2015-d-43

SPARTAN: An Improved Global Pseudospectral Algorithm for High-fidelity Entry-Descent-Landing Guidance Analysis

By Laurens HUNEKER[1)], Marco SAGLIANO[2)] and Yunus ARSLANTAS[2)]

[1)] *Astrodynamics and Space Missions Department, Technical University Delft, Delft, The Netherlands*

[2)] *Guidance, Navigation and Control Department, German Aerospace Center (DLR), Bremen, Germany*

Usage of numerical optimization for generation and analysis of optimal trajectories has rapidly increased over the years. The approach presented here allows for the generation of numerical guidance schemes, which are not possible with analytical methods. SPARTAN (Shefex-3 Pseudospectral Algorithm for Re-entry Trajectory Analysis) is a tool based on the use of the global Pseudospectral methods for the transcription of optimal control problems. This method has several advantages. It removes the Runge phenomenon and it has an exponential convergence to the corresponding Optimal Control Problem (OCP) that can be numerically validated when it is applied to smooth problems. The initial infinite dimensional OCP is transcribed into a Non-Linear Programming (NLP) problem by a proper collocation of the differential equations and the constraints for the related problem. The resulting NLP can then be solved by off-the-shelf solvers. Two realistic example problems are given to demonstrate the performance of SPARTAN, and optimal trajectories were generated for both problems.

Key Words: Pseudospectral, Entry-Descent-Landing, Optimal Control Problem

Nomenclature

a	:	Constraints
b	:	Stepsize
c	:	crossrange
d	:	Downrange
D	:	Drag
g	:	Gravity
h	:	Height
k	:	Material heating coefficient
J	:	Cost function
K	:	Lagrangian
L	:	Lift
m	:	Mass
M	:	Mach number
n	:	Load factor
P	:	Legendre polynomial
q	:	Dynamic pressure
Q	:	Heating rate
R	:	Radau quadrature
S	:	Reference surface area
t	:	Time
T	:	Thrust
u	:	Control
x	:	State
α	:	Angle of attack
β	:	Pitch angle
θ	:	Inversed atmospheric scale height
γ	:	Flight path angle
λ	:	Geocentric latitude
μ	:	Geocentric longitude
ξ	:	Heading angle
ρ	:	Atmospheric density
σ	:	Bank angle
η	:	specific propellant consumption
τ	:	Nodal values
χ	:	Yaw angle
ω	:	Roll rate
Ω	:	Earth's angular velocity
Subscripts		
D	:	Drag
l	:	Lower
L	:	Lift
m	:	Main thruster
RCS	:	Reaction control system
u	:	Upper
0	:	Initial state
1	:	Thrust in downrange direction
2	:	Thrust in crossrange direction
3	:	Thrust in height direction

1. Introduction

Pseudospectral methods (PSM) are gaining more and more popularity in the aerospace community due to their easiness of application to solve smooth problems. The idea behind PSM is that the differential equations, the cost function and the constraints are collocated at a defined set of nodes obtained from the roots of linear combination of Legendre polynomials. The set of collocation points used are indeed the roots of the Legendre-Gauss-Radau as proposed in 1). Pseudospectral methods allow to solve OCP's numerically by converting the associated problem into a Non-Linear Programming (NLP) problem. Another property of the usage of pseudospectral methods is the removal of the Runge phenomenon. SPARTAN is a tool created to obtain optimal trajectories for the SHEFEX-3 mission as done in 2). The tool has been developed to solve different set of problems and the results are repeated in 3), 4), and 5). SPARTAN uses two different ”off-the-shelf” solvers to obtain a solution for the finite dimensional NLP. The primary solver is SNOPT, a sparse solver for constrained optimization problems.[6)] Another tool is IPOPT (Interior Point OPTimizer).[7)]

SPARTAN is developed such a way that the two aforementioned solver packages could be used for computing optimal trajectories. This paper discusses mathematical background and application of SPARTAN to two problems. The paper is organized as follows. The transcription of a generic problem is given first in Section 2. The Jacobian and different contributors are explained in Section 3. Two example problems, the X-33 and a generic moon lander problem are introduced in Section 4. The validation of this program is found in Section 5. The conclusion, including a short summary, is included in Section 6.

