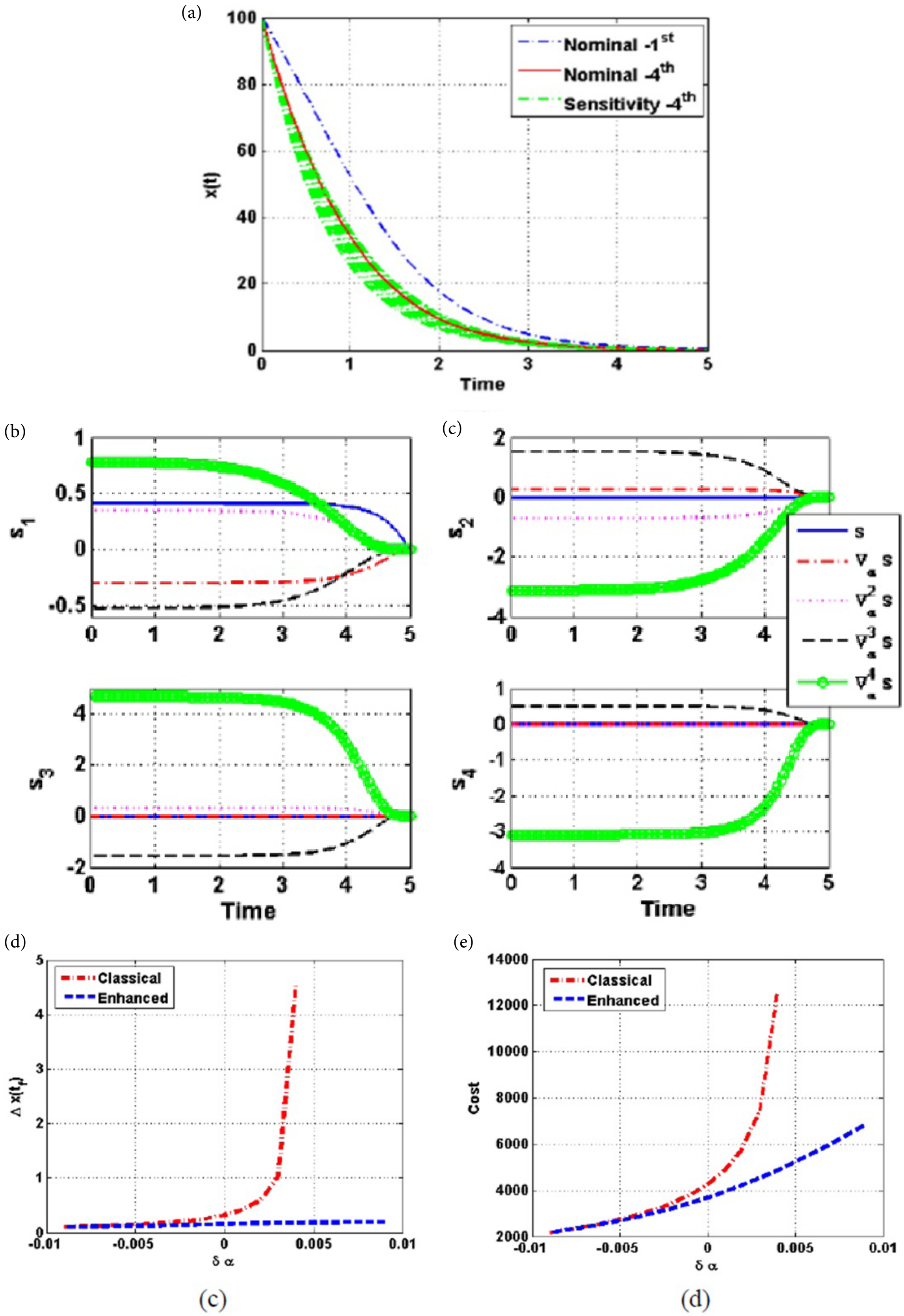

Figure 2: Enhanced Feedback Solutions for uncertain nonlinear.

