Efficient Flight Envelope Estimation for Changed Aircraft Dynamics

J.C.J. Stapel
May 22, 2015
Efficient Flight Envelope Estimation for Changed Aircraft Dynamics

MASTER OF SCIENCE THESIS

For obtaining the degree of Master of Science in Aerospace Engineering at Delft University of Technology

J.C.J. Stapel

May 22, 2015

Faculty of Aerospace Engineering · Delft University of Technology
The undersigned hereby certify that they have read and recommend to the Faculty of Aerospace Engineering for acceptance a thesis entitled “Efficient Flight Envelope Estimation for Changed Aircraft Dynamics” by J.C.J. Stapel in partial fulfillment of the requirements for the degree of Master of Science.

Dated: May 22, 2015

Readers:

Prof.dr.ir. J.A. Mulder

Dr.ir. C.C. de Visser

Dr.ir. E.J. van Kampen

Dr. Q.P. Chu

Dr.ir. D. Visser
Abstract

This report contains a study to find faster numerical methods for Hamilton-Jacobi Isaacs partial differential equations in application to model-based flight envelope estimation. This equation can be used to estimate the flight envelope through solving a reachability problem. The goal is to update the flight envelope with this method to maintain envelope protection after the occurrence of a failure. To do so the estimation methods have to become considerably faster. Useful insights have been obtained through assessing the reachable set theory associated to the problem, which permit to designate computational resources to regions of higher interest and to solve the problem with a new set of solving schemes by using the boundary value formulation of a minimal time differential game. A small literature study is held on the level set and fast marching methods. Five different techniques have been attempted to improve the computational efficiency. The applicability of a class of non-iterative schemes, known as the fast marching methods, has been evaluated both on a theoretical level as well as through simulation. The behavior of the studied methods is demonstrated on four example problems, including a simplified aircraft model.

The research has found that in application to flight envelope estimation it is not feasible to initialize the reachability problem with an estimated set, or to recursively propagate a reachable set over a failure event. The integration of the differential equation with a set of trajectories was however found to be permissible. Potential applications for this technique have been identified. It was found that the boundary value formulation of the differential equation may be used provided that the system is modeled as a time-invariant system and that the initial set consists only of trimmable states. The non-iterative Fast Marching method was demonstrated not to be applicable to envelope estimation. An extension called the safe Fast Marching method was found to give accurate but incomplete reachable sets.

It is recommended to consider the acceptance of sufficient rather than optimal control inputs to simplify the optimization problem. A continued investigation should be made on the iterative minimal time algorithms, in particular the iterative fast marching method and the fast sweeping method.
Acronyms

$L_\infty$  infinity norm
BFM              buffered Fast Marching
BLAS             Basic Linear Algebra Subprograms
C&S              Control & Simulation
CFL              Courant-Friedrichs-Lewy
DUT              Delft University of Technology
EFM              Eulerian Fast Marching
FIM              fast iterative method
FM               Fast Marching
FMM              Fast Marching method
FS               fast sweeping
FSM              fast sweeping method
GLF              global Lax-Friedrich
HIJ-(W)ENO        Hamilton Jacobi (weighted) essentially nonoscillatory
HJB              Hamilton Jacoby Bellman
HJI              Hamilton Jacoby Isaacs
ITM              iterative method
LLF              local Lax-Friedrich
LLLGF             local-local Lax-Friedrich
LOC              loss of control
LS               Level Set
M LS             Mitchell’s Level Set
MIMO             multiple input- multiple output
NB               narrow-band
OUM              ordered upwind method
PDE              partial differential equation
PFM              progressive Fast Marching
<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RMS</td>
<td>root-mean-square</td>
</tr>
<tr>
<td>SFM</td>
<td>safe Fast Marching</td>
</tr>
<tr>
<td>SL</td>
<td>semi-Lagrangian</td>
</tr>
<tr>
<td>SLF</td>
<td>stencil Lax-Friedrich</td>
</tr>
<tr>
<td>SM LS</td>
<td>self-made Level Set</td>
</tr>
<tr>
<td>SRS</td>
<td>SIMONA Research Simulator</td>
</tr>
<tr>
<td>TVD-RK</td>
<td>total variation diminishing Runge-Kutta</td>
</tr>
</tbody>
</table>
# List of Symbols

## Greek Symbols

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha$</td>
<td>CFL stability margin; angle of attack</td>
</tr>
<tr>
<td>$\tilde{\alpha}, \bar{\alpha}$</td>
<td>Critical angle of attack, mid-range angle of attack</td>
</tr>
<tr>
<td>$\beta$</td>
<td>Side-slip angle</td>
</tr>
<tr>
<td>$\chi$</td>
<td>State space domain</td>
</tr>
<tr>
<td>$\Gamma$</td>
<td>Boundary of a set</td>
</tr>
<tr>
<td>$\gamma$</td>
<td>Flight-path angle</td>
</tr>
<tr>
<td>$\lambda$</td>
<td>Normalized interpolation weights for semi-Lagrangian method</td>
</tr>
<tr>
<td>$\nu, \delta$</td>
<td>Input strategy mappings; $\delta$ also used as boundary operator</td>
</tr>
<tr>
<td>$\Phi$</td>
<td>Viscous approximation of value function</td>
</tr>
<tr>
<td>$\phi$</td>
<td>Roll angle</td>
</tr>
<tr>
<td>$\sum$</td>
<td>Dynamic system</td>
</tr>
<tr>
<td>$\tau$</td>
<td>Artificial time</td>
</tr>
<tr>
<td>$\xi$</td>
<td>Trajectory</td>
</tr>
</tbody>
</table>

## Roman Symbols

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a, \bar{a}, b, \bar{b}$</td>
<td>Generalized input signal and input strategy</td>
</tr>
<tr>
<td>$a^<em>, b^</em>, u^<em>, d^</em>$</td>
<td>Optimal input signals</td>
</tr>
<tr>
<td>$C_{D_0}, C_{D_\alpha}, C_{D_{\alpha_2}}$</td>
<td>Drag coefficients</td>
</tr>
<tr>
<td>$C_{L_0}, C_{L_{\alpha}}$</td>
<td>Lift coefficients</td>
</tr>
<tr>
<td>$C_{Y_{\alpha}}$</td>
<td>Side force coefficient</td>
</tr>
<tr>
<td>$d, \tilde{d}$</td>
<td>Disturbance input signal and Disturbance input strategy</td>
</tr>
</tbody>
</table>
List of Symbols

$D, W, L$ Drag, Weight, Lift
$F$ Magnitude of velocity component normal to the front; force
$f(.)$ ODE for dynamic system
$g, \hat{g}$ Terminal cost function; $g$ also used for input gain
$H(x, p)$ Hamiltonian
$I_{\mathcal{F}}$ Isaac’s equation
$I_{xz}$ Stencil of Lax-Friedrich dissipation scheme
$I, T, \mathcal{K}$ Initial, Target and starting sets, $\subset \chi$
$J$ Objective function
$l$ length of semi-Lagrangian integration step
$n$ Unit vector normal to the front
$O$ Neglected higher order terms
$O$ Order of magnitude
$p, p_i$ Gradient of value function and dimensional components
$\mathcal{R}_{\mathcal{K}, t_0}(t)$ Reachable set obtained at $t$ when initialized in $\mathcal{K}$ at $t_0$
$\mathcal{V}_{\mathcal{K}}(t), I_{\mathcal{K}}(t)$ Viability and invariance kernels
$S, \dot{S}$ Displacement (rate) of mass-spring-damper
$S_\mathcal{K}$ Solution set of Level Set method
$T$ Minimum arrival time surface; Thrust
$t, \tau, k, \beta$ Time, reversed time, virtual time, Kružkov transformed time
$u, \bar{u}$ Control input signal and control input strategy
$u_t, u_i, D^x$ Finite difference operations
$U, \mathcal{U}, \mathcal{D}, \mathcal{D}$ Set of permissible inputs/strategies
$V$ Optimized value of value of objective function; velocity
$v$ Kružkov transformed arrival time surface
$x, \dot{x}, x_t$ State of system and time derivative of this state
$\tilde{x}$ End position of semi-Lagrangian calculation step
List of Figures

2-1  Illustration of the backward and forward reachable set. Source: Helsen et al. (2015) 7
2-2  Illustration of the level set representation .............................................. 11
2-3  Example of the forward and backward reachable set ............................ 13
2-4  Optimal trajectory reaching toward target set ........................................ 20

3-1  Schematic for different types of maintainability ...................................... 28

4-1  Overview of scheme implementation steps ............................................. 35
4-2  Illustration of optimization using upwind gradient ............................... 37
4-3  Equivalence of forward and backward reachability ............................... 39
4-4  Illustration of semi-Lagrangian method ................................................ 45
4-5  Geometry for two-point stencil scheme .................................................. 47
4-6  Neighbor notation in considered quadrant ............................................. 47
4-7  Geometry for multi-stencil scheme ......................................................... 48
4-8  Limitations of safe Fast Marching method ............................................. 49
4-9  Paper simulation of two-point and multi-stencil scheme ......................... 50

5-1  Behavior of Narrow-Band method .......................................................... 52
5-2  Level Set results for moving line segment .............................................. 53
5-3  Numerically reproduced maintainability plots for figure 3-1 .................... 54
5-4  Results for the Eikonal problem .............................................................. 56
5-5  Results for the double integrator ............................................................. 59
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5-6</td>
<td>More results for the double integrator</td>
<td>61</td>
</tr>
<tr>
<td>5-7</td>
<td>Difference between Mitchell's and self-made Level Set scheme</td>
<td>62</td>
</tr>
<tr>
<td>5-8</td>
<td>Mass-spring-damper system</td>
<td>62</td>
</tr>
<tr>
<td>5-9</td>
<td>Response of eccentric spring excitation</td>
<td>64</td>
</tr>
<tr>
<td>5-10</td>
<td>Response of centered spring excitation</td>
<td>65</td>
</tr>
<tr>
<td>5-11</td>
<td>Leakage in Eulerian Fast Marching method</td>
<td>65</td>
</tr>
<tr>
<td>5-12</td>
<td>Reachable sets for mass-spring-damper</td>
<td>67</td>
</tr>
<tr>
<td>5-13</td>
<td>FBD for the aircraft model</td>
<td>68</td>
</tr>
<tr>
<td>5-14</td>
<td>Comparing aircraft model with Schuet et al. (2014) and Lombaerts et al. (2013b)</td>
<td>72</td>
</tr>
<tr>
<td>5-15</td>
<td>Results for nominal aircraft model</td>
<td>74</td>
</tr>
<tr>
<td>5-16</td>
<td>Illustration of two Monte Carlo principles</td>
<td>75</td>
</tr>
<tr>
<td>5-17</td>
<td>Monte Carlo analysis for aircraft model</td>
<td>76</td>
</tr>
<tr>
<td>5-18</td>
<td>Monte Carlo simulation for aircraft model</td>
<td>77</td>
</tr>
<tr>
<td>5-19</td>
<td>Reachable sets for nominal and uncertain model</td>
<td>78</td>
</tr>
<tr>
<td>5-20</td>
<td>Illustration of constraints in aircraft model</td>
<td>80</td>
</tr>
</tbody>
</table>
List of Tables

3-1 Overview of solving schemes .................................................. 32
5-1 Computational performance for the maintainability problems ........... 54
5-2 Difference measures for the Eikonal problem ............................... 57
5-3 Difference measures for double integrator problem .......................... 60
5-4 Parameters for the eccentric and centered spring-damper problem ........ 63
5-5 Bounding boxes for trimmed aircraft set .................................... 71
5-6 Level Set and safe Fast Marching method for aircraft model .......... 75
5-7 Results from the Monte Carlo simulation ................................... 79
6-1 Recommendations on considered simplification methods .................. 84
6-2 Overview of solving schemes .................................................. 84
## Contents

### Abstract

Acronyms

List of Symbols

1 Introduction
   1-1 Problem context ........................................ 1
   1-2 Problem statement and project approach ............... 2

2 Literature survey
   2-1 Flight envelope as a reachability problem ........ 5
   2-2 Reachable set theory ................................... 6
   2-3 Differential game theory ................................ 9
   2-4 Finding the numerical schemes ....................... 14
      2-4-1 Information flow and discretization of hyperbolic conservation laws .... 15
      2-4-2 Finding the correct weak solutions ............... 17
   2-5 Introducing the Boundary value formulation .......... 19
      2-5-1 Overview of solving schemes for the boundary value formulation ...... 23

3 Simplification methods for the reachability problem .... 27
   3-1 Properties of reachability problems .................. 27
   3-2 Complexity reduction assessment ...................... 30
<table>
<thead>
<tr>
<th>Chapter</th>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>4-1</td>
<td>Implementation process</td>
<td>35</td>
</tr>
<tr>
<td>4</td>
<td>4-2</td>
<td>Modeling practices</td>
<td>38</td>
</tr>
<tr>
<td>4</td>
<td>4-3</td>
<td>Implementation of the Level Set schemes</td>
<td>40</td>
</tr>
<tr>
<td>4</td>
<td>4-4</td>
<td>Update schemes for the boundary value formulation</td>
<td>41</td>
</tr>
<tr>
<td></td>
<td>4-4-1</td>
<td>Eulerian method</td>
<td>42</td>
</tr>
<tr>
<td></td>
<td>4-4-2</td>
<td>Semi-Lagrangian method</td>
<td>44</td>
</tr>
<tr>
<td>5</td>
<td>5-1</td>
<td>Case studies</td>
<td>51</td>
</tr>
<tr>
<td>5</td>
<td>5-2</td>
<td>Simplified aircraft model</td>
<td>66</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>Conclusion and Recommendations</td>
<td>81</td>
</tr>
<tr>
<td>A</td>
<td>Draft paper</td>
<td></td>
<td>85</td>
</tr>
<tr>
<td></td>
<td>Bibliography</td>
<td></td>
<td>109</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

1-1 Problem context

Safety has always been of primary importance in aviation. Even today the flight control community continues to improve safety by extending their systems' capabilities beyond the occurrence of non-critical failures or unanticipated changes in the system dynamics. Good progress has been made through adaptive model estimation and stability argumentation. An open problem is however to maintain an accurate estimate of the flight envelope, which is essential in the prevention of loss of control.

Loss of control (LOC) is considered a primary cause of fatalities in aviation. Between 2002 and 2012 almost 40% of all fatal accidents is related to LOC. CAA Accident Analysis Group (2013), Ranter (2007) In these accidents the most important causal factors were identified to be related to flight crew handling or mishandling after a technical failure.

A commonly used method to reduce the risk of these causes is to implement flight envelope protection. The main advantage of envelope protection is that when the envelope is known, a set of simple and well-behaved constraint avoidance systems can be implemented. These protection methods rely on knowing in which state-space regions the pilot and/or control systems are effective.

The current practice is to estimate accurate flight envelopes beforehand by extensive flight tests and calculations. Tang et al. (2009) Although this method suffices under nominal conditions, the resulting set is no longer valid when the system dynamics change in the event of a failure. The unavailability of this protection system in impaired situations can be considered a significant weakness in the robust and adaptive methodology.

One promising model-based approach to resolve this problem is advocated in Tang et al. (2009), where the flight envelope is described as the intersection between two non-linear,
non-convex reachability problems. These reachability problems can be described by the Hamilton Jacby Bellman (HJB) partial differential equation (PDE) or by the Hamilton Jacby Isaacs (HJI) equation when disturbances are to be incorporated in the analysis. These equations have been solved numerically for dynamic flight envelopes with the Level Set method. Lombaerts et al. (2013a,b), Allen and Kwatny (2011) Due to the high computation times, this method cannot yet be considered for realistic real-time applications. Several studies have been performed in a search for faster or better performing variants of this method.

Adalsteinsson and Sethian (1995) localize the computational domain to a narrow region around the front of the reachable set, which reduces the number of calculations per iteration. van Oort et al. (2011) uses a semi-Lagrangian Level Set method to avoid the Courant-Friedrichs-Lewy (CFL) stability criteria which is otherwise limiting the permissible time step. de Weerdt and Oort (2011) replaces the grids with interval analysis. Li et al. (2005) adapts the partial differential equation to avoid the need for re-initialization, which may be necessary in some occasions. Govindarajan et al. (2014) explores the opportunities of using simplex splines. Stipanovic and Tomlin (2003) makes over-approximations of the reachable sets with a polytopic approximation through feedback linearization. Helsen (2013) constructs the reachable set with a depth-first approach where distance fields are computed over the grid with globally optimal controlled trajectories. Good introductory reads on the propagating front theory are chapters 10-13 of van Oort (2011) and the books by Osher and Fedkiw (2004) and Sethian (2008).

This study is an attempt to explore new ways to reduce the computational complexity of the model-based approach. Two approaches to this problem are considered. The first is to propagate the set in either a more localized or recursive manner. The second approach is to identify simplifications for the equations by accepting the loss of non-essential information or unused capabilities. The interpolation between pre-calculated envelopes as suggested in Tang et al. (2009) is not assessed in this study.

This report will start with a brief overview of the reachable set theory. This theory will then be assessed on opportunities for simplification. To extend the simplification potential an alternative boundary value formulation is introduced. This formulation gives access to a set of solving schemes that to the author’s knowledge have not yet been considered for flight envelope estimation. The applicability of this class of schemes to the estimation of flight envelopes will be evaluated in the second half of this report. Finally a recommendation for further investigation will be formulated.

1-2 Problem statement and project approach

First a study will be made on the working principles of the propagating front theory and the most common methods to solve it. The potential of five simplification methods will then be assessed in a feasibility study. A demonstration of the most promising approach will be given on an example problem to verify the theoretical performance.
1-2 Problem statement and project approach

It will be assumed that the system is non-linear, but time-invariant after the failure event. Although the Level Set method can also handle time-variant problems, this restriction is necessary to properly define the notion of a flight envelope. It will be further assumed that the system model is correctly identified after the occurrence of the failure. It should be noted that this is not entirely realistic since the current adaptive system estimation techniques only provide locally correct models, and may require some iterations to converge. The main research question can be formulated as follows:

Main research question: In which ways can we reduce the solving time of non-linear, time-invariant HJI problems with an application in flight envelope estimation by simplifying the solving techniques?

In the first stages of the project, a preliminary study will be held to obtain a better insight in how the propagating front method works, how this can represent a flight envelope and which approach could be taken to reduce the computational complexity. Two methods will be considered; the well-established Level Set method and the more controversial boundary value method. The following research questions will be answered in this study:

- Which definition of a flight envelope is appropriate in this project context?
- How can the flight envelope be represented by a reachability problem?
- How is the reachability problem represented by a differential game?
- How does the level set method solve the reachability problem?
- How does the Boundary Value method solve the reachability problem?
- What are the characteristic differences between the two methods?

The remainder of this project is aimed at identifying which of these approaches can effectively contribute to a method suitable for on-line applications. The criteria for this evaluation are the ability to solve the problem, the efficiency at which this is done and the potential speed improvement that can be achieved by altering this method. Based on these outcomes, a recommendation for further study will be made. The most promising method will be investigated further with the aid of simulations and its behavior will then be demonstrated on a benchmark problem.

Five candidate techniques for computational simplification have been selected for evaluation. The selection is based on newness, generality and expected potential improvement. The first method is to divide the problem in lower-dimensional sub problems and take the intersection of their extended solutions. The second is to localize the solving calculations to regions of higher interest. The third method is to use an initialization closer to the solution. The fourth is to use the alternative boundary value formulation. The fifth approach is to simplify the computations themselves. Each of these ideas will be assessed in section 3-2.

The contribution of this paper consists of the identification of innovative simplification methods applicable to flight envelope estimation and the evaluation of applicability of a
new set of solving schemes known as the Fast Marching methods. This information results in a recommendation for the use or further investigation of several solving schemes and simplification techniques.

This report has the following structure. Chapter 2 contains a literature survey on how the flight envelope can be expressed as a partial differential equation and how the Level Set method and methods using the boundary value formulation can solve this equation numerically. Chapter 3 proves a set of new insights that is then used to assess the five candidate techniques. Chapter 4 develops further practical details for the numerical implementation which is then used in chapter 5 to subject the solving schemes to four dynamic systems. The conclusions formed during these experiments and the theoretical analysis are summarized in chapter 6. A recommendation for further study is also formulated here.
Chapter 2

Literature survey

2-1 Flight envelope as a reachability problem

Before diving into the more complex theory some concepts will be introduced that are fundamental to this study. First, a clear definition for the flight envelope has to be defined. The flight envelope represents the aircraft’s capabilities under steady flight conditions and common maneuvers and forms the collection of states in which the aircraft can be operated safely. In practice the flight envelope is carefully identified using an elaborate set of flight tests, which start conservatively and carefully extend the validated regions. The aircraft behavior is also examined for multiple configurations as well as common failures such as control surface runaways and engine failure. Walgemoed (2005) indicates that civil aviation airworthiness requires safe flight within the flight envelope. Military certification also distinguishes performance categories (mission performance, normal performance and recoverable flight regimes).

Implemented forms of envelope protection and warning systems have been around for quite some years. The Airbus A320 was the first commercial fly-by-wire controlled aircraft and included an angle-of-attack based stall protection system as well as g-load based structural protection and speed protection. Brière and Traverse (1993) Boeing also implemented a complete envelope protection system in the B777. It provided similar features as Airbus, except for the g-load limiting. In addition, the way in which the envelope is enforced differs as Boeing allows the pilot to ignore the system with excessive control forces. (In the Airbus, the pilot has to switch control mode in order to override the protection system). Rogers (1999) Each of these systems behave independent of the others and are activated when one parameter passes a certain threshold (for instance the accelerometer measuring unacceptable values), or when a particular combination of events is identified (for instance to confirm touch-down to enable reverse thrust).

This uncoupled, limit parameter-based approach has been common practice throughout applied envelope protection ever since. Rogers (1999), Koolstra et al. (2012), Lombaerts et al.
(2011) The first steps towards robust and adaptive extensions for both fixed wing and rotatory aircraft have already been made. Most of these rely on generic function approximators like neural networks Unnikrishnan (2006), Unnikrishnan and Prasad (2006), Yavrucuk et al. (2009) which are especially popular for rotor craft.

Recently however academia and experts have suggested an alternative philosophy called the moving front approach. van Oort (2011), Lygeros (2004) The idea is to calculate the dynamic aspects of the flight envelope as a reachable set problem using a high fidelity dynamic model. It is argued that this allows for better accuracy, especially in highly coupled motions. The main weakness of the uncoupled approach is that its implementation is platform specific and can become very complicated when higher dimensional models have to be described. A model-based estimation makes the method more generic and, if properly designed, easily extendable to higher dimensions.

In collaboration with the faculty of Aerospace Engineering, E.R. van Oort and colleagues have performed considerable work in this philosophy. van Oort (2011), van Oort et al. (2011), de Weerdt and Oort (2011) They formalize definitions for the flight envelope and reachability. The safe maneuver envelope is defined as the collection of states for which safe operation can be guaranteed. Of this envelope, only the constraints imposed by the system dynamics will be considered. This set is referred to as the dynamic flight envelope. Similarly, Tang et al. (2009) distinguish the immediate flight envelope, which is based solely on state reachability, and the extended flight envelope which also includes additional constraints like structural integrity. It is argued that the immediate envelope should be used for flight envelope protection on the flight control and stability level, while the extended envelope is more appropriate for path and trajectory planning.

Both van Oort and Tang et al. propose to formulate the (immediate) dynamic flight envelope as a reachability problem. To evaluate whether a state can be safely controlled, one can search for the existence of a trajectory that brings the aircraft from this state to a safe target region. The considered state should also be reachable. Hence a trajectory needs to exist that reaches to the considered state from an initially accepted region. To obtain a conservative set it is advisable to state an additional requirement; the controller should be able to reach this set even when subjected to a disturbance or to uncertainties.

2-2 Reachable set theory

Reachable set analyses have been widely used for safety analysis. Ian M. Mitchell has studied the application of reachability in this field extensively. Mitchell (2007), Kaynama et al. (2011) The main aspect of this analysis is not to consider individual trajectories emerging from a dynamic system, but to study the behavior of sets of these trajectories.

First a proper definition for a reachable set will be given. A reachable set is the collection of states that can be reached from an initial set when subjected to a dynamical system. The
following definitions of a reachable set are defined in van Oort et al. (2011):

Consider a dynamic system $\Sigma$ with dynamics $\dot{x} = f(x,u,d,t)$ where $x$ is a state in state space $\chi$, $t$ is time and $[u,d]$ are control and disturbance input signals from the permissible bounded sets $\mathcal{U}(x,t), \mathcal{D}(x,t)$. The following notation for a trajectory will be used: $\xi_{x_0,t_0,u_d(t)}(t) : t \rightarrow x \in \chi$, where $x_0$ is the initial state at time $t_0$. The initial set $\mathcal{I}$ and target set $\mathcal{T}$ are also defined such that $\mathcal{I}, \mathcal{T} \in \chi$. A backwards reachable set is the collection of states from which the target set can be reached at time $t_f$ when starting at time $t$:

$$\mathcal{R}^B_{t_f,t} := \{ x \in \chi : \forall d \in \mathcal{D}, \exists u \in \mathcal{U}[\xi_{x,t,u,d}(t_f) \in \mathcal{T}] \} \quad (2.1)$$

A forwards reachable set is the collection of states that can be reached at time $t$ when starting in the initial set at time $t_0$:

$$\mathcal{R}^F_{t_0,t} := \{ x \in \chi : \forall d \in \mathcal{D}, \exists [x_0,u] \in [\mathcal{I},\mathcal{U}][\xi_{x_0,t_0,u,d}(t) = x] \} \quad (2.2)$$

The two sets are illustrated in figure 2-1. The dynamic flight envelope for system $\Sigma$ and $\mathcal{I} = \mathcal{T} = \mathcal{K}$ can now be represented as the intersection between the forward and backward reachable set: $\mathcal{R}^F_{t_0,t} \cap \mathcal{R}^B_{t_f,t}(t)$.

![Figure 2-1: Illustration of the backward and forward reachable set. Source: Helsen et al. (2015)](image)

Besides the above definitions there exists a further classification of reachable sets. Since the set is subjected to a dynamic system, there is a vital role for the time. Special reachable sets can be defined by considering the states that can be reached within a given time horizon rather than at a specific time. Similarly existence requirements can be made weaker or stronger by replacing the $\exists$ criteria with $\forall$ symbols. These alterations lead to a large set of reachability problems. A couple of these have been classified by Mitchell in Mitchell (2007) and later in Kaynama et al. (2011). Some of these are presented here, in a slightly altered form. The sets of equations 2-1 and 2-2 are proposed in van Oort (2011), Tang et al. (2009), Lygeros (2004), de Weerdt and Oort (2011) to represent the flight envelope and are referred to as the forward and backward maximal reachability sets. Other relevant sets are defined below:
Minimal reachable set: Set of states that reach the set $\mathcal{K}$ at time $t$ regardless of the permissible trajectory used. This set is a collection of inevitable states and is of particular interest in avoidance problems. They can be defined as:
Forwards: $\mathcal{R}_{x_0}^{F}(t) := \{x \in \mathcal{X} : \forall d \in \mathcal{D}, \forall u \in \mathcal{U}, \exists x_0 \in \mathcal{I}(\xi_{x_0,t_0,u,d}(t) = x)\}$
Backwards: $\mathcal{R}_{x_0}^{B}(t) := \{x \in \mathcal{X} : \forall d \in \mathcal{D}, \forall u \in \mathcal{U}, \xi_{x,t,u,d}(t_f) \in \mathcal{I}\}$

(Finite horizon) reachability tube: set of states that can be reached at some time in the interval $t \in [t_1, t_2]$. The forward or backward reachable tube is defined as: $\mathcal{R}_{\mathcal{K}}([t_1, t_2]) := \bigcup_{t \in [t_1, t_2]} \mathcal{R}_{\mathcal{K}}(t)$. When considering $t_1 = 0$ one may also write $\mathcal{R}_{\mathcal{K}}(\leq t_2) := \bigcup_{t \in [0, t_2]} \mathcal{R}_{\mathcal{K}}(t)$.

(Finite horizon) viability kernel: set of states for which it is possible to remain in the initial set. In this case it is assumed to be known which regions are safe and that one searches for the subset for which the system can stay in this region.
$\mathcal{V}_{\mathcal{K}}([t_1, t_2]) := \{x \in \mathcal{X} : \forall t \in [t_1, t_2], \forall d \in \mathcal{D}, \exists u \in \mathcal{U}, \xi_{x,0,u,d}(t) \in \mathcal{K}\}$

(Finite horizon) invariance kernel: set of states for which it is not possible to leave the initial set regardless of the inputs used. (set for which maximal reach tubes will not leave the initial set)
$\mathcal{I}_{\mathcal{K}}([t_1, t_2]) := \{x \in \mathcal{X} : \forall t \in [t_1, t_2], \forall d \in \mathcal{D}, \forall u \in \mathcal{U}, \xi_{x,0,u,d}(t) \in \mathcal{K}\}$

The list of definitions can be further extended by also formulating the converse sets of the above (the set of states that do not satisfy the conditions). These sets can be of interest in collision avoidance, as Mitchell has studied. They can also be used to define the regions that have to be avoided, or to find the reachable set by solving the converse system. Important to note when taking the converse of a set is that any property of over or under estimation is reversed as well.

