

## Distance Fields Over Grids Method for Aircraft Envelope Determination

Helsen, R; van Kampen, Erik-Jan; de Visser, Coen; Chu, Qiping

**DOI**

[10.2514/1.G000824](https://doi.org/10.2514/1.G000824)

**Publication date**

2016

**Document Version**

Accepted author manuscript

**Published in**

Journal of Guidance, Control, and Dynamics: devoted to the technology of dynamics and control

**Citation (APA)**

Helsen, R., van Kampen, E.-J., de Visser, C., & Chu, Q. (2016). Distance Fields Over Grids Method for Aircraft Envelope Determination. *Journal of Guidance, Control, and Dynamics: devoted to the technology of dynamics and control*, 39(7), 1470-1480. <https://doi.org/10.2514/1.G000824>

**Important note**

To cite this publication, please use the final published version (if applicable). Please check the document version above.

**Copyright**

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

**Takedown policy**

Please contact us and provide details if you believe this document breaches copyrights. We will remove access to the work immediately and investigate your claim.

# Distance Fields Over Grids Method for Aircraft Envelope Determination

Roel Helsen <sup>1</sup> and Erik-Jan van Kampen <sup>2</sup> and Coen C. de Visser <sup>3</sup> and Qiping Chu <sup>4</sup>  
*Delft University of Technology, Delft, Klyuwerweg 1, Delft 2629HS, The Netherlands*

In this research, the concept of posing the flight envelope as a reachable set is explored further and it is investigated whether the "Distance Fields Over Grids (DFOG)" method for reachability computations is a viable method for envelope determination. The DFOG method approximates the reachable set of a dynamic system by solving a number of optimal control problems over a grid in the state-space. Then, the endpoints of the resulting optimal trajectories are used to provide an approximation of the reachable set. The DFOG method is improved by using a 2<sup>nd</sup> order RK method for the state discretization and developing a new adaptive grid selection approach. This method is then applied to the computation of a four-dimensional reachable set of a nonlinear high-fidelity F-16 model. The results are compared to results obtained previously with level sets and show that the DFOG method provides a less conservative approximation to the reachable set.

---

<sup>1</sup> Graduate Student, Delft University of Technology – Control & Simulation

<sup>2</sup> Assistant Professor, Delft University of Technology – Control & Simulation, AIAA member

<sup>3</sup> Assistant Professor, Delft University of Technology – Control & Simulation, AIAA member

<sup>4</sup> Associate Professor, Delft University of Technology – Control & Simulation, AIAA member

## Nomenclature

### Symbols

$\mathbf{x}(t)$  = state trajectory

$\lambda$  = adjoint trajectory

$\mathbf{x}_k$  = discretized state

$\mathbf{g}_\rho$  = grid points

$\mathbf{z}$  = optimization parameter vector

$\Phi$  = increment function

$\mathcal{R}$  = reachable set

$\mathcal{U}_m$  = set of admissible controls

$\mathcal{X}_0$  = set of initial states

$V_t$  = true airspeed [m/s]

$\alpha$  = angle of attack [rad]

$q_b$  = pitch rate [rad/s]

$q_2$  = quaternion component [-]

### Acronyms

LOC = Loss Of Control

DFOG = Distance Fields Over Grids

RK = Runge-Kutta (methods)

## I. Introduction

Envelope determination is an essential tool in aircraft design. These calculations play an important role during the design to check whether the design meets the requirements and if not, to determine what is needed to improve the design. Envelope determination is also a key part in envelope protection. Envelope protection is a global term describing all systems that help the pilot to keep the aircraft within safe operating limits, i.e. the flight envelope. This ranges from systems that warn the pilot that an envelope violation is imminent, to systems that neglect or modify the pilot's

inputs in such a way that it is guaranteed that the aircraft stays within the envelope. An example to illustrate this are stick shakers and stick pushers. A prerequisite for this kind of systems to work, is knowing when stall will occur, which demonstrates the need for envelope determination. The goal of envelope protection systems is to reduce the number of accidents caused by LOC, which is according to [1], the largest cause of fatal accidents in both commercial and general aviation. The classical and most studied example of an accident caused by LOC is the El Al flight 1862 crash in the Bijlmermeer area, [2]. The aircraft lost the two right wing engines and most of the control surfaces stopped working due to the damage to the hydraulic system. Post-crash analyses showed that, despite the extensive damage to the airframe, the aircraft had still marginal controllability left inside a severely reduced flight envelope. A detailed description is given in [3, Section 1.2.4]. Simulations, [4], have shown that when the severely reduced flight envelope would have been known, an experienced pilot could have landed the aircraft. This example clearly shows the need for knowledge of the flight envelope.

The flight envelope is described as the set of all possible flight conditions at which an aircraft can remain airborne. The flight conditions at which an aircraft can stay airborne are limited by various constraints. These constraints originate from, among others, various aerodynamic phenomena that can happen to the aircraft such as stalling and buffet characteristics. Furthermore there are constraints generated by the control limits, such as maximum thrust and maximum control deflections. Also the structural limits of the aircraft can constrain the possible flight conditions. In [5, Section 10.4], it is defined as follows: “The flight envelope describes the area of altitude and airspeed where an airplane is constrained to operate.” The flight envelope as described by this definition, is usually represented by various charts and diagrams, for example showing the possible combinations of airspeed and altitude for level flight and different climb rates. Another example is the chart displaying the possible turning rates for different airspeeds.

The dynamics of an aircraft can pose additional constraints (and possibilities) on the envelope. Furthermore, other constraints are left out, such as the terrain. This has led to a new definition of the envelope in [6]. Van Oort, [6], redefined the flight envelope as the intersection of three different envelopes: the dynamic envelope, the structural and comfort envelope, and the environmental

envelope. In [6], these three are explained thoroughly, but here the focus will be on the dynamic envelope.

The conventional flight envelope is defined for steady or quasi-steady conditions, which can be computed with the standard performance formulas. The dynamic envelope is harder to compute as it has to take the dynamics of the aircraft into account. Here, the goal is to determine all possible motions of an aircraft starting from a certain flight condition in a given time span. Furthermore it is desired to know which of these motions can be controlled back to safe flight in a given time span. These possible motions are foremost governed by the dynamics of the aircraft under investigation which determine how the aircraft will behave. Further contributing to the range of motions is the possibility to give arbitrary control inputs to the aircraft. If there would be no limit to the controls, in theory, any motion would be possible, but with the limited controls also a limited range of possible motions arises. The goal is thus to determine the set of possible motions of the aircraft that can be controlled back to safe flight. This kind of calculations are called reachability computations. To solve these questions, algorithms are needed that can quantify the set of possible motions.

Reachability algorithms are algorithms that can calculate the set of possible endpoints of a dynamic system with uncertain inputs after a certain time span starting from a set of initial points. These computations can be carried out either forward or backward in time. backward reachability is also known as controllability. This is because the resulting set contains all states that can be controlled in such a way that they end inside the set of initial states of the computation. Note that this concept of controllability differs from the traditional concept of controllability for LTI systems. For LTI systems, a system is controllable when any final state can be reached from any initial state. However, when the inputs of the system are subject to constraints, which is always the case in real life, it is often not possible to reach any final state from a given initial state. In this case, only a limited set can be reached which is called the reachable or controllable region. This will be explained in more detail in Section II.

Determining the reachable set of a dynamical system is hard and the methods that are used are still subject of ongoing research. Some authors study the reachability of linear time-(in)variant systems in order to get a better understanding of the properties of the reachable set and its structure.

The most important property of the reachable set of LTI systems is the fact that it is a convex set. This means that any two points in the set can be connected by a straight line that is part of the set. This property plays an important role in most methods for LTI systems. The backward reachable set is also called the controllability region and under this name, some methods for its determination were developed, [7, 8]. Based on the convexity of the reachable set, also another type of methods is developed, the geometric methods. These methods describe the reachable set as a collection of geometric primitives and evolve these in time. The use of different primitives leads to different methods, [9–11].

For non-linear systems there is only the small-time convexity property, which states that for small time periods the reachable set stays convex. This small time period depends on the speed of the dynamics and for larger periods, convexity is not guaranteed. This lack of convexity of the reachable set complicates the analysis and requires different methods. First there are methods that convert the nonlinear case back to a linear case such that methods for linear systems can be used, [12, 13]. Then there is the concept of level sets where the boundary of the reachable set is posed as a partial differential equation. This method is already used extensively in the past, [14–16] and also very recently, [17]. Recently, another method for the computation of reachable sets, Distance Fields Over Grids (DFOG), has been developed, [18, 19], which computes the reachable set in a different way. This method projects all grid points of a certain grid onto the reachable set and uses the resulting point cloud as an approximation to the reachable set.

The main contribution of this research is the application of DFOG, [19], to flight envelope determination. To achieve this, two extensions were made to the existing DFOG algorithm to increase its computational efficiency. First, a 2<sup>nd</sup> order RK method is used to discretize the trajectory. Second, a new iterative grid selection strategy is developed to severely reduce the number of grid points in the computation. To assess the performance of the algorithm, it is implemented for the computation of the reachable set of an F-16 model and compared to results previously obtained by level sets, [16].

The paper is organized as follows: In Section II, an introduction in the concept of reachable sets will be given. Then in Section III, the DFOG algorithm is explained. First it is shown how

the reachability problem can be posed as a set of optimal control problems. Then, it is explained how the optimal control problems are discretized into non linear programming problems. Finally, the extensions to the algorithm are described. In Section V, the application of the DFOG method to the F-16 model is discussed. Lastly, Section VI gives the conclusions.

## II. Reachable sets

In reachability analysis, a geometric interpretation of the dynamical system is used. Here a state-space representation of the system under consideration is used. In this setting, the states of the system are represented by vectors in a  $n$ -dimensional Euclidean space and all vector algebra applies. Since the states are defined in a Euclidean space, standard metrics defined for these spaces such as norms can also be used.

