Adaptive node distribution for on-line trajectory planning

David A. Anisi
Division of Optimization and Systems Theory
Royal Institute of Technology (KTH)
100 44, Stockholm, Sweden
anisi@math.kth.se

***Abstract*— Direct methods for trajectory optimization are traditionally based on *a priori* temporal discretization and collocation methods. In this work, the problem of node distribution is formulated as an optimization problem, which is to be included in the underlying non-linear mathematical programming problem (NLP). The benefits of utilizing the suggested method for on-line trajectory optimization are illustrated by a missile guidance example.**

I. Introduction

The paradigm of qualitative control design, that is associating a measure of the "utility" of a certain control action, has been a foundation of control engineering thinking. Consequently, optimal control is regarded as one of the more appealing possible methodologies for control design. However, as captivating and appealing as the underlying theory might be, real-world applications have so far been scarce. Some of the reasons for this might be the level of mathematical understanding needed, doubtful viability of optimization under uncertain conditions, and high sensitivity against measurement and modeling errors. Another particularly important factor origins from the high computational demand for solving nonlinear Optimal Control Problems (OCP). As a matter of fact, by extending their "free path encoding method" [1], Canny and Reid have demonstrated the $\mathcal{NP}$ - hardness of finding a shortest kinodynamic path for a point moving amidst polyhedral obstacles in a three dimensional environment [2]. Consequently, attention have been paid to *approximation methods* and computationally efficient algorithms that compute kinodynamically feasible trajectories that are "near-optimal" in some sense. Due to the rapid development of both computer technology and computational methods, the above picture has begun to change. Besides avionics and chemical industry, increasingly many new industrial applications of optimal control can now be observed. In this paper, the problem of missile guidance will be in focus.

It is a well-established fact in numerical analysis, that a proper distribution of grid points is crucial for both the accuracy of the approximating solution, and the computational effort (see *e.g.* [3], [4]). In general, grid adaption is carried out by some combination of re-distribution (strategically moving the nodes), refinement (adding/deleting nodes), or employing higher order numerical schemes in regions where the local accuracy needs to be improved (consult *e.g.* [5]). In most cases however, there exist a trade-off between accuracy and efficiency in terms of computational effort. In this paper, the focus is on improving accuracy for a given efficiency requirement. More precisely, once the number of nodes in the temporal discretization has been decided (depending on *e.g.* computational resources), the question of optimal node distribution is raised.

Although adaptive grid methods - which mainly concern node distribution in the *spatial* domain - have been an active field for the last couple of decades, to the best of our knowledge, utilizing them for adaptive node distribution (in the *temporal* domain) and on-line trajectory optimization has not been considered elsewhere.

This paper is organized as follows. In Section II some background material regarding computational methods for solving optimal control problems is presented. Subsequently in Section III, we advocate that in any computationally efficient method, node distribution should be a part of the optimization process and show that the receding horizon control (RHC) method can be considered as an outcome of such a paradigm. In Section IV, the benefits of utilizing the suggested method are confirmed by a missile guidance example. Finally, this paper is concluded in Section V with some expository remarks.

II. Computational Optimal Control

Consider the following trajectory optimization or Optimal Control Problem (OCP):

$$\begin{aligned} \underset{u}{\text{minimize}} \quad J &= \int_0^T \mathcal{L}(x,u)\mathrm{d}t + \Psi(x(T)) \\ \text{s.t.} \quad \dot{x} &= f(x,u) \\ g(x,u) &\leq 0 \\ x(0) &\in S_i \\ x(T) &\in S_f, \end{aligned}$$

where the state $x \in \mathbb{R}^n$, the control $u \in \mathbb{R}^m$, and the constraints $g : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^p$. All mappings in this paper are assumed to be smooth and the dynamical system complete so that every control input, $u(\cdot)$, results in a well-defined trajectory, $x(\cdot)$. An underlying assumption however is that due to imperfect information, the kinematic constraints, as well as the target set, might change drastically during the course of flight. Consequently, we can not use the family of techniques that rely on off-line generation of a trajectory database for on-line interrogation [6]–[9]. Also, assuming the problem originates from a complex, real-world application, the existence of analytical solutions is disregarded, thus seeking fast computational algorithms for solving the OCP.