In order to find these sets efficiently it is important to know how to find optimally controlled trajectories. In the search for these trajectories, it is possible to apply the principle of dynamic programming. This principle lies at the base of differential calculus and implies that a trajectory can be broken up in smaller trajectories; if $\xi_{x_0}(t_1) = x_1$, then $\xi_{x_0}(t_1 + \tau) = \xi_{x_1}(\tau) = x_2$. When this principle is applied to sets, it permits to propagate the reachable set through time and evolve from a previously calculated set. The resulting sub-problems are generally easier to solve.

In addition to the control inputs, a set of disturbance signals can be included in the problem formulation. This allows to include the influence of model uncertainties or turbulence on the flight envelope. These disturbances can be optimized as well to identify the dynamic flight envelope under worst-case disturbances. For problems involving both a controller and disturber, this optimality problem falls under the study of differential games and has been used in for instance Mitchell et al. (2005), Lomberts et al. (2013b). The differential game theory is well described in for instance Evans and Souganidis (1983), Pierre (2010). A brief
overview of the relevant insights in these papers is given in the following section.

2-3 Differential game theory

The pursuit-evader problem considers a differential game with dynamics described by
\[ \dot{x}(t) = f(s, x(t), u(t), d(t)) \]
where again x represents the state, u is a control signal and d a disturbance signal. The game also has an objective function \( J : u, d \to \mathbb{R} \) of the form
\[ J(u, d) = g(x(t_f)) \]
where g represents a terminal cost that only depends on the state reached at the time horizon \( t_f \). In a zero-sum differential game two players try to influence the same objective function. One player will attempt to minimize the value of J while the other player will try to maximize it. Whether the maximizing player uses the control or disturbance signal depends on the problem at hand. Assuming the conventions used throughout this report, the controller will have to be maximized in the case of a forward reachability problem. The disturber has to be maximized when a backward reachable set is to be found. In general, an optimal compromise between the objectives of the two players does not have to exist. In that case a strategy has to be formulated to define the information available for the optimization and the order in which the two players take their decisions.

A strategy for control input u will be denoted as \( \bar{u} \in \bar{U} \) and a strategy of disturber d as \( \bar{d} \in \bar{D} \). In this study no latency is included and the optimization only considers information of the current iteration. In order to guarantee that any pair of strategies will provide a unique motion of the ODE, only strategies will be considered for which all of the following holds and refer to this class as permissible strategies:

**Definition 1** A permissible strategy satisfies the following conditions.

- All strategies in \( \bar{U} \) and \( \bar{D} \) that give valid inputs \( u \in U \) and \( d \in D \) are permissible.
- For a given system, \( U \), \( D \) and \( x_0 \), a chosen pair of strategies \( [\bar{u}, \bar{d}] \in [\bar{U}, \bar{D}] \) gives a unique trajectory.
- Strategies can be concatenated: the strategy of first using strategy one and then strategy two is also valid.
- All strategies that can be made by delaying a valid strategy are also valid (shifting property).

For deterministic strategies, one of the objectives can have an advantage. It is possible to guarantee this advantage to the controller by making it aware of the current strategy of the disturber through the mapping \( v : d \to u \). This forces the disturber to use a minmax strategy. Similarly the disturber can be given the advantage by knowing the strategy of the controller through the mapping \( \delta : u \to d \). To generalize the discussion for the different objectives, the unspecified input signals a and b and their observing strategies \( \bar{a}(b) \) and \( \bar{b}(a) \) will be used. The following definition is taken from Pierre (2010):
Definition 2 The value of an objective function is the optimized output that can be achieved using the permissible inputs and strategies. Two extreme optima can be considered:

\[
\begin{align*}
\text{Upper value: } V^+ &= \inf_a \sup_{\bar{b}(a)} J (a, \bar{b}(a)) \\
\text{Lower value: } V^- &= \sup_b \inf_{\tilde{b}(b)} J (\tilde{a}(b), b)
\end{align*}
\] (2-3)

In general \( V^+ \succeq V^- \). Problems where \( V^+ = V^- \) are said to satisfy the Isaac’s condition. Pierre (2010) For problems that do not satisfy this condition, a justification has to be made on using either the upper or lower value, or some other rule. To guarantee a conservative flight envelope the advantage should be given to the disturber signal. As will be shown later, the forward reachable set becomes larger with lower values (provided that the conventions from this paper are used). Hence the proper choice for guaranteed safety is as follows:

\[
\begin{align*}
\text{Forward: } V &= \sup_u \inf_d J (u, d) = V^- \\
\text{Backward: } V &= \inf_u \sup_d J (u, d) = V^+
\end{align*}
\] (2-4)

It can be proven that Isaac’s condition holds for input affine problems, as the control signals are independent and do not affect each other’s sensitivity on the cost function. Sethian (2008) Now that the problem has been defined a method to find a solution is needed. This method is introduced in the following section.

Linking differential games to the HJI PDE

Evans and Souganidis (1983) have documented a paper that proves the existence of a solution for the differential game. The method also gives its value and associated optimal controls. The theories can also be extended to similar but more complex differential games. Two steps are used. First the upper and lower values are related to the HJI equation by re-writing the expressions for \( V \) to a difference equation, and rewriting it to the partial differential equation by taking the limiting differences. When only one player is used instead of two, the problem is reduced to a HJB PDE. Secondly the solution is made unique by formulating the viscosity solution of this equation. A complete derivation can be found in Pierre (2010) or Evans and Souganidis (1983). Theorem 2.1 is taken from theorem 4.1 of Evans and Souganidis (1983) and is associated to the backward reachable set in accordance to the conventions used throughout this report:

Theorem 2.1

Let \( \Phi^- \) be the viscosity solution of the (terminal) upper Isaac’s equation and \( \Phi^- \) the viscosity solution for the lower Isaac’s equation:

\[
I^+_B \left\{ \begin{align*}
\Phi^+_t + H_{\Phi}^+(t, x, \nabla_x \Phi^+) &= 0 \\
\Phi^+(t_f, x) &= g(x)
\end{align*} \right.
\] (2-5)

\[
I^-_B \left\{ \begin{align*}
\Phi^-_t + H_{\Phi}^-(t, x, \nabla_x \Phi^-) &= 0 \\
\Phi^-(t_f, x) &= g(x)
\end{align*} \right.
\] (2-6)
where \( H_B^+(t, x, p) = \minmax_u (f(t, x, u, d) \cdot p) \) and \( H_B^-(t, x, p) = \maxmin_d (f(t, x, u, d) \cdot p) \).

Then \( \Phi^+ = V^+ \) and \( \Phi^- = V^- \)

Similarly a forward reachability problem results in the initial value HJI PDE of theorem 2.2:

**Theorem 2.2**
Let \( \Phi^+ \) be the viscosity solution of the (initial) upper Isaac’s equation and \( \Phi^- \) the viscosity solution for the lower Isaac’s equation:

\[
I_F^+ \left\{ \begin{array}{l}
\Phi_t^+ + H_F^+(t, x, \nabla_x \Phi^+) = 0 \\
\Phi^+(t_0, x) = g(x)
\end{array} \right.
\]

(2-7)

\[
I_F^- \left\{ \begin{array}{l}
\Phi_t^- + H_F^-(t, x, \nabla_x \Phi^-) = 0 \\
\Phi^-(t_0, x) = g(x)
\end{array} \right.
\]

(2-8)

where \( H_F^+(t, x, p) = \minmax_u (f(t, x, u, d) \cdot p) \) and \( H_F^-(t, x, p) = \maxmin_d (f(t, x, u, d) \cdot p) \).

Then \( \Phi^+ = V^+ \) and \( \Phi^- = V^- \)

Although this theorem provides the means to solve for an objective function, it does not yet represent the reachability problem with a particular initial or target set. This is where the level set method comes into play. The principle of this method is that a set can be represented implicitly by defining its boundary as the iso-contour of the value function. This is shown in figure 2-2. This representation method turns out to be very practical as it can represent highly complex sets of arbitrary dimension.

![Figure 2-2: Illustration of the level set representation](image)

Theorem 7.1 of Evans and Souganidis (1983) shows how the level set of a value function follows the dynamics of the PDE. It states that if two different terminal cost functions \( g(x) \) and \( \hat{g}(x) \) describe the same level set: \( K = \{ x : g(x) < c \} = \{ x : \hat{g}(x) < c \} \), then solving the Isaac’s equations of theorem 2.2 for both cost functions will give the same set \( S_K(t_1) = \{ x : \Phi(t_1, x) < c \} = \{ x : \Phi(t_1, x) < c \} \), which represents the set after being evolved by the optimal control problem.
With this insight it becomes clear that if the terminal cost function is chosen to satisfy equation 2-9, the objective of minimizing this cost will be equivalent to incorporating as many states as possible in $\mathcal{S}_K(t_1)$, which implies that under these conditions $\mathcal{S}_K(t_1) = \mathcal{R}_{K,t_1}^B(t_1)$. A similar analogy holds for the forward reachable set and the initial value problem.

$$g(x) = \begin{cases} 
< c & \forall x \in \mathcal{K} \\
= c & \forall x \in \delta\mathcal{K} \\
> c & \forall x \in \mathcal{K}^{-1}
\end{cases} \quad (2-9)$$

Instead of specifying the full array of assumptions for each theorem contributing to the earlier discussions, it suffices to assume the following:

- System gives unique solution for given valid input
- System is Lipschitz continuous
- Initial/target set is closed
- Initial/target value function is Lipschitz continuous and bounded

**Interpretation of the Level Set method**

It makes sense that the search for a reachable set can be represented as a moving front. (See also the books by Sethian (2008) and Osher and Fedkiw (2004)). The front gives a clear distinction between the reachable and the unreachable domain. Suppose that the front $\Gamma$ of a set is to be modeled as the isoline on elevation $c$ of an implicit function $\Phi(x)$ as illustrated in figure 2-2. The inside of the set will be defined as the region for which $\Phi \leq c$. To make this set larger the control inputs have to be selected such that the values become smaller than $c$ in as many states as possible. (Some papers define the inside as the region larger than the chosen level, which requires opposite optimality conditions). As the problem evolves, (i.e. the front propagates), the level set of the value function will have to stay on the front of the set.

The actual movement of any front $\Gamma$ can be described by subjecting it to a velocity field, which not only has to be a function of state, but may also depend on time, current shape of the front and other parameters or system descriptions. Let $x$ be the collection of points on this front: $x(t) \in \Gamma(t)$. Since the goal is to find a description that forces $\Phi$ to deform such that its level set stays on the evolving front $\Gamma$, $\Phi(x(t), t) = 0 \ \forall x(t) \in \Gamma(t)$ has to hold. This expression relates the motion of particles in $\Gamma$ (and when generalized, any particle of the state-space) to the flow field $\dot{x}$. Since the subject of interest is the motion of a front and not of an individual particle, it is generally acceptable to only consider the motion normal to the front by using the spatial gradient as indicator for the normal and projecting the velocity field to it, resulting in the following equation:

$$\Phi_t + \nabla_x \Phi(x(t), t) \cdot x_t(t) = 0 \quad (2-10)$$
Where the subscripts indicate partial derivatives and $\nabla_x \Phi$ is the spatial gradient over the state space. Here, $x(t)$ is the velocity term that can be related to the velocity field that drives $\Gamma$.

If the velocity field is chosen such that it represents the system dynamics $x_t = f(t, x, a, b)$, and the control and disturbance inputs are selected such that this motion optimally aids (or counteracts) the flow in the direction normal to the front, the initial value HJI PDE is found: $\Phi_t + \inf_a \sup_b \{ \nabla_x \Phi \cdot f(t, x, a, b) \} = 0$.

Sethian (2008) shows a simplified reachability problem for systems that only move normal to the flow by replacing the dot-product with a metric for the gradient and the magnitude of the velocity after projecting it to the gradient. $F(\cdot)$ can be defined as the velocity field normal to the front: $F = x_t(t) \cdot n = x_t(t) \cdot \frac{\nabla_x \Phi}{|\nabla_x \Phi|}$. From this the Eikonal equation can be found:

$$\Phi_t + F|\nabla_x \Phi| = 0$$ (2-11)

**Difference between the forward and backward reachability formulation**

A clarification for the difference between the forward and backward reachability will be given here with a one-dimensional time-variant example. Consider a reachability problem on the time domain $[t_0, t_f]$ with dynamics $\dot{x} = u + d$ where $u(t) \in [-1, 1] \forall t \in [t_0, t_f]$ and $d(t) \in [0, 1] \forall t \in [t_0, t_1]$ but $d(t) \in [0, 2] \forall t \in [t_1, t_f]$. The controller $u$ aims to make the forward and backward reachable sets as large as possible, while the disturber $d$ aims for a small set. The starting set is $\mathcal{K} = [x_1, x_2]$ for both problems, as illustrated in figure 2-3.

\[\Phi_t^+ + \max_u \min_d f(x, u, d, t) \cdot \nabla_x \Phi^+ = 0 \quad \Phi_t^- + \min_u \max_d f(x, u, d, t) \cdot \nabla_x \Phi^- = 0\]

**Figure 2-3:** Example of the forwards (left) and backwards (right) reachable set

In the forwards reachable set, consider two particles $p_1(t_0) = x_1$ and $p_2(t_0) = x_2$. To grow the set, these two particles should get further away from each other. Controller $u$ will try to move $p_2$, where $\nabla \Phi < 0$ to the right and $p_1$, where $\nabla \Phi > 0$ to the left, while $d$ aims for the opposite. On $t \in [t_0, t_1]$ this results in $p_1 = 0$ and $p_2 = 1$ while on $t \in [t_0, t_1]$ it gives $p_1 = 1$. Efficiency Flight Envelope Estimation for Changed Aircraft Dynamics
and \( p_2 = 1 \).

The backwards reachable set is formulated as a terminal value problem, and should therefore be calculated backwards in time when using the HJ PDE to solve the problem. The working principles however can still be explained forwards in time when considering the particles aiming to reach \( T \) at \( t_f \). Consider again two particles, this time positioned at \( p_1(t_0) = x_a \) and \( p_2(t_0) = x_b \). Controller \( u \) again aims for a large set, and thus let the particles pursue the target set. The disturber attempts the opposite. \( P_1 \) starts well at the left of the target set, and therefore \( u \) will try to move it to the right while \( d \) attempts to move it to the left. This results in \( p_1 = 1 \). In figure 2-3, the location of \( p_1(t_0) \) is chosen such that this strategy results in \( P_1(t_f) = x_1 \) and therefore \( x_a \in \delta R_{K,t_f}^{B}(t_0) \). Points further to the left will miss the target set and therefore do not belong to the reachable set. The other extreme holds for \( p_2 \), where \( x_b \) is chosen such that \( p_2(t_f) = x_2 \) when \( u \) aims for moving the particle to the left while \( d \) attempts to move it to the right.

### 2-4 Finding the numerical schemes

Osher and Fedkiw (2004) and Sethian (2008) have well documented the schemes needed to solve the Level Set HJI PDEs numerically. A couple of notions relevant to this study will be mentioned here. The first note relates to the viscosity solution. For any differential equation to be solvable, the gradients have to be existing and unique. Unfortunately, this is generally not the case when evolving interfaces. Therefore a weak solution has to be found when the value function forms a discontinuity.

The second note also relates to the calculation of the gradient. When using a finite difference scheme to approximate the derivative, a choice has to be made on the use of neighboring nodes. In particular, the directional flow of information calls for an upwind scheme as opposed to a central differencing scheme. This will be further discussed in the next subsection.

The third and final note is related to the stability of the scheme. For the Eulerian finite difference schemes, the Fourier stability analysis boils down to the necessary CFL condition: \( \Delta t < \frac{\Delta x}{\max p} \), where \( \alpha \in [0, 1] \) is a stability margin and \( p \) the spatial gradient. Geometrically, it prevents the front from propagating a distance larger than the grid resolution in a single iteration. The main downside of this criterion is that it globally restricts the size of an integration step, making it unnecessarily small in most regions. There are a few methods to avoid the need of this criterion and to localize its influence. van Oort et al. (2011) implements a semi-Lagrangian method to circumvent the criteria, while Sethian (2008) states that the CFL stability criteria can be less stringent when using the Narrow band method Adalsteinsson and Sethian (1995) as the active domain is considerably smaller.
2-4 Finding the numerical schemes

2-4-1 Information flow and discretization of hyperbolic conservation laws

Let's start looking into numerically calculating the initial and boundary value formulations of the PDE. For a scheme to be stable, special care has to be taken with the discretized calculation of the gradients. A necessary requirement that is often overlooked is to make sure that the flow of information is not violated by the discretization. This flow of information requires some attention because the movement of the front is not truly present on a stationary grid, but is represented by deforming the height of the value functions.

The gradient on a discrete grid can be approximated as a finite difference problem with an n-th order Taylor series on every independent dimension. The time propagation on the initial value formulation can be fulfilled with a total variation diminishing Runge-Kutta (TVD-RK) scheme, the simplest of which is the well-known Taylor expansion: $u(x, t + \Delta t) = u(x, t) + u_t(x, t)\Delta t + O$, where $O$ accounts for the higher order terms that are neglected. A numerical calculation for the time derivative could be found by re-writing this equation to $u_t = \frac{u(x, t + \Delta t) - u(x, t)}{\Delta t} + O$. Higher order TVD-RK schemes tend to be more stable, but require more calculations per time step.

The gradient terms of the value function can be classified as hyperbolic conservation laws. An important property to this class is that it sends information in a characteristic direction. This is in contrast to parabolic conservation laws, as used for determining curvature of the front, where the information comes from all directions. To make a successful numerical scheme, one has to make sure that this flow of information is not disregarded. A central differencing scheme for instance uses information from both directions and is therefore ill suited for hyperbolic conservation laws, but excellent for parabolic terms.

Let's consider the discretization of hyperbolic terms. Again, a Taylor expansion is used. Special attention is needed for the selecting the neighboring grid points. For each dimension the following first order spatial derivatives and the associated first order schemes can be distinguished:

Forward difference \[ u^+_x = D^+ x u = \frac{u(x + \Delta x, t) - u(x, t)}{\Delta x} + O \]
Backward difference \[ u^-_x = D^- x u = \frac{u(x, t) - u(x - \Delta x, t)}{\Delta x} + O \]
Central difference \[ u^0_x = D^0 x u = \frac{u(x + \Delta x, t) - u(x - \Delta x, t)}{2\Delta x} + O \]  
(2-12)

Forward scheme \[ u^{n+1}_i = u^n_i - \Delta x D^+ x u^n_i \]
Backward scheme \[ u^{n+1}_i = u^n_i - \Delta x D^- x u^n_i \]
Central scheme \[ u^{n+1}_i = u^n_i - \Delta x D^0 x u^n_i \]

Where $\Delta x$ is the local grid spacing. The difference is the direction from which the information comes to calculate the gradient. Clearly, $D^+$ uses information coming from the point to the right, while $D^-$ uses information coming from the left. To comply with the flow of information, one has to choose either $D^+$ or $D^-$ for each grid point, such that the information used comes from the opposite direction of the front’s motion. This strategy of selecting the scheme that uses data “upwind” of the flow is referred to as the upwind scheme or upwinding method.
Also higher order schemes can be made. It can be proven that a numerical method works with errors of $O(\Delta x)^n$ where $n$ is the order of the scheme. For the initial value formulation, these schemes are worked out up to third order differences and then combined in a Hamilton Jacobi (weighted) essentially nonoscillatory (HJ-(W)ENO), 5-th order accurate upwind scheme in Osher and Fedkiw (2004). Although these schemes may be more accurate, the higher order gradient determination also acts as a smoothing term and removes sharp corners from the flow. This additional viscosity is actually undesired in an upwind scheme since it brings the solution further away from the limiting case of the viscosity solution. Interestingly, a second order difference induces more smoothing than a third order difference scheme.

For these higher order schemes it is important to note that the first order forward and backward differences are in fact divided differences, that is, the gradient is not calculated on the grid node under evaluation but in the middle between the two nodes used to find the slope. This is indicated by writing $D^{+x} = D_{i+1/2}^1$ and $D^{-x} = D_{i-1/2}^1$. This observation leads to a method of interpolating an essentially non-oscillatory polynomial. For the second and higher order one can define:

Second and third order divided differences:

$$D_{i+1/2}^2 u = \frac{D_{i+1/2}^1 u - D_{i+1/2}^1 u}{2\Delta x} = \frac{D_0^0 u - 2D_{i+1/2}^0 u + D_{i-1/2}^0 u}{2\Delta x^2}$$

$$D_{i+1/2}^3 u = \frac{D_{i+1}^1 u - 2D_{i+1/2}^1 u}{3\Delta x} = \frac{D_0^0 u - 3D_{i+1}^0 u + 3D_{i+1/2}^0 u - D_{i-1/2}^0 u}{6\Delta x^3}$$ (2-13)

Where $D_0^0$ (not to be confused with $D_{0x}^0$) refers to the actual grid node values of $u$. One can now define $u_x^+$ and $u_x^-$ for second order each in two ways:

$$u_x^+(x_i) = D_{i+1/2}^1 u - D_{i-1/2}^2 u \Delta x$$
$$u_x^-(x_i) = D_{i+1/2}^1 u - D_{i+1}^2 u \Delta x$$

$$u_x^+(x_i) = D_{i+1/2}^1 u + D_{i-1/2}^2 u \Delta x$$
$$u_x^-(x_i) = D_{i+1/2}^1 u + D_{i+1}^2 u \Delta x$$

It is usually advised to choose the second order term with the smallest norm. For third order accuracy, define the following possibilities:

$$u_x^+(x_i) = D_{i+1/2}^1 u - D_{i+1}^2 u \Delta x - D_{i+1/2}^3 u \Delta x^3$$
$$u_x^-(x_i) = D_{i+1/2}^1 u - D_{i+1}^2 u \Delta x - D_{i+1/2}^3 u \Delta x^3$$

$$u_x^+(x_i) = D_{i+1/2}^1 u - D_{i+1}^2 u \Delta x + 2D_{i+1/2}^3 u \Delta x^3$$
$$u_x^-(x_i) = D_{i+1/2}^1 u - D_{i+1}^2 u \Delta x + 2D_{i+1/2}^3 u \Delta x^3$$

$$u_x^+(x_i) = D_{i+1/2}^1 u + D_{i-1}^2 u \Delta x + 2D_{i+1/2}^3 u \Delta x^3$$
$$u_x^-(x_i) = D_{i+1/2}^1 u + D_{i-1}^2 u \Delta x + 2D_{i+1/2}^3 u \Delta x^3$$

$$u_x^+(x_i) = D_{i+1/2}^1 u + D_{i-1}^2 u \Delta x - D_{i+1/2}^3 u \Delta x^3$$
$$u_x^-(x_i) = D_{i+1/2}^1 u + D_{i-1}^2 u \Delta x - D_{i+1/2}^3 u \Delta x^3$$

$$u_x^+(x_i) = D_{i+1/2}^1 u + D_{i-1}^2 u \Delta x - D_{i+1/2}^3 u \Delta x^3$$
$$u_x^-(x_i) = D_{i+1/2}^1 u + D_{i-1}^2 u \Delta x - D_{i+1/2}^3 u \Delta x^3$$

J.C.J. Stapel

Efficient Flight Envelope Estimation for Changed Aircraft Dynamics
Where again the second and third divided difference terms should have the smallest norm. There are however some additional things to note. When extending all these terms, it turns out that some of them actually are the same. It is therefore possible to reduce the options to three for both forward and backward differencing. Also, instead of only choosing one of these, it is possible to use a weighted average of all three options. This leads to the fifth order accurate HJ-WENO scheme. (See also Osber and Fedkiw (2004)).

2-4-2 Finding the correct weak solutions

The need for a weak solution arises when the gradients used in the above representations are no longer uniquely defined. This can occur when a corner develops or when fronts merge or separate. In these cases a weak solution has to be found. Weak solutions formulate the problem such that the derivative terms are no longer needed. There are however several different weak solutions and therefore the correct one has to be chosen.

The first weak solution that can be used for a sharp corner is to accept the limit of the approaching gradients from both directions simultaneously. This naturally occurs in particle tracing methods, since particles may be on both sides of the corner. The result is called the swallowtail effect in which the front intersects itself and forms an additional loop. Although there are problems for which this is indeed a correct solution, it is by no means acceptable when the front represents a separation of regions.

For problems where this self-intersection is not permissible, the Huygens' principle may provide a good alternative. This principle relies on the idea that in the front of an expending wave only the first arrival is relevant. This property can also be seen as an entropy-like condition, since this principle breaks the reversibility of the problem. This solution is naturally occurring in the boundary value formulation.

A third option is to add or rely on viscosity terms. These are elements added to the solving scheme with the purpose to provide some smoothing, which prevents the occurrence of sharp corners altogether. Great care however has to be taken to make sure that the smoothing terms do not change the equation to an unrelated problem. The limiting case of adding viscosity, also called the viscous limit, approaches the desired entropy solution. This is also called the viscosity solution. These names come from shock propagation in hyperbolic conservation laws as seen in for instance fluid dynamics. This approach is often used for the level set method. There are a few ways to implement it, which will be described in further detail.

Although the viscosity solution is only presented here as a definition, it is extensively described and proven in Crandall and Lions (1984), Crandall et al. (1984) and in chapter 3.3.3 of Evans (1998); and in application to differential games in Evans and Souganidis (1983). The following properties are summarized in Sethian (2008):
• Any smooth solution of the H-J equation is a viscosity solution

• The viscosity solution is equal to the strong solution wherever it is differentiable

• For well-defined initial conditions, the viscosity solution is unique

• The solution found by taking the smoothed solution for \( \lim_{\varepsilon \to 0} \) is the viscosity solution

In the first approach, a constant dissipation term is added. Consider \( u_t + H(\nabla u, x) = 0 \) to capture both the initial and terminal value formulation. The viscous term with magnitude \( \varepsilon \) can be added on the right-hand side of the equation: \( u_t + H(\nabla u, x) = \varepsilon \Delta u \). The actual viscosity solution is the solution of this equation with \( \lim_{\varepsilon \to 0} \). On a discrete problem this value however has to be large enough to be visible on the grid.

The drawback of adding a constant dissipation term is that equal amounts of smoothing are applied over the grid, and therefore in places where this smoothing term is not required. The amount of smoothing also has to be measurable on a finite grid, which in turn enforces a minimal amount of viscosity. A smarter approach is to make the level of smoothing depend on the spatial change of the Hamiltonian. This can be achieved by using the Lax-Friedrichs scheme as an approximation of the Hamiltonian. Sethian (2008) shows that the Lax-Friedrich's scheme satisfies the correct weak solution. The scheme is given here for an n-dimensional problem:

\[
\tilde{H} \approx H \left( \frac{\Phi_{x_1}^- + \Phi_{x_1}^+}{2}, \ldots, \frac{\Phi_{x_n}^- + \Phi_{x_n}^+}{2} \right) - \sum_{i=1}^{n} \alpha^i \frac{\Phi_{x_i}^+ - \Phi_{x_i}^-}{2} \tag{2-16}
\]

Where \( \frac{\Phi_{x_i}^- + \Phi_{x_i}^+}{2} \) is the central difference approximation of the gradient in direction \( i \). \( \alpha^i \) is the dissipation coefficient which depends on the derivative terms of the Hamiltonian: \( \alpha^i = \max |H_i(\nabla \Phi)| \), where \( H_i \) is the \( i \)th derivative. This coefficient may be altered in terms of the region from which the maximum is taken. We may distinguish between the global (GLF), stencil (SLF), local (LLF) and local-local (LLLHF) Lax-Friedrich schemes, which can be summarized as follows. (See also Osber and Fedkiw (2004))

\[
\alpha^{x_i}(x) = \max_{\Phi_{x_1} \in I_{x_1}, \Phi_{x_2} \in I_{x_2}, \ldots} |H_i(\Phi_{x_1}, \Phi_{x_2}, \ldots, x)| \tag{2-17}
\]
The Lax-Friedrichs schemes differ in the set $\mathcal{I}$ of values used in equation 2-18:

\[
\begin{align*}
\text{GLF:} & \quad T_{x_j}^{\alpha_{i}}(z) = T_{x_j}^{\alpha_{i}}(z) = \left[ \min_{x \in \mathcal{D}} \{ \Phi_{\min}(x) \}, \max_{x \in \mathcal{D}} \{ \Phi_{\max}(x) \} \right] \\
\text{SLF:} & \quad T_{x_j}^{\alpha_{i}}(z) = T_{x_j}^{\alpha_{i}}(z) = \left[ \min_{x \in \mathcal{S}_i} \{ \Phi_{\min}(x) \}, \max_{x \in \mathcal{S}_i} \{ \Phi_{\max}(x) \} \right] \\
\text{LLF:} & \quad T_{x_j}^{\alpha_{i}}(x) = \left[ \Phi_{\min}(x), \Phi_{\max}(x) \right] \\
\text{LLFF:} & \quad T_{x_j}^{\alpha_{i}}(x) = T_{x_j}^{\alpha_{i}}(x) = \left[ \Phi_{\min}(x), \Phi_{\max}(x) \right]
\end{align*}
\] (2-18)

A modification of this scheme results in even smaller amounts of dissipation. The Roe-Fix scheme normally uses a simple upwind finite difference scheme as introduced in section 2-4.1. It only changes to the Lax-Friedrichs scheme when a shock or ambiguity is in the vicinity. This is detected by observing sign changes in the derivatives of the Hamiltonian as is again explained by Osher and Fedkiw (2004).