In reachability analysis, a dynamical system is considered, which is defined as follows:

$$\begin{aligned}\dot{\mathbf{x}}(t) &= f(t, \mathbf{x}(t), \mathbf{u}(t)), \\ \mathbf{u}(\cdot) &\in \mathcal{U}_m.\end{aligned}\tag{1}$$

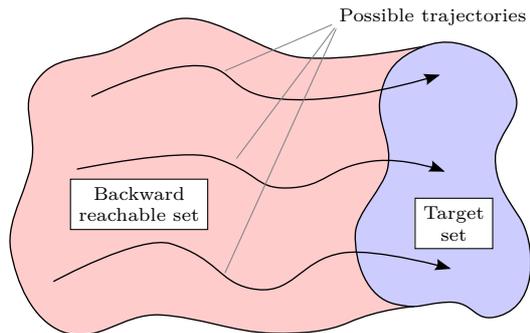
Here it is assumed that  $f(t, \mathbf{x}, \mathbf{u})$  is Lipschitz continuous,  $\dot{f}(t, \mathbf{x}, \mathbf{u}) < L$ . The set of admissible controls  $\mathcal{U}_m$ , is the most determining part in reachability analysis. Its purpose is to limit the possible control inputs to values that are actually possible in real life. When it is allowed to give arbitrarily large inputs to a controllable system, any state can be reached at any time. In real life, the possible control inputs are generally bounded in some way. Mostly, this is a mechanical limit of the actuator providing the input. With these bounded control inputs, only a certain set of states can be attained given a certain amount of time.

The interest is in finding all states that can be obtained from a certain set of initial conditions or all the states that can be controlled back to a certain set of terminal conditions. These sets can be obtained either by evolving the dynamics Eq. (1) forward or backward in time from the known initial set. Then, based on those in [20, Definition II.1, II.2], the forward and backward reachable set can be defined as follows:

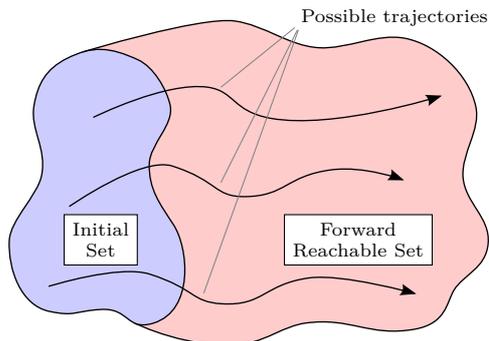
**Definition II.1 (forward Reachable Set)** *The forward reachable set  $\mathcal{R}_F(\tau)$  at time  $\tau$  ( $0 \leq \tau \leq t_f$ ) of the system Eq. (1) for the initial set  $\mathcal{S}_0$ , is the set of all states  $\mathbf{x}(\tau)$ , such that there exists*

a control input  $\mathbf{u}(t) \in \mathcal{U}_m$ , for which  $\mathbf{x}(\tau)$  is reachable from some  $\mathbf{x}(0) \in \mathcal{S}_0$  along a trajectory satisfying Eq. (1).

**Definition II.2 (backward Reachable Set)** The backward reachable set  $\mathcal{R}_B(\tau)$  at time  $\tau$  ( $0 \leq \tau \leq t_f$ ), of the system Eq. (1) for the target set  $\mathcal{T}_f$ , is the set of all states  $\mathbf{x}(\tau)$ , such that there exists a control input  $\mathbf{u}(t) \in \mathcal{U}_m$ , for which some  $\mathbf{x}(t_f) \in \mathcal{T}_f$  are reachable from  $\mathbf{x}(\tau)$  along a trajectory satisfying Eq. (1).



(a) backward



(b) forward

**Fig. 1 Reachable sets**

In essence, the forward and backward reachable sets are very similar in computation. They both start with a certain initial set and then trajectories from this set are evolved in time. The difference is that time is reversed in the computation of the backward reachable set. The concepts that will be defined hereafter can be used for both directions and the general term reachable set will be used, meaning either the forward or backward reachable set. Furthermore, reachable sets

defined in this way are known as maximal reachable sets [21], denoted as  $\mathcal{R}^\sharp$ . Therefore, the symbol  $\mathcal{R}^\sharp$  without any subscripts will be used when it applies to both directions.

These definitions only hold at specific moments in time,  $\tau$ , as states in the reachable set  $\mathcal{R}^\sharp(\tau)$  can be controlled to or from the initial set in  $\tau$  time. The reachable sets do not specify whether at any other time, this is still reachable, which might not be the case. When taking into consideration a time period instead of a specific time, an extra concept is needed, which is called the reach tube.

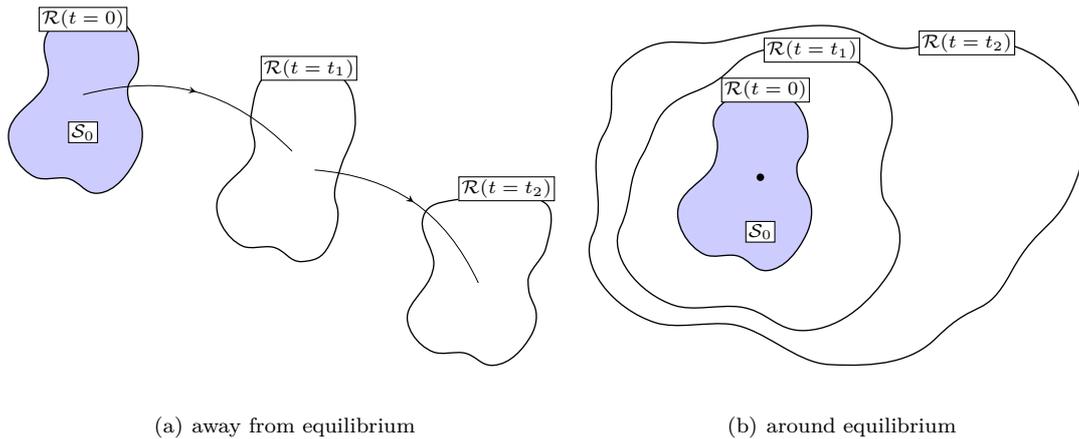
**Definition II.3 (Maximal reach tube)** *The maximal reach tube  $\mathcal{R}_{[t_1, t_2]}^\sharp(\tau)$  is the set of all states  $\mathbf{x}(\tau)$  for which there exists an input  $\mathbf{u}(\cdot) \in \mathcal{U}_m$  such that  $\mathbf{x}(\tau)$  is reachable from some  $\mathbf{x}_0 \in \mathcal{S}_0$  along a trajectory satisfying Eq. (1) at a time  $\tau \in [t_1, t_2]$ .*

It can be proven, [21, proposition 3], that this reach tube can be computed in the following way:

$$\mathcal{R}_{[t_1, t_2]}^\sharp = \bigcup_{\tau \in [t_1, t_2]} \mathcal{R}(\tau). \quad (2)$$

The nature of the reachable sets and reach tube is determined to a large extent by the location of the initial set in the state-space and the relation between the control authority and the natural flow of the system. When the initial set is an equilibrium point of the system or situated around it, the reachable set is likely to grow in time around this equilibrium point depending on the control restrictions. When the initial set is situated away from the equilibrium point and the natural flow dominates the possible control inputs, the reachable set will move through the state-space and change shape when looking at different times. These two different behaviors can be seen in Figure 2. In the latter case, the nature of the system plays an important role in where the reachable set will end up after some time. When the system is stable, the reachable set will converge around the equilibrium point and stay there. When the system has a stable limit cycle, the reachable set will move towards it and keep following it. When the system is unstable, the reachable set will only grow and diverge. Different problems will yield a different type of system with its corresponding type of reachable set.

The manoeuvre envelope of an aircraft was defined as the intersection of the set of states that can be reached from safe flight and the set of states that can be controlled back to safe flight. The important part here is the definition of “*Safe Flight*”. How should this be defined? The best



**Fig. 2 Reachable set evolution in time**

definition would probably be: “*The set of states in which the aircraft can be kept indefinitely.*” This set is called the viability kernel, which is a concept closely related to reachability analysis as explained in [17, 21]. Here, safe flight will be defined as steady, straight and level flight, i.e. trimmed flight. This set intuitively satisfies the definition of the viability kernel, as simply giving the trim inputs will keep the aircraft within the set. However, in reality, the true viability kernel will be larger, [17].

The problem with using a reachable set to determine the set of states which can be reached from safe flight is that it only says something about a specific time instant and not for other time instants or periods of time. To have a meaningful definition of the set, it should be defined as a reach tube, so that information about a certain time period can be given. Here, the construct of the continual reachable set [21] over a horizon  $[t_m, \infty]$  can be used. This set contains all states that can be controlled to the initial set in any time  $\tau$  within the horizon.

Hence, a modified version of their definition of the continual reachable set, [21, Def 7], can be used for the dynamic flight envelope.