2. Transcription of the problem

Consider a continuous system with the following dynamics

$$\dot{x} = f(x(t), u(t), t), x \in \mathbb{R}^n \text{ and } u \in \mathbb{R}^m \tag{1}$$

where x are the states, u is the control and t is the time and suppose that we have a cost function with the form

$$J = \phi(x(t_f)) + \int_{t_0}^{t_f} K(x(t), u(t), t)dt \tag{2}$$

where J is the cost function, ϕ is the Mayer term related to the final states and K is the Lagrange term. The states and the controls are bounded by

$$x_l \leq x(t) \leq x_u \tag{3}$$

$$u_l \leq u(t) \leq u_u \tag{4}$$

The constraints g, associated with the states and controls, are defined in the following manner

$$a_l \leq a(x, t, u) \leq a_u \tag{5}$$

The goal is to minimize the cost Eq. (1) while satisfying the relationships of Eqs. (2)-(5). Multiple methods exist for solving NLP for the OCP defined above. Pseudospectral methods are a class of methods that are very powerful when it comes to approximating the solution of a NLP. The method used here is called Legendre-Gauss-Radau (LGR). In this method, the continuous functions are approximated by collocating them using discretization nodes (also called collocation points). The cost function, equations for system dynamics, and the constraints are collocated into these nodes and they are subsequently treated as a set of nonlinear algebraic constraints. In other words, the infinite dimensional OCP is converted into finite dimensional NLP. Pseudospectral methods transform the physical domain of a variable (time t, in this case) into a normalized independent variable τ in the interval [−1,1] instead of [t_0,t_f]. It is possible to transform back and forth between the independent variable τ and t with the following affine transformations

$$t = \frac{t_f - t_0}{2}\tau + \frac{t_f + t_0}{2} \tag{6}$$

$$\tau = \frac{2}{t_f - t_0}t - \frac{t_f + t_0}{t_f - t_0} \tag{7}$$

The scaling of the time is reverted at the end of all the calculations. The states and controls that are converted to discrete points are described with:

$$x(t_i) \cong X_i, i \in [0, N] \tag{8}$$

$$u(t_j) \cong U_j, j \in [1, N] \tag{9}$$

These points substitute the continuous functions at the discretization nodes. The location of the nodes are calculated with the Radau Quadrature and the Newton-Raphson method which comes with a slight modification from 9). Given by

$$(\tau_n)_{i+1} = (\tau_n)_i - \frac{R_{n-1}(\tau_n)}{R'_{n-1}(\tau_n)} \tag{10}$$

with

$$R_{n-1}(\tau_n) = \frac{P_{n-1}(\tau_n) + P_n(\tau_n)}{1 + \tau_n} \tag{11}$$

$$R'_{n-1}(\tau_n) = \frac{2n}{1 - \tau_n^2} P_{n-1}(\tau_n) \tag{12}$$

where P_n represents the Legendre polynomial of order n.

$$P_i(\tau) = \prod_{j=0, j\neq i}^{N} \left(\frac{\tau - \tau_j}{\tau_i - \tau_j}\right) \tag{13}$$

Dividing Eqs. 11-12 and using $P_{(n-1)}(\tau_n) = -P_n(\tau_n)$ as used with the Radau Pseudospectral Method (RPM) as seen in 10) results in

$$(\tau_n)_{i+1} = (\tau_n)_i - \left(\frac{1 - \tau_n}{2}\right) \frac{P_{n-1}(\tau_n) + P_n(\tau_n)}{P_{n-1}(\tau_n) - P_n(\tau_n)} \tag{14}$$

The nodal values $\boldsymbol{\tau}$ is obtained until $|\boldsymbol{\tau}_{(i+1)} - \boldsymbol{\tau}_i|$ reaches machine zero. The initial guess $\boldsymbol{\tau}_0$ for this method is provided by using the Chebyshev Polynomials. [11] Initial guess using the Chebyshev polynomials is calculated with

$$\boldsymbol{\tau}_0 = -\cos\left(\frac{2\pi(n_j - 1)}{2(n_j - 1) + 1}\right) \tag{15}$$

where n_j is the number of the root, with $j \in [0, \ldots, N - 1]$, with N_0 being 0. A stable node-placement composition of a higher number of nodes can be achieved in this way. 600 nodes were obtained in a numerically stable sense with computations in MATLAB.

3. Structure of the Jacobian

The quality of the results and the computation time are strongly affected by the Jacobian generated from the transcription of the OCPs. This Jacobian is sparse by nature due to the non-dependency of the different states with one another when considering the dynamical equations of the problem. Taking advantage of this inherent sparseness can reduce the computational time. [10] The Jacobian can be expressed as the sum of three parts

$$Jac = Jac_{Ps} + Jac_{Nu} + Jac_{Th} \tag{16}$$

where the pseudo-spectral part Jac_{Ps} is inherent to the use of the Flipped Radau Pseudospectral method (FRPM), and describes the differential relationship between the different nodes. [10] The numerical Jacobian Jac_{Nu} is computed using the complex-step method or the dual-step. The complex-step approximates the differential and it has the following form, [12]

$$f'(x) = \frac{im[f(x + ib)]}{b} \tag{17}$$

The dual-step method has an exact solution, [13] and it makes use of the dual-class method to compute first derivatives

$$f'(x) = \frac{dual[f(x + \epsilon b)]}{b} \tag{18}$$

Both methods are much more extensively discussed and compared in 8).