### 2-5 Introducing the Boundary value formulation

In the previous sections the reachability problem was formulated as an initial and terminal value problem and solved with the HJI PDE: $\Phi_t + H(x, \nabla_x \Phi) = 0$. Together with some extensions, the initial value formulation of the Level Set method is the most commonly used form as it covers a very general class of problems. It can be solved numerically with the schemes by Osher and Fedkiw (2004) which are summarized in section 2-4. What will be seen next is that there is an alternative representation for stationary problems: the boundary value representation. This approach observes the arrival time for states instead of the velocity of the front.

In a SIAM paper from 1994 Falcone et al. (1994), Falcone and colleagues note that the burn equation, which is normally computed as a level set problem can also be solved with the stationary boundary value formulation of the HJB PDE. The trick is that the reachability problem is not modelled as a pursuit-evasion differential game, but as a minimal time problem: $J(x, u, d) = \min(t : \xi_{x, u, d}(t) \in K)$. When a state is never reached, the assigned value defaults to $+\infty$. The generalized version of this formulation directly forms the HJI PDE given in theorem 2.3. Bardi et al. (1999), Cristiani (2006), Falcone (2006) Additionally, Falcone’s paper gives a nice theorem (Theorem 3.1) stating that any boundary value formulated problem can also be solved with the more general dynamic Level Set method. Sethian states in Sethian (2008) that the only requirement for the reverse of this theorem is that the problem has to be stationary, or equivalently either monotonically extending or contracting.

Falcone et al. (1994) also highlights that the boundary value formulation can be much lighter in memory use compared to the level set formulation since the boundary value method can describe the complete evolution of the front in a single value surface while
the level set method needs a new surface for each time increment. Further advantages of the minimum-arrival time formulation are that it satisfies the correct weak solution without additional dissipation terms by utilizing Huygens' principle of first arrival, that each grid point has to be solved only once and that no CFL stability criterion has to be satisfied since the front is progressed with a fixed spatial step instead of a fixed time step. This also makes the boundary value formulation more convenient for localized propagation. Depending on the solving method, the computations themselves can however be harder to solve since the unknowns in the boundary value formulation are described implicitly.

![Figure 2-4: Optimal trajectory reaching toward target set](image)

A simplified derivation of the boundary value formulation for the backward reachable set is summarized here. Cristiani (2006) Although not verified here, the resulting equation also holds in a viscous scene. Consider a trajectory \( \xi_x(t, a^*, b^*) \) that optimally transitions state \( x \) into the target set \( T \), as illustrated in figure 2-4. The objective function \( J \) and optimized value function \( T \) are given below:

\[
J(x, a, b) = \inf_t \{ t : \xi_x(t, a, b) \in T \} \\
T(x) = \inf_a \sup_b \{ J(x, a, b) \}
\]  

(2-19)

After following this optimal trajectory for a duration of \( t_1 \), the new state can be described as \( \xi_x(t_1, a^*, b^*) \). The dynamic programming principle states that:

\[
T(x) = T(\xi_x(t_1, a^*, b^*)) + t_1 \\
= t_1 + \inf_a \sup_b \{ T(\xi_x(t_1, a, b)) \}
\]  

(2-20)

This equation can be re-written:

\[
\frac{T(x) - \inf_a \sup_b \{ T(\xi_x(t_1, a, b)) \}}{t_1} = 1
\]  

(2-21)

\(^1\)Cristiani and Falcone mention a "CFL-like" criteria to avoid irrational values for the boundary value formulation.Cristiani and Falcone (2007) This condition however is only relevant for the Eikonal equation. The formulations used in this report are not subject to it as the distance norm is replaced with a dot product, effectively avoiding the root term in the equation. As Falcone points out, the Semi-Lagrangian formulation also is not able to produce imaginary outcomes.
The following properties hold:

\[
\inf_a \sup_b \{f(a, b)\} = -\sup_a \inf_b \{-f(a, b)\}
\]

\[
-\inf_a \sup_b \{f(a, b)\} = \sup_a \inf_b \{-f(a, b)\}
\]

\[
t + \sup_a \inf_b \{-f(a, b)\} = \sup_a \inf_b \{t - f(a, b)\}
\]

\[
T - \inf_a \sup_b \{f(a, b)\} = \sup_a \inf_b \{T - f(a, b)\}
\]

(2-22)

Using these it is possible to re-write equation 2-21:

\[
\sup_a \inf_b \left\{ \frac{T(x) - T(\xi(t_1, a, b))}{t_1} \right\} = 1
\]

(2-23)

The fraction contained in this equation has the form of a finite difference. The limit forms a derivative: \( f(x) = \lim_{h \to 0+} \frac{f(x+h) - f(x)}{h} \)

This limit can be applied when the Taylor expansion of the trajectory is used: \( \xi(x, t, a, b) \approx x + \int_0^{t_1} \Delta f \cdot \Delta t \). This approximation becomes exact for \( \lim_{t_1 \to 0^+} \), provided that \( \xi \) is smooth. Substituting this leads to the Hamiltonian:

\[
\lim_{t_1 \to 0^+} \frac{T(x) - T(\xi(t_1, a, b))}{t_1} = -\lim_{t_1 \to 0^+} \frac{T(x + f \cdot t_1) - T(x)}{t_1} = \frac{f(x, a, b) \cdot \nabla_x T(x)}{f t_1}
\]

(2-24)

Substituting this result in equation 2-23 gives the boundary value formulation of the HJI PDE for backward reachability:

\[
\max_a \min_b \{-f(x, a, b) \cdot \nabla_x T(x)\} = 1
\]

(2-25)

A similar result can be obtained for the forward reachable set and is given in theorem 2.3.

**Theorem 2.3**

Let \( T \) be the viscosity solution of the boundary value formulated HJI PDE, where \( H^+(x, p) = \inf_a \sup_b \{f(x, a, b) \cdot p\} \):

\[
\begin{cases}
H^+(x, \nabla_x T) = 1 \\
T(x \in K) = t_0
\end{cases}
\]

(2-26)

Then \( \mathcal{R}_K(t_1) = \{x : T(x) \leq t_1\} \).

The monotonicity requirement forms a restriction on the applicability of this method. Since the method tracks time of first arrival, each grid point may only be passed once in the problem horizon. It can therefore only represent problems in which the front strictly moves either inwards or outwards. (In other words, either \( F \leq 0 \forall x, t \) or \( F \geq 0 \forall x, t \) has to hold).
In order to utilize this alternative solving approach it has to be proven that this is the case for flight envelope estimation. This will be done in chapter 3 for time invariant systems initialized with a maintainable set.

Similar to the previous section, a heuristic description can be given for the boundary value formulation. Consider again the front $\Gamma$ of a set. For any point on this front the change in position over time can be described by $Dx = x_t(t)Dt$, which can be re-written to $1 = x_t(t)\frac{Dt}{Dx}$. Substituting $x_t = f(t, x, a, b)$ and $T(x)$ for time on each position, and using the gradient notation for higher dimensions results in $1 = f(t, x, a, b) \cdot \nabla_x T = H(t, x, \nabla_x T)$. Here, $T(x)$ can be seen as a new value function where the value at each state $x$ is the time at which the front passes. Hence, each level set of $T(x)$ gives the location of the front at a particular time. Important to note is that this is different from the level set representation, where only the (zero) level set defines the front and a different value function is calculated for each time instant.

The boundary value formulation also provides a fresh set of solving schemes. Besides the iterative solvers, there is a set of schemes that avoid unnecessary iterations. Sethian (2008), Sethian and Vladimirsky (2003) have developed a local single-pass scheme for a certain class of reachability problems. The method is referred to as the Fast Marching method and was originally developed for the Eikonal equation. The classical Fast Marching method is only applicable to problems where the characteristics (the paths traversed by a particle) are pointing in the direction normal to the front. Several extensions to the fast marching method have been developed to make it applicable to more general problems. Some of these will be introduced in section 2.5.1.

So far two representation methods have been discussed for analysis of the reachability problem (see also Sethian (2008)):

\[
\begin{align*}
\text{Initial/Terminal value:} & & \text{Boundary value:} \\
\Phi_t + f(x, u^*, d^*, t) \cdot \nabla_x \Phi(x, t) &= 1 & f(x, u^*, d^*, t) \cdot \nabla_x T(x) = 0 \\
\text{Front } \Gamma(t) &= \{x(t)|\Phi(x, t) = 0\} & \text{Front } \Gamma'(t) &= \{x(t)|T(x) = t\} \\
\text{General motion} & & \text{Provided } F > 0 \forall x, t
\end{align*}
\]

Important to note is that both formulations are special cases of the general (in this case first order) Hamilton-Jacobi equation, as is shown below:

\[
\begin{align*}
\alpha U_t + H(x, \nabla_x U) &= 0 & \alpha = 1, \ H = f \cdot \nabla_x \Phi, \ U = \Phi \\
\Phi_t + f \cdot \nabla_x \Phi(x, t) = 0 & & \alpha = 0, \ H = f \cdot \nabla_x T - 1, \ U = T \\
\end{align*}
\]

The total derivative $D$ refers to the partial derivatives over all parameters of $u$. 

J.C.J. Stapel

Efficient Flight Envelope Estimation for Changed Aircraft Dynamics
2-5-1 Overview of solving schemes for the boundary value formulation

There exist some good reviews on boundary value schemes. Cacace, Christiani and Falcone review some of these methods in a recent paper Cacace et al. (2013) and provide a methodology to assess which properties a scheme needs in order to solve a particular class of problems. They state that the general static HJI PDE cannot be solved by schemes that are both local and non-iterative. A brief overview of some of the methods will be given here. The methods can be roughly classified in three subsets: fully iterative methods, iterative fast marching methods and ordered upwind methods.

Iterative schemes

Classical iterative method (ITM). This method is the simplest way to solve the boundary value formulation. It directly uses the principle of dynamic programming by iteratively updating all nodes using the estimated arrival times of neighboring nodes. Sethian (2008) derives a scheme for the Eikonal equation, but the method is generally applicable. Cacace et al. (2013) For n-dimensional problems and N nodes per dimension, it requires approximately $O(N)$ iterations for each node, resulting in a complexity of order $O(N^{n+1})$ and is therefore comparable to the level set method. The lack of CFL stability requirements however could make this method a little faster.

The fast sweeping method (FSM) Tsai et al. (2003), Falcone et al. (2011), Cacace et al. (2013) is similar to the classical iterative method. It does however improve the rate of convergence by updating the nodes in a more directional manner. In each iteration, the nodes are updated in a different order such that in each region the characteristic direction is exploited in at least one of the 'sweeps' over the grid. Applicability and order of complexity are the same as those of the iterative method in all other aspects. Although no mention was found in literature it is expected that it is possible to develop a narrow band version of the fast sweeping method. This will however not be considered in this project.

Ordered upwind schemes

The Fast Marching method (FMM) Sethian (2008) avoids the necessity for iteration completely by incorporating the causality of arrival times in the algorithm. The assumption that the gradient of the value function coincides with the direction of characteristics is used to find this causality property. The grid is divided in three regions; a 'far' region where front of the reachable set has not yet been, a 'final' region where the front has passed and the corresponding arrival times are known and a 'considered' or 'narrow band' region that forms the interface between the former two regions.

Candidate values are calculated for the nodes in the narrow-band region. The node with the smallest value is then accepted to the 'final' region after which its direct neighbors that are not yet in 'final' are re-calculated. This update process is summarized in the following steps by Sethian (2008):
1. Initialization:
   (a) All nodes of the initial set belong to the final set 'A' and get \( T = 0 \)
   (b) All direct neighbors belong to the considered set 'C' and get \( T = \infty \)
   (c) All other nodes get \( T = \infty \) and belong to the far set 'F'

2. March forward
   (a) Select node \( X \) with \( T_{\text{min}} \) from considered and identify surrounding nodes
   (b) All neighboring nodes living in \( F \) are transferred to \( C \)
   (c) All neighboring nodes living in \( C \) are tested for a shorter arrival time when reached from \( X \)
   (d) \( X \) is transferred to \( A \)

3. Repeat until all nodes are accepted or until some other stopping criteria.

The complexity of this approach is of order \( O(N^n \ln N) \), where the logarithmic term originates from the value ordering. The Fast marching method also has the advantage that nodes well outside of the steady reachable set will never be calculated. Especially in sparse, high dimensional problems this can result in considerable savings. As mentioned earlier, this method only works correctly for Eikonal-like problems, where the characteristics and gradients coincide thanks to the isotropy of the vector field. Cristiani (2008) as well as Sethian and Vladimirsky (2003) demonstrates the effect of applying the Fast Marching method on problems that do not satisfy this property. Both demonstrations however only consider problems that are fully reachable and only indicate incorrect arrival times.

There are currently two main methods for extending the fast marching method to more general static HJI PDEs. The first approach is to permit nodes to be re-calculated in a (quasi) iterative scheme. In this approach, the Fast Marching principles are used to minimize the number of iterations per node and the number of nodes that have to be calculated. The second approach is to reduce the locality of the value calculations. The classical Fast Marching method only has to consider direct neighboring nodes in the value calculation thanks to the isotropy assumption of the Eikonal equation. When this assumption does not hold a larger stencil of nodes has to be incorporated in the calculation, which makes the scheme non-local and increases the computational complexity in terms of overhead. For near-isotropic problems, the required stencil size can be bounded. The size of this region is referred to as the 'near-field' region.

The use of a larger stencil or near-field region in the calculation of candidate values for a node generalizes the Fast Marching method to the ordered upwind method (OUM). This only affects step two of the Fast Marching scheme, where not only the neighbors are observed but all nodes in the stencil. The need for iterations is therefore replaced by a larger calculation for the candidate value.
Iterative fast marching methods

The paper by Christiani also introduces a generalization called the buffered Fast Marching (BFM) Cristiani (2008). It sacrifices the single-pass property to become capable of solving any static HJI PDE. The method uses an extra set after the considered region to buffer accepted nodes. The size of this buffer scales with the anisotropy of the problem. The nodes in this buffer are re-computed iteratively. Every now and then, it will be checked if a node in the buffered set changes when the extreme values, Tmin and Tmax, are assumed for all nodes in the considered set. Only if this is the case will these nodes be permanently accepted. Since Tmax is already assumed in the far zone, no additional calculation has to be performed for this part of the evaluation. The order of complexity for the BFM method depends on the level of isotropy. For Near-isotropic problems, the order of complexity is comparable to that of the fast marching method. For non-isotropic cases it can however be even less efficient than the classical iterative method, due to the additional bookkeeping. The method is therefore only practical for problems that are almost isotropic and only locally deviate from this property.

The fast iterative method (FIM) Cacace et al. (2014) also modifies the FMM to an iterative version. A node is removed from the NB set once it is converged with respect to its direct neighbors, but can also enter the set again when any of the neighbors on which it depends changes its value. The method tends to be faster than the Fast Sweeping method or the BFM for complex problems.

An alteration of the BFM method is known as the progressive Fast Marching (PFM) Falcone et al. (2011). For convex Hamiltonians, it allows to perform the BFM scheme on the NB set and hence does not need the buffered set. Currently however this method is slower than the ITM. Cacace et al. (2013)
In this chapter, the obtained theories are used to develop and identify properties of reachable sets that to the author’s knowledge have not been explicitly mentioned in literature. These properties are then used in a search for potential approaches to reduce the computational complexity.

3-1 Properties of reachability problems

Besides the various sets mentioned in section 2-2, an additional property for individual states can be considered, which is of great importance for dynamic systems as it represents the trimmable states. It is called the maintainable set, and is defined as the collection of maintainable states. A state is maintainable if there exists a permissible input for which the system can remain in that state: \( \{ x | 3[u, d] \in [U, D] : \dot{x} = 0 \} \). When the reachable set is initialized as a fully maintainable set, it is possible to proof the following statements for time invariant systems:

**Proposition 3.1**
Consider a reachability problem for a time-invariant dynamic system \( \dot{x} = f(x, u, d) \). Let \( \mathcal{K} \) be a maintainable set, then \( \mathcal{R}_K(t_1) = \mathcal{R}_K(t_2) \) (The set reachable at a specific time equals the reachable tube with the same horizon).

*Proof* Consider the case where a state is reached for the first time on \( t = t_1 \). It is then also possible to reach this state at an arbitrary moment \( t_2 \geq t_1 \) by staying in the initial maintainable state for a period of \( \Delta t = t_2 - t_1 \) after which the same trajectory is followed. This is a permissible trajectory because of the shifting property mentioned in section 2-3.

**Corollary 3.1.1**
Consider the same reachability problem as in proposition 3.1. For \( t \geq 0 \) one can state that \( \mathcal{R}_K(t_1) \subseteq \mathcal{R}_K(t_1 + \tau) \) (The reachable set can only grow monotonically).
Proof: This follows logically from proposition 3.1, since every reached state can still be reached every moment thereafter.

When the system is time-variant, these properties do not hold in general. In fact, for these conditions to hold in general not only the initial set has to be maintainable, but every other state reached has to be maintainable as well. These properties are illustrated in figure 3-1. Here a set is subjected to a velocity field that moves the set to the right in time period \([t_0 - t_2]\) and \([t_4 - t_5]\). In the period \([t_2 - t_3,]\) the flow field moves the set downwards. When the initial set is not maintainable, the set will simply translate along with the flow. When however the initial set consists of maintainable states, the reachable set at a specific time \(t_f \leq t_2\) will be equal to the set reachable tube on \([t_0 - t_f]\). For time variant problems, this will not work on \([t_2 - t_5]\) as is also shown in figure 3-1. In this case not only the initial set has to be maintainable for the property to hold, but every reached state.

![Diagram of reachable sets and tubes for different types of maintainability](image-url)

**Figure 3-1**: Evolution of reachable sets and tubes for different types of maintainability. \(\mathcal{K}\) is the starting set and \(\mathcal{R}_\mathcal{K}\) is the reachable set resulting from it.

These properties have already been used in Mitchell et al. (2005). Here Mitchell highlights a method for calculating \(\mathcal{R}_\mathcal{K}(t)\). Although not explicitly using this term, he then augments global maintainability to efficiently solve a reachability tube.
Some more observations can be made. Although the following proofs are relatively simple, they give vital insights in the behavior of reachability problems

**Proposition 3.2**

For a reachability problem where the front represents a cluster of permissible trajectories, consider two initial sets $K_1$ and $K_2 \subseteq K_1$. Then their respective reachable sets satisfy $\mathcal{R}_{K_1}(t) \subseteq \mathcal{R}_{K_2}(t)$.

**Proof** Since $\mathcal{R}(t) = \bigcup_{x_0, t_0, u} \mathcal{R}_{x_0, t_0, u}(t)$, one can simply split the problem $K_1 = K_2 \bigcup (K_1 \setminus K_2)$, solve $\mathcal{R}_{K_2}(t)$ and $\mathcal{R}_{K_1 \setminus K_2}(t)$. Now $\mathcal{R}_{K_1}(t) = \mathcal{R}_{K_2}(t) \bigcup \mathcal{R}_{K_1 \setminus K_2}(t) \supseteq \mathcal{R}_{K_2}(t)$

**Proposition 3.3**

Consider a subset $V$ of the reachable set $\mathcal{R}_K(t_1)$. Then $\mathcal{R}_V(t) \subseteq \mathcal{R}_K(t_1 + \tau)$

**Proof** Since $V \subseteq \mathcal{R}_K(t_1)$, $\exists \xi_{x_0, t_0, u}(t_1) = x \in V \\forall x \in V$. Extending these trajectories over a period of $\tau$ in accordance with the dynamic programming principle gives a new set of states equal to $\mathcal{R}_V(t)$.

**Lemma 3.3.1**

Consider a stationary reachability problem with a maintainable initial set. Then on a bounded domain of finite dimensions there exists a time horizon $t_f$ such that $\mathcal{R}_K(t_f) = \mathcal{R}_K(t_2)$ for all $t_2 > t_f$. This set is referred to as the steady set and $t_f$ is defined as the infinite horizon.

**Proof** Since the set can only grow (corollary 3.1.1), the trivial case of reaching the full domain proofs existence of this horizon

**Proposition 3.4 (inclusion theorem)**

Consider a solution set $\mathcal{R}_{K_1}(t_1)$ found by solving the time invariant system which is initialized from a maintainable initial set $K_1$ over a time horizon $t_1$. Consider an arbitrary valid trajectory (or set of trajectories) $\xi$ such that $\xi(t = 0) \in K_1$ and $\xi \in \mathcal{R}_{K_1}(t_1)$. Then there exists a $t_2 \leq t_1$ and $t_3 \geq t_1$ such that $\mathcal{R}_{K_1}(t_1) \subseteq \mathcal{R}_{(K_2 \cup \xi)(t_2)} \subseteq \mathcal{R}_{K_2}(t_3)$

**Proof** The proof consists of two parts. First it can be shown that $\mathcal{R}_{K_1}(t_1) \subseteq \mathcal{R}_{K_2}(t_2)$. This follows from lemma 3.2 since $K_1 \subseteq K_2$.

To proof that $\mathcal{R}_{K_1}(t_2) \subseteq \mathcal{R}_{K_2}(t_1)$, consider reaching set $\mathcal{R}_{K_1}(t)$ by first Reaching $K_2$ from $K_1$ over a time horizon $\Delta t$ by restricting the set of permissible inputs such that only $\xi$ can be made. Then the set is grown normally till time $T + \Delta t$. Since a smaller set of permissible inputs is used one can say that $\mathcal{R}_{K_1}(T + \Delta t) \subseteq \mathcal{R}_{K_1}(T + \Delta t)$.

Since by definition, $\mathcal{R}_{K_1}(\Delta t) = K_2$ through the principle of dynamic programming it can also be stated that $\mathcal{R}_{K_1}(T + \Delta t) = \mathcal{R}_{K_1}(\Delta t)(T) = \mathcal{R}_{K_2}(T)$ and hence arrive at $\mathcal{R}_{K_2}(T) \subseteq \mathcal{R}_{K_1}(T + \Delta t)$

**Corollary 3.4.1**

The two problems of proposition 3.4 with initial sets $K_1$ and $K_2$ have the same steady set, but may have different infinite horizons.

**Remark** These proofs only hold when the reachable sets represent clusters of trajectories. When the shape of the set has a physical meaning, like in the burn equations where temperature is depending on radius of curvature of the front, these proofs do not hold.
3-2 Complexity reduction assessment

In section 1-2, five candidate approaches have been identified that have the potential to reduce the computational complexity of the reachability functions. In this section the information obtained from the literature study and proofs mentioned earlier in this chapter will be used to assess each of these.

Combine solutions of subspaces

The idea to project the problem on a collection of subspaces and to then extrapolate the sub-solutions back to the full time domain has been explored by Mitchell and Tomlin (2003). Although they successfully demonstrate this method for problems requiring an over-estimated solution, they also highlight that this method is not suited for problems requiring an under-estimate of the set. Unfortunately the reachability problems that form the flight envelope should be under-estimated in order to obtain a conservative solution Mitchell et al. (2005) and hence this approach is not a good candidate for flight envelope estimation.

Prioritize regions of interest

The second approach suggested was to find a way to only perform calculations in the regions of interest, for instance the regions near the aircraft's current state, or close to where the limits are expected. In general, a reachability calculation has to be initialized in the safe initial region and is propagated without any means to influence the preferred direction other than what the system and available strategies dictate.

In section 3-1 however a couple of interesting propositions have been deduced that hold under two conditions. The system has to be time invariant on the observed time domain and the set has to represent a cluster of trajectories. Proposition 3.2 and the inclusion theorem 3.4 show that it is than possible to correctly identify reachable states from a larger initial set, provided that this set is contained in the reachable set of the original problem. The only difference with the original solution is a change in the estimated arrival times.

This insight proves that it is possible to prioritize. A promising implementation is to first calculate one or multiple trajectories to regions of interest and then to include these to the initial set. Proposition 3.4 indicates that this set will find the reachable states sooner than the initial set alone. This also means that the steady set from proposition 3.3.1 can be found earlier. This method is therefore recommended for further development. An interesting notion would be to see if this method is better suited for the Level Set method or for the boundary value formulations.
Initialize from estimate

It was further expected that the above results might be extended to a situation where a somewhat arbitrary initial set will converge to the intended reachable set. The main problem however is that this initial set no longer guarantees to be contained in the original reachable set. Since some of these states may not be reachable, also proposition 3.1 and its corollaries no longer hold.

It is possible to give a counter example in which this method fails. Consider a highly simplified aircraft model where only the bank angle can be changed. When working properly the aircraft can reach any angle within finite time. Once a failure occurs, all means to control the bank angle will be lost, leaving the aircraft stationary in the condition it was when the failure occurred. When after the failure the reachability problem is initialized as the old flight envelope, the full domain would remain part of this set since each state is maintainable in the new situation. Clearly this is not in accordance with the true solution of the steady set which would be only the single state in which the aircraft was when the failure occurred.

An additional issue arises when considering recursively calculating a reachable set over a time-variant event. This is the reversed causality of the backwards reachable set. This property makes it impossible to calculate the backwards reachable set without a-priory knowledge of the changing dynamics. Instead the backward calculation would be solving a problem where the system is first in a failure with unknown flight envelope and after some time repairs to the known nominal situation. Since the backwards reachable set is the most important set when concerning safety, this problem makes it impossible to perform the calculations recursively. Since no convincing arguments have been found for initializing from an estimated solution, this approach will not be considered in the remainder of this project and is not recommended for further study.

Use of boundary value formulation

In section 2-5 the boundary value formulation of the HJI PDE was introduced. Efficient iterative and fast marching-like schemes have recently been developed for general HJI problems. The only requirement is that the front grows monotonic, which is proved to be the case for time invariant systems initialized with a maintainable set.

From a quality perspective the use of ordered upwind methods is somewhat discouraged by Mitchell et al. (2005) as the method is limited to first order sub-grid accuracy, can form discontinuities and lacks additional information outside the reachable set. Mitchell also cautions that the computational advantages can be less rosy than the theory suggests. The possibility for discontinuities is however intentional as the use of Huygens’ principle of first arrival avoids the augmented viscosity needed in the Level Set method to find the correct weak solution. Although additional information outside the reachable set can be useful, it is not essential for determining the flight envelope itself and may therefore be sacrificed for an increase in
computational performance. The boundary value methods are also positively advocated by Cacace et al. (2013, 2014), Cristiani (2006).

Since this new set of schemes may still prove to be faster as it increases the minimal step size and reduces the number of calculations, it is recommended for further study.

Section 2-5-1 gave an overview of the several solving methods for the boundary value formulation. These results are summarized in table 3-1. It should be noted that the order of computational complexity assumes that an iterative method requires $N$ iterations to converge while a narrow-band scheme removes one dimension. These assumptions however are not appropriate for close comparison. In addition, the logarithmic terms in the fast marching method assume worst-case sorting, while the naturally occurring order tends to be more favorable. Cacace et al. (2013) makes some further remarks on relative performance which are also shown in the table.

The table also contains an expectation for the applicability of the schemes to flight envelope estimation. The Level set methods have been proven and demonstrated before. The Iterative method is not recommended here because the FSM is able to find the same solution but converges faster. Although the Fast Marching method is not able to find a correct reachability tube in anisotropic problems, it may still be able to correctly identify the steady reachable set. If this turns out not to be the case, the Ordered Upwind method and the Fast Iterative method may provide good alternatives. Although the BFM should also be able to find a correct solution, the method is not recommended as Cacace et al. (2013) states that it tends to be slower than the iterative method for anisotropic problems.

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Complexity</th>
<th>CFL</th>
<th>Upwind</th>
<th>Iterative</th>
<th>Anisotropic</th>
<th>Speed</th>
<th>Suitable</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSM</td>
<td>$O(N^{n+1})$</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>NB-LSM</td>
<td>$O(kN^n)$</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>ITM</td>
<td>$O(N^{n+1})$</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>FSM</td>
<td>$O(N^{n+1})$</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>$&lt; ITM$</td>
<td>Maybe</td>
<td></td>
</tr>
<tr>
<td>FMM</td>
<td>$O(N^n \ln N)$</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>Maybe</td>
<td></td>
</tr>
<tr>
<td>OUM</td>
<td>$O(kN^n \ln N)$</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>Maybe</td>
<td></td>
</tr>
<tr>
<td>BFM</td>
<td>$O(kN^n \ln N)$</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>$&gt; ITM$</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>PFM</td>
<td>$O(kN^n \ln N)$</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>$&gt; ITM$</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>FIM</td>
<td>$O(kN^n \ln N)$</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>$\approx FSM$</td>
<td>Maybe</td>
<td></td>
</tr>
</tbody>
</table>

Localizing the problem

Corollary 3.3 proves that is is possible to find valid set regions when propagating only a subset of the set’s front. This gives potential application for localized front propagation. An attempt can be made to spend more computational resources to propagate the front in the vicinity of the current aircraft state and along the intended state transition. There is however no guarantee that this approach efficiently identifies reachable states. It may however work in
locally controllable regions. Since aircraft are typically not locally controllable this approach is not recommended for this application. It should further be noted that any form of local propagation would be very hard to accomplish with the level set method, as the algorithm is supposed to iterate with globally equal time increments.