**Definition II.4 (Continual reachable set)** *The continual reachable set for horizon  $[t_1, t_2]$ , denoted as  $\mathcal{R}_{[t_1, t_2]}^\gamma$ , is the set of states for which, for any  $t \in [t_1, t_2]$ , there exists a control  $\mathbf{u}(\cdot) \in \mathcal{U}_m$  such that they can be controlled back to the initial set along a trajectory satisfying Eq. (1).*

Analogous to [21, Theorem 1] this can be linked to the maximal reachable sets as follows:

$$\mathcal{R}_{[t_1, t_2]}^\gamma = \bigcap_{t \in [t_1, t_2]} \mathcal{R}^\sharp(t). \quad (3)$$

Since the initial/target set is a viability kernel, the reachable set cannot shrink. All states part of  $\mathcal{R}^\sharp(t)$  are also reachable in  $t + \tau$ , by adding the control required to stay in the initial set for  $\tau$  seconds, thus they are also part of  $\mathcal{R}^\sharp(t + \tau)$ . Now, reasoning that the reachable set can only grow, it can be stated that:

$$\mathcal{R}_{[t_1, t_2]}^\gamma = \bigcap_{t \in [t_1, t_2]} \mathcal{R}^\sharp(t) = \mathcal{R}^\sharp(t_1). \quad (4)$$

This framework allows to compute the set of states that can be controlled back to the initial set and kept there within at least  $t_1$ , which is the desired set. For this, only the reachable set at  $t_1$  has to be computed as this gives the largest set fulfilling all constraints.

Now the set of potential manoeuvres which can be executed within at least  $t_{m,F}$  can be determined by computing  $\mathcal{R}_{[t_{m,F}, \infty]}^\gamma = \mathcal{R}_F^\sharp(t_{m,F})$ . Also, the set of states which can be controlled back to level flight within at least  $t_{m,B}$  can be obtained by computing  $\mathcal{R}_{[t_{m,B}, \infty]}^\gamma = \mathcal{R}_B^\sharp(t_{m,B})$ . The dynamical envelope of the aircraft is then the intersection between these two sets  $\mathcal{E}_{dyn} = \mathcal{R}_F^\sharp(t_{m,F}) \cap \mathcal{R}_B^\sharp(t_{m,B})$ . In this way, the manoeuvre envelope computed as the intersection of the forward and backward reachable set says something about all potential manoeuvres that can be made in  $t_{m,F}$  and can be controlled back to the safe set in at least  $t_{m,B}$  by computing the maximal reachable sets at  $t_{m,F}$  and  $t_{m,B}$ . Now that the framework is defined and all the reachable sets used here are maximal reachable sets, the  $\mathcal{R}^\sharp$  notation will be dropped for convenience.

To enable the use of the continual reachability framework as described above, it is essential that the state can remain in its initial set indefinitely, as shown in Figure 2(b). Therefore, this approach only works for dynamically coupled states. When the state cannot remain in the initial set, a different framework will be needed.

### III. Distance Fields Over Grids algorithm

In this section, the DFOG algorithm developed by Baier et al. in [19, 22] used for the computation of the reachable sets is explained. First it is explained how the DFOG method uses optimal

control problems to compute an approximation to the reachable set. Thereafter, it is explained how the optimal control problems are solved. Then, the numerical errors of the method and its computational complexity are discussed. Finally, some extensions to the algorithm are presented.

### A. Optimal control problem formulation

The reachable set is defined as the total collection of endstates of a dynamical system starting at a certain initial point, given that they are the result of an admissible control function.

$$\mathcal{R}(t_f, t_0, \mathbf{x}_0) := \left\{ \mathbf{y} = \mathbf{x}(t_f) \in \mathbb{R}^n \mid \exists \mathbf{u}(\cdot) \in \mathcal{U}_m, \exists \mathbf{x}(\cdot), \dot{\mathbf{x}}(t) = f(t, \mathbf{x}(t), \mathbf{u}(t)) \right\} \quad (5)$$

The DFOG method computes points belonging to this set by projecting points in the state space onto the reachable set. The projection onto the reachable set  $\mathcal{R}$  is defined as follows:

$$\Pi_{\mathcal{R}}(\mathbf{x}) := \{ \mathbf{y} \in \mathcal{R} \mid \|\mathbf{x} - \mathbf{y}\| = \text{dist}(\mathbf{x}, \mathcal{R}), \mathbf{x} \in \mathbb{R}^n \} \quad (6)$$

where  $\text{dist}(\mathbf{x}, \mathcal{R}) := \inf_{\mathbf{y} \in \mathcal{R}} \|\mathbf{x} - \mathbf{y}\|$ .

For any given point  $\mathbf{x} \in \mathbb{R}^n$ ,  $\Pi_{\mathcal{R}}$  yields a point  $\mathbf{y} \in \mathcal{R}$  closest to  $\mathbf{x}$ , thus effectively projecting  $\mathbb{R}^n$  onto  $\mathcal{R}$ . When  $\mathbf{x}$  is outside  $\mathcal{R}$ , it is projected orthogonally onto  $\mathcal{R}$  and when  $\mathbf{x}$  is inside  $\mathcal{R}$ , it maps onto itself. This definition is nice in theory, but has no practical use since the projection would need to be calculated for all states in the state space,  $\mathbb{R}^n$ . To make this practically computable, a grid is placed in the state space to which the computations will be restricted. This grid is defined as follows:

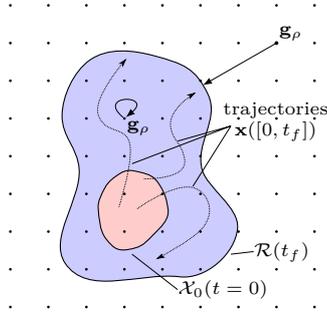
$$\mathcal{G}_{\rho} := \rho \cdot \mathbb{Z}^n \quad (7)$$

where  $\rho$  is the gridsize.

The projection at each grid point  $\mathbf{g}_{\rho}$  can then be computed by solving the following optimal control problem:

$$\begin{aligned} \text{Minimize} \quad & \frac{1}{2} \|\mathbf{x}(t_f) - \mathbf{g}_{\rho}\|_2^2 \\ \text{Subject to:} \quad & \dot{\mathbf{x}}(t) = f(t, \mathbf{x}(t), \mathbf{u}(t)) \quad \text{a.e. in } [t_0, t_f], \\ & \mathbf{x}(0) \in \mathcal{X}_0 \\ & \mathbf{u}(t) \in \mathcal{U}_m \quad \text{a.e. in } [t_0, t_f]. \end{aligned} \quad (8)$$

Solving the optimal control problems for all grid points,  $\mathbf{g}_\rho$ , results in a point cloud defining an approximation of the reachable set. A schematic representation of this can be seen in Figure 3. In [19], three strategies to use this point cloud are described. The strategy used here, is the one approximating the reachable set as the collection of all the projections of the grid points on the reachable set.



**Fig. 3 Concept of DFOG algorithm**

The resulting optimal control problem can be solved in many different ways, a number of which are described in the surveys, [23, 24]. Here the reduced direct transcription method, also known as single shooting, is used, as it yields a compact optimization problem which can be solved with standard nonlinear programming software.

### B. Problem discretization

In order to solve the optimal control problem, the state equation,  $\dot{\mathbf{x}} = f(t, \mathbf{x}, \mathbf{u})$  is discretized by a one-step method as follows:

$$\hat{\mathbf{x}}(t_{k+1}) = \hat{\mathbf{x}}(t_k) + h_k \Phi(t_k, \hat{\mathbf{x}}(t_k), \mathbf{u}(\cdot), h_k), \quad k = 0, 1, \dots, N - 1, \quad (9)$$

where  $\Phi$  is the increment function. This function can be seen as the ‘mean’ gradient of  $\mathbf{x}(t)$  over the interval  $[t_k, t_{k+1}]$  and it links the state at the beginning and end of the interval. Here, the increment function of the discretization is chosen to be an explicit RK method. The explicit RK method is chosen because, unlike implicit RK methods, it is deterministic and allows the computation of the derivatives of the steps with respect to the state and control inputs. These derivatives are required for efficiently solving the resulting minimization problem. For the use in optimal control, an RK

scheme has to fulfill a number of conditions, as described in [25, 26].

Also the control function is parameterized as follows:

$$\mathbf{u}(\cdot) = u_M(t; \mathbf{w}), \quad (10)$$

where  $\mathbf{w}$  is a vector containing the  $M$  parameters that describe the discretized control function. The control function is discretized on the same mesh as the trajectory. Here it is chosen to use a piecewise linear discretization of the control function, because it allows for the simple implementation of, for instance, rate constraints on the control function. The piecewise constant approximation can be seen as a 1<sup>st</sup> order B-spline. When higher order continuity of the control functions is desired, higher order B-splines could be used to discretize the control function. The choice of a control parameterization function limits the set of possible control functions. However, since the set of parameterized control functions is a subset of all admissible controls, this parameterization will result in a conservative approximation of the reachable set.

Now that both the state trajectory and the control function are discretized, the state at each discrete step can be described as a function of only the control parameters and the parameters defining the initial state as follows:

$$\begin{cases} \mathbf{x}_0 = p(\mathbf{z}_0) \\ \mathbf{x}_{k+1} = \mathbf{x}_k + h\Phi(\mathbf{x}_k, \mathbf{w}, h) \quad k = 0, \dots, N-1 \end{cases} \quad (11)$$

This furthermore allows the cost function and all constraints of the optimal control problem to be described as a function of the following parameters:

$$\mathbf{z} := (\mathbf{z}_0, \mathbf{w})^\top \in \mathbb{R}^{n_{z_0} + M}. \quad (12)$$

With this, the optimal control problem, Eq. (8), is transformed in a dense nonlinear programming problem of  $\mathbf{z}$  as follows:

$$\begin{aligned} \text{Minimize} \quad & J = \frac{1}{2} \|\mathbf{x}_N - \mathbf{g}_\rho\|_2^2, \\ \text{Subject to:} \quad & \mathbf{z}_{min} \leq \mathbf{z} \leq \mathbf{z}_{max}, \end{aligned} \quad (13)$$

with  $\mathbf{g}_\rho$  the gridpoints and  $\mathbf{z}_{min/max}$  the bounds of the optimization parameters  $\mathbf{z}$ , Eq. (12).