The theoretical contribution of the Jacobian is only applied if there is an unknown final time. If the final time is defined as free parameter in the OCP, another column corresponding to the final time is added to the Jacobian. [10]

Poor scaling of the Jacobian and states can generate numerical difficulties during the optimization, even in the presence of a proper transcription method. For that reason, an automatic scaling technique is added. A linear technique called Projected Jacobian Rows Normalization (PJRN) is used. [2] The scaled Jacobian is used by SNOPT or IPOPT to calculate the optimal trajectories while minimizing the cost function. In this paper, solutions only obtained by SNOPT are given. The resulting states and controls are then scaled inversely and the nodes are linked together using Lagrange polynomials. [10]

4. Entry-Descent-Landing Mission Examples

4.1. The X-33

The first example simulates a landing sequence for a generic X-33 re-entry vehicle. [14] Standard models such as the terminal area and a heading alignment cylinder are not used. Instead, the craft is to pass through a flight approach corridor. The simulation environment is discussed first before moving to the mission specifics. The equations of motion used for the X-33 are:

$$\dot{h} = V \sin\gamma \tag{19}$$

$$\dot{\mu} = \frac{V \cos\gamma \cos\eta}{r \cos\lambda} \tag{20}$$

$$\dot{\lambda} = \frac{V \cos\gamma \sin\xi}{r} \tag{21}$$

$$\begin{aligned}\dot{V} =& -\frac{D}{m} - g \sin\gamma + \Omega^2 r \cos\lambda(\sin\gamma\cos\lambda - ...\\ & \cos\gamma \sin\lambda \sin\xi)\end{aligned} \tag{22}$$

$$\begin{aligned}\dot{\gamma} =& \frac{L\cos\sigma}{mV} + \left(\frac{V}{r} - \frac{g}{V}\right)\cos\gamma + ...\\ & 2\Omega\cos\lambda\cos\xi + ...\\ & \frac{\Omega^2 r}{V}\cos\lambda(\cos\gamma\cos\lambda + sin\gamma\sin\lambda\sin\xi)\end{aligned} \tag{23}$$

$$\begin{aligned}\dot{\xi} =& \frac{L\sin\sigma}{mV\cos\gamma} - \frac{V}{r}\cos\gamma\cos\xi\tan\lambda + ...\\ & 2\Omega(\tan\gamma\cos\lambda\sin\xi - ...\\ & \sin\lambda) - \frac{\Omega^2 r}{V\cos\gamma}\sin\lambda\cos\lambda\cos\xi\end{aligned} \tag{24}$$

It can be seen that a standard 3-DOF model is used where h is the height of the plane, μ is the geocentric longitude, λ is the geocentric latitude, V is the airspeed, γ is the flight path angle and ξ is the heading angle, and r is the summation of the altitude and the equatorial radius of the Earth ($r = h + R_e$).The gravity is subsequently calculated with the inverse square law ($g = GM/r^2$) . The two control inputs are the angle of attack α and bank angle σ. The lift and drag are calculated with

Table 1. Constants used in the X-33 re-entry scenario.

m	2455 slugs	Vehicle's empty mass
S	1608 ft^2	Vehicle reference surface area
R_e	20902900 ft	Equatorial radius of the Earth
g_0	32.174 ft/s^2	Surface gravity
Ω	$7.2722 \cdot 10^{-5}$ rad/s	Earth's angular velocity
ρ_0	0.002378 slugs/ft^2	Vehicle's empty mass
GM	$0.14076539 \cdot 10^{-5}$ ft^3/s^2	Earth's gravitational constant
θ	$4.20168 \cdot 10^{-5}$ ft^{-1}	Inverse atmospheric scale height

$$L = \frac{1}{2}\rho(h)V^2 C_L(\alpha, M)S \tag{25}$$

$$D = \frac{1}{2}\rho(h)V^2 C_D(\alpha, M)S \tag{26}$$

The density of the atmosphere is calculated with the exponential model:

$$\rho(h) = \rho_0 e^{-\theta h} \tag{27}$$

The lift- and drag-coefficient are functions of the angle of attack and the Mach number generated by fitting the aerodynamic data of the X-33 with a 2nd order polynomial.

$$\begin{aligned}C_L(\alpha, M) =& -0.0005225\alpha^2 + 0.03506\alpha - ...\\ & 0.04857M + 0.1577\end{aligned} \tag{28}$$

$$\begin{aligned}C_D(\alpha, M) =& 0.0001432\alpha^2 + 0.00558\alpha - ...\\ & 0.01048M + 0.2204\end{aligned} \tag{29}$$

Imperial units are used in order to compare the results with existing literature . [14] Values of the constants mentioned above can be found in Table 1.