Reducing the reachability problem

The second proof of the inclusion theorem 3.4 shows that sub-optimal control can be used to find the reachable set faster. In essence, this idea can be extended by realizing that it is not required to reach all states in an optimal way. Instead of investing computational efforts in finding the one optimal input, it suffices to identify any control that reaches the state. Once a state is reached, the principle of dynamic programming dictates that from there on the next state can be found irrespective of how the previous state was reached.

This approach reduces the optimal reachability problem to a sufficient reachability problem. The sub-optimality permits non-unique solutions on all time horizons (except the infinite horizon). Still, since the actual arrival time is of a lesser interest than the quick availability of a correct flight envelope, this inconsistency may be acceptable in on-line envelope estimation.

It is stressed that there is an important difference between not considering the optimal control and accepting a sufficient control input. In the first case, one of the players is restricted in its capabilities and hence the steady set may be altered. The second case allows to stop searching for a better control input once the reachability of a new state is verified by a possibly sub-optimal control input.

It is expected that this observation can make an improvement in the selection criteria for the control inputs, and in case of the fast marching method may even circumvent the need to causally order the narrow band nodes, reducing the number of calculations by a factor \( \log(N^n) \) and also improving the degree of parallelism.

Since the boundary value formulation has not been investigated in the field of flight envelope estimation, the remainder of this study will consider stationary schemes for solving the HJI PDE and compare the performance to the level set method. In particular the non-iterative Fast Marching methods will be investigated as they are expected to result in the largest reduction of calculations, provided that the methods can correctly identify the reachable states. Although literature Sethian and Vladimirsky (2003), Cristiani (2008) indicates that only the extended Fast Marching methods may be used to achieve correct minimum arrival time surfaces, no definite argument has been found that would prevent this method from finding the steady reachable set. It is hypothesized that not satisfying the anisotropy condition may induce errors that are of a similar nature as those created when using sub-optimal control. This will be tested by subjecting the methods to an anisotropic problem with a bounded stationary set. The next chapter will provide further details on the implementation of the Level Set and Fast Marching schemes.
Chapter 4

Scheme implementation

This chapter contains an additional discussion needed to implement the algorithms. The first section gives an overview of the required implementation steps. The second section gives a couple of practical insights in how to efficiently solve the different reachable sets with a single model set-up. Section 4-3 describes the implementation process of the Level Set method as well as the Fast Marching method. Section 4-4 works out the further details of the update schema for the boundary value formulated problems.

4-1 Implementation process

Many elements have to be considered in order to solve a reachability problem with one of the discussed schemes. This section will give an overview of the steps that have to be taken and the considerations that have to be made. The overview is illustrated in figure 4-1.

![Figure 4-1: Overview of scheme implementation steps](image-url)
The first step that has to be taken is to define the dynamics of the system under consideration. The dynamics should be described as a set of ordinary differential equations. Some care should be taken in the decision of which parameters will be used to span the state-space, the control inputs and the disturbance inputs. Parameter uncertainties can be modeled by implementing them as disturbance inputs. If the system dynamics are time-variant, the time can be added to the state-space or to the disturbance inputs.

The second aspect to be defined is the problem that has to be solved for this system. The required reachable sets should be identified and the behavior of the control and disturbance inputs should be defined. The differential game can be defined to give an advantage to the controllers or the disturbers, can simulate unpredictable strategies to achieve a stochastic game, or use a strategy that corresponds to a specific problem.

There is a large set of schemes that have to be defined, all of which have to satisfy consistency, stability and convergence criteria. This overview distinguishes between iterative algorithms like the fast marching method and non-iterative upwind algorithms like the Fast Marching method. Depending on the problem class, different information is available and required. The iterative schemes have the gradient readily available from previous iterations. This is not the case for the upwind schemes where the unknown is implicit to the gradient. Although it is possible to make approximations of this gradient, this option is not considered in this study.

The Level Set methods also require a dissipation scheme to obtain a correct weak solution. A selection has to be made among the several methods. The Lax-Friedrichs schemes are popular because of their simplicity. The dissipation is not needed for the boundary value formulation as Huygens' principle of minimum arrival time is used to obtain the weak solution.

The iterative schemes require a good update scheme once the Hamiltonian is found. For the Level Set method, this update consists of an integration scheme that calculates a new value using the Hamiltonian. The update scheme for the boundary value problems require some derivation as the unknown is defined implicitly. Section 4.4.1 and 4.4.2 work out particular cases for the Eulerian and semi-Lagrangian approach.

These schemes provide the information necessary to solve a reachability problem when the inputs are known beforehand. In most problems however this is not the case and an optimization algorithm is needed that optimizes the Hamiltonian. Many of the examples in literature are defined such that this optimization can be solved easily by first determining the local gradient of the value function and then finding the inputs that optimize the Hamiltonian. This approach can however be challenging when the gradient is calculated using an upwind difference scheme. In this case the approximated gradient can depend on the inputs as is illustrated in figure 4-2. The Lax-Friedrichs scheme does not have this problem as it uses central differences. The calculation of the dissipation terms however requires the largest possible partial derivative of the Hamiltonian. For trajectory-based problems, this derivative is equal to the system dynamics. Since this term is only used for a dissipation coefficient, this calculation can be simplified at the cost of a more conservative value. The Roe-Fix scheme can use upwind differencing to reduce the amount of dissipation in states.
where this approximation is not complicated by an ambiguity.

The gradient is not available for optimizing the inputs in the boundary value problem. The general upwind direction however is available. For the Eulerian Fast Marching Method an implicit optimization scheme is derived for input-affine problems. This is not done for the Semi-Lagrangian method. In general the optimization has to be solved iteratively by testing several combinations of inputs. In this study, no attempt has been made to search for these inputs efficiently.

![Figure 4-2: Illustration of optimization using upwind gradient](image)

Once the optimization scheme has been found all the elements necessary for the calculations are available. To define a proper simulation some extra considerations have to be made. The problem needs an initialization. This consists of an initial set and a value function in case of the iterative methods. There are several ways to define a value function around an initial set. The most general method is to generate a signed distance function. An efficient method is to solve the Eikonal equation using the fast marching method. For some problems it may be necessary to re-initialize the problem. In this case similar techniques can be used. Extra care has to be taken to conserve the sub-grid accuracy of the intermediate solution, however.

The schemes need a finite grid to operate in. The nodes in this grid have to be related to the state space of the problem. Appropriate resolutions have to be selected as well as a domain of interest. Additional bookkeeping is required on the boundary of the grid. Extrapolated ghost nodes can be used to supply the missing information to stencils. Alternatively, the schemes themselves can be built in such a way that they automatically handle the boundaries of the grid. This is practical for the ordered upwind methods that only seldomly reach the boundary of the grid and can already handle the absence of available nodes. The ghost cells are more practical for the iterative methods as they perform calculations over the full boundary on every iteration.

Finally, the simulation requires a stopping criteria. This can be a number of iterations, a time horizon or the front reaching a particular state. Depending on the problem, some post-processing may be desired. Examples are the extraction of a level set from the value function, the conversion of the reachability tube to a minimal arrival surface, intersections between the computed sets, comparison of different sets and conversion to difference metrics as well visualization of the results.
4-2 Modeling practices

There are a few tricks that prevent the need for setting up a new problem for the forward reachable set, the backward reachable set and the viability or invariance kernel. Instead of programming all problems separately, these sets can be expressed with only a few parameter changes.

The first insight is related to the use of augmented reachability. Apart from problems with maintainable initial sets, augmenting reachability is necessary for the sets with additional constraints on the time aspects:

- $\forall t$ constraint $\rightarrow$ set may only become smaller as the set evolves through time
- $\exists t$ constraint $\rightarrow$ set may only become larger as the set evolves through time
- Sets without time constraints should not use augmented maintainability.

The second insight allows changing between forward and backward reachable problems. The ideas will be made intuitive with a one dimensional, time variant example.

In chapter 5 of Evans and Souganidis (1983) as well as in Melikyan et al. (2007) it is realized that the terminal reachability problem can be re-written to an initial reachability problem and vice-versa through the transform $H^T(x, \nabla_x \Phi, t) = -H^T(x, -\nabla_x \Phi, t)$. There are two elemental changes. First, the optimality conditions are 'fooled' by using opposite gradients in the optimization process. Note that this is equivalent to turning the value function upside down, provided that the converse set is used. The second change is to reverse the temporal direction of integration. Since the dynamics are described as an ordinary differential game, one can simply multiply the optimal velocity vector with $-1$. The result allows to formulate the backward problem as an initial problem as is illustrated in figure 4-3. Note that the minus sign in front of the Hamiltonian disappears by using the reversed time $\tau = t_f - t$, which is positive in the direction of set evolution.

These insights allow to express the reachable sets, tubes and kernels. In this example either the Forward or backward reachable set will be used and the other sets will be expressed based on either one. The forward and backward reachable sets are:
\( \Phi_i^+ + \min_u \max_d f(x, u, d, t) \cdot \nabla_x \Phi^+ = 0 \quad \Phi_i^- + \max_u \min_d (-f(x, u, d, t_f - \tau) \cdot \nabla_x \Phi^+) = 0 \)

**Figure 4-3:** Equivalence between forwards (left) and backwards (right) reachability problems

\[
\mathcal{R}_F(t) = \begin{cases} 
\Phi_{i+1} = \Phi_i - \Delta t H_F \\
\{u^*, d^*\} = \arg_{u,d} \max_{u,d} \min_{x} \{ f(x, u, d) \cdot p \} \\
H_F = f(u^*, d^*) \cdot p \\
\Phi(t_0, x) = +g(x) \\
\mathcal{R}_{\mathcal{K},t_0}^F(t) = \{ x | \Phi(\tau, x) < c \} 
\end{cases}
\]

(4-1)

\[
\mathcal{R}_B(\tau) = \begin{cases} 
\Phi_{i+1} = \Phi_i - \Delta \tau H_F \\
\{u^*, d^*\} = \arg_{u,d} \min_{u,d} \max_{x} \{ f(x, u, d) \cdot p \} \\
H_F = -f(u^*, d^*) \cdot p \\
\Phi(t_0, x) = +g(x) \\
\mathcal{R}_{\mathcal{K},t_0}^B(t) = \{ x | \Phi(\tau, x) < c \} 
\end{cases}
\]

Using the transformations described before, it is possible to express the reachability problems in several ways, which gives the opportunity to reuse optimization schemes and to set up a single solving scheme that can be configured to calculate all sets. The most relevant formulations are worked out below.

The sign conventions as well as the equivalence of all these sets have been derived on paper and were then verified in simulation. This verification has been performed on the double integrator problem as well as on the aircraft model which will be described in sections 5-2 and 5-1.
\[ \begin{align*}
\mathcal{R}_F([t_0, t]) &= \left\{ \begin{array}{l}
\Phi_{t+1} = \Phi_t - \Delta t H_F \\
\{u^*, d^*\} = \arg_{u,d} \min_{u,d} \max_{x,u} \{f(x,u,d) \cdot -p \} \\
H_F = \max(f(u^*, d^*) \cdot p, 0) \\
\Phi(t_0, x) = +g(x) \\
\mathcal{R}_F^{t_0}(t) = \{x|\Phi(t, x) < c\} 
\end{array} \right. \\
= \left\{ \begin{array}{l}
\Phi_{t+1} = \Phi_t - \Delta t H_F \\
\{u^*, d^*\} = \arg_{u,d} \min_{u,d} \max_{x,u} \{f(x,u,d) \cdot p \} \\
H_F = \min(f(u^*, d^*) \cdot p, 0) \\
\Phi(t_0, x) = -g(x) \\
\mathcal{R}_F^{t_0}(t) = \{x|\Phi(t, x) > c\} 
\end{array} \right. \\
\mathcal{R}_B([t_0, \tau]) &= \left\{ \begin{array}{l}
\Phi_{t+1} = \Phi_t - \Delta t H_F \\
\{u^*, d^*\} = \arg_{u,d} \max_{u,d} \min_{x,u} \{f(x,u,d) \cdot -p \} \\
H_F = \max(-f(u^*, d^*) \cdot p, 0) \\
\Phi(t_0, x) = +g(x) \\
\mathcal{R}_B^{t_0}(t) = \{x|\Phi(t, x) < c\} 
\end{array} \right. \\
= \left\{ \begin{array}{l}
\Phi_{t+1} = \Phi_t - \Delta t H_F \\
\{u^*, d^*\} = \arg_{u,d} \max_{u,d} \min_{x,u} \{f(x,u,d) \cdot p \} \\
H_F = \min(-f(u^*, d^*) \cdot p, 0) \\
\Phi(t_0, x) = -g(x) \\
\mathcal{R}_B^{t_0}(t) = \{x|\Phi(t, x) > c\} 
\end{array} \right. \\
\mathcal{I}([t_0, \tau]) &= \left\{ \begin{array}{l}
\Phi_{t+1} = \Phi_t - \Delta t H_F \\
\{u^*, d^*\} = \arg_{u,d} \min_{u,d} \max_{x,u} \{f(x,u,d) \cdot -p \} \\
H_F = \max(-f(u^*, d^*) \cdot p, 0) \\
\Phi(t_0, x) = +g(x) \\
\mathcal{I}^{t_0}(t) = \{x|\Phi(t, x) < c\} 
\end{array} \right. \\
= \left\{ \begin{array}{l}
\Phi_{t+1} = \Phi_t - \Delta t H_F \\
\{u^*, d^*\} = \arg_{u,d} \min_{u,d} \max_{x,u} \{f(x,u,d) \cdot p \} \\
H_F = \max(-f(u^*, d^*) \cdot p, 0) \\
\Phi(t_0, x) = +g(x) \\
\mathcal{I}^{t_0}(t) = \{x|\Phi(t, x) < c\} 
\end{array} \right. \\
\mathcal{V}([t_0, \tau]) &= \left\{ \begin{array}{l}
\Phi_{t+1} = \Phi_t - \Delta t H_F \\
\{u^*, d^*\} = \arg_{u,d} \min_{u,d} \max_{x,u} \{f(x,u,d) \cdot p \} \\
H_F = \max(-f(u^*, d^*) \cdot p, 0) \\
\Phi(t_0, x) = +g(x) \\
\mathcal{V}^{t_0}(t) = \{x|\Phi(t, x) < c\} 
\end{array} \right. \\
= \left\{ \begin{array}{l}
\Phi_{t+1} = \Phi_t - \Delta t H_F \\
\{u^*, d^*\} = \arg_{u,d} \min_{u,d} \max_{x,u} \{f(x,u,d) \cdot -p \} \\
H_F = \max(-f(u^*, d^*) \cdot p, 0) \\
\Phi(t_0, x) = +g(x) \\
\mathcal{V}^{t_0}(t) = \{x|\Phi(t, x) < c\} 
\end{array} \right. 
\end{align*} \]

\[ (4-2) \]

### 4.3 Implementation of the Level Set schemes

For the purpose of verifying and demonstrating the behavior and properties of the Level Set method, a set of numerical schemes was implemented in MatLab. Although Mitchell has already made a convenient toolbox available Mitchell and Templeton (2005), it was decided to develop an independent tool based on Osher’s book. Osher and Fedkiw (2004) The reason for this is that the development process would give a more thorough understanding of the vast set of details and concerns that are involved with the fast marching schemes and numerical calculations in general. It was found that this approach aided the understanding of the reachability problem on the numerical level and it has contributed to some of the insights mentioned in this report.

The Level Set algorithms were first implemented in its simplest form on a one-dimensional domain with first order upwind differencing, no dissipation and without a control optimization scheme. They were then extended to two dimensions and more advanced capabilities.
when they were found to be required to solve the problem. In this process several versions of the algorithm were made to improve the structure of the program.

The capabilities of the algorithms were extended to solve two-dimensional problems with three dissipation methods, second order Runge-Kutta integration and several options for handling differential games, augmenting maintainability (including the Roe-Fix scheme) and solving different reachable sets. The narrow-band methods were not extended to the two-dimensional schemes. The main reason for this was that it was found that this method was already well understood and implemented in literature and would therefore not contribute to this search for new methods.

Each implementation has been subjected to a variety of small tests to verify the operation and cooperation of the contributing functions. The algorithms were then used to solve some simple test problems that are described in section 5.1. In this section the self-made scheme is also compared to Mitchell’s toolbox on the double integrator problem.

The schemes needed for the Fast Marching methods have also been developed. No satisfying external toolbox was found to be available. Two sets of algorithms have been developed. The first is capable of solving the classical Eulerian level set method for n-dimensional problems and has an integrated solving scheme for multiple input- multiple output (MIMO), input affine differential games. The second set is limited to two-dimensional problems but uses semi-Lagrangian approach. Several variants have been implemented, which are further described in the next chapter. Further details on the implemented numerical schemes can be found in section 4.4.

### 4.4 Update schemes for the boundary value formulation

The boundary value formulation of the HJI PDE will be worked out a bit further here. In section 1.2 of Bardi et al. (1999) the HJI PDE for the boundary value formulation is derived for the backward reachable set using the dynamic programming principle. This paper also states that the viscosity solution of this PDE results in a valid minimal arrival time. The forward reachable set is derived using the reversion theories mentioned in section 4.2. When considering the forward and backward reachable set, the boundary value formulation can be written as:

\[
\begin{align*}
\text{Backward: } & 1 = \max_u \min_d \{ - \nabla T(x) \cdot f(x, u, d) \} = - \min_u \max_d \{ \nabla_s T(x) \cdot f(x, u, d) \} \\
\text{Forward: } & 1 = \max_u \min_d \{ \nabla T(x) \cdot f(x, u, d) \} 
\end{align*}
\]  
(4.3)

In case of the Eikonal equation, Sethian (2008) simplifies these equations. The PDE then becomes:
\begin{equation}
| \nabla T | = \frac{1}{F(x)} \quad \text{s.t.} \\
\mathcal{K} = \{ x \mid T(x) = 0 \}
\end{equation}

The upwind scheme can be represented as:

\begin{equation}
\frac{1}{F_i} = | \max (T_x^-(x_i), 0) | + | \min (T_x^-(x_i), 0) |
\end{equation}

Note that since it is assumed that $F \geq 0 \ \forall \ x, t$ there is no need to add extra min or max functions for the velocity. In fact, this property can be used to adjust the scheme a little further (see also Rouy and Agnès Tourin (1992)):

\begin{equation}
\frac{1}{F_i} = | \max (T_x^-(x_i), -T_x^+(x_i)) |
\end{equation}

All of these formulations are implicit expressions for the unknown $T_{i,j}$. In application to the Ordered Upwind methods, two approaches to solve this equation are given in this section. The first approach is worked out for multi-dimensional input-affine problems and is referred to as the Eulerian method. The second approach is worked out only for two-dimensional problems and does not implicitly find the optimal inputs. This method is the semi-Lagrangian method.

### 4.4.1 Eulerian method

The Eulerian approach is a well-understood method to solve PDEs on a sampled grid. The problem is made numerical by splitting it in one-dimensional components along the principle axes of the grid. These sub-problems are then combined to a single solution. Sethian has used the Eulerian approach extensively for the Eikonal equations. Here these schemes are extended to a more general IHJ PDE.

\begin{equation}
\min_u \max_d \left( \begin{array}{c}
    f_1(x_{i,j}, u, d) \\
    f_2(x_{i,j}, u, d) \\
    \vdots
\end{array} \right) \cdot \left( \begin{array}{c}
    \nabla^\pm_{x_1} T_{ij} \\
    \nabla^\pm_{x_2} T_{ij} \\
    \vdots
\end{array} \right) = 1
\end{equation}

Here, $\nabla^\pm_{x_1} T_{ij}$... is either the lower ($\nabla^-_{x_1} = \frac{T_{i+1,j} - T_{i,j}}{\Delta x_1}$) or upper ($\nabla^+_1 = \frac{T_{i,j} - T_{i-1,j}}{\Delta x_1}$) spacial difference, depending on the location of the considered accepted node. In this implicit function the value $T_{i,j}$... will be the unknown and either the left or right neighboring value will be known. Assuming that the optimal inputs $u^*, d^*$ have been found, the PDE reduces to $f_{i,j}(u^*, d^*) \cdot \nabla_x T_{i,j}$ and it becomes straightforward to express $T_{i,j}$ explicitly:

\begin{equation}
T_{i,j} = \frac{1 + \sum_i f_i(u^*, d^*) \frac{T_{i+1,j} - T_{i,j}}{\Delta x_1} - \sum_i f_i(u^*, d^*) \frac{T_{i,j} - T_{i-1,j}}{\Delta x_1}}{
\sum_i f_i(u^*, d^*) \Delta x_i - \sum_i f_i(u^*, d^*) \Delta x_i}
\end{equation}
Where \( i^- \) are the dimensions in which \( \nabla_{x_1}^- \) is used, \( i^+ \) the dimensions with \( \nabla_{x_1}^+ \) and in which \( T_{i-1} \) and \( T_{i+1} \) are the left and right neighbor in dimension \( i \) respectively. Things are a bit more challenging when the optimal inputs are not known beforehand. The combined scheme for input optimization and updating \( T_{i,j} \) is worked out here for non-linear, input affine MIMO problems. The scheme will be using first order upwind finite differencing.

The following structure will be assumed for the dynamics associated with this class of problems:

\[
\min_u \max_d \left( \begin{bmatrix} f_1 \\ f_2 \\ \vdots \end{bmatrix} + \begin{bmatrix} g_{1u_1} & g_{1u_2} & \cdots & u_1 \\ g_{2u_1} & g_{2u_2} & \cdots & u_2 \\ \vdots & \vdots & \ddots & \vdots \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \\ \vdots \end{bmatrix} + \begin{bmatrix} g_{1d_1} & g_{1d_2} & \cdots \\ g_{2d_1} & g_{2d_2} & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \\ \vdots \end{bmatrix} \right) = 1
\]

(4-9)

It is possible to re-write the left-hand side of the equation to identify the behavioral criteria for the optimal control inputs:

\[
\sum_{m=1}^{\text{dim}} (f_m \nabla_{x_m}^\pm T_{ij}) + \max_{u_1} \left( g_{1u_1} \nabla_{x_1}^\pm T_{ij} \cdots + g_{2u_1} \nabla_{x_2}^\pm T_{ij} \cdots + \cdots \right) u_1 + \max_{u_2} (\cdots) + \cdots = 1
\]

(4-10)

Each of the optimality terms will have this structure. There is one unknown in the optimality criteria, and that is the value for \( T_{i,j} \). Writing this a bit more explicit results in the following structure:

\[
\cdots + \max_{u_1} \left( T_{i,j} \cdots + g_{1u_1} \frac{\Delta y_1}{\Delta x_1} + g_{2u_1} \frac{\Delta y_1}{\Delta x_2} \cdots \right) + \left( \pm T_{i\pm1,j} \cdots + g_{1u_1} \frac{\Delta y_1}{\Delta x_1} \pm T_{i,j\pm1} \cdots + g_{2u_1} \frac{\Delta y_1}{\Delta x_2} \cdots \right) u_1 + \cdots = 1
\]

(4-11)

Here, \( \pm \) is positive when using an upper difference and negative when using a lower difference. The opposite holds for \( \mp \). The following can be stated when the known terms are simplified:

\[
\cdots + \max_{u_1} \left( TA_{u_1} + B_{u_1} \right) u_1 + \max_{u_2} \left( TA_{u_2} + B_{u_2} \right) u_2 + \min_{d_1} \left( TA_{d_1} + B_{d_1} \right) d_1 + \cdots = 1
\]

(4-12)

Or assuming that the optimal inputs are known:

\[
T \left( A_f + \sum_{i=1}^{nu} A_{u_i} u_i^* + \sum_{i=1}^{nd} A_{d_i} d_i^* \right) + B_f + \sum_{i=1}^{nu} B_{u_i} u_i^* + \sum_{i=1}^{nd} B_{d_i} d_i^* = 1
\]

(4-13)

Since the problem is linear in the control inputs, only the extremes of the control signals have to be used. Also, existence of a value is proven for differential games with input
affine dynamics Bardi et al. (1999). The selection can be based on the sign of $T_{ij}A + B$. If $T_{ij} = T$ is treated as a variable, and approach the equality as the intersection between the functional on the left hand side and the constant of the right hand side, the problem becomes straightforward to solve.

The critical values of $T_{ij}$ for which an optimal control would switch to its other extreme are calculated by $T^*_{ui} = \frac{-B_i}{A}$. The sign of $A$ dictates whether the upper extremes are used for $T_{ij} > T^*_{ui}$ or $T_{ij} < T^*_{ui}$:

- For $\min_{ui}$: when $\text{sign}(A) > 0$, the lower control input $u^-$ should be used for $T_{ij} > T^*_{ui}$ and the upper control limit $u^+$ should be used for $T_{ij} < T^*_{ui}$. When $\text{sign}(A) < 0$, the opposite holds.

- For $\max_{di}$: when $\text{sign}(A) > 0$, the upper control input $d^+$ should be used for $T_{ij} > T^*_{di}$ and the lower control limit $d^-$ should be used for $T_{ij} < T^*_{di}$. When $\text{sign}(A) < 0$, the opposite holds.

This observation shows that normally, the functional is a piecewise affine function of $T_{ij}$ that changes its slope on every critical value $T^*_{ui}$. Because of this, one could sort these critical points and verify if the equality condition is reached on any of the piecewise affine domains. The smallest value satisfying this condition must be the correct solution.

There are however a couple of special considerations to take into account. An extra set of boundary conditions should be implemented to guarantee that the upwind direction is not violated. This can be achieved by setting $\nabla_{x_1}^- = \frac{(T_{ij} - T_{i-1,j})}{\Delta x_1} \geq 0$ and $\nabla_{x_1}^+ = \frac{(T_{ij} - T_{i+1,j})}{\Delta x_1} \leq 0$. These conditions also guarantee that at any intermediate calculation step the causality of front propagation is maintained.

It may occur that $A$ is zero. One than directly base the control value on the sign of $B$. If also $B$ is zero, the choice of the control input will not influence the arrival time and an arbitrary selection may be made. A convenient choice would be to place the associated critical point such that it would not induce an additional evaluation. This is for instance at infinity, or at least outside the considered time horizon.

4.4.2 Semi-Lagrangian method

The semi-Lagrangian method also has been used extensively. Cacace, Christiani and Falcone have been studying it, but also Sethian has made use of this approach. Van Oort has implemented the semi-Lagrangian approach to the level set method to avoid the CFL stability criteria. This section will describe its workings. The theory below has put together by using mainly Cacace et al. (2013) and Cacace et al. (2014).

The semi-Lagrangian method is illustrated in figure 4.4. The idea is to select a grid point to update (for instance $T_{ij}$). From this starting position a trajectory can be calculated
backwards in time until a certain distance criteria is met. On this end position, which is denoted as $\tilde{x}_{-\tau}$, the value $T_{\tilde{x}}$ is estimated by interpolation between the neighboring nodes (for instance $T_{t_{i+1,j}}$ and $T_{t_{i-1,j}}$). The new value at the starting node than is this interpolated value, plus the time $\tau$ spend for the trajectory to reach $x_{ij}$ from $\tilde{x}_{-\tau}$.

![Figure 4-4: semi-Lagrangian method. a: interpolation between two neighboring nodes. b: interpolation between three neighboring nodes](image)

The simplest implementation is to calculate a single vector at the starting node and linearly follow this vector back to the stopping distance. More accurate would be to integrate the dynamics over multiple steps, but this also would require multiple control optimizations.

The semi-Lagrangian scheme can be seen as a discrete model. It can be derived by discretizing the PDE, or by using the dynamic programming principle on a discrete system. (See chapter 1.5 of Cristiani (2006) for a comprehensive derivation for several cost functions). In this paper, only the minimal time problem will be used.

For the derivation, consider the discretized system dynamics with $\Delta t$ a (virtual) time step.

$$y_{n+1}(t) = y_n + \Delta t f(y_n, a, b)$$

$$y_0 = x_0 \in I$$

(4-14)

For the minimal time problem, the value function is repeated here:

$$T(x) = \left\{ \inf_a \sup_b \min_{t \to \infty} \left\{ t \mid y_{x_0}(t, a, b) \in x \right\} \quad \forall x \in R \right\}$$

(4-15)

The finite difference PDE associated with this problem is repeated here for comparison. Also the Kruzkov transformed PDE is given, which transforms $v = 1 - e^{-T}$ to bound the time domain to $t \in [0, 1]$; Falcone (2006)

$$\inf_a \sup_b f(x, a, b) \cdot \nabla_x T(x) - 1 = 0$$

$$v(x) + \inf_a \sup_b f(x, a, b) \cdot \nabla_x v(x) - 1 = 0$$

(4-16)

These expressions can be approximated with a first-order discretization as is shown in equation 4-17 with $\beta = e^{-h}$. It should be noted that these results are formulated a little
different than those found in Cristiani (2006). This is caused by the different perspective of reachability; Cristiani defines reachability as the ability to reach a target set from a particular state, while this paper follows the perspective from Osher and Fedkiw (2004), which defines reachability as the ability to reach a particular state from a given initial set. The two perspectives are interchangeable with the appropriate transformations as discussed before.

\[
T_h(x) = \inf_a \sup_b \{T_h(x + hf(x, a, b)) - h \}
\]

\[
v_h(x) = \inf_a \sup_b \left\{ \frac{1}{\beta} v_h(x + hf(x, a, b)) + 1 - \frac{1}{\beta} \right\}
\]

(4-17)

It is easy to see that these equations correspond to the graphical interpretation. The finite time step \( h \) represents the time \(-\tau\) necessary to let the (linearized) trajectory reach the line or distance where it is possible to interpolate the grid values.