However, methods to solve this optimization problem require the gradient of the cost with respect to the optimization parameters for efficient solution. Since the dynamics of the systems

are now implicitly integrated, determining the effect of changing the control parameters becomes a complex task. Different approaches exist to determine these effects. The two major approaches are the ‘Sensitivity ODE’ approach, which mainly depends on the number of parameters and the ‘Adjoint ODE’ approach which mainly depends on the number of nonlinear constraints, [27, 28]. Since no nonlinear constraints are present, the ‘Adjoint ODE’ approach, [29], is the most suitable and thus the one used. If nonlinear state and/or control constraints are present, an adjoint trajectory has to be computed for every constraint, which can make this method inefficient. In this case, the sensitivity approach needs to be used.

The key part of this approach is the backward integration of the adjoint ODE. For the adjoint method, a Lagrangian approach is used and the cost function is changed in the following way:

$$J_a = J + \sum_{i=0}^{N-1} \boldsymbol{\lambda}_{i+1}^\top (\mathbf{x}_{i+1} - \mathbf{x}_i - h_i \Phi_i) \quad (14)$$

When computing the gradient of this cost function, the adjoint variable,  $\boldsymbol{\lambda}$ , is used to eliminate the sensitivity of the endpoint,  $S_N$ , and thus circumvent the need of solving the sensitivity ODE. For this, the adjoint variable has to satisfy the following trajectory:

$$\left\{ \begin{array}{l} \boldsymbol{\lambda}_N^\top = J' \\ \quad = -(\mathbf{x}_N - \mathbf{g}_\rho)^\top \\ \boldsymbol{\lambda}_k^\top = \boldsymbol{\lambda}_{k+1}^\top - h \boldsymbol{\lambda}_{k+1}^\top \Phi'_x(\mathbf{z}) \end{array} \right. \quad (15)$$

This adjoint trajectory can then be used to compute the influence of the different parameters as follows:

$$J'(\mathbf{z}) = J'_a = -\boldsymbol{\lambda}_0^\top p'(\mathbf{z}) - \sum_{k=0}^{N-1} h \boldsymbol{\lambda}_{k+1}^\top \Phi'_z[t_k] \quad (16)$$

If state and/or control constraints are present, an adjoint trajectory has to be computed for every constraint, which can make this method inefficient. In this case, the sensitivity approach needs to be used.

In this research, the cost function is implemented as a MATLAB<sup>®</sup> function which also computes and returns the derivatives if required. The resulting optimization problems are solved with the SQP-algorithm in the *fmincon* function of MATLAB<sup>®</sup>. For practical use, a dedicated implementation is advised.

### C. Numerical errors in the algorithm

To obtain results which are as accurate as possible, it is important to investigate the sources of error. The results obtained with the DFOG method suffer errors from three major causes. These are: the errors resulting from the control parameterization, the errors resulting from the numerical integration of the trajectories, and the errors resulting from approximating the reachable set with a point cloud. These errors will be shortly discussed below and are illustrated at the hand of the reachable sets for the following system, also used in [19]:

$$\dot{\mathbf{x}} = \begin{bmatrix} \pi x_2 \\ -\pi x_1 u \end{bmatrix}, \quad \mathbf{x}(0) = \begin{bmatrix} -1 \\ 0 \end{bmatrix}, \quad u(t) \in [0, 1], \quad t \in [0, 1] \quad (17)$$

First there is the integration error, which is the error made in the numerical integration of all the individual trajectories. The error of interest here is the global error, being the accumulation of the local truncation and round-off errors at each step in the integration. It is impossible to pose any meaningful bounds on this error, it can however be estimated at extra computational cost, [30, 31]. Since this error is random in state-space, it can lead to either an under- or overapproximation of the reachable set. The magnitude of this error depends on the chosen numerical integration method and the chosen step-size in the integration. The effect of using different RK methods on the integration error can be seen in Figure 4(a).

The second contributor to the total approximation error is the control function, which stems from two sources. First, the control functions are limited to a set of piecewise continuous functions in the discretization of the optimal control problems. These discretized control functions are used to approximate the continuous optimal control function. The accuracy of this approximation depends on the number of steps in the discretization, which can be seen in Figure 4(b). Second, the control function is the result of an optimization procedure, thus it is possible that it is not yet entirely optimal or it can be stuck in a local optimum. This error also interacts with the integration error in the way that a large integration error will make the optimization process harder, yielding a larger control error. Despite these errors, the discrete control functions are limited to feasible controls, so even if they are not entirely optimal, they yield a trajectory that ends inside the reachable set. This results in the fact that with control errors only, the result of the reachability computation would be an underapproximation of the true reachable set. This error is hard to quantify, since the control

function approximation error also passes through the numerical integration to yield the total control error.

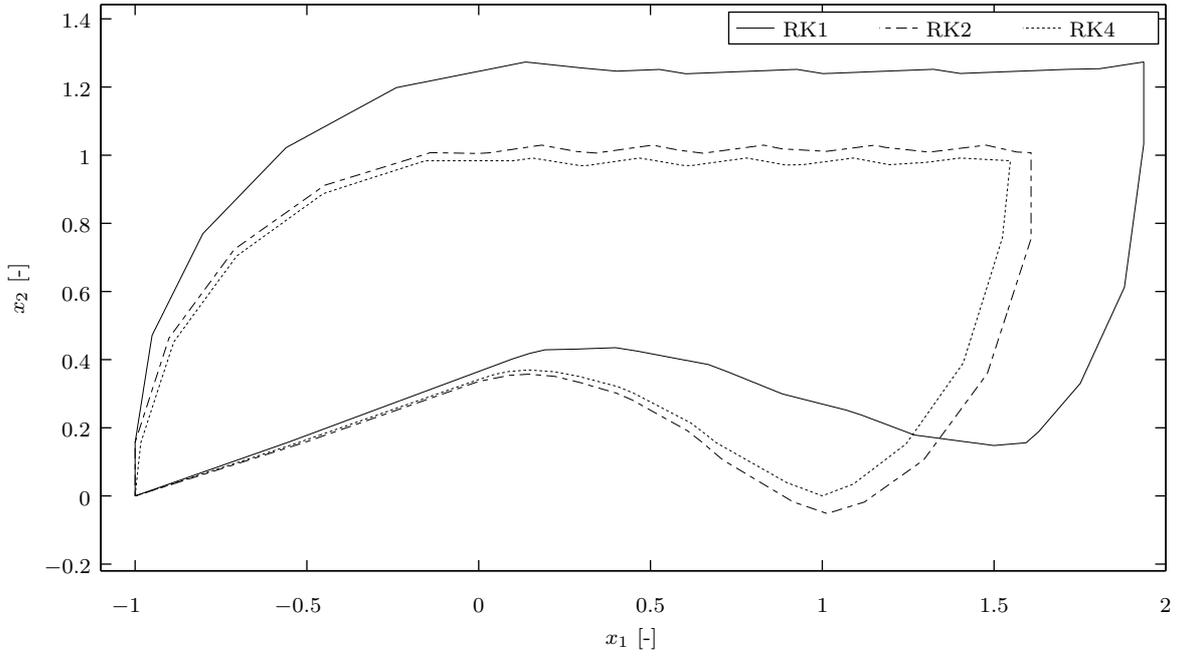
Another error in the approximation of the reachable set, results from the approximation of the continuous boundary of the set by a point cloud, which is denoted as the representation error. This error is highly dependent on the geometry of the set and should be defined in terms of curvature of the boundary and the distances between the points in the cloud. On non-convex parts of the set, a slight over-approximation of the set will be made and on convex parts, the set is under-approximated. This error can easily be bounded by the grid resolution but will be much smaller in practice.

The combination of the three errors discussed above yields the total error in the approximation. This results in an approximation of the reachable set that is neither an under- or overapproximation, which converges to the real set with increasingly accurate computations. As stated above, the different errors depend on different parameters. The integration error depends on the RK method used for the discretization and the number of steps in the integration. The control error depends solely on the number of steps in the integration and the approximation error only depends on the grid step. For an efficient algorithm, the contributions of the three errors to the total error have to be balanced, so that no effort is wasted on a non-dominant error source. First, the integration and control error have to be balanced by choosing the numerical integration method. Afterwards, these errors have to be balanced with the representation error by relating the integration step-size and grid resolution. This is investigated further in Section III E.

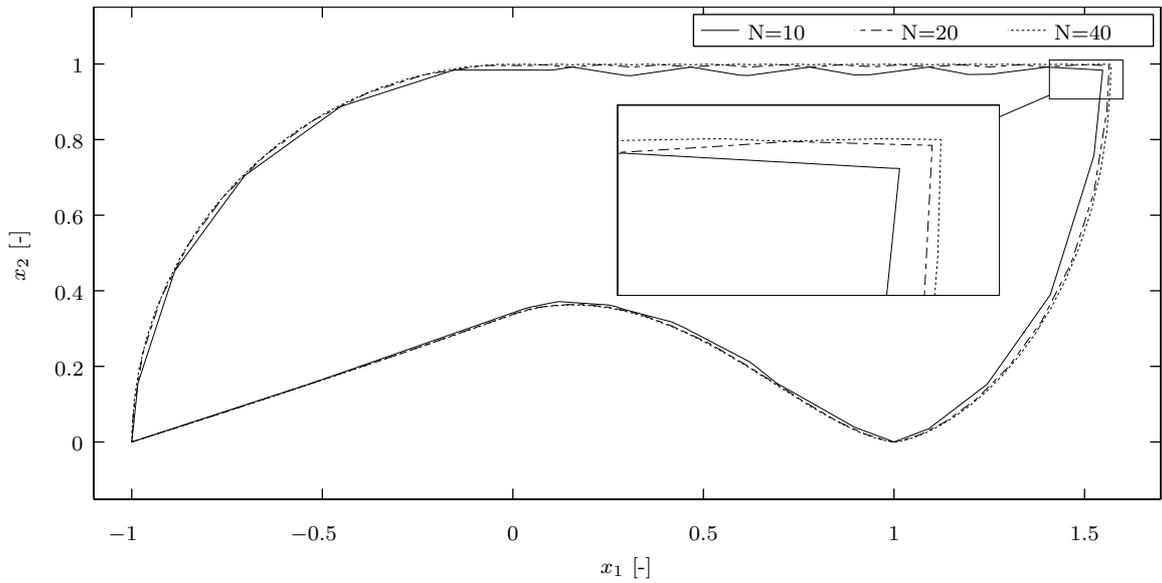
#### **D. Complexity & Scalability**

There are four main parameters that determine the total computational complexity of the algorithm. These parameters are the simulation time, the grid resolution, the number of integration steps and the state dimension.