For this problem, the following state vector is used.

$$\mathbf{x} = \begin{bmatrix} h, & \mu, & \lambda, & V, & \gamma, & \xi, & \alpha, & \sigma \end{bmatrix}^T \tag{30}$$

The derivatives $\dot{\alpha}$ and $\dot{\sigma}$ are used as control inputs to realistically simulate the movements of the vehicle. The control vector is thus defined as

$$\mathbf{u} = \begin{bmatrix} \dot{\alpha}, & \dot{\sigma} \end{bmatrix}^T \tag{31}$$

The box constraints on the states are defined as:

$$\begin{bmatrix} 0 \text{ ft} \\ -90 \text{ deg} \\ -90 \text{ deg} \\ 1 \text{ ft/s} \\ -89 \text{ deg} \\ -180 \text{ deg} \\ -10 \text{ deg} \\ -80 \text{ deg} \end{bmatrix} \leq \begin{bmatrix} h \\ \mu \\ \lambda \\ V \\ \gamma \\ \xi \\ \alpha \\ \sigma \end{bmatrix} \leq \begin{bmatrix} 400000 \text{ ft} \\ 90 \text{ deg} \\ 90 \text{ deg} \\ 26000 \text{ ft/s} \\ 89 \text{ deg} \\ 180 \text{ deg} \\ 50 \text{ deg} \\ 80 \text{ deg} \end{bmatrix} \tag{32}$$

The angular rates of the control inputs are limited by

$$\begin{bmatrix} -5 \text{ deg/s} \\ -5 \text{ deg/s} \end{bmatrix} \leq \begin{bmatrix} \dot{\alpha} \\ \dot{\sigma} \end{bmatrix} \leq \begin{bmatrix} 5 \text{ deg/s} \\ 5 \text{ deg/s} \end{bmatrix} \tag{33}$$

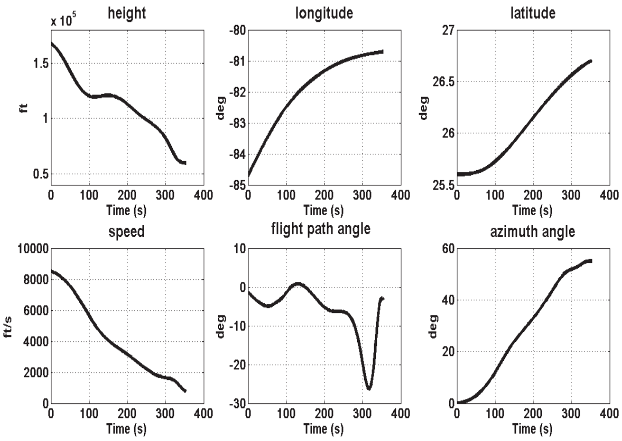

Fig. 1. States of the X-33 during re-entry.

Three different path constraints are imposed on the vehicle. The load factor n_z, the dynamic pressure q, and the heating rate Q, they are calculated with

$$n_z = \frac{|L\cos\alpha + D\sin\alpha|}{m} \tag{34}$$

$$q = \frac{1}{2}\rho(h)V^2 \tag{35}$$

$$Q = k\sqrt{\rho(h)}V^{3.15} \tag{36}$$

The value of the material heating coefficient k is $4.47228 \cdot 10^{-9}$. [14] The three path constraints have the following upper and lower bounds:

$$\begin{bmatrix} -2.5\text{ g's} \\ 0 \\ 0 \end{bmatrix} \leq \begin{bmatrix} n_z \\ q \\ Q \end{bmatrix} \leq \begin{bmatrix} 2.5\text{ g's} \\ 300\text{ lb/ft}^2 \\ 70\text{ BTU/ft-s} \end{bmatrix} \tag{37}$$

The plane should be approaching the runway at the end of the trajectory. The cost function is designed in such a manner that the final location of the trajectory is at Cape Canaveral.

$$J = \sqrt{FAC_r - r_f}^2 + \sqrt{FAC_\lambda - \lambda_f}^2 + \sqrt{FAC_\mu - \mu_f}^2 \tag{38}$$

FAC stands for Flight Approach Corridor. It is an imaginary box that, if properly entered, aligns the vehicle to the runway. [14] The initial conditions are set south-west of Cape Canaveral,

$$\mathbf{x}_0 = \left[167323, -84.7112, 25.6, 8530.2, -1.3, 0, 20, 2\right]^T \tag{39}$$

The X-33 should roughly approach the Kennedy Space Center under the following set of approximated end conditions

$$\mathbf{x}_f = \left[2000, -80.7112, 26.7, 600, -6, 0, 170\right]^T \tag{40}$$

The results of SPARTAN for this particular problem can be seen in Fig. (1), Fig. (2), and Fig. (3). The results proved to be smooth. The final states are successfully reached within 353 seconds with a slow turn and smooth control inputs. The numerical integration of the control inputs by the Runge-Kutta are inline with the solution from SPARTAN.