Problem Transcription

For the actual design of the computational algorithm, the *infinite dimensional* problem of choosing a control function in a given space, has to be turned into a *finite dimensional* optimal parameter selection problem, *i.e.* a non-linear mathematical programming problem (NLP). This process of representing the continuous time functions by a finite number of parameters, is referred to as *transcription* and is typically achieved by either temporal discretization or finite sum of known basis functions[1] [12]. Since this latter transcription method leads to implicit constraints and gradient expressions, which in turn may give increased computational complexity, the focus in this paper will be on transcription methods based on *temporal discretization*.
It is further conceptually important to differ between *direct* and *indirect* transcription methods (see Figure 1). For a given

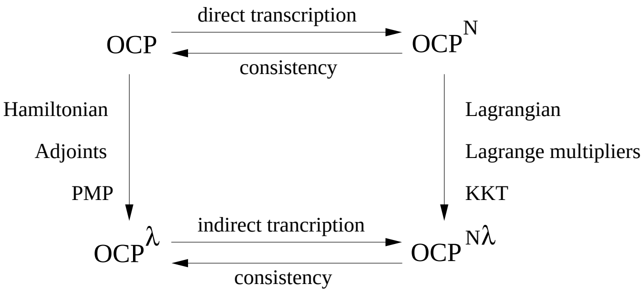

Fig. 1. Direct and in-direct transcription methods.

OCP, indirect methods, which are based on the calculus of variations, start off by introducing the Hamiltonian and formulating the optimality conditions according to the Pontryagin Maximum Principle (PMP). They then proceed by transcribing the associated two point boundary value problem (TPBVP) (denoted OCP$^\lambda$ in Figure 1).
In contrast, direct methods transcribe the OCP directly, hence turning it into a large NLP (denoted OCP^N in Figure 1). The dual to this NLP and the Lagrange multipliers may be achieved by way of the Lagrangian and the Karush-Kuhn-Tucker (KKT) conditions. The direct- and indirect methods have a particular simple relation for the so called *complete* methods [13], for which transcription and dualization indeed commutes, so that the Lagrange multipliers of the NLP are a multiple of the discretized values of the adjoint variables associated with the PMP.

Although indirect methods are considered to produce more accurate results, they are not typically used to solve problems having complex dynamics or constraint set. Neither are they suitable for problems where the underlying OCP is considered to be changeable in terms of the final manifold, S_f and/or the constraint set, $g(x, u)$. This is mainly due to the inherent ill-conditioned properties of the TPBVP, but also the occasionally tedious derivation of the necessary conditions via PMP. Bearing in mind the type of problems considered in this paper, the focus will therefore be on direct transcription methods.
In most direct methods (see *e.g.* [12] and the references therein), transcription is achieved by *a priori* partition of the time interval into a prescribed number of subintervals whose endpoints are called *nodes*. The NLP variables may then be taken as the value of the controls and the states at these nodes. The integral cost functional and the constraint set are discretized similarly and approximated by any preferred quadrature rule (consult *e.g.* [3], [14]). Finally, additional constraints are imposed on the NLP variables so that the state equations are fulfilled at the so called collocation points.

III. Adaptive Node Distribution

It is a well-established fact in numerical analysis, that a proper distribution of grid points is crucial for both the accuracy of the approximating solution, and the computational effort (see *e.g.* [3], [4]). Consequently, the use of adaptive grid methods has for long been an essential element in the sphere of numerical solution of partial differential equations (PDE) as well as ordinary differential equations (ODE) [15]. Despite being an active field for the last couple of decades, to the best of our knowledge, utilizing adaptive grid methods for finding on-line solutions to the trajectory optimization problem has not been considered elsewhere. The basic idea is that by concentrating the nodes and hence computational effort in those parts of the grid that require most attention (*e.g.* areas with sharp non-linearities and large solution variations), it becomes possible to gain accuracy whilst retaining computational efficiency.
This is in fact one of the explanations to the success of the receding horizon control (RHC) or model predictive control (MPC) methods (see *e.g.* [16], [17]). Here, the doubtful viability of long term optimization under uncertain conditions is adhered, so that instead of solving the OCP on the full interval $[0, T]$, one repeatedly solves it on the interval $[t_c, t_c+T_p]$ instead. Here t_c denotes the current time instance and T_p is the planning horizon. However, even in the RHC case, the sub-horizon OCP on $[t_c, t_c + T_p]$ is most often solved based on, if not equidistant (uniform), but at least *a priori* temporal discretization techniques.