$$\dot{S} = -2\alpha s + S^2 - 1$$

Even for this simple problem, it is noteworthy that even a small change in the plant parameter (say $\alpha = \alpha + \delta\alpha$) Forces a recalculation of the state feedback gains for maintaining the desired controlled system response. Mathematically, parameter sensitivity is handled by developing Taylor series expansions for the control gains about prescribed nominal values, leading to

$$u(t) = -\left\{ s + \sum_{n=1} \frac{1}{n!} \nabla^n_{\alpha} s(\delta\alpha)^n \right\} x(t)$$

The time histories for the state and gain sensitivities are shown in Figure 1 (a and b respectively). Figure 1(a) presents the nominal solution of the system, where $\alpha^* = 1$. The gain partial derivatives are generated automatically by OCEA during the backward integration of the Riccati gains. It is obvious that the gain sensitivities approach steady state at initial time. Figure 1 (c and d) shows parameter variations where the derivative enhanced control easily handles this challenging range in model uncertainty. Particularly, Figure 1c which demonstrates that the terminal error for the fixed time control problem is virtually not impacted by the large parameter variations even though the classical control approach experiences quadratic growth in the observed error. Figure 1d demonstrates similar behavior for the classical (nominal) and the enhanced solutions in terms of the variations in the performance index. This suggests that the new control gain approach is much more robust to modeling errors that classical control designs.

In the next example, we augment the state equation, given by Eq.(9), to become nonlinear

$$\dot{x} = -(1+\alpha)x + \alpha x^2 + u$$

The system nonlinearity is handled by assuming that the co-state is expressed as following Taylor series

$$\lambda = \sum_{j=1} s_j x^j$$

where the control gains $\{S_1, S_2, S_3, \ldots, S_j\}$ must be recovered. The new co-state terms are for handling the plant model nonlinearities. For this particular example, fourth order co-state model is used. Similarly, parameter sensitivity is handled by developing Taylor series expansions for the control gains $\{S_1, S_2, S_3, S_4\}$ about prescribed nominal values, leading to

$$u(t) = -\sum_{j=1} \left\{ s_j + \sum_{n=1} \text{—} \nabla^n \ s_j(\delta\alpha) \right\} x^j(t)$$

In the proposed scheme, the feedback gain solution is stored for the nominal gain, as well as the first-through-fourth partial derivatives. Those partials are generated automatically by OCEA, see Figure 2b. This allows a large family of nearest neighbor perturbations in the parameter to be handled by a single preliminary calculation. Figure 2a exhibits the sensitivity in the approximate/calculated solution as a time history. It is obvious that the perturbations in da are well-handled at the boundary conditions. Figure 2 (c and d) show that the approximate/calculated final state solution and cost are valid for larger da values when compared to the nominal solution.

Conclusions

It is well known that the closed loop system dynamics is highly sensitive to plant parameters. Generalizations of classical control results are presented in this paper that mitigates the sensitivity of plant models to both parameter and state nonlinearities. Computational differentiation is introduced as a methodology for automatically generating the required partial derivatives for implementing Taylor series-based generalization of classical feedback control results. Several examples are presented that demonstrate the effectiveness and robustness of the Taylor series-based approach for the feedback gains for nonlinear applications. The derivative enhanced feedback gain solutions are shown to handle large parameter variations with little impact on the resulting terminal errors for the problem. These promising results suggest that the proposed new control strategies can have a significant impact on addressing complex real world applications where parameter uncertainty and model nonlinear effects are present.