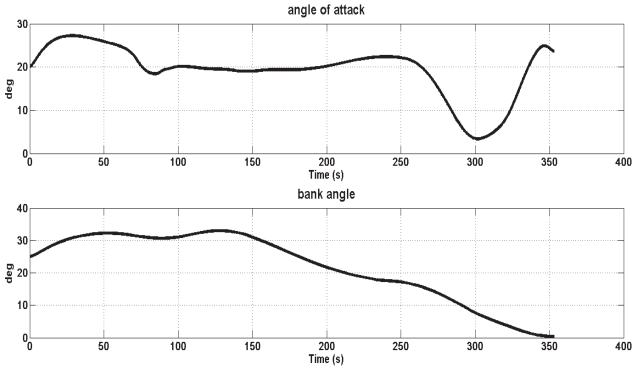

Fig. 2. Controls of the X-33 during re-entry.

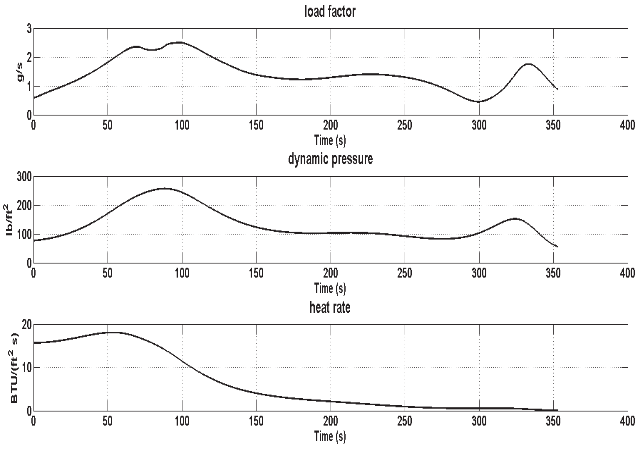

Fig. 3. The values of the constraints of the X-33.

4.2. Lunar Lander

The second example is a lunar landing scenario, with a S/C having a non-throttable engine and reaction control system composed of three throttable orthogonal thrusters. The goal is to land the lunar lander on the surface of the moon from orbit at a pre-determined position. [4]

Two different initial reference frames are defined. These are the body fixed coordinate system and the moon-centred moon-fixed coordinate system (dhc-frame). [4] The state vector is

$$\mathbf{x}(t) = \left[\dot{d}, \dot{h}, \dot{c}, d, h, c, \beta, \chi, m\right]^T \tag{41}$$

The other variables are the pitch angle β, the yaw angle χ, and the mass m. The control vector is given by

$$\mathbf{u}(t) = \left[T_u, T_s, T_q, \omega_\beta, \omega_\chi\right]^T \tag{42}$$

The thrust values are taken in non-dimensional form with respect to mass of the lunar lander. The three thruster vectors are transformed from the body-frame to the dhc-frame .

$$T_1 = \frac{m_0}{m}\left[\cos\left(\beta - \frac{d}{r}\right)\left((T_m + T_u)\cos\chi + T_q\sin\chi\right) - ... \sin\left(\beta - \frac{d}{r}\right)T_s\right] \tag{43}$$

$$T_2 = \frac{m_0}{m}\left[\sin\left(\beta - \frac{d}{r}\right)\left((T_m + T_u)\cos\chi + T_q\sin\chi\right) + ... \cos\left(\beta - \frac{d}{r}\right)T_s\right] \tag{44}$$

$$T_3 = -\frac{m_0}{m}\left((T_m + T_u)\cos\chi + T_q\sin\chi\right) \tag{45}$$

The main thruster T_m is constant and is set to 1 N/kg. The equations of motion can then be described as

$$\ddot{d} = \frac{r}{m(r+h)\cos\frac{c}{r}}\left(-T_1\cos\frac{d}{r} + T_2\sin\frac{d}{r}\right) + ... 2\dot{d}\left(\frac{\dot{c}}{r}\cdot\tan\frac{c}{r} - \frac{\dot{h}}{r+h}\right) \tag{46}$$