In general, there exist three types of grid adaption techniques [5]:

1) *h-refinement*: strategically adding extra nodes to the existing grid in order to improve local grid resolution.
2) *p-refinement*: employing higher order numerical schemes in regions where the local accuracy needs to be improved.
3) *r-refinement*: maintaining a fixed number of nodes, but relocating them strategically over the interval.

Generally, trajectory optimization run-times are critically depending on the number of variables in the NLP. These in turn, are proportional to the number of nodes in the temporal discretization, hence-forth denoted N. How the solution time varies as a function of N is depending on the

[1] Certain choices for basis functions, blur the distinction between the two mentioned transcription methods (see *e.g.* [10], [11]).

particular NLP solver used. Figure 2 illustrates the average, and maximum run-times of NPOPT; the solver used for all simulations here-within. NPOPT is an updated version of NPSOL; a sequential quadratic programming (SQP) based method for solving NLPs [18]. It it worth mentioning, that the average and maximum have been taken both over a number of planning horizons (typically 10 different values) and iterations (typically $100-150$ iterations per planning horizon).

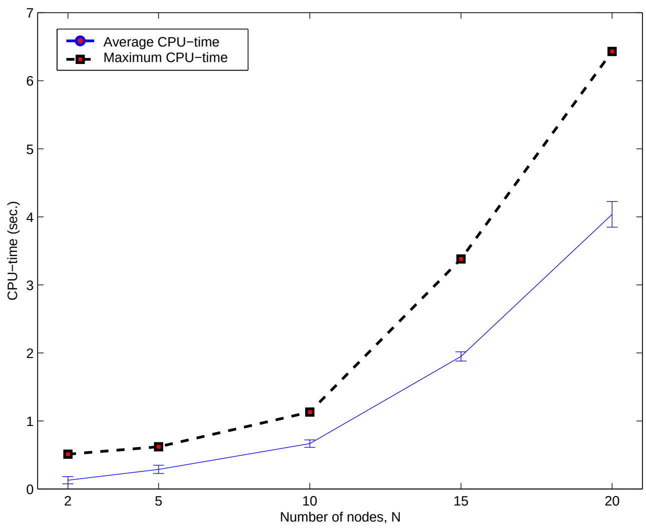

Fig. 2. The increasing average and maximum run-times of NPOPT as a function of N. Computations are performed on a shared Linux cluster, using one of its four 2.80 GHz Intel® Xeon processors.

The essence of Figure 2 is that the choice of N, is to a large extent restricted by real-time computational requirements. Hence, it is extremely important to keep N as low as possible when aiming at constructing computationally efficient methods for trajectory optimization. Therefore, it is the idea of r-refinement that suits our purposes best. To this end, let $p=[t_1,\cdots,t_N]\in\mathbb{R}^N$ denote a partition of $[0,T]$,

$$0=t_1<t_2<\cdots<t_{N-1}<t_N\leq T.$$

Adaptive grid methods are then based on either *equidistribution* of a *monitor function*, or *functional minimization* (FM) [4], [5], [19].

The equidistribution principle (EP) requires a chosen positive definite monitor function (or weight), w, to be equidistributed over all subintervals. Mathematically, the EP can be expressed in various equivalent forms, *e.g.*:

$$m_i(p)=\int_{t_i}^{t_{i+1}} w\ \mathrm{d}t-\frac{\int_0^T w\ \mathrm{d}t}{N-1}=0, i=1,\cdots,N-1,$$
$$m_i(p)=\int_{t_{i-1}}^{t_i} w\ \mathrm{d}t-\int_{t_i}^{t_{i+1}} w\ \mathrm{d}t=0, i=2,\cdots,N-1.$$

As an example, $w\equiv 1$ gives rise to the oftenly used uniform (equidistant) discretization method. Other commonly employed monitor functions include the "arclength monitor function", $w=\sqrt{\varepsilon+\dot{x}^2}$ (claimed to be the most efficient among all choices), and "curvature monitor function", $w=(\varepsilon+\ddot{x}^2)^{\frac{1}{4}}$. Here the design-parameter, $\varepsilon\geq 0$, decides how dense the nodes are lumped in the circumvent of areas with large solution variations.