These expressions can be implemented on a sampled grid in several ways. Here the schemes will be worked out for the two-point stencil (figure 4-4.a) and for a 3x3 multi-stencil scheme. The two-point stencil is implemented for the normal and Kružkov transformed case. The multi-stencil has been built only for the normal case and is meant for the safe fast marching scheme. For the upcoming discussion it will be assumed that the velocity \( f(x, u, d) \) is known.

**Two-point stencil schemes**

The two-point stencil scheme is the simplest to implement. An illustration is given in figure 4-5. The equations needed for calculating a candidate value are given here:

\[
l = \frac{\Delta x_1 \Delta x_2}{\Delta x_1 \cos \alpha + \Delta x_2 \sin \alpha}
\]

\[
\lambda_2 = \frac{l^2 + \Delta x_2^2 - 2l \Delta x_2 \sin \alpha}{\Delta x_1^2 + \Delta x_2^2}
\]

\[
\lambda_1 = \lambda_2 - 1
\]

\[
T_{i,j} = T_{n_1} \lambda_1 + T_{n_2} \lambda_2 + \frac{l}{|f|}
\]

(4-18)

These equations change a little when considering the Kružkov transform. The expression for \( T_{i,j} \) has to be replaced by the following:

\[
\lambda_1 = \lambda_2 - 1
\]

\[
\beta = e^{\frac{r}{r}}
\]

\[
V_{i,j} = \beta (V_{n_1} \lambda_1 + V_{n_2} \lambda_2) + 1 - \beta
\]

(4-19)
Multi-Stencil for safe Fast Marching scheme

The multi-stencil scheme adapts its neighbor usage depending on the availability of converged nodes and the direction of the velocity vector. In this case, the stencil is limited to the eight neighboring nodes. Without loss of generality, the discussion can be limited to the three neighboring nodes of one quadrant. Using the notation given in figure 4-6, five situations can be distinguished:

S.1 No neighbors are available
S.2 Neighbor $n_1, n_2$ or $n_3$ is available
S.3 Neighbors $\{n_1, n_3\}$ (or $\{n_2, n_3\}$) are available
S.4 Neighbors $\{n_1, n_2\}$ are available
S.5 Neighbors $\{n_1, n_2, n_3\}$ are available

In situation S.1 no neighbors exist with an accepted value. This means that for the considered velocity it is not possible to calculate a candidate arrival time. Therefore the candidate value is set to $+\infty$. 
In situation $S.2$ only one of the neighbors is available. The safe fast marching scheme dictates that no interpolation is possible. The scheme can therefore only find a value if the negative velocity vector points exactly towards this neighbor. In all other cases, the candidate value is set to $+\infty$.

Situation $S.3$ has two neighbors and therefore allows to interpolate. This however is not advisable, since the third node is not accepted. This situation can occur when a node is approached by two fronts, in which case the interpolation leads to unnecessary dissipation. Therefore a value may only be calculated if the vector points to one of the accepted nodes.

In situation $S.4$ it is possible to calculate a value if the negative velocity vector points somewhere between the two accepted nodes. In all other cases the value is set to $+\infty$.

In situation $S.5$ all the required nodes are available to use the two- or three-point stencil. In this situation a finite value is guaranteed to exist.

![Figure 4-7: Geometry for semi-Lagrangian multi stencil](image)

All cases but the first are illustrated in the quadrants of figure 4-7a, where green nodes have converged values and red nodes are unknown. For situations $S.4$ and $S.5$, two steps have to be taken in order to calculate $T_{i,j}$. First of all, the value $T_x$ has to be estimated by a weighted average of the accepted neighbors. Secondly the distance $|x_{i,j} - \bar{x}|$ has to be found. The solution for the two-point stencil is already known. Situation $S.4$ is illustrated in figure 4-7b. The update equations for a candidate value are:
\[ l = \frac{\Delta x_2}{\cos \alpha} \]
\[ \lambda_2 = \frac{\sqrt{l^2 - \Delta x_2^2}}{\Delta x_1} \]
\[ \lambda_1 = 1 - \lambda_2 \]
\[ T_{i,j} = T_{n_1} \lambda_1 + T_{n_2} \lambda_2 + \frac{l}{|J|} \]  

(4-20)

**Limitations to the safe semi-Lagrangian Fast Marching method**

Unfortunately, the safe Fast Marching (SFM) method is limited by the stencil size in its capabilities of identifying reachable nodes. Consider the two situations visualized in figure 4-8. The evaluated node is not reached according to the safe semi-Lagrangian fast marching method, while in reality a trajectory can be found that connects the reachable set to the node. In the left case, the flow vector points directly to the front of the set, but does not reach it within the stencil space around the considered node. In the right case, the linearized trajectory misses the front altogether, while the continuous (and in this example, also the piecewise linearized) trajectory reaches the front.

The capability of identifying nodes as reachable depends on the selected stencil size. Figure 4-9 shows the results of a paper simulation of the two-point stencil and the multi-stencil scheme for a system with an isotropic dynamics on a grid of five by five nodes. For each node, three candidate velocities (colored red, blue and green) are evaluated. For both stencils, the problem is initialized with the five nodes in the center of the grid. While the two-point stencil is only capable of finding two extra nodes, the multi-stencil is able to reach a considerably larger set.

![Figure 4-8: Two situations where the considered node \( x_{i,j} \) is incorrectly ignored by the safe SL FM scheme](image)

Despite these limitations, the safe semi-Lagrangian Fast Marching method is well capable of calculating an accurate reachable set, as long as these exceptions do not occur. In practice this means that for some problems this single pass method can be used to quickly find part of the reachable set, after which the solution can be refined with a more robust but iterative method.
It may further be possible to extend the method with (partial) fixes for the limitations mentioned here. One method is to use larger stencils as is suggested for the Ordered Upwind method. On very large stencils it may however be necessary to find more efficient schemes to identify the optimal controls.

![Two-point stencil](image1)

![Multi stencil](image2)

**Figure 4-9:** Reachability behavior of the safe two-point and multi-stencil scheme on a dynamic system with anisotropic flow. Each node has three candidate velocities (red, blue and green). Consecutive steps show geometry in which reachability is found through shape. Used velocity is indicated by color of geometry. The two-point stencil cannot find new regions after one iteration while the multi-stencil continues to find new nodes over seven iterations.
Chapter 5

Tests and demonstrations

A number of simulations have been performed to raise the understanding of the several solving methods highlighted in this report. These tests serve two purposes. The implementation process as well as the simulations give a more practical comparison of the different methods under evaluation. Secondly, they form a verification and demonstration method for the theorems and implementations as given in this report as well as to gain a better insight in the statements found in literature.

Five classes of dynamic systems have been implemented in four different solving schemes. Three of these schemes have been built from scratch in MatLab. The Level Set schemes are mainly based on the work of Osher and Fedkiw (2004) and are fully Eulerian. Three variants of the Fast Marching method have been implemented as well, namely the Eulerian method with a two-point stencil, the semi-Lagrangian method and the safe Fast Marching method. Finally a Level Set scheme has been implemented that uses the LStoolbox by Mitchell. Mitchell and Templeton (2005). This implementation has been used to verify the self-made algorithms and is also implemented for the aircraft model.

This chapter will give the problem descriptions and the results of the computations. The first section provides four dynamic models that have been used for assessing and testing the behavior of the different solving schemes. The second section verifies and demonstrates the behavior of the safe Fast marching scheme and the level set method in an actual flight envelope estimation process.

5-1 Case studies

This section contains four dynamic problems with varying complexity. The different schemes are tested and compared on these problems to get a better view on the relative applicability
for different problem types. The objectives for each test as well as the results will be explained in the subsequent subsections.

Set in external flow field

For the Level Set method, two experiments have been performed where a set is subjected to an externally defined flow field. These tests were made to obtain a better insight in the Level Set method and to validate the statements on maintainability. Since no control inputs are present, the PDE reduces to $\Phi_t + \hat{x} \cdot \nabla_x \Phi = 0$.

First, a one-dimensional problem was made for a first experience with the Level Set method to get a better insight in the numerical mechanics. For this problem the initial set, in this case a line segment, is translated by a constant velocity. The implicit function is initialized as a quadratic curve. Besides the classical Level Set method, two versions of the narrow band principle have also been implemented. The first implements a narrowband in a small boundary around the front with a thickness of three nodes in each direction. The second method includes all nodes inside the set as well as four node layers away from the set. No dissipation is added. Integration over time is accomplished with a first order forward difference scheme.

The intended set-up and behavior is illustrated in figure 5-1. The analytic solution for the reachable set is the line segment shifting in the direction of the velocity. The narrow band methods should correctly represent the motion of the zero level set, but also deform the value function. The deformations are not caused by active calculation, but are a consequence of not updating it in regions outside the boundaries of the narrowband.

![Figure 5-1: Schematic of initial set (top) and evolved set for the classical (left) and narrow band (right) Level Set method.](image)

The 1D problem is set up on domain $x = [-4, 4]$ and with a resolution of $\Delta x = 0.05$. The velocity is $\hat{x} = 0.5$. The initial set $\mathcal{K} = [-1, 1]$. A stability margin of 0.75 is used. The motion of the set through time is shown in figure 5-2a for the three Level Set methods. The overall motion of the set boundaries is in agreement with the expected motion, although minor differences exist among the three methods.
The Terminal value functions are shown in figure 5-2b. Here the difference between the three methods becomes apparent. For the classical Level Set method, the value function has simply moved one unit of distance to the right. The narrow-band methods however show some deformation. The set narrow-band method is closely following the normal level set method for all states inside (and close to) the set. Outside this region the set is left unchanged. The method leaves a horizontal line behind the trailing edge of the set, where nodes leave the narrow band region as the set progresses forward. The boundary narrow band method updates the value function only in the region near the iso-contour. As a consequence there is one extra region in \( x = [0, 1] \) where the value has not changed. Also a second horizontal line is trailing from the leading edge of the set.

An important insight is that the set is traversed to the right, while the value function is not. The value function merely gives apparent motion by changing the magnitude of the value at each node.

The second problem was made to verify the behavior of set propagation under various levels of maintainability as discussed in section 3-1. It was set up for two dimensions and involves an initial set moved around in an arbitrary time-variant flow field. The flow field used for this demonstration represents a uniform, constant motion that changes direction after some time. A diamond-shaped initial set is traversed to the right for two seconds, then moved down for one second and finally to the right for two seconds. In all phases the velocity is kept at unity. The intention of this flow field is to reproduce figure 3-1 numerically. Only the first order upwind scheme has been used. The grid has a resolution of \( \Delta x = [0.02, 0.02] \) and the stability margin is set to 0.9. The value function is initialized with \(|x_1 + x_2| - 1\). Again no dissipation terms were added.

For the purpose of observing maintainability characteristics, some simple changes to the scheme have been made that allow to argument various forms of maintainability. Initial
Figure 5-3: Numerically reproduced maintainability plots for figure 3-1

set maintainability is enforced by freezing the values of this set to the initial ones. Global
maintainability is enforced by not allowing the Hamiltonian to become smaller than zero.
(See also Mitchell et al. (2005)). Both the reachable set as the reachability tube are recorded
to make a qualitative observation on the effect of reachability. The reachable end set (black)
and reachability tube (gray) are shown for the three cases in figure 5-3. The computational
performance is summarized in table 5-1.

| Table 5-1: Computational performance for the maintainability problems on 251x401 grid |
|---------------------------------|-----------------|-----------------|-----------------|
| Iterations                     | Not Maintainable | Initial set maintainable | Fully maintainable |
| Iterations                     | 240             | 285              | 222             |
| CPU time [s]                   | 2.98            | 3.44             | 2.70             |

Without maintainability, the initial set moves around in the state-space like an object.
With the initial set maintainable, a smoke-trail is emitted by the initial set. With global
maintainability, the reachable set and the reachable tube are the same.

Despite the absence of augmented dissipation, there appears to be a significant amount of
dissipation as the edges of the set become smoother. This behavior is a common phenomenon
for low-order schemes and is also examined in chapter 12.1 of Sethian (2008). Here the
diffusion is associated to the coarseness of the mesh and to the order of numerical differencing.
When observing figures 5-3b, 5-3 the dissipation becomes less when maintainability is present.
The main reason is that the set is present for a longer period at a given state, giving more
iterations to converge.

Furthermore, the initial maintainability argumentation caused a large reduction in the time
step size. This is most likely caused by the violation of the continuity requirement for the
implicit function. The discontinuity caused by freezing the initial set locally increases the
Hamiltonian, which negatively affects the CFL stability criteria. The problem did not occur
with globally maintainable sets.
Uniformly expanding front

The uniformly expanding front problem is used to compare the performance of the Level Set and Ordered Upwind methods. The uniformly expanding front is an Eikonal problem and should therefore be well suited for the Fast Marching schemes. The dynamics of this system are given below:

\[
\dot{x} = f(x) = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \frac{1}{\sqrt{x_1^2 + x_2^2}} = \frac{\nabla \Phi}{|\nabla \Phi|} \tag{5-1}
\]

Assuming that the initial set is a point in the origin of the domain, the analytic solution to this problem is \( T = \sqrt{x_1^2 + x_2^2} \). It should be noted that the dynamics in equation 5-1 are provided as an external function of state, which is only valid for the initial set used in this example, as well as the more general expression that uses the gradient of the front. In this study the Fast Marching methods have not been implemented to accept gradient-dependent problems, as this phenomenon is not occurring in reachability problems of trajectory based dynamic systems. The Level set method has been implemented to accept dynamics that use gradient terms and can therefore use either expression.

The problem has been implemented as a Level Set problem as well as a Fast Marching problem. Both the safe Fast Marching scheme as the Eulerian Fast Marching scheme has been used. The Level Set method has been implemented with the externally defined dynamics expressed as a function of state as well as the internally defined dynamics that use the gradient. Both problems are solved using a Local Lax-Friedrichs’s scheme as well as a Roe-Fix scheme for dissipation and are initialized with \( \Phi = \sqrt{x_1^2 + x_2^2} - 0.08 \). The results are shown in figure 5-4. The calculations have been performed on a \( N = [49, 49] \) grid and a domain of \( x_1, x_2 = [-2, 2] \).

Figure 5-4a shows the analytic solution of the 2D Eikonal equation. The remaining figures show the difference between the analytic and the numerical solutions. Figure 5-4b shows the difference pattern for the Eulerian Fast Marching method. No error is present on the principle axis, but the Fast Marching method predicts later arrival times elsewhere. This error is a direct consequence of the way in which the method estimates a node update. The estimation on the primary axis is exact since the problem is linear on these axes and the projection of the velocity field is exact in these regions. Figure 5-4c shows the difference pattern for the safe Fast Marching method. In contrast to the Eulerian method, the error is about four times smaller in magnitude. Also the diagonals coincide with the exact solution. This is because the SFM method uses the larger multi-stencil which also incorporates the diagonal axis in the calculation.

Figure 5-4d shows the difference surface of the Level Set method with local Lax-Friedrichs scheme using the externally defined dynamics and is therefore comparable with the results of the Fast Marching methods. Although using the same stencil as the Eulerian Fast Marching method, the explicit Level Set method is less accurate. The level set method has a lower performance as it does not obtain exact solutions on the primary axis and since the grid resolution is relatively low. Figure 5-4e shows the difference surface for the same scheme using
Figure 5-4: Results for the Eikonal problem
the implicitly defined dynamics. Comparing this surface to figure 5-4d gives an indication on the difference between evolving the set with pre-calculated dynamics and following the actual Eikonal equation with the numerical inaccuracies. The differences are larger in this case. Figure 5-4f and 5-4g show the differences for the Roe-Fix scheme. The results for the internally defined dynamics are a bit better when compared to the local Lax-Friedrichs scheme, which is caused by the reduced dissipation. The Roe-Fix scheme however induces oscillations on the externally defined dynamics. The Roe-Fix scheme assumes that the derivatives of the Hamiltonian are a good indicator for discontinuities. This derivative is equal to the optimal velocity vector in case of a trajectory-based system. Since the velocity vectors of the externally defined flow field are not affected by the gradient, this method fails as an indicator for induced oscillations. It is therefore suggested to add a second fix for problems that do not have the gradient in the derivative of the Hamiltonian. A good candidate may be to observe sign changes in the upper and lower derivatives themselves. The overall performance data is given in table 5-2, where e and i stand for internal and external dynamics respectively and E and s represent the Eulerian and safe Semi-Lagrangian method.

| Table 5-2: Difference measures for the Eikonal problem; $N = [49, 49] \text{ and } \Delta x = 1/12$ |
|--------------------------------------------------|-----------|-----------|-----------|
| $L_{\infty}$ | Forward | RMS | RMS % |
| Analyst - E FSM | $9.24e^{-02}$ | $5.35e^{-02}$ | 1.89 |
| Analyst - s FSM | $2.23e^{-02}$ | $1.22e^{-02}$ | 0.43 |
| Analyst - e LS LLF | $3.93e^{-01}$ | $2.53e^{-01}$ | 8.94 |
| Analyst - i LS LLF | $5.42e^{-01}$ | $3.82e^{-01}$ | 13.5 |
| Analyst - e LS RF | $3.93e^{-01}$ | $2.49e^{-01}$ | 8.80 |
| Analyst - i LS RF | $4.20e^{-01}$ | $2.86e^{-01}$ | 10.1 |

Double integrator

The double integrator forms a typical element found in many dynamic systems. It is used here to compare the behavior of the Level Set method and the Fast Marching methods on a problem that is not of the Eikonal type. The double integrator problem will also be used to verify the behavior of the self-made Level Set method. A comparison will be between this scheme and Mitchel’s Level Set Toolbox Mitchell and Templeton (2005). The double integrator has the following dynamics:

$$x = f(x, u) = \begin{bmatrix} x_2 \\ u \end{bmatrix}, \quad u \in [-1, 1]$$  \hspace{1cm} (5-2)

The analytic solution for the associated minimal time problem is given in equation 7.26 of Athans (2007):

$$T = \begin{cases} x_2 + \sqrt{4x_1 + 2x_2^2}, & x_1 \geq -\frac{x_2|x_2|}{2} \\ -x_2 + \sqrt{-4x_1 + 2x_2^2}, & x_1 \leq -\frac{x_2|x_2|}{2} \\ |x_2|, & x_1 = -\frac{x_2|x_2|}{2} \end{cases} \hspace{1cm} (5-3)$$
Tests and demonstrations

Mitchell's toolbox comes with a set of example problems, one of which is a performance demonstrator that compares the analytic and the numerically calculated minimum arrival times of this double integrator problem. For a good comparison, the evolutionary level set value-function data from the self-built solver has to be converted to a single minimal arrival time value function. This has been done by finding the time iteration on which the value of a node changes sign and then improving the accuracy by linear interpolation.

The results are shown in figure 5-5 and figure 5-6 for a grid size of $N = [101, 101]$ and a domain of $x_1, x_2 = [-1, 1]$. A Local Lax-Friedrich scheme with first order finite differencing is used for the Level Set methods.

The comparisons have been made for both forward and backward reachability. The results are summarized in table 5-3, where the percentage of the RMS values is taken with respect to the largest analytic arrival time. The calculations have not been performed for the backward Eulerian Fast Marching method. M LS refers to Mitchell's level set method while SM LS refers to the self-made Level Set method. A couple of important observations can be made. Comparing the methods to the analytic values form figure 5-5a and figure 5-5b show that the different solving methods have slightly different features. When comparing between forward and backward reachable sets, no difference in performance is observed for any of the approximation methods, as can also be seen in table 5-3. In all cases except the Eulerian Fast Marching method, the estimated arrival times are conservative.

The self-made Level Set method (figure 5-5c and figure 5-5d) follows the characteristic shapes of the level set but the additional dissipation seems to smooth out the valley coinciding with the switch curve $x_1 = -x_2|x_2|/2$. The difference between the analytic and Level Set results is shown in figure 5-6c and figure 5-6d. These figures clearly show that the valleys are also regions with additional error. A comparison between the results from the self-made implementation and the results of Mitchell's toolbox is made in figure 5-6e. Interesting to see is that the two implementations result in different arrival times for regions close to the edge of the domain. In all other regions the average difference is in the order of $O(e^{-16})$. A closer inspection reveals that Mitchell's toolbox modifies the boundary conditions on the edges of the domain. Mitchell extrapolates ghost cell values such that they move away from the zero level set. This is done to avoid situations where a set appears from the boundary of the domain when the flow moves information from these ghost cells into the domain. When the physical set however reaches the domain boundaries, the same approach will promote the growth of the reachable set instead of preventing it, which again leads to nonphysical growth of the set in these regions. Figure 5-7 shows the same difference plot when Mitchell's extrapolation method is used in the self-made scheme. The numeric difference is given in table 5-3 under "M LS - SM* LS". These differences fall in the order of magnitude of computational inaccuracy.

The Eulerian Fast Marching method is not able to capture the characteristics of the problem, as can be seen in figure 5-6a and figure 5-6f. The Eulerian Fast Marching (EFM) method gives smaller arrival times in most regions than the analytic solution allows. It also fails to follow the curvature of the switch curve, resulting in large overestimations in these regions.
Figure 5.5: Results for the double integrator.
An explanation for this will be given in the next section.

The safe Fast Marching method performs considerably better when compared to the Eulerian method. It is not under-estimating the arrival times anywhere. It has less dissipation in the switch-curve valley when compared to the Level Set methods and captures this part of the problem better. The safe Fast Marching method however has more difficulty in the regions away from the valley, where the estimated arrival times are considerably larger. The SFM method also stops pre-maturely and there are two segments missing from the arrival time surface. In contrast to the results obtained for the Eikonal problem of section 5-1 the overall accuracy of the Level Set Method outperforms the safe Fast Marching method.

<table>
<thead>
<tr>
<th></th>
<th>$L_\infty$</th>
<th>RMS</th>
<th>RMS%</th>
<th>Backward</th>
<th>$L_\infty$</th>
<th>RMS</th>
</tr>
</thead>
<tbody>
<tr>
<td>M LS - SM* LS</td>
<td>3.197e-01</td>
<td>1.025e-02</td>
<td>3.197e-01</td>
<td>1.025e-02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>M LS - SM* LS</td>
<td>4.411e-16</td>
<td>3.700e-17</td>
<td>4.411e-16</td>
<td>3.700e-17</td>
<td></td>
<td></td>
</tr>
<tr>
<td>E FM - Analyt</td>
<td>1.294e00</td>
<td>3.309e-01</td>
<td>10.18</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>

**Table 5-3:** Difference measures for double integrator problem; $N = [101, 101]$ and $dx = 0.02$

**Mass-spring-Damper system**

The mass-spring-damper system is a generalization of the double integrator. Although the implementation here is linear, the system is easily extendable to problems with non-linear spring and damping coefficients. The system is particularly interesting for testing Ordered Upwind methods as the damping term causes the angle between the front and the characteristic curves to decrease when reaching away from the neutral position, converging to a situation where the characteristics are parallel to the front. Consequently, there are regions inside the state-space that can never be reached, provided that the permissible inputs are bounded and the damping coefficient is sufficiently large. The system is illustrated in figure 5-8 and the equation of motion is given below:

\[
\begin{pmatrix}
\dot{S} \\
\ddot{S}
\end{pmatrix} = \begin{pmatrix}
0 & 1 \\
-k/M & -c/M
\end{pmatrix} \begin{pmatrix}
S \\
\dot{S}
\end{pmatrix} + \begin{pmatrix}
0 \\
1/M
\end{pmatrix} F
\]  

(5-4)

For linear coefficients, the optimal input for obtaining maximal internal energy corresponds to applying a bang-bang controller with $F = F_{max}$ while $\dot{s} \geq 0$ and switch to $F = F_{min}$ once $\dot{s} < 0$. When the aim is to make the reachable set as large as possible, a more complicated control scheme has to be applied. The bang-bang controller however defines an outer boundary for the set.
Figure 5-6: More results for the double integrator
The system can be formulated as a HJB-PDE:

\[
\Phi_t + \max_{F} \left\{ \nabla \Phi, \left\langle \frac{\partial F}{\partial S}, \frac{\partial \Phi}{\partial S} \right\rangle \right\} = 0 \\
\Phi(t_0) = g_0(S, \dot{S})
\]  

For the initial set a diamond-shaped region around the neutral state is chosen and is described as \(g_0(S, \dot{S}) = |S| + |\dot{S}| - \varepsilon\) with \(\varepsilon\) a small reduction to make sure that the level set is seen by the discretization. The time increment is chosen to satisfy the CFL stability criteria and is implemented as \(dt = \frac{\alpha}{\max(\frac{\partial F}{\partial S}, \frac{\partial F}{\partial S'})} \). The CFL stability margin and is set to 0.75.

This system has been used to study a couple of problems. First of all a not-maintainable reachability problem has been solved with the Level Set method by initializing the set away from the neutral point. The result is an oscillating and growing set. The problem is made stationary by initializing on the neutral point. With this setting, five more implementations of the Ordered Upwind methods are tested.

The first method is to apply the Eulerian Fast Marching method. The dimensions without converged nodes will be ignored in the update scheme. The second method is the semi-Lagrangian fast marching method, where extrapolation between converged and not
Table 5-4: Input parameters for the eccentric (1) and centered (2) spring-damper problem

<table>
<thead>
<tr>
<th></th>
<th>Problem 1</th>
<th>Problem 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k/M$</td>
<td>0.2</td>
<td>0.2</td>
</tr>
<tr>
<td>$c/M$</td>
<td>0.2</td>
<td>0.2</td>
</tr>
<tr>
<td>$F/M$</td>
<td>[0.6, 0.9]</td>
<td>[−0.15, 0.15]</td>
</tr>
<tr>
<td>$\varepsilon$</td>
<td>1</td>
<td>0.2</td>
</tr>
<tr>
<td>$X_{\text{min}}$</td>
<td>[−4.5, −3.5]</td>
<td>[−3, −3]</td>
</tr>
<tr>
<td>$X_{\text{max}}$</td>
<td>[11.5, 3.5]</td>
<td>[3, 3]</td>
</tr>
<tr>
<td>$N$</td>
<td>[321, 141]</td>
<td>[121, 121]</td>
</tr>
</tbody>
</table>

converged values is made possible by using the Kružkov transform. The third method uses the same principles as the second, except that is avoids the Kružkov transform by replacing infinite arrival times with a large finite value. This method is also extended to an iterative scheme by repeating the computations with the previous results as the initial value function. The fourth method is to use the safe Fast Marching method with a multi-stencil of 3x3 nodes. Finally, this method is extended to include a trajectory in the initial set to observe the behavior of trajectory usage as described in section 3-2. It should be noted that even the safe Fast Marching method is a rather limited version of an ordered upwind method, since it is only implemented with a small stencil.

Two implementations for the Level Set method were made with the mass-spring-damper system. The input parameters of both are summarized in table 5-4. Problem 1 is eccentric in the sense that the control input limits are not symmetric around the neutral input. As a consequence the set extends in an oscillative fashion, which means that a minimum arrival time cannot reproduce the reachability tube. (The opposite is always possible, as can be derived from section 2-5).

The results of the first problem are shown in figure 5-9. The reachability tube of figure 5-9a is calculated with the Level set method with 2000 iterations over 10.5 seconds. It shows a spiraling motion through time. Figure 5-9b shows the minimum arrival time as derived from the reachability tube. The two green spirals in this plot represent two optimally excited and damped trajectories and are calculated with a Lagrangian forward Euler integration. The outer trajectory is excited in the resonant frequency with a permissible bang-bang input while the inner trajectory is controlled for maximum dissipation. This trajectory becomes stationary after reaching the neutral point. This figure shows two things that are of importance. First of all, the reachable set is conservative compared to the two trajectories. Secondly it is clear that although the reachability tube contains all information needed to build a first arrival time surface, this surface does not contain the information needed to reconstruct the reachability tube.

The second problem shows the behavior of a centrally excited system. In this case, the algorithm produced non-physical sets on the boundary of the computational domain after approximately 300 iterations. Due to the smaller input signals there now is a region inside the domain where the damping is stronger than the applied force. As a consequence information flows into the domain as opposed of away from it. To resolve this, a very
crude implementation of a re-initialization scheme has been implemented, which after every 200 iterations generates a signed distance function using the nodes with negative value as intermediate set. By re-initializing some errors in the progression of the set, may occur. Typically, the arrival times will become somewhat conservative because of this.

The results of the second problem are shown in figure 5-10. The calculations were performed over 2000 iterations, which took 10.9 seconds to calculate. Comparing the reachability tube and minimum arrival time surface shows that for this problem, where the initial set is (essentially) maintainable, the two figures contain the same information. The set is again conservative, although not as much as is the case in problem 1.

Problem 2 has also been used as a test-bed for the Level Set methods. The same grid resolution has been used for these calculations. The initial set however has been chosen a little smaller. The minimum arrival times are shown in figure 5-12. Each method is compared to two optimally expanding trajectories. Each will be discussed briefly.