The simulation time determines the size of the reachable set and thus the number of grid points that need to be computed. However, the influence of this parameter on the complexity is rather convoluted. For a stable linear system, the forward set will converge to a bounded set, but



(a) Integration error (computed with  $N=10$ , RK1 = 1<sup>st</sup> order Euler's method, RK2 = 2<sup>nd</sup> Heun's method, RK4 = classic 4<sup>th</sup> order RK method)



(b) Control error (computed with RK4 for state discretization, to reduce the effects integration errors as much as possible)

**Fig. 4 Numerical errors in the DFOG method ( $\mathcal{R}(t = 1s)$  for the system in Eq. (17) )**

the backward set will diverge to the entire state-space and vice-versa for unstable systems. For nonlinear systems this relation is much more complicated and impossible to determine in advance, but it is safe to say that with increasing time the complexity will increase, possibly exponential.

The grid resolution has a polynomial complexity in the form of  $\mathcal{O}(\rho^n)$ , with  $n$  the state dimen-

sion.

The number of integration steps influences the complexity in two ways. First there will be more decision variables, making the optimization harder to solve. The SQP-algorithm used here is assumed to have an, on average, polynomial complexity w.r.t. the decision variables. Second, because there are more integration steps, each evaluation of the cost function will require more evaluations of the model. This influence is a linear one, so the total complexity remains polynomial on average.

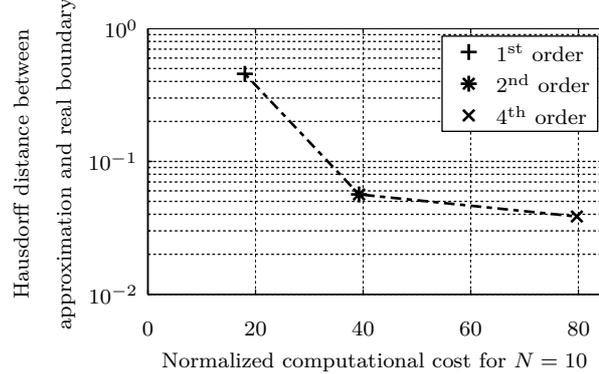
Inherent to the use of a grid is the exponential complexity w.r.t. the state dimension, which is known as the curse of dimensionality. This problem is present in most reachability algorithms and prevents their use for higher state dimensions. The key point in making practical reachability algorithms will be solving this issue. Either by developing an algorithm without grids, or developing a way to recombine the reachable sets of decoupled dynamics.

#### **E. Algorithm extensions**

The DFOG algorithm essentially consists of two components. First there is the solver used to solve the optimal control problems at each grid point. Second there is the problem of choosing the grid points used in the approximation. As a contribution, two improvements were made to the algorithm, which increase the efficiency and the accuracy of the algorithm

First, different RK methods were investigated for the discretization of the trajectory. This resulted in the choice for the 2<sup>nd</sup> order Heun method. This method is chosen as a compromise between accuracy and speed. It has a much higher accuracy than the first order Euler method used in [19]. The 4<sup>th</sup> order RK method was also tested and this reduced the integration error significantly, making the control error dominant. The disadvantage of this is the significantly higher computational cost, as the ODE has to be evaluated more and more iterations of the optimization routine are required. Still, the resulting optimized control sequences were not very different from those obtained with the 2<sup>nd</sup> order method. This makes the 2<sup>nd</sup> order method a better choice for discretizing the trajectory, which can be seen in Figure 5. Here the average number of function evaluations required to solve the optimal control problems is plotted against the Hausdorff distance

between the approximated boundary of the reachable set and the boundary of the reachable set computed with a much higher accuracy. To further increase the accuracy, it is believed that a final



**Fig. 5 Accuracy versus function cost**

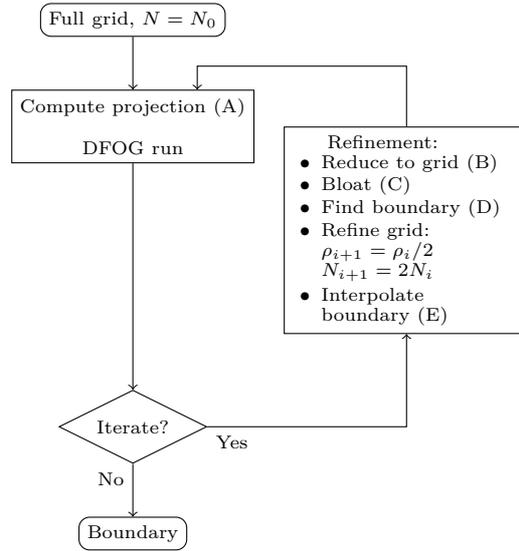
reintegration of the optimized control sequences would be the best option.

Second, a new approach for selecting the grid points was developed. Since solving the optimal control problems at each grid point is very costly, one way of improving the efficiency of the algorithm is to optimize the grid points used. Under fairly general assumptions, it can be assumed that the reachable set is a connected, compact set. In this case, the reachable set can be described by its boundary. Therefore a new strategy is developed that aims to construct only an approximation of the boundary. The benefit of this, is a greatly reduced number of grid points.

The most efficient way to compute an accurate estimate of the boundary would be to start with a set of grid points neatly distributed around the reachable set. The DFOG algorithm projects these points onto the reachable set and this would yield an accurate estimate. The problem is that the boundary has to be known for this, creating a chicken/egg problem. Here, the observation is made that computing an approximation with a lower resolution already gives a reasonable idea of the boundary. This can then be used to generate a set of grid points for the computation of a higher resolution approximation to the boundary. The low resolution computation yields an approximation of the boundary. Then it is known that the actual boundary of the set will be in some uncertainty band around this estimate. The size of this uncertainty band depends on the integration and control errors of the low-resolution discretization. A new set of grid points is then created at the outer boundary of the this band and these are outside the reachable set. Running the DFOG algorithm with these grid points will then create a more accurate approximation of the

boundary. This leads to an iterative approach which consecutively refines the approximation of the reachable set's boundary.

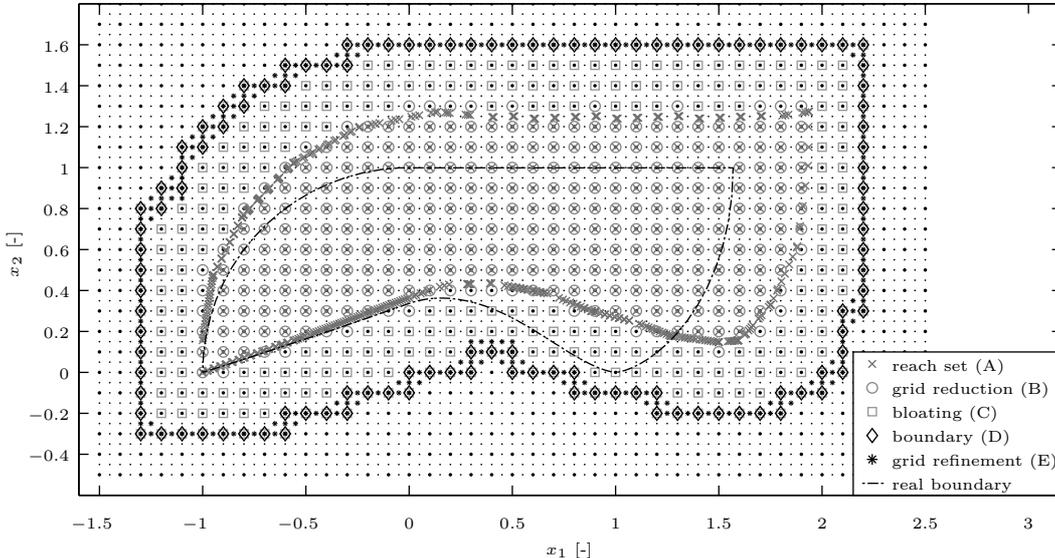
This iterative approach is now explained in more detail at the hand of the example used earlier, (17). A diagram of the steps is shown in Figure 6 and they are further illustrated with the first step for the test system in Figure 7. Note here that the letters next to the steps in the figures will return in the text to indicate the respective steps.



**Fig. 6 Iterative approach to determine the reachable set's boundary**

First, the low resolution approximation, (A), is computed. For the refinement procedure, the projection of the grid points is converted to a  $n$ -D binary image on the grid, (B), which allows the use of image processing software for its manipulation. For this, the nearest grid point is computed for all projected points and they form a binary image w.r.t. the grid. This image is then used for further processing. The image processing software needs a filled object, e.g. a disk instead of a circle, so a filling operation is performed when needed. Now the image is expanded outwards with an arbitrary number of gridsteps, (C), so that its boundary is outside the reachable set. The software does this by replacing each point in the image with an array of points and then it unites these to form the new image. Afterwards the boundary of the bloated image is computed, (D). Then the grid step and time step are halved to increase the accuracy of the next iteration. Finally the image's boundary is transformed from the low resolution grid to the refined grid, (E). These boundary points will then

serve as the grid points for a new, more accurate DFOG run. This process can then be iterated. Note that in this way relatively few grid points of the refined grid are actually used for the higher resolution DFOG computation. This reduction justifies the cost of computing the low resolution approximation.