The functional framework to grid generation (FM), is based on the principle of specifying a measure of the grid quality. Traditionally, principles as smoothness, orthogonality and clustering properties of the grid are included in the functional, $I(p)$, [4], [19]. Minimizing $I(p)$ will produce an optimal partition with respect to the chosen grid quality measure.

Based on the two existing frameworks for adaptive grid generation (EP and FM), we now outline a generalized approach. Regardless the choice of w, we remark that node allocation by the EP, can be determined by imposing a number of *grid constraints*, $m(p)\leq 0$. These constraints are to be augmented with the original constraints, $g(x,u)$. Note that this approach introduces constraints and state variables (namely p) in the augmented NLP. However, it also enable us to use a partition with smaller number of nodes compared with an *a priori* and fixed discretization method, so that the total number of variables and constraints might still be reduced.

The idea is then to formulate the problem of node distribution as a constrained optimization problem:

$$\begin{aligned} \underset{p}{\text{minimize}} \quad & I(p) \\ \text{s.t.} \quad & m(p)\leq 0, \end{aligned} \tag{1}$$

which is to be augmented with the underlying NLP. From (1) it is plainly seen that EP and FM are merely special cases of the suggested approach. We conclude this section by giving examples of the usage of this approach.

Example 1: *Setting $d_i=t_{i+1}-t_i, i=1,\cdots,N-1$, the solution to the following optimization problem:*

$$\begin{aligned} \underset{d}{\mathit{minimize}} \quad I(d) &= \textstyle\sum_{i=1}^{N-1} d_i-\varepsilon\ln d_i \\ s.t. \quad m(d) &= \textstyle\sum_{i=1}^{N-1} d_i-T\leq 0 \quad (d_i\geq 0), \end{aligned}$$

is the equidistant RHC discretization scheme with ε deciding the step length (and hence planning horizon). This follows since if $(N-1)\varepsilon\leq T$, then

$$\nabla_i I(d)=1-\frac{\varepsilon}{d_i}=0\Longrightarrow d_i=\varepsilon.$$

Example 2: *The linear constraint*

$$m(d)=\sum_{i=1}^{\varepsilon_1(N-1)} d_i-\varepsilon_2 T\leq 0,$$

reflects the objective of distributing ε_1 parts of the nodes in the first ε_2 parts of the time interval.

The main reason for being interested in this types of constraints lies along the line of thought of RHC/MPC approaches; that is considering current information as perishable so that it is favorable to concentrate the nodes in the near future.

IV. Design Study: Missile Guidance

Traditionally, the problem of steering a missile to its target is broken down into (at least) two subproblems: the problem of *trajectory optimization* and the problem of *auto-pilot design*. This can be viewed as a control system having two degree of freedom; an inner loop (the auto-pilot) and an outer loop (the trajectory optimizer) (see Figure 3). The

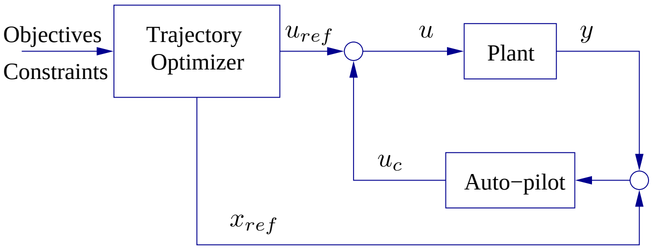

Fig. 3. Two level separation of the missile guidance problem.

trajectory optimizer provides a feasible feed-forward control and reference trajectory that is optimal in some specified sense with respect to *e.g.* time to intercept or intercept velocity, and subject to constraints on *e.g.* terminal aspect angle (given by warhead efficiency and target vulnerability) or path segment location (dictated by tactical considerations). It is then the task of the auto-pilot to perform the trajectory following.
By virtue of this separation, only *suboptimal* solutions can in general be found, but the advantage is that the details of the dynamics of the missile only enters into the trajectory optimization part of the problem as (relatively simple) conditions on the reference trajectory. In this work, the existence of an auto-pilot is assumed, so that the focus will solely be on the trajectory optimization part.