Figure 5-12a shows the results of the Eulerian fast marching method, which ignores the dimensions without converged values. The estimation does not follow the trajectories and even for $T < 10$ the surface is not even remotely approximating the Semi Lagrangian approximation. It appears that the set “leaks”. After some investigation it was discovered that the projection of the problem to a lower dimension is the main cause of failure for this approach. This is illustrated in figure 5-11. By reducing the dimensionality of the problem, it becomes possible to incorrectly conclude that a node is reachable. If the projected problem is extended back into the original problem, the missing dimensions are extrapolated as flat, infinite extensions. It therefore ignores the finiteness of the set that would otherwise prevent a node from being reached. This approach is therefore only applicable to Eikonal equations, where the direction of flow is guaranteed to coincide with the gradient of the front.
Figure 5-10: Results for problem 2 as estimated using the Level Set method

(a) Reachability tube in time
(b) Minimum arrival time surface

Figure 5-11: Illustration of cause of leakage. Left: node will not be reached from set. Right: Projected problem is reachable.

Figure 5-12b shows the value function of the Kružkov transformed semi-Lagrangian fast marching method with a two-point stencil. Instead of ignoring dimensions, this implementation accepts to use nodes that are not yet converged. The Kružkov transform allows interpolating between finite and infinite values. The solution produces conservative values until $T \approx 18$, after which the arrival times are under-estimated. A large region of the domain outside the steady set is assigned a value before the arrival time of $T = 36.74$ is reached for the remaining domain. This latter phenomenon can be attributed to the numerical resolution of the floating-point operator. For $T = 35.5$ a time difference of 0.1 results in an increment of $O10^{-17}$ after applying the Kružkov transformation. This is smaller than the resolution of a double floating point of magnitude one, and hence is neglected when transformed back to the time domain. The over-approximation is expected to be caused by the interpolation between not-converged nodes. Mathematically, this approach will always produce a finite value, even if the considered node is not reachable. This approach therefore only works for problems that are fully reachable.

Figure 5-12c shows the arrival time surface of the two-point stencil semi-Lagrangian Fast Marching method. Instead of a Kružkov transform, the nodes in the 'far' region are initialized with a value larger than the infinite arrival time. Here this value is set to $T_{\text{max}} = 60$. Since
the values are not transformed, the interpolation is linear. The resulting surface is very conservative. The maximum arrival time also provides a stopping criteria as no node can become larger.

Figure 5-12d shows an attempt to implement an iterative scheme. For each iteration, the nodes in the far region are given the values of the previous iteration and thereby reduce the interpolation penalty. The result is shown for 50 iterations. The value function now follows the two trajectories more closely. The arrival times are conservative everywhere except when approaching $T_{\text{max}}$. Here the interpolation still allows unreachable nodes to receive a value smaller than $T_{\text{max}}$. One attempt to overcome this is to use a threshold value $T_i < T_{\text{max}}$. At the end of each iteration, all values larger than this threshold are set to the maximum value: $T(x) = T_{\text{max}}, \forall x|T(x) > T_i$. Provided that the difference between $T_i$ and $T_{\text{max}}$ cannot be overcome in one iteration for unreachable nodes, this threshold may prevent over-estimation of the reachable set. When $T_i$ is too small it can also prevent reachable states from being assigned a value. The main challenge is therefore to find an appropriate threshold. Figure 5-12e shows the result for $T_i = 0.95T_{\text{max}}$. The reachable set is now conservative everywhere while still approximating the two trajectories. It is suspected that these issues can also arise in similar iterative boundary value schemes like the fast sweeping method.

Figure 5-12f shows the results from the semi-Lagrangian safe Fast Marching method with a multi-stencil. Since the method is not allowed to use not-converged nodes, it stops when there are no candidate nodes inside the observed stencil. For the 3x3 stencil used here, the calculated set is relatively small. The nodes that are found however closely follow the trajectories.

Figure 5-12g demonstrates the proposed idea of combining the depth-first and breadth-first methods by including a trajectory in the initial set. During implementation, it was found that the initial values of the nodes visited by the trajectory can be set to the arrival times of this trajectory. An interesting note is that since a permissible trajectory is always slower than or as fast as an optimal controlled system an iterative minimal time method should be able to find the correct arrival time. The single-pass method used here however will result in conservative arrival times when the used trajectories are not optimal. This conservation of arrival time knowledge is not possible for the Level Set method where the front of the set must have the same arrival time. As an example, one trajectory is made over a horizon of $t = [0, 40]$ and projected on the grid. Using this initial set the safe Fast Marching method is solved again. The added trajectory greatly improves the region on which the method can solve the problem.

## 5-2 Simplified aircraft model

To demonstrate the performance of the schemes on a more aerospace related system, the aircraft model used in Lygeros (2004), Lombaerts et al. (2013b), Schnet et al. (2014) has been implemented. It consists of a simplified model of the slower symmetric motions. The state space is described by the velocity $V$ and the flight path angle $\gamma$. The relevant axis and
Figure 5-12: Results for problem 2 as estimated using the Fast Marching methods
vectors are defined in figure 5-13.

![Figure 5-13: FBD for the aircraft model](image)

Apart from the angles and forces shown here, the influence of the sideslip angle $\beta$ is also included in this model but not used as an input to be optimized. The considered equations of motion are given here:

$$
\begin{bmatrix}
\dot{V} \\
\dot{\gamma}
\end{bmatrix}
= 
\begin{bmatrix}
-\frac{\rho S}{2m} V^2 C_{D_0} - g \sin \gamma + \cos \alpha \cos \beta \frac{T}{m} - \frac{\rho S}{2m} V^2 \left( C_{D_\alpha} \alpha + C_{D_{\alpha^2}} \alpha^2 \right) \\
-\frac{S}{V} \cos \gamma + (\cos \phi \sin \alpha \cos \beta - \sin \phi \cos \beta) \frac{T}{V m} + \frac{\rho S}{2m} V \left( C_{L_0} + C_{L_\alpha} \alpha \right) \cos \phi - \frac{\rho S}{2m} V C_{Y_\beta} \beta \sin \phi
\end{bmatrix}
$$

(5-6)

This model can be simplified by assuming small angles:

$$
\begin{bmatrix}
\dot{V} \\
\dot{\gamma}
\end{bmatrix}
= 
\begin{bmatrix}
-\frac{\rho S}{2m} V^2 C_{D_0} - g \sin \gamma + \frac{T}{m} - \frac{\rho S}{2m} V^2 \left( C_{D_\alpha} \alpha + C_{D_{\alpha^2}} \alpha^2 \right) \\
-\frac{S}{V} \cos \gamma + \frac{\rho S}{2m} V \left( C_{L_0} + C_{L_\alpha} \alpha \right) \cos \phi - \frac{\rho S}{2m} V C_{Y_\beta} \beta \sin \phi
\end{bmatrix}
$$

(5-7)

The control parameters are $T \in [20546N, 410920N]$, and $\alpha \in [0^\circ, 14.5^\circ]$. The following aerodynamic derivatives will be used: $C_{D_0} = 0.1599$, $C_{D_\alpha} = 0.5035$, $C_{D_{\alpha^2}} = 2.1175$, $C_{L_0} = 1.0656$, $C_{L_\alpha} = 6.0723$ and $C_{Y_\beta} = -1.6$. The remaining terms are: $m = 120 \cdot 10^3$ kg, $\rho = 1.225$ kg/m$^3$, $S = 260$ m$^2$ and $g = 9.81$ m/s$^2$.

There are a couple of challenges to this problem. The dynamics are non-linear and quadratic in the control inputs. When a model uncertainty is implemented as a disturbance these will be coupled to the control inputs. All of this complicates the optimization problem. For the Level Set method it is possible to find the optimal control inputs when assuming there are no disturbance terms. Under this condition, all inputs are decoupled, and all signals but $\alpha$ are input affine. For the affine terms the optimal input can be based directly on the sign of the value function’s gradient. For the quadratic term, one can identify the optimal input by taking the derivative of the involved terms. Then either this value has to be chosen or an input as far away from it, depending on whether the optimum is a maximum or a minimum. This second criteria is easily found using the second derivative of the involved terms and turns out to be only depending on the first dimension of the value-function’s gradient. The resulting control logic is worked out in Lombaerts et al. (2013a) and is repeated here. Define the critical value $\bar{\alpha} = \frac{2 \rho S C_{L_\alpha} \cos \phi - \rho V C_{D_{\alpha}}}{2 \rho V C_{D_{\alpha}}}$ and mid-range angle $\bar{\alpha} = \frac{\alpha_{\text{min}} + \alpha_{\text{max}}}{2}$. Then to minimize the Hamiltonian, one has to choose the optimal inputs in accordance to the following rules:

- If $p_1 > 0$ then $T^* = T_{\text{min}}$ and

J.C.J. Stapel
Efficient Flight Envelope Estimation for Changed Aircraft Dynamics
- if \( \dot{\alpha} \geq \dot{\alpha} \) then \( \alpha^* = \alpha_{\min} \)
- if \( \dot{\alpha} < \dot{\alpha} \) then \( \alpha^* = \alpha_{\max} \)

- If \( p_1 = 0 \) then \( T^* = T_{\min} \) and
  - if \( p_2 \geq 0 \) then \( \alpha^* = \alpha_{\min} \)
  - if \( p_2 < 0 \) then \( \alpha^* = \alpha_{\max} \)

- If \( p_1 < 0 \) then \( T^* = T_{\max} \) and
  - if \( \dot{\alpha} \leq \alpha_{\min} \) then \( \alpha^* = \alpha_{\min} \)
  - if \( \dot{\alpha} \leq \alpha_{\min} \) then \( \alpha^* = \alpha_{\min} \)
  - if \( \alpha_{\min} \leq \dot{\alpha} \leq \alpha_{\max} \) then \( \alpha^* = \dot{\alpha} \)

- If \( p_2 \sin \phi \geq 0 \) then \( \beta^* = \beta_{\min} \)
- If \( p_2 \sin \phi < 0 \) then \( \beta^* = \beta_{\max} \)

Here \( p_1, p_2 \) are the components of the value-function’s gradient. When model parameter uncertainty is included in the problem, most of these conditions do not change as long as the coefficients to not change sign. An exception to this rule are the \( \alpha \)-related coefficients which may alter the value of \( \dot{\alpha} \). The disturbance inputs should therefore be calculated before the control inputs when the controller is expected to account for model uncertainties. For the disturbance inputs, the following rules apply in order to maximize the Hamiltonian:

- If \( p_1 \geq 0 \) then \( C^*_{D_0} = C_{D_0 \min}, C^*_{D_\alpha} = C_{D_\alpha \min}, C^*_{D_{\alpha^2}} = C_{D_{\alpha^2 \min}} \)
- If \( p_1 \leq 0 \) then \( C^*_{D_0} = C_{D_0 \max}, C^*_{D_\alpha} = C_{D_\alpha \max}, C^*_{D_{\alpha^2}} = C_{D_{\alpha^2 \max}} \)
- If \( p_2 \geq 0 \) then \( C^*_{L_0} = C_{L_0 \max}, C^*_{L_\alpha} = C_{L_\alpha \max} \)
- If \( p_2 \leq 0 \) then \( C^*_{L_0} = C_{L_0 \min}, C^*_{L_\alpha} = C_{L_\alpha \min} \)
- \( C^*_{Y_\beta} = C_{Y_\beta \max} \)

The simple condition for \( C^*_{Y_\beta} \) follows from the fact that \( \beta \) will always be selected to satisfy \( p_2 \sin \phi \beta < 0 \). Alternatively to solving the conservative minmax problem, one may also consider solving a problem with a non-robust controller. This can be done by forcing the control logic to assume a model without uncertainties and use the disturbances when calculating the actual Hamiltonian.

For the Ordered upwind methods, determining the optimal inputs is a bit more complicated as no gradient is readily available for the node under evaluation. To resolve this, an estimation has to be made of the gradient or an alternative optimization approach should be used. For the purpose of demonstration, a brute-force optimization will be implemented. Although this is one of the least elegant approaches, it is also the simplest to implement. The controller can select from ten values for \( \alpha \) and choose between the upper and lower limits.
for the remaining controls. This results in a total of 40 different combinations of control inputs.

The model is implemented in the Level Set method as well as in the safe Fast Marching method. To verify that the model is implemented correctly some of the results from Lombaerts et al. (2013b) and Schuet et al. (2014) are re-produced with the Level Set scheme. All calculations have been made on a uniform 150x150 grid with the dimensions as shown in figure 5-14. A local Lax-Friedrichs scheme has been used for the dissipation and both the spatial derivative and the temporal integration were performed with second order schemes. As these settings may differ from the ones used in Lombaerts et al. (2013b) and Schuet et al. (2014), small differences can be expected between the results produced here, and those published by Lombaerts and his colleagues.

The first paper represents reachable sets but also invariance and viability kernels of the aircraft model on a time horizon of two seconds and observes the influence of different lift and drag coefficients on these sets. It also studies the influence of parameter uncertainty. The initial set used in this paper however does not form a maintainable set. The green sets of figures 5-14c, 5-14f, 5-14g, 5-14h and 5-14i are taken from figures 7, 8 and 13 of Lombaerts et al. (2013b) respectively.

A reasonable correspondence can be seen in figure 5-14c. The backward reachable set (largest set) coincides on the upper and lower face. The left and mostly the right face show a slight difference. These are expected to be caused by a different set of solving parameters. The viability kernels coincide as well, although Lombaerts’ set also shows signs of slightly rounded corners, which are most likely caused by the used visualization techniques. The invariance kernel (smallest set) is again a little larger when compared to Lombaerts’ set but resembles the same features.

The forward reachable sets as shown in figure 5-14f do not coincide as accurately. The set published by Lombaerts is shifted to the right compared to the one that is produced here. Lombaerts also appears to use augmented maintainability for this set. No explanation for this difference has been found.

Figures 5-14g, 5-14h and 5-14i compare the backward reachable sets for three roll angles. Each figure also shows the shrinking of these sets as a consequence of adding 10% and 20% model uncertainty. Again there is a large difference with the results from Lombaerts. For the larger roll angles, even the upper edge shows a large offset. The simulations performed to replicate Lombaerts figures assume that the same model is used. Lombaerts’ backward reachable set without model uncertainty of figure 5-14g however is different from what is assumed to be the same set depicted in figure 5-14c. No clue on what causes this difference was found in the paper.

The second paper uses a trimmable initial set on the same system dynamics. Here the nominal flight envelope is compared to situations with 20% extra drag and 20% reduced lift (figures 5-14a and 5-14d), as well as an additional 50% decrease in available maximum thrust.
Table 5-5: Bounding boxes for trimmed initial set

<table>
<thead>
<tr>
<th>Case</th>
<th>Speed [m/s]</th>
<th>Gamma [deg]</th>
</tr>
</thead>
<tbody>
<tr>
<td>nominal</td>
<td>[54, 82]</td>
<td>[-6.5, 11]</td>
</tr>
<tr>
<td>20% L/D</td>
<td>[60, 92]</td>
<td>[-10.2, 6.5]</td>
</tr>
<tr>
<td>50% Tmax</td>
<td>[60, 92]</td>
<td>[-10.2, -3.9]</td>
</tr>
</tbody>
</table>

(figures 5-14b and 5-14e). For these sets, a temporal horizon of five seconds is used. Also the model without small angle assumptions was used in the calculation of the Hamiltonian, as this was found to provide a better match with the results in Schuet et al. (2014). The initial sets used in the paper are simplified to rectangular boxes. In an attempt to replicate this, the boxes described in table 5-5 were assumed.

The forward reachable sets again show a good correspondence, although a small difference can be observed in the upper and right faces. Also the backward reachable sets compare well. One exception is the impaired set in figure 5-14d, where the calculated set drifts off when approaching the right edge of the domain. A brief investigation showed that this was caused by the boundary conditions imposed on the edges of the domain. The anomaly disappears when a larger domain is used.

From these comparisons, it can be concluded that the model is correctly implemented for the nominal backward reachable set. Although the forward set appears to be unsatisfying when compared to the result presented in Lombaerts et al. (2013b), a reasonable match with the results in Schuet et al. (2014) was found. Since no successful comparison was achieved for non-zero roll angles, this aspect of the model cannot be considered to be verified. Despite this, the implemented model can still serve as demonstration problem.

After this verification step, the nominal model is also implemented in the safe Fast Marching method and compared to these results. To make a better comparison, the Level Set method is configured to use first order spatial differencing and a first order runge-Kutta integration scheme. Figure 5-15 shows the forward and backward reachable sets for ten arrival times to a horizon of five seconds. The same grid resolution has been used as before. The maintainable initial set is no longer simplified on a rectangular box but is calculated on the same grid. The difference between the safe Fast Marching method and the Level Set method is shown in figures 5-15e and 5-15f.

A couple of interesting observations can be made. As before, the safe Fast Marching method stops prematurely in regions where the gradient of the set and the characteristic curves are not sufficiently aligned, leaving distinctive edges. The safe Fast Marching method arrives a little later in most regions, except on the upper right edge of the forward reachable set, where it arrives earlier. It is suspected that the dissipation of the Level Set method has a significant impact on how the upper right corner of the initial set is extended. The Fast Marching method seems to extend this set further to the right.

The backward reachable set has two regions where the safe fast marching method arrives
Figure 5-14: Reachable sets as produced with the Level Set method, in comparison with figures 9, 10 from Schuet et al. (2014) and figures 7, 8, 13 from Lombaerts et al. (2013b)
considerably later than the Level Set method. There are also two small regions where the Fast Marching method is a little faster. Especially the right region experiences a sudden increase in arrival times. There are a couple of possible explanations for this. First of all it may be possible that the anisotropy prevents the safe fast marching method from progressing the front in the same manner as the Level Set method. A given node may only be solved once it is approached from a different direction at a later iteration or is only reachable form the current front when using a control input that is sub-optimal compared to what the Level set method can use. The safe Fast marching method also has to bridge the space between grid nodes in a single iteration while the Level Set method may use several steps. A final reason for different arrival times is the fact that the Level Set method uses an initial value function whose overall shape influences the progression of the front through stability criteria, dissipation and numerical derivatives.

The data in table 5-6 has been collected to compare the computational performance. Even with more efficient optimization, the Level Set method requires 226 times as many node evaluations for the forward reachable set and 158 times as many for the backward reachable set. The difference for the number of model evaluations is less impressive in this example due to the inefficient optimization method used in the Fast Marching method, where 40 input combinations are tested for optimization. The Level Set method requires only nine model evaluations to figure out the correct upwind and dissipation terms. An additional advantage of the safe fast marching method is that no computations are performed on nodes outside the reachable set. The computation time however is considerably larger for the fast marching method. There are a few reasons for this. First of all, the Level Set method has been implemented to make maximum use of the BLAS libraries. The safe Fast Marching method is less suitable for parallelization and has to do with the more involved for-loop. Since MatLab is not pre-compiling, there is a lot more overhead involved with this method. A second reason is that the Semi-Lagrangian method requires a considerable amount of geometric calculations. These could however be simplified. An indication for the potential computational speed can be found in table 1 of Cacace et al. (2014). Here the Fast Iterative method requires 5.3 seconds for a 101x101 grid. Apart from the computational performance, one can also observe the quality of the result. Table 5-6 also shows the maximum and average difference between the two methods. The percentage of missed nodes of either method is measured against the union set of reached nodes. The surplus of nodes reached in the Level Set method is caused by the inability of the safe Fast Marching method to capture the regions with high anisotropy. The few nodes that are uniquely identified by the safe Fast Marching method are most likely not found by the Level Set method due to the shrinking of the set caused by dissipation.

As an additional verification step, two sets of Monte Carlo analysis have been made on the reachable sets. The first set considers a large set of randomly generated trajectories initialized throughout the state-space. The second set considers randomly generated optimal control through sets of candidate increments using the estimated arrival time surfaces of the Level Set and safe Fast Marching methods as optimality criteria. Both methods are illustrated in figure 5-16.

The first method generates 1000 random sample states. Only those that have an estimated arrival time larger than one second or that lie outside the identified reachable domain will
Tests and demonstrations

(a) backward LS
(b) forward LS
(c) backward sFM
(d) forward sFM
(e) backward sFM-LS
(f) forward sFM-LS

Figure 5-15: Arrival time iso-lines and difference surface for the forwards and backward reachable sets
Table 5-6: Operational performance of the Level Set method and safe Fast Marching method on the aircraft model

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Forward LS</th>
<th>Backward LS</th>
<th>Forward sFM</th>
<th>Backward sFM</th>
</tr>
</thead>
<tbody>
<tr>
<td># iterations</td>
<td>364</td>
<td>400</td>
<td>3.84</td>
<td>3.89</td>
</tr>
<tr>
<td># model evaluations</td>
<td>3276</td>
<td>3600</td>
<td>1443640</td>
<td>2269000</td>
</tr>
<tr>
<td># model evaluations/node update</td>
<td>9</td>
<td>9</td>
<td>40</td>
<td>40</td>
</tr>
<tr>
<td># nodes/evaluation</td>
<td>22500</td>
<td>22500</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>tot # node evaluations</td>
<td>8190000</td>
<td>9000000</td>
<td>36091</td>
<td>56725</td>
</tr>
<tr>
<td>tot # model evaluations</td>
<td>73710000</td>
<td>81000000</td>
<td>144340</td>
<td>2269000</td>
</tr>
<tr>
<td>nodes evaluated</td>
<td>22500(100%)</td>
<td>22500(100%)</td>
<td>8940(39.7%)</td>
<td>14582(64.81%)</td>
</tr>
<tr>
<td>CPU time(^1) [s]</td>
<td>15.397</td>
<td>17.191</td>
<td>102.976</td>
<td>164.020</td>
</tr>
<tr>
<td>nodes missed by sFM</td>
<td>10.7%</td>
<td>7.89%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Nodes missed by LS</td>
<td>1.21%</td>
<td>0.89%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RMS</td>
<td>0.158</td>
<td>0.207</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(L_\infty)</td>
<td>1.511</td>
<td>1.487</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 5-16: Illustration of the two Monte Carlo principles: a cluster of random trajectories (left) and a single trajectory constructed from sets of random candidate increments (right)

be used for evaluation. Each sample is first tested with 80 unique sets of constant input signals. This results in 80 trajectories running over five seconds. If the end state of one of these trajectories lies inside the initial set the node is verified as a reachable state. For all nodes where this first verification step is unsuccessful a second batch of 10000 randomly selected piece-wise constant input signals is evaluated that change inputs half way through the simulation.

The results of one realization is shown in figure 5-17. Although the trajectories are chosen arbitrarily, most nodes inside the estimated reachable sets are verified. A few nodes on the right side of the forward in figure 5-17c are verified outside the set estimated by the Level set method. These nodes do fall inside the set estimated by the safe Fast marching method. Both safe Fast Marching sets contain large regions for which the estimated set is smaller than what the verified samples suggest. In turn, these samples are captured by the Level Set method. Both observations give an indication that the two estimation techniques result in a conservative solution.

\(^1\)Computed on an EliteBook 8570w Mobile Workstation running Windows@7 Home Premium and MatLab R2013a. Logged with MatLab's Profile toolbox
The First Monte Carlo method has not been able to verify the bottom-left limit of the two backward reachable sets. Since it is unlikely that the optimal trajectory is found by random inputs, a more sophisticated Monte Carlo analysis has been set up to verify arrival times more accurately.

The second Monte Carlo simulation produces a single, stochastically optimal trajectory for any sample state inside the estimated reachable set. Instead of producing a large number of global trajectories, the principle of dynamic programming is applied by first generating a set of candidate trajectory increments and then selecting the one that results in the best performance. On each time increment of 0.02 seconds, the dynamic system is solved for a set of 51 random permissible inputs. The candidate end state of this increment with the smallest value on the arrival time surface as estimated by the Level Set or Fast Marching method is accepted. The corresponding set of inputs is included in the set of trial inputs for the next iteration.

Figure 5-18 shows the resulting trajectories and initial states. The initial states for which the trajectory successfully reaches the initial set within 110% of the estimated arrival time are colored green while the remaining initial states are red. A small number of trajectories escape from the region in which the estimated arrival time surface is properly defined and
can no longer use it to optimize the control inputs. These trajectories should closely resemble the time-optimal paths. Figure 5-18a (and also figure 5-17a) have a set of unverified nodes on the left edge of the reachable set. This concentration of unverified nodes could indicate an overestimate. It is expected that the the Level Set method has incorrectly extended the lower-left corner of the trimmable set due to dissipation. Alternatively, the considered states can be particularly unforgiving for using sub-optimal inputs. The trajectories found from the unverified states pass close to the bottom-left corner. A truly optimal trajectory might have been able to reach this corner.

Table 5-7 gives an overview of these trajectories. The reach-rate indicates the fraction of initial states for which the trim set was reached successfully. The Success rate gives the fraction of these trajectories that reach the trim set within the time estimated by the arrival time surface. The remaining values give information on the arrival times obtained by the trajectories in comparison to the estimated arrival time. The standard deviation of the spread in arrival time fractions is given as well. Interesting to note is that on average the successful trajectories of the Monte Carlo simulation predict approximately 60% faster arrival times than the Level Set and safe Fast Marching method. Although only first-order approximations were used in the estimation process and although no design effort was put in optimizing the estimation accuracy, these results give a strong indication that the estimation techniques tend to be very conservative. It is therefore recommended to obtain further insight in techniques to improve accuracy of the arrival time estimations.
Figure 5-19: Full nominal and uncertain reachable sets and envelope for arrival time of 5 s (red) and trimmed initial sets (green)
Table 5-7: Statistical results from the Monte Carlo simulation with randomly generated optimal inputs.

<table>
<thead>
<tr>
<th>problem</th>
<th>Reach rate</th>
<th>Success rate</th>
<th>Minimum T</th>
<th>Maximum T</th>
<th>mean T</th>
<th>std</th>
</tr>
</thead>
<tbody>
<tr>
<td>LS forward</td>
<td>79.86%</td>
<td>96.93%</td>
<td>20.40%</td>
<td>108.00%</td>
<td>60.13%</td>
<td>21.69%</td>
</tr>
<tr>
<td>LS backward</td>
<td>98.18%</td>
<td>93.98%</td>
<td>19.20%</td>
<td>109.60%</td>
<td>60.68%</td>
<td>23.01%</td>
</tr>
<tr>
<td>sFM forward</td>
<td>99.05%</td>
<td>98.57%</td>
<td>17.60%</td>
<td>105.60%</td>
<td>56.64%</td>
<td>21.89%</td>
</tr>
<tr>
<td>sFM backward</td>
<td>99.80%</td>
<td>96.30%</td>
<td>14.80%</td>
<td>108.00%</td>
<td>57.51%</td>
<td>22.71%</td>
</tr>
</tbody>
</table>

Figure 5-19 shows the full reachable sets and flight envelopes as calculated by the Level Set method as well as the Fast Marching method. The results are given for both the nominal dynamics as well as for a system with 20% uncertainty in all aerodynamic coefficients, ranging from 80% to 120% of the nominal values. The surfaces in the center of each set represent the set of trim conditions for zero sideslip. In all cases, the safe Fast Marching method is more conservative than the Level Set method in most regions. A brief explanation for the emerging features will be given here. Only the main contributing elements are considered to simplify the discussion.

The trimmable set for the nominal dynamics shifts to smaller flight path angles and higher velocities as the roll angle increases. A larger lift is required to maintain the same flight path angle because a smaller component of the Lift counteracts the aircraft Weight. To Compensate, a Larger Lift force is obtained by increasing the velocity. This also moves the angle of attack constraints to larger velocities. Since this velocity increase also increases the total drag, a smaller power margin is left to invest in a climb rate. Both the upper and lower limit of the maintainable flight path angle move down because of this.

The trimmable set for the model with 20% uncertainty is smaller than the nominal case for zero roll angle and becomes even smaller for larger values for \( \phi \). When \( C_L \) becomes smaller the consequences are similar to flying under a larger roll angle. A larger velocity is needed to maintain the same Lift and more power has to be invested in compensating the larger drag. This reduces the upper flight path constraint and increases the lower velocity constraint. Similarly the lower limits for \( \gamma \) moves up and upper limit for \( V_{\gamma=0} \) moves down when \( C_L \) becomes larger. Decreasing \( C_D \) mainly lowers the power required to compensate the drag, leaving a larger margin that has to be invested in a climb rate. The lower limit for \( \gamma \) therefore has to move up. Similarly the upper limit moves down when a larger \( C_D \) value is used.

When the roll angle is increased for the uncertain model, the Trim set becomes significantly smaller. Combining the effects of the separate phenomena explains this. Both the model uncertainty and the larger roll angle lower the upper bound on \( \gamma \). For the lower \( \gamma \) bound the increase caused by the model uncertainty is counteracted by the reduction due to increased \( \phi \). Similarly the lower \( V_{\gamma=\text{max}} \) boundary is increased by both changes while they counteract for the upper \( V_{\gamma=\text{min}} \) boundary. The combined effect makes it impossible to maintain both flight path angle and velocity with a bank angle of 60 degrees.
The forward and backward reachable sets produced from these trim sets are also influenced by the roll angle and model uncertainty. The reachable sets are mainly extended through two processes. First of all, more power can be invested/drained from one of the two dimensions by accepting a chance in the other dimension. Larger velocities can be achieved by accepting an increasing flight path angle, or made smaller than the trim set by accepting a decreasing angle. This is illustrated in figure 5-20

![Figure 5-20: Illustration of the input constraints on the maintainable set of the aircraft model](image)

The second process is to rapidly exchange potential and kinematic energy. If a sufficient velocity is obtained it can be exchanged with altitude by pitching up, temporarily increasing the flight path angle. Once $\gamma = 0$, the highest point in the maneuver is achieved. If this state is not maintainable the potential energy will be exchanged back to kinematic energy, increasing the velocity and temporarily reducing $\gamma$.