**Fig. 7** First step of the iterative approach. ( $\mathcal{R}(t = 1s)$  for the system in Eq. (17). For clarity, the 1<sup>st</sup> order approximation is shown here.)

#### IV. F16 model

The DFOG method will be used to determine the forward and backward reachable sets of a high-fidelity F-16 model. Because reachability calculations are computationally expensive and this only becomes worse in higher dimensions, the complexity of the model is reduced by only considering the longitudinal part of it. Here the same model as used in [16] is used. This is a four-dimensional model with the following states: the true airspeed,  $V_t$  [m/s], the angle of attack,  $\alpha$  [rad], the pitch rate,  $q_b$  [rad/s] and the second quaternion component,  $q_2$  [-]. Two other quaternion components ( $q_1, q_3$ ) remain 0 since only longitudinal flight is considered and the final component depends directly on  $q_2$  as in Eq. (22). The control variables are the elevator deflection,  $\delta_e$ , limited to  $\pm 25$  deg, and the thrust,  $T$ , limited between 0 and 75000 Newton. The leading edge flap available in the model is not used and set to zero.

## A. Dynamics

The longitudinal equations of motions of the F-16 are:

$$\dot{V}_t = \frac{1}{m} [-D + T \cos \alpha + mg_1], \quad (18)$$

$$\dot{\alpha} = q_b + \frac{1}{mV_t} [-L - T \sin \alpha + mg_3], \quad (19)$$

$$\dot{q}_b = \frac{M}{I_{yy}}, \quad (20)$$

$$\dot{q}_2 = \frac{1}{2} q_b q_0. \quad (21)$$

Where:

$$q_0 = \sqrt{1 - q_2^2} \quad (22)$$

$$g_1 = g_0 [-2q_0 q_2 \cos \alpha + (q_0^2 - q_2^2) \sin \alpha], \quad (23)$$

$$g_3 = g_0 [2q_0 q_2 \sin \alpha + (q_0^2 - q_2^2) \cos \alpha]. \quad (24)$$

The lift, drag, and pitching moment are computed from the look-up tables in [32]. For this, multivariate tensor product spline interpolation instead of piecewise linear interpolation was used, so that continuous analytic derivatives of pitch, drag, and pitching moment can be computed.

## B. Trimcurve

In Section II, safe flight was defined as trimmed, straight, and level flight. This means that the trim curve will be used as the initial set for the optimal control problems used to compute the reachable sets. Therefore, it has to be encoded as a set of constraints on the decision variables. Furthermore, these constraints have to be differentiable, so that the influence of changing the initial state on the cost function can be computed. For efficiency, the trimcurve is parameterized, so that a trimmed initial point can be obtained with one optimization parameter. This parameterization has to be differentiable so that the influence of changing the parameter(s) on all components of the initial state can be computed analytically. First, the trimcurve of the F-16 model was determined by solving the trim equations for  $V_t \in [50, 200]$  at 0m altitude with a 1  $m/s$  increment. This was done with a simple nonlinear optimizer, but more robust and efficient methods are available, e.g. [33, 34]. The obtained trim curve can be seen in Figure 8. Then the curve was parameterized as

function of airspeed by spline interpolation. A less computationally intensive approximation can be obtained by carefully fitting a spline with less intervals to the trimcurve. It is assumed that any remaining discrepancy between the interpolated curve and the real trimcurve falls inside the reachable set, as it is really close to the equilibrium curve. Since quaternions are used in the model, a separate spline was made for the angle of attack and the attitude.

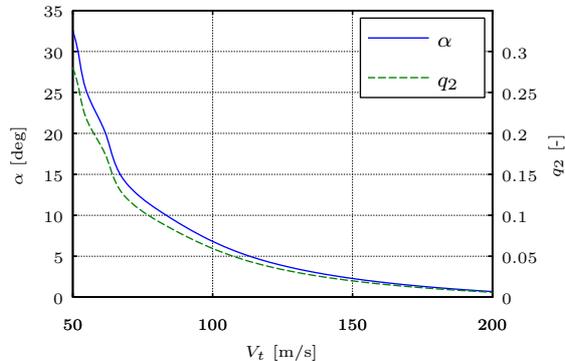


Fig. 8 F-16 trim curve for straight and level flight at sea-level

## V. Application of DFOG to envelope determination of the F-16

First, the setup of the computation is explained. Then, the results obtained with the DFOG method are shown and discussed. Thereafter, the results obtained with the DFOG method are compared to those obtained with the level set method in [6].

### A. Experiment setup

The computation of the reachable set is carried out with the extended DFOG method as described in Section III E. The computations will be limited to the domain shown in Table 1. This domain is chosen such that comparisons with the sets obtained in [6] can be done. For the same reason and for the sake of keeping the computations practically possible, the end time is set to  $t_e = 1$  s.

Computations with the DFOG method should be done on a square grid, so that the optimization problems resulting from the optimal control problems are conditioned as well as possible. For this, an initial grid is defined with certain step sizes for each state, also shown in Table 1. This grid

**Table 1 Computational domain of the reachability computations**

Variable	lower bound	upper bound	initial step size
$V_t$ [m/s]	50	200	10
$\alpha$ [deg]	-20	50	5
$q_b$ [deg/s]	-60	60	10
$q_2$ [-]	-1	1	0.2

is then scaled to a square grid, which still leaves a free parameter in the scaling matrix. This parameter is chosen such that the state grid step coincides with the time step. Furthermore, both control inputs were scaled to the  $[-1, 1]$  interval for further conditioning of the optimization. In the initial computation, the trajectory and control function is divided in 25 steps. Thereafter, two refinements are made, each doubling the number of steps, yielding a final number of 100 steps.

## B. Results

### 1. forward reachable set

From the forward reachable set, shown in Figs. 9 and 10, some logical results that can be expected from flight dynamics can be seen. First of all, it can be seen that higher angles of attack can be reached for positive pitch rates and lower angles of attack for negative pitch rates. Furthermore, one can see that the angle of attack,  $\alpha$ , and the pitch angle,  $\theta$  are closely linked. This is natural, as the aircraft starts from level flight and then begins changing its attitude. In Figure 9(b), it can also be seen that the reachable set is very narrow in the  $(\alpha-\theta)$ -plane. This means that for a certain pitch angle, only a limited amount of flight path angles can be obtained within a second. However, since the mean relation between  $\alpha$  and  $\theta$  is slightly different from one-to-one, still a reasonable range of flight path angles can be obtained when considering all pitch angles. Naturally, positive flight path angles correspond to positive pitch angles and they lag behind the pitch angle as it takes time to change the trajectory of the aircraft.

### *2. backward reachable set*

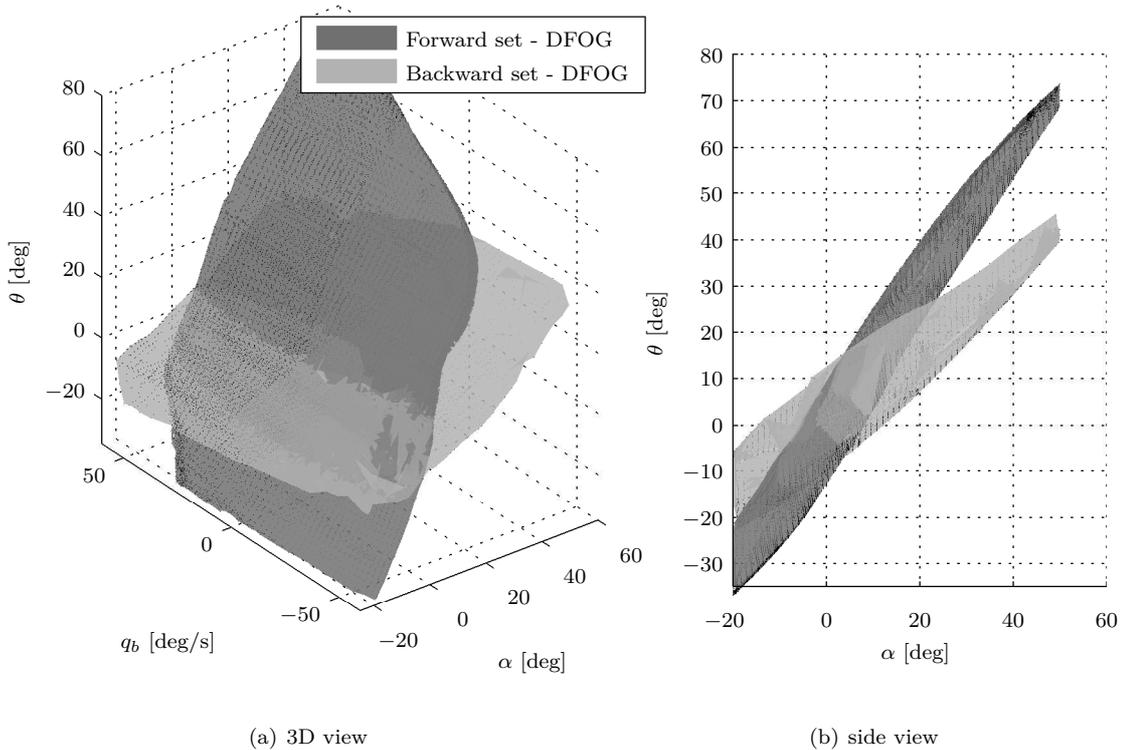
For the backward reachable set, shown in Figs. 9 and 10, similar relations as for the forward reachable set can be observed. One major difference here is that the highest values of the pitch angle,  $\theta$ , are combined with the most negative pitch rates. At first, this might seem contradictory, but after realizing that the backward set are the states that can be controlled back to the equilibrium, it makes sense. To recover from a strongly negative pitch rate, one must pull-up the aircraft. However, during this process, the pitch angle will decrease as the pitch rate is being brought back to zero. Therefore, strongly negative pitch rates can only be returned to trimmed level flight when starting at a large pitch angle. This effect is most noticeable in the reachable sets at lower airspeeds, as it is visible in the domain, due to the smaller manoeuvrability at these speeds. For higher airspeeds, this effect is still present, but not really visible as the higher pitch rates are cut off by the domain boundaries.