By means of standard approximation procedures in flight-community (see *e.g.* [20], [21]), the six-degree-of-freedom (6DoF) equations of motion of the missile in $\mathbb{R}^3$, can be reduced to 3DoF planar movement in two orthogonal subspaces, namely the pitch-, and yaw-plane. Since the 3DoF equations of motions in these planes are similar and decoupled, in what follows, just the pitch-plane dynamics will be considered.
The 3DoF equations of motion in the pitch plane consider the rotation of a body-fixed coordinate frame, (X_b, Z_b) about an Earth-fixed inertial frame, (X_e, Z_e) (see Figure 4).
The governing dynamic equations are

$$\begin{aligned}
\dot{u} &= \frac{F_x}{m} - qw - g\sin\theta \\
\dot{w} &= \frac{F_z}{m} + qu + g\cos\theta \\
\dot{q} &= \frac{M}{I_y} \\
\dot{\theta} &= q \\
\dot{x}_e &= \quad u\cos\theta + w\sin\theta \\
\dot{z}_e &= -u\sin\theta + w\cos\theta,
\end{aligned}$$

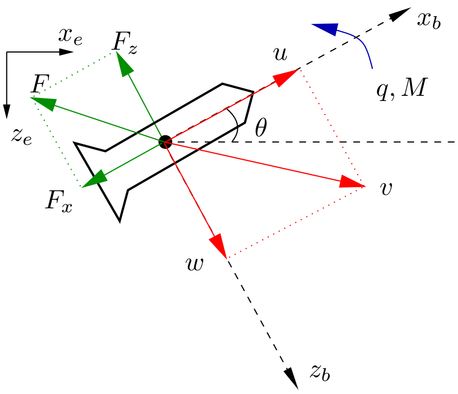

Fig. 4. Missile system variables.

where u and w are the X_b and Z_b components of the velocity vector, x_e and z_e denote the position of the missile in the inertial frame (X_e, Z_e), q is the pitch angular rate, θ denotes the pitch angle, m is the missile mass, g is the gravitational force, while I_y denotes the pitching moment of inertia. The system inputs are the applied pitch moment, M, together with the aerodynamic forces, F_x, F_z, acting along the X_b and Z_b axis respectively. During the simulations we adopt the constants given in Reference [22] and set $m = 204.02\ kg$, $g = 9.8\ m/s^2$ and $I_y = 247.437\ kg\ m^2$.
Referring to Figure 5 and 6, the first simulation shows the terminal guidance part of a missile trajectory optimization problem. The missile starts off horizontally from $(0, 10)$ aiming at a target in $(700, 0)$ with terminal aspect angle $-\frac{\pi}{2}$. Figure 5 depicts the reference trajectories with the missile velocities (in the inertial frame) indicated by small line segments.

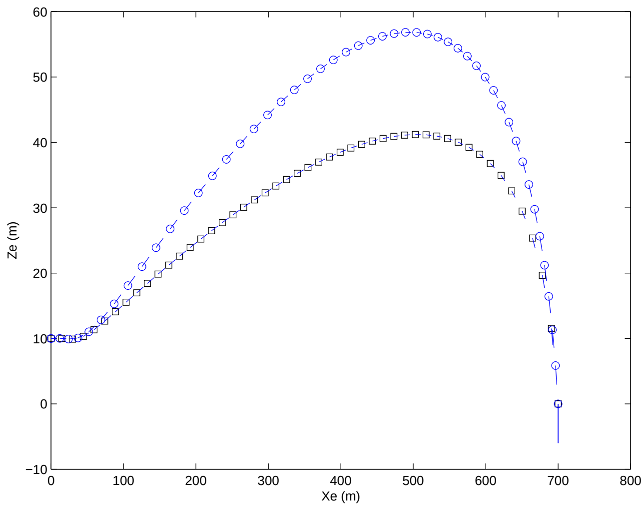

Fig. 5. Reference trajectories: static (○) and adaptive (□).

In the adaptive case, an EP based on the arclength monitor function together with a linear $I(p)$ is used. Seeing beyond the unequal axis scales, the nodes have been distributed more evenly over different path segments. In fact, there are 7 nodes/100 m path segment in the adaptive case, while the same figure varies between $5 - 13$ in the static case.