$$\ddot{h} = \frac{r}{m(r+h)\cos\frac{c}{r}}\left(-T_1\cos\frac{d}{r} + T_2\sin\frac{d}{r}\right) + ... 2\dot{d}\left(\frac{\dot{c}}{r}\cdot\tan\frac{c}{r} - \frac{\dot{h}}{r+h}\right)\cdot ... \left[\left(-T_1\sin\frac{d}{r} - T_2\cos\frac{d}{r}\right)\cos\frac{c}{r} - T_3\sin\frac{c}{r}\right] + ... \left[\left(\dot{d}\cos\frac{c}{r}\right)^2 + \dot{c}^2\right]\frac{r+h}{r^2} - \frac{GM}{(r+h)^2} \tag{47}$$

$$\ddot{c} = \frac{r}{m(r+h)}\left[\left(T_1\sin\frac{d}{r} + T_2\cos\frac{d}{r}\right)\sin\frac{c}{r} - T_3\cos\frac{c}{r}\right] - \frac{\dot{d}^2}{r}\sin\frac{c}{r}\cos\frac{c}{r} - \frac{2\dot{c}\dot{h}}{r+h} \tag{48}$$

The standard gravitational parameter of the Moon GM is $4.9044 \cdot 10^{12}$ m^3/s^2 and its radius r is 1737,000 m. The states of the lander itself are described in the following manner

$$\dot{\beta} = \omega_\beta \tag{49}$$

$$\dot{\chi} = \omega_\chi \tag{50}$$

$$\dot{m} = |T_m|\cdot\eta_m - (|T_u| + |T_s| + |T_q|)\cdot\eta_{RCS} \tag{51}$$

where η_m and η_{RCS} are the fuel coefficients for the main thruster and reaction control system with values $5 \cdot 10^{-4}$ N/(kg s) and $3.75 \cdot 10^{-4}$ N/(kg s) respectively. The goal is to land at a predetermined position, the cost function is therefore

$$J = \sqrt{(d_{final} - d_{pos})^2 + (h_{final} - h_{pos})^2} \tag{52}$$

The initial conditions of the lunar lander are defined as

$$\mathbf{x}_0 = [5, -19, 0, 0, 300, 0, -86.0583, 0, 0.5397]^T \tag{53}$$

And the final states are restricted to

$$\mathbf{x}_f = [0, 0, 0, free, 0, free, -90, 0, free]^T \tag{54}$$

The constraints on the controls are defined as

$$\begin{bmatrix} 0 \\ -0.4 \\ -0.4 \\ -2 \\ -2 \end{bmatrix} \leq \begin{bmatrix} T_u \\ T_s \\ T_q \\ \omega_\beta \\ \omega_\chi \end{bmatrix} \leq \begin{bmatrix} 0.222 \\ 0.4 \\ 0.4 \\ 2 \\ 2 \end{bmatrix} \tag{55}$$