### *3. backward set larger than forward*

From the figures 9 and 10, it can be seen that the backward reachable sets are typically larger than the forward reachable sets. This is due to the fact that the dynamics of the F-16 aircraft are still mostly stable. When evolving these dynamics backward, they become unstable and as such, a much larger set can be obtained. In terms of safety, this is beneficial as the backward reachable set represents the states that can be steered back to an equilibrium state.

## **C. Comparison with level set data**

Here, a comparison is made to the results obtained with the level set method as described in [6]. These results were obtained with the same model used here, only a slightly different initial set was used. The initial (final) set used in the level set approach is the same trimcurve as used here, except that it was not implemented as a curve through 4D-space. The methods used to solve the level set problem require that at least one node of the grid falls inside the initial set to enable the propagation of the interface in time. For this reason, the trimcurve was bloated by adding a certain weighted distance to the trimcurve. This creates a 4D tube around the trimcurve which is used as

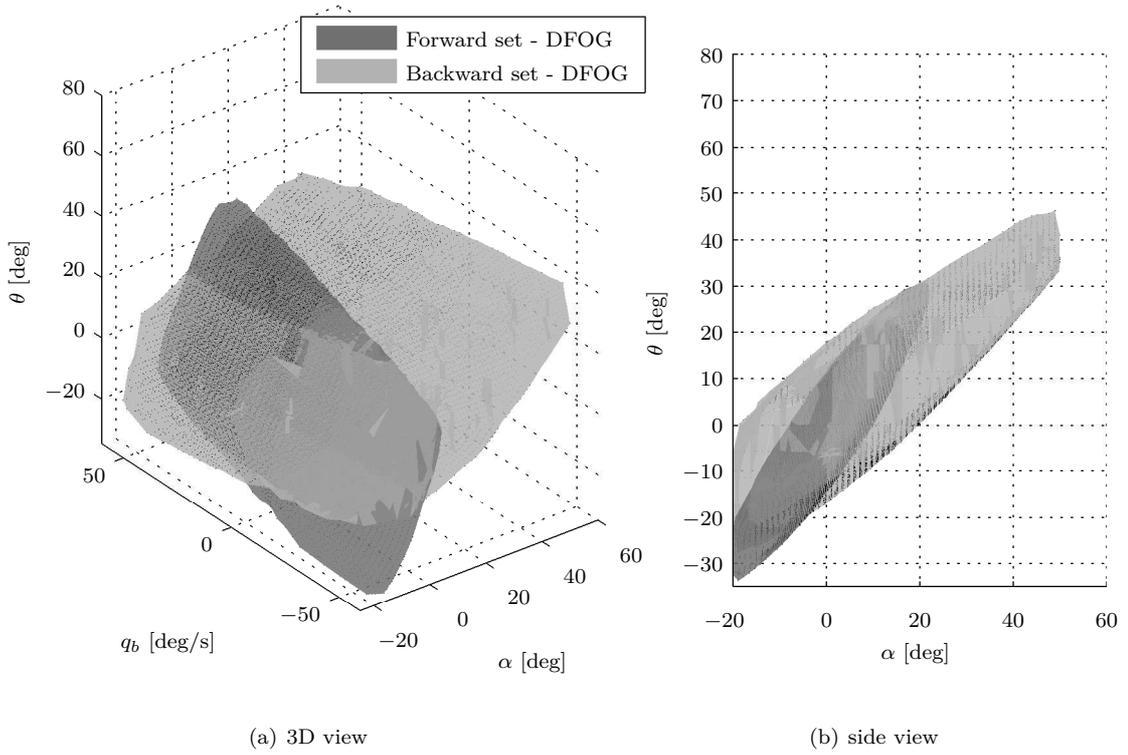


**Fig. 9 Cross-section of the reachable sets of the F-16 model at  $V_t = 150$  m/s ( $t_f = 1$ s,  $dt = 0.01$ s, RK2 discretization)**

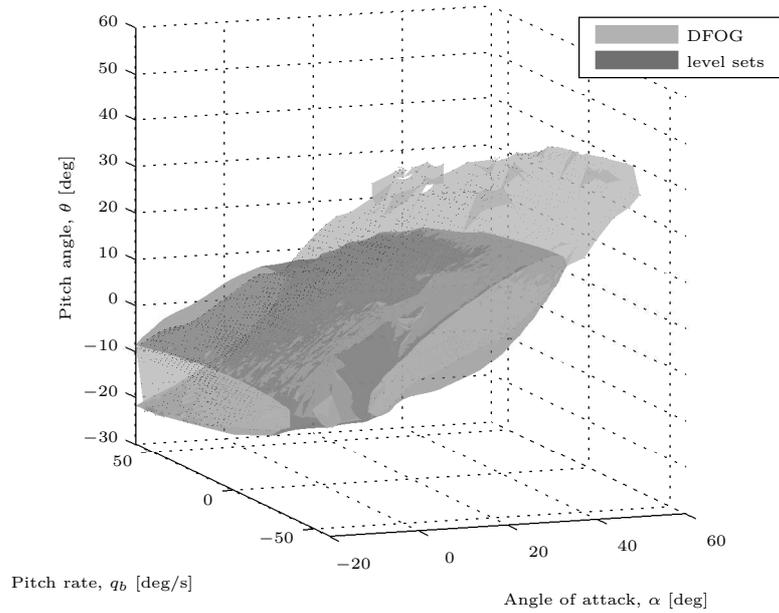
the initial set. In [6], it is assumed that the states added in this way are all reachable.

From Figure 11, it can be seen that the reachable set obtained with the level set method are considerably smaller than that obtained with the DFOG method, although it is somewhat thicker. This difference in thickness can easily be explained by the use of different initial sets. One possible explanation for the discrepancy between the level sets and DFOG results is given in [17, Section 3]. Here it is stated that level sets yield conservative results due to the need of a dissipation function to guarantee numerical stability.

Another advantage of the DFOG method is that it readily generates the control sequences and trajectories of the points making up the boundary of the reachable set. Especially when the method is used for performance assessment and the study of the dynamics of an aircraft, these can yield very useful information. For instance, these outputs can be used to identify which control sequences will steer the aircraft into an unrecoverable state. Thereafter, it can be made sure that these control sequences are avoided. Also an advantage of the DFOG method is that with some extra computational and storage costs, an accurate estimate of the error in the approximation of



**Fig. 10** Cross-section of the reachable sets of the F-16 model at  $V_t = 200 \text{ m/s}$  ( $t_f = 1\text{s}$ ,  $dt = 0.01\text{s}$ , RK2 discretization)



**Fig. 11** Comparison of the cross-section of the backward reachable set of the F-16 model at  $V_t = 150 \text{ m/s}$  ( $t_f = 1\text{s}$ ,  $dt = 0.01\text{s}$ ). The level set data are taken from [6].

the reachable set can be obtained. This estimate could be used to shrink the resulting set, to ensure

that the remaining set is actually reachable.

Both algorithms have their own specific problems considering computational speed. The level set method only has to compute the local optimal control, but it has to approximate the state and costate over the entire domain and then propagate the entire interface forward in time. This can be done more efficiently with the adaptive approach developed in [6]. For the DFOG method, an optimal control problem has to be solved at every grid point, which is computationally expensive. Again, the number of grid points can be reduced by using an adaptive method.

At the moment, the level set implementation is computationally less intensive compared with the DFOG implementation. A more efficient implementation of the DFOG algorithm could significantly reduce the computational cost and make it more tractable.

## VI. Conclusions

In this paper, the newly proposed Distance Fields Over Grids algorithm is extended and investigated by applying it to the computation of the safe manoeuvring envelope of a nonlinear high-fidelity F-16 model. This has shown that Distance Fields Over Grids is capable of generating an accurate approximation to the forward and backward reachable sets. The obtained results provide a less conservative estimate than previously obtained results.

The advantage of the Distance Fields Over Grids method is that generates an accurate approximation to the reachable set in an easily implemented way. Also, the current framework allows easy implementation of (non)linear state and/or control constraints. Furthermore, the state and control trajectories corresponding to the points describing the boundary of the reachable set can also be obtained. This allows the possibility of detailed analysis of specific parts of the boundary. The nature of the algorithm makes it highly parallelizable, as it consists of a large number of independent optimal control problems.

The algorithm also has a number of drawbacks. There is the computational cost and the increase thereof when enlarging the time horizon. Also the algorithm produces an approximation which is neither an over- or underapproximation and no simple way exists to alter the algorithm such that one is produced. Finally, no disturbances in the model or disturbance inputs can be taken into

account in the current framework.

The computation of safe envelopes with reachability algorithms is currently very computationally intensive. Therefore they will not be useful for online applications in the near future. At the moment, the benefits of these computations lie in the design phase of aircraft. Here they can be used to assess the performance of different configurations and the remaining performance in the case of failures. Furthermore, they can aid in controller design. This can be done in two ways. First, the combination of the forward and backward reachable envelopes, allows to identify potentially hazardous control sequences that can steer the aircraft to unrecoverable states. This knowledge can be used in the design of adequate envelope protection systems. Second, the obtained backward reachable set can be compared with the region of attraction of different controller designs as an extra performance metric.