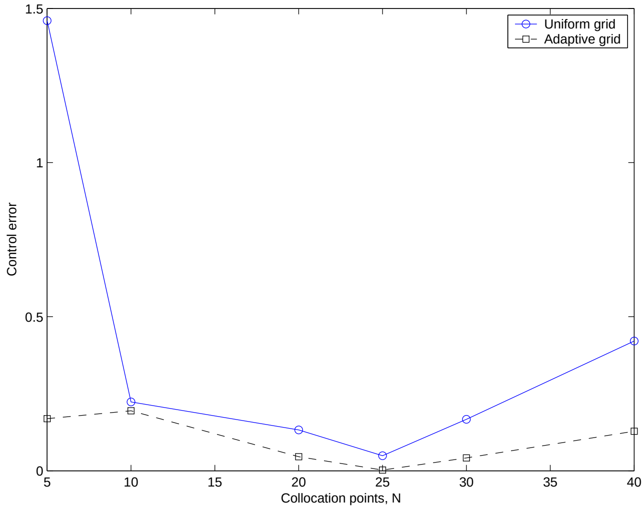

Fig. 6. The accuracy of the approximating control.

Figure 6 shows the optimal control approximation error as a function of N. It can be noted that, for a given N, the extra degree of freedom provided by distributing the nodes is used constructively to improve accuracy. This illustrates the soundness of the proposed approach. Moreover, Figure 6 reveals the nonuniform convergence rate of the approximation error which - in our particular case - is seen to be minimized for $N = 25$. The reason for this is the pronounced nonlinearity of the considered NLP together with the fact that the used optimization routine (NPOPT) is a local optimizer, *i.e.* does not guarantee convergence to a *global* minimum. It is therefore not possible to expect that a higher value on N should always yield a better trajectory approximation.
As previously mentioned, in general, there is a trade-off between accuracy and efficiency in terms of computational effort. Once we have observed that re-distributing the nodes improves the accuracy of the approximation, one might wonder how this effects the computation time. Figure 7 shows the average CPU-time used in the simulations for different values on N. It can be noted that adopting the proposed adaptive grid generation scheme, does *not* bring any increase in the average computational time. We believe that the nonlinearity of the original set of equations describing the motion of the missile, is one of the main reasons for this.

V. Concluding Remarks

The main purpose of this paper have been to advocate the use of adaptive grid generation techniques for on-line trajectory planning. In this work, we have chosen to concentrate on the use of the so called r-refinement technique; that is strategically re-distributing a given number of nodes over the time domain. The main reason for this have been the pronounced inter-relation between the number of nodes in the temporal discretization and trajectory optimization run-times.

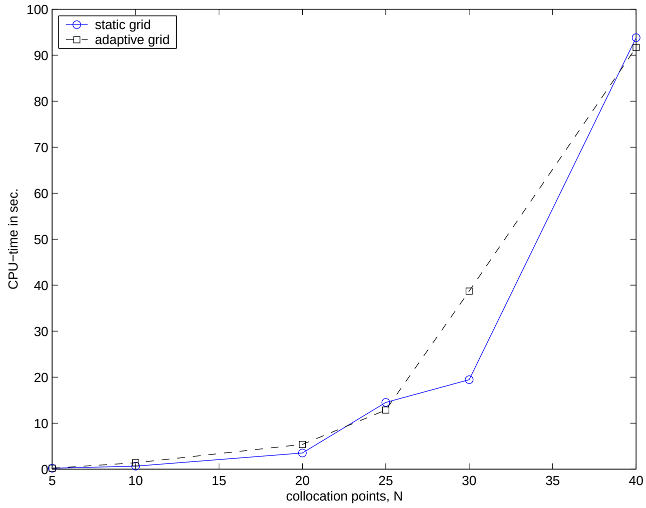

Fig. 7. The average CPU-time for the uniform- and adaptive grid generation scheme as a function of N.

It is argued that in any computationally efficient method, node distribution should be a part of the optimization process. This, in order to minimize the discretization error and gain accuracy, *without* bringing any drastic increase in the computational effort. Here-within, re-distributing the nodes have been formulated as a constrained optimization problem which is to be included in the underlying NLP.
The missile guidance problem considered, showed that the extra degree of freedom provided by distributing the nodes is used constructively to improve accuracy. These advantages accrue particularly in the case when having a nonlinear dynamic system at hand. The reason for this being that having the node positions as variables in the underlying NLP, turns a linear system into a bilinear one, which may then give rise to an undesirable increase in computational complexity.

Acknowledgments

The author would like to gratefully acknowledge the financial support by the Department of Autonomous Systems at the Swedish Defence Research Agency.