References

1. Junkins JL, Turner JD, Majji M (2009) Generalizations and applications of the lagrange implicit function theorem. J Astronaut Sci 57: 313-345.
2. Turner J, Majji M, Junkins J (2008) High order state and parameter transition tensor calculations. AIAA/AAS Astrodynamics Specialist Conf.
3. Malanowski K, Maurer H (1996) Sensitivity analysis for parametric control problems with control-state constraints. Comput Optim App 5: 253-283.
4. Malanowski K, Maurer H (2001 Sensitivity analysis for optimal control problems subject to higher order state constraints. Ann Oper Res 101: 43-73.
5. De Pinho MdR, Rosenblueth JF (2006) Mixed constraints in optimal control: an implicit function theorem approach. IMA J Math Control Inf.
6. McCrate CM (2010) Ms thesis: High order methods for determining optimal control and their sensitivities. PhD thesis, Texas A&M University, Department of Aerospace Engineering.
7. Carrington CK, Junkins JL (1984) Nonlinear Feedback Control for Spacecraft Slew Maneuvers. J Astron Sci 32.
8. Carrington CK, Junkins JL (1986) Optimal Nonlinear Feedback Control for Spacecraft Attitude Maneuvers. J Guid Control Dyn 9: 99-107.
9. Bani Younes A, Turner J, Majji M, Junkins J (2010) An investigation of state feedback gain sensitivity calculations. AIAA/AAS Astrodynamics Specialist Conf.
10. Bani Younes A, Turner J, Majji M, Junkins J (2011) Nonlinear tracking control of maneuvering rigid spacecraft. 21st AAS/AIAA Space Flight Mechanics Meeting.
11. Bani Younes A, Turner J, Majji M, Junkins J (2012) High-order state feedback gain sensitivity calculations using computational differentiation. Jer-Nan Juang Astrodyn Sym.
12. Turner JD (2003) Automated generation of high-order partial derivative models. 1590-1599.
13. Bani Younes A, Turner JD (2015) Generalized Least Squares and Newton's Method Algorithms for Nonlinear Root-Solving Applications. AAS J Astron Sci 60: 517-540.
14. Griffith DT, Turner JD, Junkins JL (2005) Automatic generation and integration of equation of motion for flexible multibody dynamical systems. AAS J Astron Sci 53: 251-279.
15. Graham A (1981) Kronecker products and matrix calculus: with applications. Ellis Horwood series in mathematics and its applications. Horwood.
16. Lewis F, Syrmos V (1995) Optimal Control. A Wiley-Interscience publication.
17. Blaha G (1977) A few basic principles and techniques of array algebra. Bulletin godsique 51: 177-202.
18. Bischof CH, Carle A, Hovland PD, Khademi P, Mauer A (1998) ADIFOR 2.0 user's guide (Revision D). Tech. rep., Mathematics and Computer Science Division Technical Memorandum no. 192 and Center for Research on Parallel Computation Technical Report CRPC-95516-S.
19. Griewank A (1989) On automatic differentiation. In Mathematical Programming. Iri M Tanabe K (eds) Kluwer Academic Publishers 34: 83-108.

20. Wengert RE (1964) A simple automatic derivative evaluation program. 463-464.

21. Wilkins RD (1964) Investigation of a new analytical method for numerical derivative evaluation. Commun ACM 7: 465-471.

22. Bischof C, Eberhard P (1996) Automatic differentiation of numerical integration algorithms. Tech Rep ANL/MCS-P621-1196, Mathematics and computer Science Division, Argonne National Laboratory.

23. Bischof C, Carle A, Corliss G, Griewank A, Hovland P (1992) Adifor: Generating derivative codes from fortran programs. Sci Program 1-29.

24. Griffith DT, Turner JD, Junkins JL (2004) An embedded function tool for modeling and simulating estimation problems in aerospace engineering. AAS/AIAA Spaceflight Mechanics Meeting.

25. Majji M, Junkins J, Turner J (2009) A perturbation method for estimation of dynamic systems. Nonlinear Dyn.

26. Griffith DJ, Turner JD, Junkins JL (2005) Some applications of automatic differentiation to rigid, flexible, and constrained multibody dynamics. Proceedings of the 5th International Conference on Multibody Systems, Nonlinear Dynamics and Control.

27. Pett (1996) Macsyma, Symbolic/ numeric/graphical mathematics software: Mathematics and System Reference Manual. 16th ed.

28. Turner JD, Majji M, Junkins JL (2010) Keynote paper: Fifth-order exact analytic continuation numerical integration algorithm. Proceed Int Conf Comput Exp Eng Sci.

29. Turner J (2006) OCEA User Manual. Amdyn System.