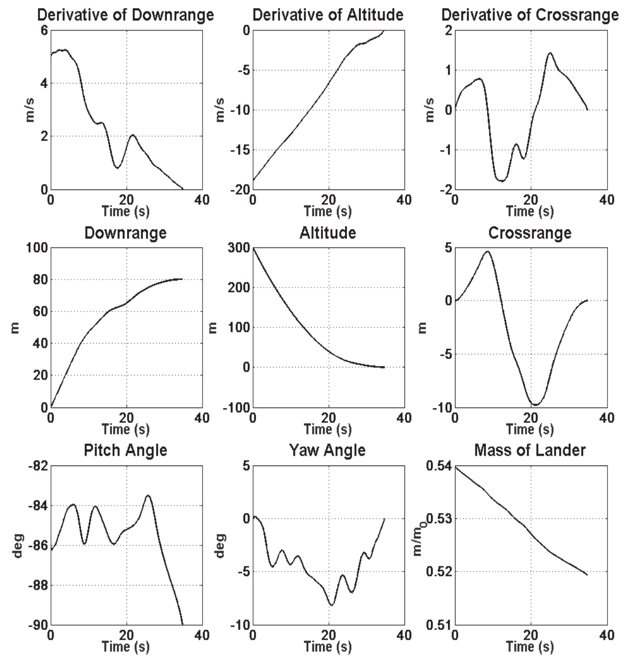

Fig. 4. The states of the lunar lander

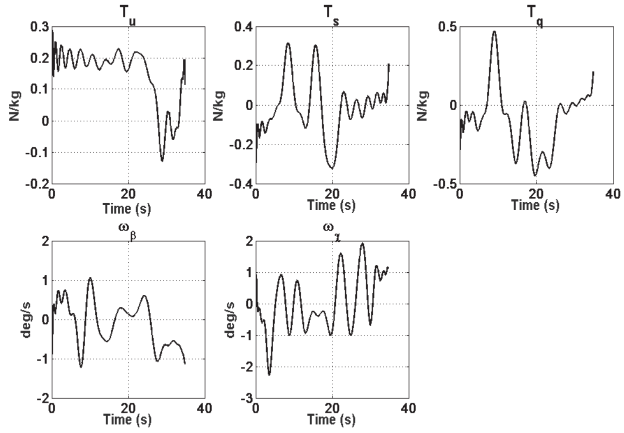

Fig. 5. The controls of the lunar lander

The results of the entire simulation are found in Fig (4) and Fig (5). The final state is reached for given equality and inequality constraints and the Runge-Kutta integration returned satisfactory results. The difference between solution of NLP and propagation of control inputs is smaller than 0.05 meters for final position and altitude.

5. Validation

An auto-validating feature based on the use of Runge-Kutta integration schemes is used at each run to validate the results. The controls that are generated by SPARTAN are inserted into MATLAB's ODE45 solver and the states obtained by propagation of control inputs are compared with the solution found with SPARTAN. The Space Shuttle example from 3) has been used as a benchmark to assess the improved accuracy with the increase of nodes. The results can be seen in Fig. (6). Theoretically, the difference between the solution of the OCP and states obtained by the propagation of the control inputs should ap-

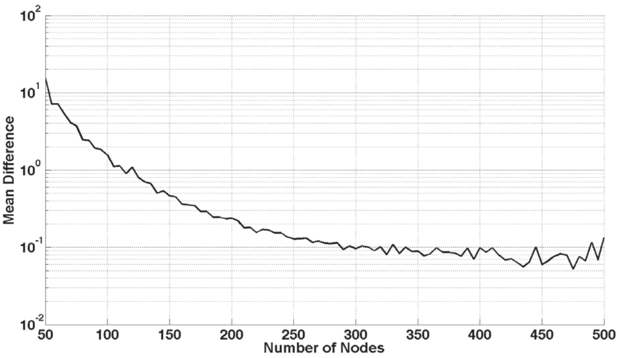

Fig. 6. The difference between SPARTAN and a Runge-Kutta integration scheme over the number of nodes.

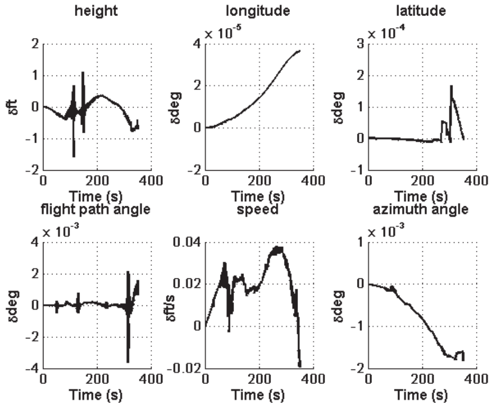

Fig. 7. The difference between SPARTAN and a Runge-Kutta integration scheme for X33 reentry example.

proach zero with the increase of discretisation points. Simulations, however, show an asymptotic behaviour after 300 points, possibly due to truncation errors. The difference between the two methods is still acceptable, especially considering that the two dominant factors are the velocity and height. The computation time for this case is 15.20 seconds for the 100 nodes case and 270 seconds for 300 nodes obtained by a desktop computer with quad-core CPU at 2.66 GHz and 4 GB of RAM.

Similarly, the solutions obtained for Lunar Lander example and X33 is also verified with the same methodology. The results are shown in Fig. (7) and Fig. (8). For the lunar lander the error is uniform for all states except mass of the lander. In the second case the maximum error is experienced for the height of the X33.

6. Conclusion

This paper demonstrated the capabilities of SPARTAN, a tool that employs the Lagrange-Gauss-Radau pseudospectral method to discretise OCP. It has numerous mathematical advantages such as scaling, numerical integration with the dual step, and stable node placement to be able to solve optimal control problems by discretizing the given OCP. Two entry-descent-landing examples were given for demonstration purposes and both solutions were smooth and satisfy constraints within defined error margins. The system is able to handle a different amount of collocation points with increased stability if more points are used, with up to 600 points used whilst still proving to be numerically stable.