References

[1] Canny, J. and Reif, J., "New lower bound techniques for robot motion planning problems," *Proc. of the* 28^{th} *Annual IEEE Symposium on Foundations of Computer Science*, Los Angeles, CA, USA, Oct. 1987, pp. 49–60.

[2] Canny, J., Reif, J., Donald, B., and Xavier, P., "On the complexity of kinodynamic planning," *Proc. of the* 29^{th} *Annual IEEE Symposium on Foundations of Computer Science*, Oct. 1988, pp. 306–316.

[3] Stoer, J. and Bulirsch, R., *Introduction to numerical analysis*, Vol. 12 of *Texts in Applied Mathematics*, Springer-Verlag, New York, 2nd ed., 1993, Translated from the German by R. Bartels, W. Gautschi and C. Witzgall.

[4] Liseikin, V. D., *Grid generation methods*, Scientific Computation, Springer-Verlag, Berlin, 1999.

[5] Blake, K., *Moving mesh methods for non-linear parabolic partial differential equations*, Ph.D. thesis, Department of Mathematics, the University of Reading, U.K., 2001.

[6] Chen, Y. and Huang, J., "A new computational approach to solving a class of optimal control problems," *International Journal of Control*, Vol. 58, No. 6, 1993, pp. 1361–1383.

[7] Grancharova, A. and Johansen, T. A., "Explicit approaches to constrained model predictive control: a survey," *Modeling, Identification and Control*, Vol. 25, No. 3, 2004, pp. 131–157.

[8] Milam, M., "A homotopy technique for constrained trajectory generation," Research notes, Division of Engineering and Applied Science, California Institute of Technology, Pasadena, CA, Sep. 1998.

[9] Schierman, J., Hull, J., and Ward, D., "On-line trajectory command reshaping for reusable launch vehicles," *Proc. of the AIAA Guidance, Navigation, and Control Conference and Exibit*, Aug. 2003.

[10] Elnagar, G., Kazemi, M. A., and Razzaghi, M., "The pseudospectral Legendre method for discretizing optimal control problems," *IEEE Trans. Automat. Control*, Vol. 40, No. 10, 1995, pp. 1793–1796.

[11] Fahroo, F. and Ross, M., "Direct trajectory optimization by a Chebyshev pseudospectral method," *Journal of Guidance, Control, and Dynamics*, Vol. 25, No. 1, 2002, pp. 160–166.

[12] Betts, J. T., "Survey of numerical methods for trajectory optimization," *Journal of guidance, control, and dynamics*, Vol. 21, No. 2, Mars–April 1998, pp. 193–207.

[13] Fahroo, F. and Ross, M., "A perspective on methods for trajectory optimization," *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, Aug. 2002.

[14] Davis, P. J. and Rabinowitz, P., *Methods of numerical integration*, Computer Science and Applied Mathematics, Academic Press Inc., Orlando, FL, 2nd ed., 1984.

[15] Gottlieb, D., Hussaini, M., and Orszag, S., "Theory and applications of spectral methods," *Spectral Methods for PDEs*, SIAM, 1984.

[16] Mayne, D. Q., Rawlings, J. B., Rao, C. V., and Scokaert, P. O. M., "Constrained model predictive control: stability and optimality," *Automatica*, Vol. 36, No. 6, 2000, pp. 789–814.

[17] Qin, J. S. and Badgwell, T. A., "A survey of industrial model predictive control technology," *Control Engineering Practice*, Vol. 11, No. 7, Jul. 2003, pp. 733–764.

[18] Gill, P. E., Murray, W., Saunders, M. A., and Wright, M., "User's guide for NPSOL 5.0: a fortran package for nonlinear programming," Tech. rep., Systems Optimization Laboraty, Stanford University, Stanford, CA. 94 305, 1998.

[19] Kennon, S. and Dulikravich, G., "Comparison of grid generation techniques," *IEEE Fall Joint Computer Conference*, Dallas, TX, USA, 1986, pp. 568–575.

[20] Miele, A., *Flight mechanics*, Addision Wesley, Massachusetts, MA, 1962.

[21] Stevens, B. and Lewis, F., *Aircraft control and simulation*, Wiley, 1992.

[22] Leith, D. J. and Leithead, W. E., "Gain-scheduled control: relaxing slow variation requirements by velocity-based design," *Journal of Guidance, Control, and Dynamics*, Vol. 23, No. 6, 2000, pp. 988–1000.