### References

- [1] “Annual Safety review,” Tech. rep., EASA, 2011.
- [2] Netherlands Aviation Safety Board, “El Al flight 1862,” Tech. rep., 1994.
- [3] Lombaerts, T., *Fault Tolerant Flight Control*, Phd thesis, Delft University of Technology, 2010. Section 1.2.4, p. 32-34.
- [4] Smaili, M. and Mulder, J., “Flight data reconstruction and simulation of the 1992 Amsterdam Bijlmermeer airplane accident,” in “Modeling and Simulation Technologies Conference,” American Institute of Aeronautics and Astronautics, Reston, Virginia, August, 2000, doi:10.2514/6.2000-4586.
- [5] Ruijgrok, G., *Elements of Airplane Performance*, Delft University Press, 2004. Section 10.4, p. 261-262.
- [6] van Oort, E. R., *Adaptive Backstepping Control and Safety Analysis For Modern Fighter Aircraft*, Phd thesis, Delft University of Technology, 2011. Part III, p. 215-257.
- [7] Hu, T., Qiu, L., and Lin, Z., “The controllability and stabilization of unstable LTI systems with input saturation,” in “Proceedings of the 36th IEEE Conference on Decision and Control,” IEEE, Vol. 5, 1997, pp. 4498–4503, doi:10.1109/CDC.1997.649679.
- [8] Goman, M. G. and Demenkov, M. N., “Computation of Controllability Regions for Unstable Aircraft Dynamics,” *Journal of Guidance, Control, and Dynamics*, Vol. 27, No. 4, 2004, pp. 647–656, doi:10.2514/1.4038.

- [9] Guernic, C. L. and Girard, A., “Reachability analysis of linear systems using support functions,” *Non-linear Analysis: Hybrid Systems*, Vol. 4, No. 2, 2010, pp. 250–262, doi:<http://dx.doi.org/10.1016/j.nahs.2009.03.002>.
- [10] Kurzhanskiy, A. a. and Varaiya, P., “Ellipsoidal Toolbox (ET),” in “Proceedings of the 45th IEEE Conference on Decision and Control,” IEEE, 2006, pp. 1498–1503, doi:[10.1109/CDC.2006.377036](https://doi.org/10.1109/CDC.2006.377036).
- [11] Girard, A., “Reachability of Uncertain Linear Systems Using Zonotopes,” in “Hybrid Systems: Computation and Control,” Springer Berlin Heidelberg, Vol. 3414 of *Lecture Notes in Computer Science*, pp. 291–305, 2005, doi:[10.1007/978-3-540-31954-2\\_19](https://doi.org/10.1007/978-3-540-31954-2_19).
- [12] Hwang, I., Stipanović, D. M., and Tomlin, C. J., “Polytopic approximations of reachable sets applied to linear dynamic games and a class of nonlinear systems,” in Abed, E., ed., “Advances in Control, Communication Networks, and Transportation Systems,” Birkhäuser Boston, Systems and Control: Foundations & Applications, pp. 3–19, 2005, doi:[10.1007/0-8176-4409-1\\_1](https://doi.org/10.1007/0-8176-4409-1_1).
- [13] Dang, T., Maler, O., and Testylier, R., “Accurate hybridization of nonlinear systems,” in “Proceedings of the 13th ACM international conference on Hybrid systems: computation and control - HSCC '10,” ACM Press, New York, New York, USA, 2010, p. 11, doi:[10.1145/1755952.1755956](https://doi.org/10.1145/1755952.1755956).
- [14] Mitchell, I., Bayen, A., and Tomlin, C., “A Time-Dependent Hamilton-Jacobi Formulation of Reachable Sets for Continuous Dynamic Games,” *IEEE Transactions on Automatic Control*, Vol. 50, No. 7, 2005, pp. 947–957, doi:[10.1109/TAC.2005.851439](https://doi.org/10.1109/TAC.2005.851439).
- [15] Mitchell, I. M., “The Flexible, Extensible and Efficient Toolbox of Level Set Methods,” *Journal of Scientific Computing*, Vol. 35, No. 2-3, 2007, pp. 300–329, doi:[10.1007/s10915-007-9174-4](https://doi.org/10.1007/s10915-007-9174-4).
- [16] van Oort, E. R., Chu, Q. P., and Mulder, J. A., “Maneuver Envelope Determination through Reachability Analysis,” in “Advances in Aerospace Guidance, Navigation and Control,” pp. 91–102, 2011, doi:[10.1007/978-3-642-19817-5\\_8](https://doi.org/10.1007/978-3-642-19817-5_8).
- [17] Lombaerts, T., Schuet, S., Wheeler, K., Acosta, D. M., and Kaneshige, J., “Safe Maneuvering Envelope Estimation based on a Physical Approach,” in “AIAA Guidance, Navigation, and Control (GNC) Conference,” American Institute of Aeronautics and Astronautics, Boston, MA, 2013, pp. 1–20,

doi:10.2514/6.2013-4618.

- [18] Baier, R., Büskens, C., Chahma, I. a., and Gerdtts, M., “Approximation of reachable sets by direct solution methods for optimal control problems,” *Optimization Methods and Software*, Vol. 22, No. 3, 2007, pp. 433–452,  
doi:10.1080/10556780600604999.
- [19] Baier, R., Gerdtts, M., and Xausa, I., “Approximation of reachable sets using optimal control algorithms,” *Numerical Algebra, Control and Optimization*, Vol. 3, No. 3, 2013, pp. 519–548,  
doi:10.3934/naco.2013.3.519.
- [20] de Weerd, E., van Oort, E., Van Kampen, E.-J., Chu, P., and Mulder, J., “Level Set Method Based on Interval Analysis,” in “AIAA Guidance, Navigation, and Control Conference,” American Institute of Aeronautics and Astronautics, Reston, Virginia, August, 2011, pp. 1–19,  
doi:10.2514/6.2011-6258.
- [21] Kaynama, S., Oishi, M., Mitchell, I. M., and Dumont, G. a., “The continual reachability set and its computation using maximal reachability techniques,” in “IEEE Conference on Decision and Control and European Control Conference,” IEEE, Vol. 0, 2011, pp. 6110–6115,  
doi:10.1109/CDC.2011.6161424.
- [22] Baier, R., Büskens, C., Chahma, I. A., and Gerdtts, M., “Approximation of reachable sets by direct solution methods for optimal control problems,” *Optimization Methods and Software*, Vol. 22, No. 3, 2007, pp. 433–452,  
doi:10.1080/10556780600604999.
- [23] Betts, J. T., “Survey of Numerical Methods for Trajectory Optimization,” *Journal of Guidance, Control, and Dynamics*, Vol. 21, No. 2, 1998, pp. 193–207,  
doi:10.2514/2.4231.
- [24] Rao, A. V., “A Survey of Numerical Methods for Optimal Control,” in “AAS/AIAA Astrodynamics Specialist Conference,” Univelt Inc., Vol. 135 of *Advances in the Astronautical Sciences*, 2010, pp. 497–528.
- [25] Dontchev, A. L., Hager, W. W., and Veliov, V. M., “Second-Order Runge-Kutta Approximations in Control Constrained Optimal Control,” *SIAM journal on numerical analysis*, Vol. 38, No. 1, 2000, pp. 202–226,  
doi:10.1137/S0036142999351765.
- [26] Hager, W. W., “Runge-Kutta Methods in Optimal control and the Transformed Adjoint System,” *Numerische Mathematik*, Vol. 87, 2000, pp. 247–282,

doi:10.1007/s002110000178.

- [27] Gerdtts, M., “Gradient evaluation in DAE optimal control problems by sensitivity equations and adjoint equations,” in “PAMM,” Vol. 5, 2005, pp. 43–46,  
doi:10.1002/pamm.200510012.
- [28] Baier, R. and Gerdtts, M., “A computational method for non-convex reachable sets using optimal control,” in “European Control Conference (ECC),” Budapest, Hungary, 2009, pp. 97–102.
- [29] Cao, Y., Li, S., Petzold, L., and Serban, R., “Adjoint Sensitivity Analysis for Differential-Algebraic Equations: The Adjoint DAE System and its Numerical Solution,” *SIAM Journal on Scientific Computing*, Vol. 24, No. 3, 2003, pp. 1076–1089,  
doi:10.1137/S1064827501380630.
- [30] Cao, Y. and Petzold, L., “A Posteriori Error Estimation and Global Error Control For Ordinary Differential Equations by the Adjoint Method,” *SIAM Journal on Scientific Computing*, Vol. 26, No. 2, 2004, pp. 359–374.
- [31] Kulikov, G. Y., “Cheap global error estimation in some Runge-Kutta pairs,” *IMA Journal of Numerical Analysis*, pp. 1–28,  
doi:10.1093/imanum/drr060.
- [32] Nguyen, L. T., Ogburn, M. E., Gilbert, W. P., Kibler, K. S., Brown, P. W., and Deal, P. L., “Simulator study of stall/post-stall characteristics of a fighter airplane with relaxed longitudinal static stability,” Tech. Rep. 1538, NASA, 1979.
- [33] van Kampen, E.-J., Chu, Q. P., Mulder, J. A., and van Emden, M. H., “Nonlinear Aircraft Trim Using Interval Analysis,” in “AIAA Guidance, Navigation and Control Conference and Exhibit,” American Institute of Aeronautics and Astronautics, Reston, Virginia, August, 2007,  
doi:10.2514/6.2007-6766.
- [34] Ananthkrishnan, N. and Sinha, N. K., “Level Flight Trim and Stability Analysis Using Extended Bifurcation and Continuation Procedure,” *Journal of Guidance, Control, and Dynamics*, Vol. 24, No. 6, 2001, pp. 1225–1228,  
doi:10.2514/2.4839.