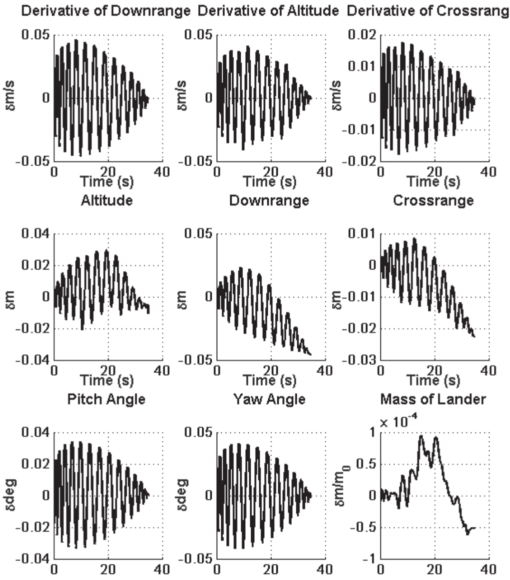

Fig. 8. The difference between SPARTAN and a Runge-Kutta integration scheme for Lunar Lander example.

SPARTAN has proved to be useful to generate trajectories for nonlinear problems having fixed or free final time.

Acknowledgments

The author of this document would like to thank the German Aerospace Centre for offering an internship at Bremen and assist in preparation of the paper. I would especially like to thank my co-authors whom were my supervisors at the GNC department. The freedom they gave me to perform my assignment essentially led to this study.

References

1) Fahroo, F. and Ross, I.M.X., *Pseudospectral Methods for Infinite-Horizon Optimal Control Problems*, Journal of Guidance, Control, and Dynamics, vol. 31, no. 4, pp. 927-936, 2008, doi: 10.1016/j.automatica.2011.01.085.

2) Sagliano, M., *Performance analysis of linear and nonlinear techniques for automatic scaling of discretized control problems*, Operations Research Letters, vol. 42, no. 3, pp. 213 - 216, 2014, doi: 10.1016/j.orl.2014.03.003.

3) Betts, J.T., *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming*, Society for Industrial and Applied Mathematics (SIAM), 2010.

4) Arslantas, Y. E., Oehlschlägel, T., Sagliano, M., Theil, S. and Braxmaier, C., *Approximation of attainable landing area of a moon lander by reachability analysis*, in Hybrid Systems: Computation and Control, Bremen, 2014.

5) Arslantas, Y. E., Oehlschlägel T., Sagliano, M., Theil, S. and Braxmaier, C., *Safe Landing Area Determination For a Moon Lander by Reachability Analysis*, the 65th International Astronautical Congress 2014, Toronto/Canada

6) Gill, E.P., Murray, W. and Michael, A., *Snopt: An SQP Algorithm For Large-Scale Constrained Optimization*, Society for Industrial and Applied Mathematics (SIAM), Vol. 47,No. 1,pp. 9913,1 2005, doi:10.1137/S0036144504446096

7) Laird, C. and A. Wachter, A., *On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming*, Journal of Mathematical Programming, vol. 106, no. 1, pp

25-27, 2006, doi: 10.1007/s10107-004-0559-y

8) D'Onofrio, V., *Implementation of Advanced Differentiation Methods for Optimal Trajectory Computation*, University Federico II, Naples, Italy, 2015.

9) Hildebrand, F.B., *Introduction to numerical analysis*, Dover Publicatons Inc, 1987.

10) Sagliano, M., *Hybrid Jacobian Computation for Fast Optimal Trajectories Generation*, in AIAA, Guidance, Navigation, and Control (GNC) Conference, Boston, 2013, doi:10.2514/6.2013-4554.

11) Berrut, J.P. and Trefethen, L.N., *Barycentric Lagrange Interpolation*, Society for Industrial and Applied Mathematics (SIAM) review, vol. 46, no. 3, pp. 501-517, 2004, doi: 0.1137/S0036144502417715.

12) Martins, J., Sturdza, P. and Alonso, J., *The complex-step derivative approximation*, in ACM transactions on mathematical software, Vol. 29, No. 3, Pages 245262, September 2003, doi:10.1145/838250.838251.

13) Fike, J., Jongsma, S., Alonso, J. and van der Weide, J., *Optimization with gradient and hessian information calculated using hyperdual numbers*, in AIAA applied aerodynamics conference, Honolulu, Hawai, 2011, doi: 10.2514/6.2011-3807.

14) Bollino, K.,Ross, M. and Doman, D., *Optimal Nonlinear Feedback Guidance for Reentry Vehicles*, in AIAA applied aerodynamics conference, Honolulu, Hawai, 2011. doi: 10.2514/6.2006-6074

15) Sagliano, M., Samaan, M., Theil, S. and Mooij, E., *SHEFEX-3 Optimal Feedback Entry Guidance*, in AIAA SPACE 2014 Conference and Exposition, 4-7 August 2014, San Diego, CA 2014, doi: 10.2514/6.2014-4208