Looking back at figure 5-18 shows that both methods are used for the optimal control. In both forward reachable sets the upper-right nodes are reached by first increasing the velocity with a slow increase of $\gamma$ after which the trajectory rapidly moves up by exchanging energy, passing through the evaluation node at the desired velocity and flight-path angle.
Chapter 6

Conclusion and Recommendations

To aid the need for real-time flight envelope estimation, this research has identified and explored new methods to calculate reachable sets in the application of model-based flight envelope estimation. The approach advocated in Tang et al. (2009) estimates the flight envelope by solving the Hamilton Jacoby Isaacs (HJI) partial differential equation (PDE) with the Level Set method of Osher and Fedkiw (2004). The high dimensionality of the problem however makes this approach too slow for on-line applications.

This report describes an investigation in new methods to improve the computational performance of this method. The main research question is formulated as follows: In which ways can we reduce the solving time of non-linear, time-invariant HJI problems with an application in flight envelope estimation by simplifying the solving techniques?

In a search for methods that improve the solving speed, five candidate approaches were examined on applicability. The first method considered was to split the problem in a set of lower-dimensional sub problems. The second was to localize the calculations to regions of higher interest. The third method was to see if an estimated set could converge to the correct set. The fourth was to use the boundary value formulation. The fifth approach was to simplify the computations themselves. The use of the boundary value formulation was assessed for different solving schemes.

To make an early assessment on the applicability of these methods a literature study has been performed. The investigation was then extended to prove a set of properties that were then used to assess the applicability of these approaches. A set of calculation methods have been developed and implemented to test a new hypothesis on the usability of the Fast Marching schemes and to demonstrate some of the identified concepts. The developed algorithms were subjected to standard problems like a convection problem, a double integrator and a mass-spring-damper system. Finally the safe Fast Marching (SFM) method and the Level set method were implemented on simplified aircraft model of the slow maneuver dynamics.
The following conclusions were found during this research.

It was found in Mitchell and Tomlin (2003) that splitting the problem in lower dimensional sub-problems will result in an over-approximation of the reachable set and is therefore not recommendable for flight envelope estimation.

A simple thought experiment has shown that it is not possible to guarantee convergence of an arbitrary initial set. Similarly it has been shown that a set found by recursively propagating a reachable set over a time-variant event cannot be used for the backward reachable set due to causality and does not produce a reachable set suitable for flight envelope estimation.

It was argued that permitting sub-optimal solutions by simplifying the optimization problem will result in more conservative minimal arrival times but should not affect the accuracy of the steady set at infinite horizon. This simplification could save considerable computation time in the optimization aspect of the problem solving, which is of particular interest when considering more complex control problems. The simplification may be extended to the update schemes as well. Since no application of this idea has been tested, it is still open for further investigation.

It was further proven that under the assumption of a time-invariant system and a maintainable initial set the reachable tube can be found from the minimal arrival time. This time can be found by the stationary boundary formulation. The schemes for this method are not subjected to the Courant-Friedrichs-Lewy (CFL) stability criteria and do not require augmented dissipation in order to find a weak solution.

A variety of Fast Marching methods have been tested for applicability on a set of simple test problems. Although it was hypothesized that the classical Fast Marching method may be able to correctly identify the steady reachable set, it was found that this is not the case. A physical explanation for this was found for both the Eulerian and the semi-Lagrangian approach. The safe Fast Marching method has been demonstrated to find correct but incomplete reachable sets. An iterative upwind method was shown to require special modifications to prevent over-estimated reachable sets. The performance of these methods is compared to the Level Set method on a variety of test systems and a simplified aircraft model. The behavior of the safe Fast Marching method was demonstrated on a simplified aircraft model for the slower dynamics. The method required fewer calculations despite an inefficient optimization scheme. The estimated arrival times differ from those calculated with the level set method. This is partially attributed to the low resolution of attempted control input samples. The results of the Eikonal equation demonstrate that the SFM method can outperform the Level Set method in terms of accuracy. The method is expected to perform better in anisotropic problems when a larger stencil size is used, as suggested for the ordered upwind method Sethian and Vladimirsky (2003). Since the stencil size can give considerable overhead in larger dimensions, it is not recommended to use this method to find solutions in highly anisotropic regions. Combined with the depth-first trajectories it may however serve as a method to quickly identify a first accurate region of the flight envelope that can be extended afterwards with another scheme. An overview on the applicability of the solving
schemes is given in table 6-2.

In a search for more localized solving schemes it was proven that a set of depth-first trajectories can be combined with the propagating front methods to quickly reach out to state-regions of immediate interest. This technique can be applied to both the Level Set method and the Minimal time method. Although this method may alter the computed arrival times when applied to the Level Set method it can be guaranteed to correctly identify the reachable states. This approach has been demonstrated with the safe Fast Marching scheme on a mass-spring-damper system where it was found to aid the solution finding of the safe ordered upwind method (OUM). During implementation, it was realized that this technique is better suited for the minimal time method as the notion of arrival time can be maintained in the initialization. Iterative minimal arrival schemes should be able to converge to the true minimal arrival time even when the selected trajectories are not optimal. The advantages of this technique are earlier verification of important regions, faster convergence and better handling of anisotropy for the safe OUMs.

Based on the experience gained in this study, the following recommendations can be made:

- Continue the exploration of Minimal arrival time calculation methods. In particular, investigate the fast iterative method (FIM) and fast sweeping method (FSM) as they should result in competitive computational efficiency. The observations made with the mass-spring-damper problem suggest that a modification may be required to prevent the occurrence of over-estimation when approaching the steady reachable set. The FSM may be improved by extending it to a narrow band method.

- Of all the single-pass methods only the safe upwind method is believed suitable for producing correct reachable sets in anisotropic problems. The SFM method may have applications for problems where an incomplete but otherwise accurate reachable set has to be quickly available.

- It is expected that the Level Set method, but in particular the boundary value scheme can result in significant computational savings when combing the methods with a depth-first method. Particular applications and implementation methods could be explored further.

- The idea to accept sub-optimal solutions by simplifying the optimization problem at the loss of accurate arrival times has not been assessed beyond the theoretical concept and remains open for further investigation. No indications were found that would restrict the applicability.

- The narrow band Level Set method is still a good competitor to many of the alternative methods explored in this study and should not be forsaken in further research.

- The Roe-Fix scheme may be extended to also handle non-smooth value surfaces and ambiguities that are independent of the flow field.

- The performed demonstrations have shown indications of excessive under-estimation of arrival times for all approximation schemes. Some investigation may be in order to obtain further insight in techniques to improve accuracy of the arrival time estimations.
The recommendations are summarized in table 6-1.

Finally it is stressed that in order to use any form of model-based moving front theory it is required that the model is accurate in all regions where the front is propagated. Although a more localized front propagation technique may make this requirement easier to achieve, it is stressed that the current adaptive model estimation techniques are only accurate near the states where measurements have been made. In order to safely use any of the proposed techniques, the model estimation method has to be able to predict the occurrence of bifurcation in unexplored states. Unless a suitable approximation technique is found, the techniques discussed in this study will be of limited use on on-line applications.

Table 6-1: Recommendations on considered simplification methods.
- - = not possible, - = not recommended, + = acceptable with loss of information, ++ = recommended

<table>
<thead>
<tr>
<th>Method</th>
<th>Moving front</th>
<th>Minimal time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Split to problems of lower dimension</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Convergence of estimated set</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Recursion on time-invariant system</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Localized front propagation</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>collaboration with trajectories</td>
<td>+</td>
<td>++</td>
</tr>
<tr>
<td>Acceptance of sub-optimal inputs</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Use of formulation</td>
<td>++</td>
<td>+</td>
</tr>
<tr>
<td>Use of classic Fast Marching method</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Use of Narrow band method</td>
<td>++</td>
<td>++</td>
</tr>
</tbody>
</table>

Table 6-2: Overview of solving schemes

<table>
<thead>
<tr>
<th></th>
<th>Complexity</th>
<th>CFL</th>
<th>Upwind</th>
<th>Iterative</th>
<th>Anisotropic</th>
<th>Speed</th>
<th>Suitable</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSM</td>
<td>$O(N^{n+1})$</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>NB-LSM</td>
<td>$O(kN^n)$</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>ITM</td>
<td>$O(N^{n+1})$</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td>No</td>
</tr>
<tr>
<td>FSM</td>
<td>$O(N^{n+1})$</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>$&lt; ITM$</td>
<td>Maybe</td>
</tr>
<tr>
<td>NB-FSM</td>
<td>$O(kN^n)$</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td>Maybe</td>
</tr>
<tr>
<td>E FMM</td>
<td>$O(N^n \ln N)$</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td></td>
<td>No</td>
</tr>
<tr>
<td>SL FMM</td>
<td>$O(N^n \ln N)$</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td></td>
<td>No</td>
</tr>
<tr>
<td>E OUM</td>
<td>$O(kN^n \ln N)$</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td></td>
<td>No</td>
</tr>
<tr>
<td>SL OUM</td>
<td>$O(kN^n \ln N)$</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td></td>
<td>No</td>
</tr>
<tr>
<td>s OUM</td>
<td>$O(kN^n \ln N)$</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>BF M</td>
<td>$O(kN^n \ln N)$</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>$&gt; ITM$</td>
<td>No</td>
</tr>
<tr>
<td>PFM</td>
<td>$O(kN^n \ln N)$</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>$&gt; ITM$</td>
<td>No</td>
</tr>
<tr>
<td>FIM</td>
<td>$O(kN^n \ln N)$</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>$\approx FSM$</td>
<td>Maybe</td>
</tr>
</tbody>
</table>
Appendix A

Draft paper
Efficient Methods for Flight Envelope Estimation through reachability analysis

J.C.J. Stapel*

This paper contains a study to find faster numerical methods for Hamilton-Jacobi Isaacs partial differential equations in application to model-based flight envelope estimation. The aim is to identify new methods capable of providing the basic information needed for flight envelope protection. Useful insights have been obtained though assessing the reachable set theory associated to the problem, which permit to integrate depth-first and breadth first estimation methods and to estimate the flight envelope as a minimal time problem. The applicability of a class of non-iterative schemes, known as the fast marching methods, has been evaluated. The behavior of the studied methods is demonstrated on different example problems, including a simplified aircraft model. A recommendation is given for continued research.

I. Introduction

Safety has always been of primary importance in aviation. Even today the flight control community continues to improve safety by extending their systems' capabilities beyond the occurrence of non-critical failures or unanticipated changes in the system dynamics. Good progress has been made through adaptive dynamic model estimation and stability argumentation. An open problem is however to maintain an accurate estimate of the flight envelope, which is an essential tool in the prevention of loss of control.

Loss of control (LOC) is considered a primary cause of fatalities in aviation.\(^1\) Between 2002 and 2012 almost 40% of all fatal accidents is related to LOC.\(^2\) In these accidents the most important causal factors were identified to be related to flight crew handling or mishandling after a technical failure.

The current practice is to estimate accurate flight envelopes beforehand by extensive flight tests and calculations. Although this method suffices under nominal conditions, it fails when the system dynamics change in the event of a failure. The unavailability of this protection system in impaired situations can be considered a significant weakness in the robust and adaptive methodology.

One promising model-based approach to resolve this problem is advocated in Tang et al.\(^3\) where the flight envelope is described as the intersection between two non-linear, non-convex reachability problems. These reachability problems can be described by the Hamilton Jacoby Bellman (HJB) partial differential equation (PDE) or by the Hamilton Jacobi Isaacs (HJI) equation when disturbances are to be incorporated in the analysis. These equations have been solved numerically for dynamic flight envelopes with the Level Set method.\(^5,^6\) Due to the high computation times, this method cannot yet be considered for realistic real-time applications. Several studies have been undertaken in a search for faster or better performing variants of this method.

Adalsteinsson and Sethian\(^7\) localize the computational domain to a narrow region around the front of the reachable set which reduces the number of calculations per iteration. Van Oort et al.\(^8\) use a semi-Lagrangian Level Set method to avoid the Courant-Friedrichs-Lewy (CFL) stability criteria which is otherwise limiting the permissible time step. De Weerdt and van Oort\(^9\) replace the grids with interval analysis. Li et al.\(^10\) adapt the partial differential equation to avoid the need for re-initialization, which may be necessary in some occasions. Govindaarajan et al.\(^11\) explore the opportunities of using simplex splines. Stipanovic and Tomlin\(^12\) make over-approximations of the reachable sets with a polytopic approximation through feedback linearization. Helsen\(^13\) constructs the reachable set with a depth-first approach where distance fields are

*Job Title, Department, Address, and AIAA Member Grade.
computed over the grid with globally optimal controlled trajectories. Good introductory reads on the propagating front theory are chapters 10-13 of van Oort\textsuperscript{14} and the books by Osher and Fedkiw\textsuperscript{15} and Sethian\textsuperscript{16}

This study is an attempt to explore new ways to reduce the computational complexity of the model-based approach. It will be assumed that the system is non-linear, but time-invariant after the failure event. It will be further assumed that the system model is correctly identified after the occurrence of the failure. It should be noted that this assumption is not entirely realistic since the current adaptive system estimation techniques only provide locally correct models, and may require some iterations to converge.

This paper proposes to consider stationary schemes for solving the HJI PDE and proves applicability to the search for flight envelopes. From a quality perspective the use of this approach is discouraged by Mitchell\textsuperscript{17} as the method is limited to first order sub-grid accuracy, can form discontinuities and lacks additional information outside the reachable set. Mitchell also cautions for disappointing computational advantages. The possibility for discontinuities is however intentional as the use of Huygens principle of first arrival avoids the augmented viscosity needed in the Level Set method to find the correct weak solution. Although additional information outside the reachable set can be useful, it is not essential for determining the flight envelope itself and may therefore be sacrificed for an increase in computational performance. Additional advantages of the boundary value formulation is that this method is not sensitive to the CFL stability criteria, which restricts the permissible step size in the Level Set method. The absence of global integration steps also makes the method more suitable for integration with depth-first methods or for localized set propagation, which may proof beneficial in real-time applications.

This paper will start with a brief overview of the reachable set theory. A formal proof will be given for the applicability of the boundary value formulation, along with some further insights on localized set propagation. The behavior of some of the solving schemes for the boundary value formulation will be investigated on a set of test problems. Finally a recommendation for further investigation will be formulated.

II. Description of the reachable set theory

The reachable sets can be defined as follows.\textsuperscript{8} Consider a dynamic system $\Sigma$ with dynamics $\dot{x} = f(x, u, d, t)$ where $x$ is a state in state space $\chi$, $t$ is the time variable and $[u, d]$ are control and disturbance input signals from the permissible bounded sets $[\mathcal{U}(x, t), \mathcal{D}(x, t)]$. A trajectory is denoted as $\xi_{x_0, t_0, u(.), d(.)}(t) : t \rightarrow x \in \chi$, where $x_0$ is the initial state at $t = t_0$. The initial set $\mathcal{I}$ and target set $\mathcal{T}$ are defined such that $\mathcal{I}, \mathcal{T} \in \chi$.

A backwards reachable set is the collection of states from which the target set can be reached at time $t_f$ when starting at time $t$ while a forwards reachable set can be reached at time $t$ when starting in the initial set at time $t_0$:

\begin{equation}
\begin{array}{l}
\quad \text{Backwards reachable set}: R_{\mathcal{T}, t_f}^B (t) := \{x \in \chi : \forall d \in \mathcal{D}, \exists u \in \mathcal{U}(\xi_{x, t, u, d}(t_f) \in \mathcal{T}) \}
\end{array}
\end{equation}

\begin{equation}
\begin{array}{l}
\quad \text{Forwards reachable set}: R_{\mathcal{T}, t_0}^F (t) := \{x \in \chi : \forall d \in \mathcal{D}, \exists x_0, u \in [\mathcal{I}, \mathcal{U}] | \xi_{x_0, t_0, a, b}(t) = x \}
\end{array}
\end{equation}

The dynamic flight envelope for system $\Sigma$ and $\mathcal{I} = \mathcal{T} = \mathcal{K}$ can now be represented as the intersection between the forward and backward reachable set: $R_{\mathcal{K}, t_0}^F (t) \cap R_{\mathcal{K}, t_f}^B (t)$. Apart from the reachable set, one may also observe the reachability tube, which is the union of all reachable sets over the observed time horizon. The reachability tube will be denoted as $\mathcal{R}_\mathcal{K}(\leq t) = \cup \mathcal{R}_\mathcal{K}(t)$. This study also uses the concept of a maintainable set which is the union of all states $\{x | \exists u, d \in [\mathcal{U}, \mathcal{D}] : x = 0\}$. Further set types and their properties have been identified by Mitchell\textsuperscript{18} and Kaynama et al.\textsuperscript{19}

II.A. Level set method

To find the reachable sets efficiently the optimally controlled trajectories have to be identified. In the most general case the problem involves both a controller as a disturber. This optimality problem forms a differential game and has been used by Mitchell\textsuperscript{17} and Lombokaerts.\textsuperscript{5} The differential game theory as well as the related differential equations are well described for instance Evans\textsuperscript{20} or Pierre.\textsuperscript{21}

For the Level Set method the reachable set is described by a pursuit-evader zero-sum differential game with an objective function $J : u, d \rightarrow \mathbb{R}$ of the form $J(u, d) = g(\bar{x}(t_f))$, where $g$ is chosen such that it satisfies
equation 2. The reachable sets can then be found as stated in theorem 1.

\[
g(x) = \begin{cases} 
  < c & \forall x \in K \\
  = c & \forall x \in \delta K \\
  > c & \forall x \in K^{-1}
\end{cases}
\]  

(2)

**Theorem 1.** Let $\Phi^+$ be the viscosity solution of the (terminal) upper Isaac’s equation and $\Phi^-$ the viscosity solution for the (initial) lower Isaac’s equation:

\[
\begin{align*}
I_B^+ \left\{ \Phi^+(t, x, \nabla_x \Phi^+) = 0 \right\} & \quad I_F^+ \left\{ \Phi^+(-t, z) = g(x) \right\} \\
I_B^- \left\{ \Phi^-(t, x, \nabla_x \Phi^-) = 0 \right\} & \quad I_F^- \left\{ \Phi^-(t, z) = g(x) \right\}
\end{align*}
\]

where $H_B^+(t, x, p) = \minmax_u \{ f(t, x, u, d) \cdot p \}$ and $H_F^-(t, x, p) = \maxmin_u \{ f(t, x, u, d) \cdot p \}$. Then $R_{F,F,t}^U(t) = \{ x : \Phi(x, t) \leq 0 \}$ and $R_{F,F,t}^L(t) = \{ x : \Phi(x, t) \leq 0 \}$.

The theory states that the reachable set can be found by tracing the intersection between the defined value function and a horizontal plane at elevation $c$. The set is initialized such that this intersection represents the boundary of the initial set. This concept is illustrated in figure 1.

![Figure 1: Illustration of the level set representation](image)

Together with some extensions, the Level Set method is a commonly used reachability method as it covers a very general class of problems. It can be solved numerically with the schemes by Osher.\textsuperscript{15} The reachable sets of stationary problems can also be described by a boundary value formulation.

II.B. Minimal time problem

In a SIAM paper\textsuperscript{22} Falcone and colleagues note that the burn equation, which is normally computed as a level set problem, can also be solved with the stationary boundary value formulation of the HJB PDE. The trick is that the reachability problem is not modeled as a pursuit-evasion differential game, but rather as a minimal time problem: $J(x, u, d) = \min\{ t : \xi_{x, u, d}(t) \in K \}$. When a state is never reached, the assigned value defaults to $+\infty$ (or $-\infty$ in case of backward reachability). This formulation directly forms the HJI PDE given in theorem 2.\textsuperscript{23,24,25}

**Theorem 2.** Let $T^+$ and $T^-$ be the viscosity solution of the upper and lower Isaac’s equation:

\[
\begin{align*}
I_B^+ \left\{ -H_B^+(x, \nabla_x T) = 1 \right\} & \quad I_F^+ \left\{ H_F^+(x, \nabla_x T) = 1 \right\} \\
T(x \in K) = t_f & \quad T(x \in K) = t_0
\end{align*}
\]

where $H_B^+(x, p) = \minmax_u \{ f(x, u, d) \cdot p \}$ and $H_F^+(x, p) = \maxmin_u \{ f(x, u, d) \cdot p \}$. Then $R_{F,F,t}^U(\geq t) = \{ x : T(x) \geq t \}$ and $R_{F,F,t}^L(\leq t) = \{ x : T(x) \leq t \}$.
Note that the minus sign in the backward reachability problem disappears when a virtual time scale \( \tau = -t \) is used. Since the method only tracks minimal arrival time, the boundary value formulation can only represent problems in which either \( H < 0 \forall x, \tau \) or \( H > 0 \forall x, \tau \) holds.

The minimal time problem comes with a larger assortment of numerical schemes; each with its own advantages and limitations. A good review is given in.\(^{26}\) The simplest scheme is referred to as the iterative method (ITM). It is generally applicable and the order of complexity is comparable to that of the Level Set method. Assuming \( N \) nodes in each dimension and a comparable number of iterations the computational complexity is of the order \( O(N^{n+1}) \). The fast sweeping method (FSM)\(^{27}\) is a variant that updates the nodes in a directional manner. This modification improves the rate of convergence by exploiting the characteristic direction in at least one of the 'sweeps' over the grid. Apart from this advantage, the methods have comparable complexity and applicability.

The Fast Marching method (FMM)\(^{15}\) avoids the necessity for iteration completely by tracking the causality in which nodes will be reached. The complexity is \( O(N^n \log N^n) \). The original method is only suitable for Eikonal problems, but has been extended to the ordered upwind method (OUM)\(^{28,27}\) which uses larger stencils to better cope with anisotropy, but also increases the overhead. An alternative extension is the iterative Fast Marching method (IFM).\(^{26,29}\) The order of complexity of the IFM lies between that of the ITM and the FMM. The safe Fast Marching (SFM)\(^{26}\) only progresses the front if the used Ordered upwind method is capable of finding a valid approximation.

II.C. Proofs for properties and usability

The following theorems prove that the flight envelope can be formulated as a stationary reachability problem. They also provide further insights that allow to progress the front locally, that permit to combine depth-first methods with the breadth-first front propagation and that show how to simplify the optimization problem.

**Proposition 1.** \( \text{Consider a reachability problem for a time-invariant dynamic system } \dot{x} = f(x,u,d). \text{ Let } K \text{ be a maintainable set, then } R_K(t_1) = R_K(\leq t_1). \) \( \text{(The set reachable at a specific time equals the reachable tube with the same horizon).} \)

**Proof.** Consider the case where a state is reached for the first time on \( t = t_1 \). It is then also possible to reach this state at an arbitrary moment \( t_2 \geq t_1 \) by staying in the initial maintainable state for a period of \( \Delta t = t_2 - t_1 \) after which the same trajectory is followed. This trajectory is permissible because of the shifting property.

**Corollary 1.1.** \( \text{Consider the same reachability problem as in proposition 1. For } \tau \geq 0 \text{ we can state that } R_K(t_1) \subseteq R_K(t_1 + \tau) \) \( \text{(The reachable set can only grow monotonically).} \)

**Proof.** This property follows logically from Proposition 1, since every reached state can still be reached every moment thereafter.

**Proposition 2.** \( \text{For a reachability problem where the front represents a cluster of permissible trajectories, consider two initial sets } K_1 \text{ and } K_2 \subseteq K_1. \text{ Then their respective reachable sets satisfy } R_{K_1}(t) \subseteq R_{K_2}(t) \)

**Proof.** Since \( R(t) = \bigcup_{t_o: t_o \leq (t)} (K_1 \setminus K_2) \), one can simply split the problem \( K_1 = K_2 \cup (K_1 \setminus K_2) \), solve \( R_{K_2}(t) \) and \( R_{K_1 \setminus K_2}(t) \). Now \( R_{K_1}(t) = R_{K_2}(t) \cup R_{K_1 \setminus K_2}(t) \) \( \supseteq R_{K_1}(t) \)

**Proposition 3.** \( \text{Consider a subset } V \text{ of the reachable set } R_K(t_1). \text{ Then } R_V(\tau) \subseteq R_K(t_1 + \tau) \)

**Proof.** Since \( V \subset R_K(t_1) \), \( \exists \exists_{x_o \in K_1(t_1)} x \in V \forall x \in V \). Extending these trajectories over a period of \( \tau \) in accordance with the dynamic programming principle gives a new set of states equal to \( R_V(\tau) \).

**Lemma 1.** \( \text{Consider a stationary reachability problem with a maintainable initial set. Then on a bounded domain of finite dimensions there exists a time horizon } t_f \text{ such that } R_K(t_f) = R_K(t_2) \text{ for all } t_2 > t_f. \text{ this set is referred to as the steady set and } t_f \text{ is defined as the infinite horizon.} \)

**Proof.** Since the set can only grow (corollary 1.1), the trivial case of reaching the full domain proofs existence of this horizon.

American Institute of Aeronautics and Astronautics
Proposition 4 (inclusion theorem). Consider a solution set $\mathcal{R}_{K_1}(t_1)$ found by solving the time invariant system which is initialized from a maintainable initial set $K_1$ over a time horizon $t_1$. Consider an arbitrary valid trajectory (or set of trajectories) $\xi$ such that $\xi(t = 0) \in K_1$ and $\xi \subseteq \mathcal{R}_{K_1}(t_1)$. Then there exists a $t_2 \leq t_1$ and $t_2 \geq t_1$ such that $\mathcal{R}_{K_1}(t_2) \subseteq \mathcal{R}_{\mathcal{R}_{K_1}(t_1) \cup \xi}(t_2) \subseteq \mathcal{R}_{K_1}(t_3)$

Proof. The proof consists of two parts. First it can be shown that $\mathcal{R}_{K_1}(t_1) \subseteq \mathcal{R}_{K_2}(t_2)$. This property follows from lemma 2 since $K_1 \subseteq K_2$.

To proof that $\mathcal{R}_{K_2}(t_2) \subseteq \mathcal{R}_{K_1}(t_3)$, consider reaching set $\mathcal{R}_{K_1}(t)$ by first reaching $K_2$ from $K_1$ over a time horizon $\Delta t$ by restricting the set of permissible inputs such that only $\xi$ can be made. Then the set is grown normally till time $T + \Delta t$. Since a smaller set of permissible inputs is used one can say that $\mathcal{R}_{K_1}(T + \Delta t) \subseteq \mathcal{R}_{K_1}(T + \Delta t)$.

Since by definition, $\mathcal{R}_{K_1}(\Delta t) = K_2$ through the principle of dynamic programming it can also be stated that $\mathcal{R}_{K_1}(T + \Delta t) = \mathcal{R}_{\mathcal{R}_{K_2}(\Delta t)}(T) = \mathcal{R}_{K_1}(T)$ and hence arrive at $\mathcal{R}_{K_2}(T) \subseteq \mathcal{R}_{K_1}(T + \Delta t)$.

Corollary 4.1. The two problems of proposition 4 with initial sets $K_1$ and $K_2$ have the same steady set, but may have different infinite horizons.

Remark. These proofs only hold when the reachable sets represent clusters of trajectories. When the shape of the set has a physical meaning, like in the burn equations where temperature is depending on radius of curvature of the front, these proofs do not hold.

II.D. Proof consequences

Prioritize regions of interest

In general a reachability calculation has to be initialized in the safe initial region and is propagated without any means to influence the progression of the front towards a region of interest. For real-time implementations, this requirement entails that it may take considerable time before the reachability of a state far away from the initial set is verified. Proposition 2 and the inclusion theorem 4 provide the means to avoid this delay.

They show that it is possible to correctly identify reachable states from a larger initial set, provided that this set is contained in the reachable set of the original problem. The inclusion theorem also indicates that this set will find the reachable states sooner than the initial set alone. This also means that the steady set from proposition 1 can be found earlier. The only difference with the original solution is a change in the estimated arrival times.

It is therefore possible to combine depth-first trajectory optimization with the breadth-first dynamic programming principles of the moving front. A set of trajectories can be used to initialize the moving front in the vicinity of states of interest; for instance the regions near the aircraft’s current state or close to where the limits are expected. The level set method can then start expanding the envelope directly in these regions. Even when there are no states of particular interest, a set of depth-first trajectories can be used to increase the area of the starting front, which also may lead to faster conversion of the sets. Although this trick may severely distort the original reachability tube, it will correctly identify whether a state is reachable or not. This method is therefore recommended for further development.

Use of boundary value formulation

Corollary 1.1 proves that the monotonicity requirement can be satisfied by initializing the reachability problem with a trimmable initial set. The problem can therefore be fully described by the minimal time problem. The applicability of various solving schemes will be evaluated later in this paper.

Localizing the problem

Corollary 3 proves that it is possible to find valid set regions when propagating only a subset of the set’s front. This property gives potential application for localized front propagation. An attempt can be made to spend more computational resources to propagate the front in the vicinity of the current aircraft state and along the intended state transition. Unfortunately there is no guarantee that this approach efficiently identifies reachable states. It may however work in locally controllable regions. Since aircraft are typically
not locally controllable, this approach is not recommended for this application. It should further be noted that any form of local propagation will be very hard to accomplish with the level set method, as the algorithm has to iterate with globally equal time increments.

Reducing the reachability problem

The second proof of the inclusion theorem 4 shows that sub-optimal control can be used to find the reachable set faster. This insight shows that it is not required to reach all states in an optimal way. Instead of investing computational efforts in finding the optimal input, it suffices to identify any control that reaches the state. Once a state is reached, the principle of dynamic programming dictates that from there on the next state can be found irrespective of how the previous state was reached.

This approach reduces the optimal reachability problem to a sufficient reachability problem. The sub-optimality permits non-unique solutions on all time horizons. Still, since the actual arrival time is of a lesser interest than the quick availability of a correct flight envelope, this inconsistency may be acceptable in on-line envelope estimation.

It is expected that this observation can make an improvement in the selection criteria for the control inputs, and in case of the fast marching method may even circumvent the need to causally order the narrow band nodes, reducing the number of calculations by a factor log(N^o) and also improving the degree of parallelism.

III. Implementation of the Level Set schemes

For the purpose of verifying and demonstrating the behavior and properties of the Level Set method, a set of numerical schemes was implemented in MatLab. Although Mitchell has already made a convenient toolbox available, it was decided to develop an independent tool based on Osher and Fedkew. It was found that this approach aided the understanding of the reachability problem on the numerical level and it has contributed to some of the insights mentioned in this paper.

The implemented algorithms can solve two-dimensional problems with three different dissipation methods, second order Runge-Kutta integration and several options for handling differential games, augmenting maintainability and solving different reachability problems. Each implementation has been subjected to a variety of small tests to verify the operation and cooperation of the contributing functions. The algorithms were also compared to Mitchell's toolbox on a double integrator problem. Apart from a minor difference in the handling of the domain boundaries, the two implementations were found to produce the same results.

IV. Implementation of the minimal time schemes

The schemes needed for the Fast Marching methods have also been implemented. No satisfying external toolbox was found to be available. Two sets of algorithms have been developed. The first is capable of solving the classical Eulerian Fast Marching method for n-dimensional problems and has an integrated solving scheme for multiple input- multiple output (MIMO), input affine differential games. The second set is limited to two-dimensional problems but uses the semi-Lagrangian approach.

In section 1.2 of the IIJ PDE for the boundary value formulation is derived for the backward reachable set using the dynamic programming principle. This paper also states that the viscosity solution of this PDE results in valid minimal travel time.

IV.A. Euler method

The Eulerian scheme has been combined with an optimization scheme for non-linear, input affine MIMO problems. The scheme will be using first order upwind finite differencing. The assumed structure of the IIJ PDE is as follows:

\[
\min_{u} \max_{d} \left( \begin{bmatrix} f_1 \\ f_2 \\ \vdots \end{bmatrix} + \begin{bmatrix} g_{11} & g_{12} & \cdots \\ g_{21} & g_{22} & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \end{bmatrix} + \begin{bmatrix} g_{31} & g_{32} & \cdots \\ g_{41} & g_{42} & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \\ \vdots \end{bmatrix} \right) \cdot \begin{bmatrix} \nabla_x^+ T_{ij} \\ \nabla_y^+ T_{ij} \\ \vdots \end{bmatrix} = 1
\]  

(5)
Where $\nabla_{x_i}^{\pm} T_{i,j...}$ is either the lower ($\nabla_{x_i}^{-} = \frac{(T_{i,j-1} - T_{i,j})}{\Delta x_i}$) or upper ($\nabla_{x_i}^{+} = -\frac{(T_{i+1,j} - T_{i,j})}{\Delta x_i}$) spatial difference. In this implicit function the value $T_{i,j...}$ will be the unknown and either the left or right neighboring value will be available. Assuming that the optimal inputs $u^*, d^*$ have been found, the PDE reduces to $f_{i,j}(u^*, d^*) \cdot \nabla_{x} T_{i,j} = 1$ and it becomes straightforward to express $T_{i,j}$ explicitly:

$$T_{i,j} = \frac{1 + \sum_{i^-} f_i(u^*, d^*) \frac{T_{i,j+1}^-}{\Delta x_i} - \sum_{i^+} f_i(u^*, d^*) \frac{T_{i,j+1}^+}{\Delta x_i}}{\sum_{i^-} \frac{f_i(u^*, d^*)}{\Delta x_i} - \sum_{i^+} \frac{f_i(u^*, d^*)}{\Delta x_i}}$$

(6)

Where $i^-$ are the dimensions in which $\nabla_{x_i}^{-}$ is used, $i^+$ the dimensions with $\nabla_{x_i}^{+}$, and in which $T_{i-1}$ and $T_{i+1}$ are the left and right neighbor in dimension $i$ respectively.

### IV.B. Semi-Lagrangian method

The semi-Lagrangian method is illustrated in figure 2. The theory below has been put together by using mainly the works of Cacacce et al.\textsuperscript{26,29} The idea is to select a grid point (for instance $T_{i,j}$). From this starting position a trajectory can be calculated backwards in time until a certain distance criteria is met. On this end position, which is denoted as $\tilde{x}_{-\tau}$, the value $T_{\tilde{x}}$ is estimated by interpolation between the neighboring nodes (for instance $T_{i+1,j}$ and $T_{i,j-1}$). The new value at the starting node than is this interpolated value, plus the time $\tau$ spend for the trajectory to reach $x_{ij}$ from $\tilde{x}_{-\tau}$.

![Figure 2: Semi-Lagrangian method](image)

The semi-Lagrangian (SL) scheme can be seen as a discrete model. (See chapter 1.5 of\textsuperscript{24}) For the minimal time problem the first order discretized PDE is given in equation 7, where $h$ represents the time $-\tau$ necessary to reach the line or distance where the grid values are interpolated. Also the Kružkov transformed PDE is given with $\beta = e^{-h}$, which transforms $v = 1 - e^{-T}$ to bound the time domain to $t \in [0, 1]$:

$$T_h(x) = \inf_a \sup_b \{T_h(x + hf(x, a, b)) - h \}
\begin{equation}
(7)
\end{equation}$$

$$v_h(x) = \inf_a \sup_b \left\{ \frac{1}{\beta} v_h(x + hf(x, a, b)) + 1 - \frac{1}{\beta} \right\}$$

These expressions can be implemented on a sampled grid in several ways. Here the two-point stencil form figure 2 will be used as well as a 3x3 multi-stencil scheme. The two-point stencil is implemented for the normal and Kružkov transformed case. The multi-stencil has been built for the safe fast marching scheme. It adapts its neighbor usage depending on the availability of converged nodes and the direction of the velocity vector. In this case, the stencil is limited to the eight neighboring nodes. Without loss of generality, the discussion can limited to the three neighboring nodes of one quadrant. Using the notation given in figure 3, five situations can be distinguished:
1. No neighbors are available
2. Neighbor n₁, n₂ or n₃ is available
3. Neighbors {n₁, n₂} are available
4. Neighbors {n₁, n₃} (or {n₂, n₃}) are available
5. Neighbors {n₁, n₂, n₃} are available

Figure 3: Neighbor notation in considered quadrant

In the first situation no neighbors exist with an accepted value. For the considered velocity it is not possible to calculate a candidate arrival time. Therefore the candidate value is set to $+\infty$. In the second situation only one of the neighbors is available. The safe fast marching scheme dictates that no interpolation is possible. The scheme can therefore only find a value if the negative velocity vector points exactly towards this neighbor. In all other cases, the candidate value is set to $+\infty$. The third situation has two neighbors and therefore allows to interpolate. This interpolation however is not advisable, since the third node is not accepted. This situation can occur when a node is approached by two fronts, in which case the interpolation leads to unnecessary dissipation. Therefore a value may only be calculated if the vector points to one of the accepted nodes. In the fourth case it is possible to calculate a value if the negative velocity vector points somewhere between the two accepted nodes. In all other cases the value is set to $+\infty$. In the final situation all the required nodes are available to use the two- or three-point stencil. In this situation a finite value is guaranteed to exist.

Figure 4: Geometry for semi-Lagrangian multi stencil

All cases but the first are illustrated in the quadrants of figure 4, where green nodes have converged values and red nodes are unknown.

IV.C. applicability of the FM schemes

Both the Eulerian and the semi-Lagrangian Fast Marching method were found not to be applicable to anisotropic problems with uncontrollable regions. Even when an incorrect arrival time is acceptable, the Fast marching method will not be able to separate reachable states from unreachable states. The update schemes assume that all upwind neighboring nodes have a converged value. In case of an anisotropic flow this assumption is not necessarily the case. For the classical Fast Marching methods there are two methods to resolve situations where the neighboring nodes are not available.

Sethian\textsuperscript{16} suggests for the Eulerian method to ignore the dimensions in which no converged neighbor is available. The consequence is illustrated in figure 5. By reducing the dimensionality of the problem, it becomes possible to incorrectly conclude that a node is reachable. If the projected problem is extended back into the original size, the missing dimensions are extrapolated as flat, infinite extensions. It therefore ignores the finiteness of the set that would otherwise prevent a node from being reached.

For the semi-Lagrangian method Falcon\textsuperscript{17} suggests to use the Kružkov transform. This transformation permits to interpolate between converged and unknown states. This interpolation will fail in unreachable regions as the interpolation will always result in a finite arrival time. The same argument holds when a finite
time ceiling is used instead of a Kružkov transform.

The SFM method handles the absence of converged nodes by not calculating a solution when the required information is not available. This abstinence avoids the risk of calculating a finite value for an unreachable state but also limits the capabilities of identifying the reachable states. Consider the two situations visualized in figure 6. The evaluated node is not reached according to the safe semi-Lagrangian fast marching method, while in reality a trajectory can be found that connects the reachable set to the node. In the left case, the flow vector points directly to the front of the set, but does not reach it within the stencil space around the considered node. In the right case, the linearized trajectory misses the front altogether, while the continuous (and in this example, also the piecewise linearized) trajectory reaches the front. The ability to identify reachable nodes improves with the used limited by the stencil size.

Figure 6: Two situations where the considered node is incorrectly ignored by the safe SL FM scheme

Despite these limitations, the safe semi-Lagrangian Fast Marching method is well capable of calculating an accurate reachable set, as long as these exceptions do not occur. In practice this result means that for some problems this single pass method can be used to quickly find part of the reachable set, after which the solution can be refined with a more robust but iterative method.

V. demonstrations and evaluations

The derived properties are demonstrated on three example cases. A simple Eikonal equation is solved with the Level Set method as well as the Fast Marching and safe Fast Marching method in comparison to the analytic solution. A mass spring damper system is then used to show the behavior of these methods on an anisotropic problem with uncontrollable regions. Finally the safe Fast Marching method is applied to a simplified aircraft model.

Uniformly expanding front

The uniformly expanding front problem is used to compare the performance of the Level Set and ordered upwind methods. It is an Eikonal problem and is therefore well suited for the Fast Marching schemes. The dynamics of this system are given below.
\[
\dot{x} = f(x) = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \frac{1}{\sqrt{x_1^2 + x_2^2}} \frac{\vee \Phi}{|\vee \Phi|}
\]

(8)

Assuming that the initial set is a point in the origin of the domain, the analytic solution to this problem is \( T = \sqrt{x_1^2 + x_2^2} \). The dynamics in equation 8 are also provided as an external function of state. In this study the Fast Marching methods have not been implemented to accept gradient-dependent problems, as this phenomenon is not occurring in reachability problems of trajectory based dynamic systems. The Level set method has been implemented to use either expression.

The Level Set method is solved using a Local Lax-Friedrichs scheme as well as a Roe-Fix scheme for dissipation and are initialized with \( \Phi = \sqrt{x_1^2 + x_2^2} - 0.08 \). The results are shown in figure 7. The calculations have been performed on a \( N = [49, 49] \) grid and a domain of \( x_1, x_2 = [-2, 2] \).

Figure 7a shows the analytic solution of the Eikonal equation. The remaining figures show the difference between the analytic and the numerical solutions. Figure 7b shows the difference pattern for the Eulerian Fast Marching method. No error is present on the principle axis, but the Fast Marching method estimates later arrival times elsewhere. Figure 7c shows the difference pattern for the safe Fast Marching method. In contrast to the Eulerian method, the error is about four times smaller in magnitude. Also the diagonals coincide with the exact solution. This improvement in accuracy is because the SFM method uses the larger multi-stencil which also incorporates the diagonal axis in the calculation.

Figure 7d shows the difference surface of the Level Set method with local Lax-Friedrichs scheme using the externally defined dynamics and is therefore comparable with the results of the Fast Marching methods. Although using the same stencil as the Eulerian Fast Marching method, the explicit Level Set method is less accurate in this Eikonal problem. Figure 7e shows the results for the same scheme using the implicitly defined dynamics. Comparing this surface to figure 7d gives an indication on the difference between evolving the set with pre-calculated and internal dynamics. The latter case uses different inputs in response to the approximation error. Figure 7f shows the difference for the Roe-Fix scheme. The results for the internally
defined dynamics are a bit better when compared to the local Lax-Friedrichs scheme due to the reduced dissipation. The Roe-Fix scheme was found to induce oscillations on the externally defined dynamics. The Roe-Fix scheme assumes that the derivatives of the Hamiltonian are a good indicator for discontinuities. Since the velocity vectors of the externally defined flow field are not affected by the gradient, this method fails as an indicator for induced oscillations. It is therefore suggested to add a second fix for problems that do not have the gradient in the derivative of the Hamiltonian. A good candidate may be to observe sign differences in the upper and lower derivatives themselves. The overall performance data is given in table 1. e and i stand for external and internal dynamics.

\[
L_\infty \quad \text{RMS}
\]

<table>
<thead>
<tr>
<th>Method</th>
<th>L_\infty</th>
<th>RMS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analyt - E FSM</td>
<td>9.24e^{-02}</td>
<td>5.35e^{-02}</td>
</tr>
<tr>
<td>Analyt - s FSM</td>
<td>2.23e^{-02}</td>
<td>1.22e^{-02}</td>
</tr>
<tr>
<td>Analyt - e LS LLF</td>
<td>3.93e^{-01}</td>
<td>2.53e^{-01}</td>
</tr>
<tr>
<td>Analyt - i LS LLF</td>
<td>5.42e^{-01}</td>
<td>3.82e^{-01}</td>
</tr>
<tr>
<td>Analyt - e LS RF</td>
<td>3.93e^{-01}</td>
<td>2.49e^{-01}</td>
</tr>
<tr>
<td>Analyt - i LS RF</td>
<td>4.20e^{-01}</td>
<td>2.86e^{-01}</td>
</tr>
</tbody>
</table>

**Table 1: Difference measures for the forward Eikonal problem; N = [49, 49] and dx = 1/12**

*Figure 8: Schematic of the mass-spring-damper system with force excitation*

**Mass-spring-Damper system**

The mass-spring-damper system forms a good test for the minimal time methods as the damping term provides unreachable regions inside the state-space. The characteristics lie parallel to the reachable front on boundary of this region. The system is illustrated in figure 8 and the equation of motion is given below:

\[
\begin{bmatrix}
\dot{S} \\
\dot{S}
\end{bmatrix} = \begin{bmatrix}
0 & 1 \\
-k/M & -c/M
\end{bmatrix}
\begin{bmatrix}
S \\
1/M
\end{bmatrix} + \begin{bmatrix}
0 \\
1/M
\end{bmatrix} F
\]

(9)

The problem is set up with a specific spring and damping constant of 0.2 and a permissible specific forcing function of $[-0.15, 0.15]$. The calculations are performed on a 121x121 grid on domain $x_1, x_2 = [-3, 3]$. The initial level set is described by $g_0(S, S) = |S| + |S| - 0.2$.

The results for the minimal time problem are shown in figure 9. Two optimally exited trajectories are included to indicate the boundary of the reachability tube. Figure 9a shows the arrival time as calculated by the Eulerian Fast Marching method. The dimensions without converged nodes are ignored in the update scheme, which causes the solution to underestimate arrival times and to "leak" into the unreachable region.

Figure 9b shows the semi-Lagrangian fast marching method where extrapolation between converged and not converged values uses the Kružkov transform. For $T = 35.5$ the increments are no longer visible to the floating-point resolution after the Kružkov transformation and the remaining state-space is reached with that arrival time. The solution produces conservative values until $T \approx 18$, after which the arrival times are underestimated. The over-approximation is expected to be caused by the interpolation between not-converged nodes.

Figure 9c shows the arrival time surface when the Kružkov transform is replaced by a finite time ceiling. The nodes in the 'far' region are initialized with a value of $T_{max} = 60$. Since the values are not transformed the interpolation is linear. The resulting solution is very conservative in its arrival times. The maximum arrival time also provides a stopping criteria as the arrival time cannot increase.

Figure 9d shows the result for an iterative scheme, where the semi-Lagrangian method is repeated 50 times. For each iteration, the nodes in the far region are given the values that were found in the previous iteration and thereby reduce the interpolation penalty. The value function now follows the two trajectories more closely. The arrival times are conservative everywhere except when approaching $T_{max}$. Here the interpolation still allows unreachable nodes to receive a value smaller than $T_{max}$. One attempt to overcome
this problem is to use a threshold value \( T_i < T_{\text{max}} \). At the end of each iteration, all values larger than this threshold are set to the maximum value: \( T(x) = T_{\text{max}}, \forall x | T(x) \geq T_i \). Provided that the difference between \( T_i \) and \( T_{\text{max}} \) cannot be overcome in one iteration for unreachable nodes, this threshold prevents overestimation of the reachable set. When \( T_i \) is too small it can also prevent reachable states from being assigned a value. The main challenge is therefore to find an appropriate threshold. Figure 9c shows the result for \( T_i = 0.95T_{\text{max}} \). The reachable set is now conservative everywhere while still approximating the two trajectories. It is suggested that these issues can also arise in similar iterative schemes like the fast sweeping method.

Figure 9f shows the results from the semi-Lagrangian safe Fast Marching method with a multi-stencil. Since the method is not allowed to use not-converged nodes, it stops when there are no candidate nodes inside the observed stencil. For the 3x3 stencil used here, the calculated set is relatively small. The nodes that are found however closely follow the trajectories.

Figure 9g demonstrates the proposed idea of combining the depth-first and breadth-first methods by including a trajectory in the initial set of the safe Fast Marching scheme. During implementation it was found that the initial values of the nodes visited by the trajectory can be set to the arrival times of this trajectory. An interesting note is that since a permissible trajectory is always slower than or as fast as an optimal controlled system an iterative minimal time method should be able to find the correct arrival time. The single-pass method used here will result in conservative arrival times when the used trajectories are not optimal. This conservation of arrival time knowledge is not possible for the Level Set method where the front of the set must have the same value. One trajectory is used over a horizon of \( t = [0, 40] \) and projected on the grid. The added trajectory greatly improves the region on which the method can solve the problem.

### VI. Simplified aircraft model

To demonstrate the performance of the schemes on a more aerospace related system, the aircraft model used in\(^3\) has been implemented. It consists of a simplified model of the slower symmetric motions. The state space is described by the velocity \( V \) and the flight path angle \( \gamma \). The relevant axis and vectors are defined in figure 10. In addition, the influence of the side-slip angle \( \beta \) and roll angle \( \phi \) is also included in this model. The considered equations of motion are given here:

\[
\begin{bmatrix}
\dot{V} \\
\dot{\gamma}
\end{bmatrix} = \begin{bmatrix}
-\frac{c_5}{2m} V^2 C_{D_0} - g \sin \gamma + \cos \alpha \cos \beta \frac{T}{m} - \frac{c_5}{2m} V^2 \left( C_{D_\alpha} \alpha + C_{D_\beta} \beta^2 \right) \\
-\frac{c_7}{2m} \cos \gamma + \left( \cos \phi \sin \alpha \cos \beta - \sin \phi \cos \beta \right) \frac{T}{m} + \frac{c_7}{2m} V \left( C_{L_\alpha} + C_{L_\beta} \alpha \right) \cos \phi - \frac{c_7}{2m} V C_{Y_\beta} \beta \sin \phi
\end{bmatrix}
\]  

(10)

This model can be simplified by assuming small angles:

\[
\begin{bmatrix}
\dot{V} \\
\dot{\gamma}
\end{bmatrix} = \begin{bmatrix}
-\frac{c_5}{2m} V^2 C_{D_0} - g \sin \gamma + \frac{T}{m} - \frac{c_5}{2m} V^2 \left( C_{D_\alpha} \alpha + C_{D_\beta} \beta^2 \right) \\
-\frac{c_7}{2m} \cos \gamma + \frac{c_7}{2m} V \left( C_{L_\alpha} + C_{L_\beta} \alpha \right) \cos \phi - \frac{c_7}{2m} V C_{Y_\beta} \beta \sin \phi
\end{bmatrix}
\]  

(11)

The control parameters are \( T \in [20546N, 410920N] \), and \( \alpha \in [0^\circ, 14.5^\circ] \). The following aerodynamic derivatives will be used: \( C_{D_0} = 0.1599 \), \( C_{D_\alpha} = 0.5035 \), \( C_{D_{12}} = 2.1175 \), \( C_{L_0} = 1.0656 \), \( C_{L_{12}} = 0.60723 \) and \( C_{Y_\beta} = -1.6 \). The remaining terms are: \( m = 120.10^4 \text{ kg} \), \( \rho = 1.225 \text{ kg/m}^3 \), \( S = 260 \text{ m}^2 \) and \( g = 9.81 \text{ m/s}^2 \).

For the Level Set method it is possible to find the optimal inputs. The control logic including model uncertainty is worked out in\(^3\) and is repeated here. Define the critical value \( \bar{\alpha} = \frac{c_7 V C_{L_\beta}}{2m} \) and mid-range angle \( \bar{\alpha} = \frac{c_7\alpha_{\text{max}}}{2m} V C_{L_\beta} \). Then to minimize the Hamiltonian, one has to choose the optimal inputs in accordance to the following rules:
Figure 9: Results for problem 2 as estimated using the Fast Marching methods
Control logic:

- If $p_1 > 0$ then $T^* = T_{\text{min}}$ and
  - if $\dot{\alpha} \geq \alpha$ then $\alpha^* = \alpha_{\text{min}}$
  - if $\dot{\alpha} < \alpha$ then $\alpha^* = \alpha_{\text{max}}$
- If $p_1 = 0$ then $T^* = T_{\text{min}}$ and
  - if $p_2 \geq 0$ then $\alpha^* = \alpha_{\text{min}}$
  - if $p_2 < 0$ then $\alpha^* = \alpha_{\text{max}}$
- If $p_1 < 0$ then $T^* = T_{\text{max}}$ and
  - if $\dot{\alpha} \leq \alpha_{\text{min}}$ then $\alpha^* = \alpha_{\text{min}}$
  - if $\dot{\alpha} \leq \alpha_{\text{max}}$ then $\alpha^* = \alpha_{\text{max}}$
  - if $\alpha_{\text{min}} \leq \dot{\alpha} \leq \alpha_{\text{max}}$ then $\alpha^* = \dot{\alpha}$
- If $p_2 \sin \phi \geq 0$ then $\beta^* = \beta_{\text{min}}$
- If $p_2 \sin \phi < 0$ then $\beta^* = \beta_{\text{max}}$

Disturbance logic:

- If $p_1 \geq 0$ then $C_{D_0}^* = C_{D_0, \text{min}}$, $C_{D_a}^* = C_{D_a, \text{min}}$, $C_{D_{a2}}^* = C_{D_{a2, \text{min}}}$
- If $p_1 \leq 0$ then $C_{D_0}^* = C_{D_0, \text{max}}$, $C_{D_a}^* = C_{D_a, \text{max}}$, $C_{D_{a2}}^* = C_{D_{a2, \text{max}}}$
- If $p_2 \geq 0$ then $C_{L_0}^* = C_{L_0, \text{max}}$, $C_{L_a}^* = C_{L_a, \text{max}}$
- If $p_2 \leq 0$ then $C_{L_0}^* = C_{L_0, \text{min}}$, $C_{L_a}^* = C_{L_a, \text{min}}$
- $C_{Y_p}^* = C_{Y_{\beta, \text{max}}}$

Figure 10: FBD for the aircraft model

Here $p_1, p_2$ are the components of the value-function’s gradient. The simple condition for $C_{Y_p}^*$ follows from the fact that $\beta$ will always be selected to satisfy $p_2 \sin \phi \beta < 0$.

Determining the optimal inputs for the ordered upwind methods is a bit more complicated as no gradient is readily available. For the purpose of demonstration, a brute-force optimisation will be implemented. The controller can select from ten samples for $\alpha$ and choose between the upper and lower limits for the remaining controls. This approach results in a total of 40 different combinations of control inputs. 160 combinations are needed when model-uncertainty is considered.

To make a better comparison, Mitchell's Level Set toolbox uses first order spatial differences and a first order runge-Kutta integration scheme. The implementation has been verified by re-producing some of the results from\textsuperscript{2} and\textsuperscript{12} with the Level Set scheme. All calculations have been made on a uniform 150x150 grid with the dimensions as shown in the figures.

Figure 12 shows the forward and backward reachable sets for zero roll on ten arrival times to a horizon of five seconds. The difference between the safe Fast Marching method and the Level Set method is shown in figures 12c and 12f.

The safe Fast Marching method stops prematurely in regions where the gradient of the set and the characteristic curves are not sufficiently aligned, leaving distinctive edges. In contrast to the Eikonal equations the safe Fast Marching method arrives a little later in most regions, except on the paths visited by the corners of the initial set and on the upper right edge of the forward reachable set, where it arrives earlier. It is suspected that the dissipation of the Level Set method has a significant impact on how these regions are extended.

The backward reachable set has two regions where the safe fast marching method arrives considerably later than the Level Set method. There are also two small regions where the Fast Marching method is a little faster. Especially the right region experiences a relative increase in arrival times. There are a couple of possible explanations for this increase. First of all the anisotropy prevents the safe fast marching method from progressing the front in the same manner as the Level Set method. A given node may only be solved once it is approached from a different direction at a later iteration or is only reachable form the current front when using a control input that is sub-optimal compared to what the Level set method can use. The safe Fast marching method also has to bridge the space between grid nodes in a single iteration while the Level Set method may use several steps. A final reason for different arrival times in general is the fact that the Level Set method uses an initial value function whose overall shape influences the progression of the front.
through stability criteria, dissipation and numerical derivatives.

The data in table 2 has been collected to compare the computational performance. Even with more efficient optimization the Level Set method requires 226 times as many node evaluations for the forward reachable set and 158 times as many for the backward reachable set. The difference for the number of model evaluations is less impressive in this example due to the inefficient optimization method used in the Fast Marching method, where 40 input combinations are tested for optimization. The Level set method requires only 9 model evaluations to figure out the correct upwind and dissipation terms. An additional advantage of the fast marching method is that no computations are performed on nodes outside the reachable set. The computation time however is considerably larger for the fast marching method. There are a few reasons for this anomaly. First of all, the Level Set method has been implemented to make maximum use of the BLAS libraries. The safe Fast Marching method is less suitable for parallelization and has to do with the more involved for-loop. Since MatLab is not pre-compiling, there is a lot more overhead involved with this method. A second reason is that the Semi-Lagrangian method requires a considerable amount of geometric calculations. These could however be simplified. Apart from the computational performance, one can also observe the quality of the result. Table 2 also shows the maximum and average difference between the two methods. The percentage of missed nodes of either method is measured against the union set of reached nodes. The surplus of nodes reached in the Level Set method is caused by the inability of the safe Fast Marching method to capture the regions with high anisotropy. The few nodes that are uniquely identified by the safe Fast Marching method are most likely not found by the Level Set method due to the shrinking of the set caused by dissipation.

Table 2: Operational performance of the Level Set method and safe Fast Marching method on the aircraft model

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Forward LS</th>
<th>Backward LS</th>
<th>Forward sFM</th>
<th>Backward sFM</th>
</tr>
</thead>
<tbody>
<tr>
<td># iterations</td>
<td>364</td>
<td>400</td>
<td>3.84</td>
<td>3.89</td>
</tr>
<tr>
<td># model evaluations/node update</td>
<td>3276</td>
<td>3600</td>
<td>1443640</td>
<td>2269000</td>
</tr>
<tr>
<td># nodes/evaluation</td>
<td>9</td>
<td>9</td>
<td>40</td>
<td>40</td>
</tr>
<tr>
<td>tot # node evaluations</td>
<td>8190000</td>
<td>6000000</td>
<td>36091</td>
<td>56725</td>
</tr>
<tr>
<td>tot # model evaluations</td>
<td>73710000</td>
<td>81000000</td>
<td>1443640</td>
<td>2269000</td>
</tr>
<tr>
<td>nodes evaluated</td>
<td>22500(100%)</td>
<td>22500(100%)</td>
<td>8940(39.7%)</td>
<td>14582(64.81%)</td>
</tr>
<tr>
<td>CPU time* [s]</td>
<td>15.397</td>
<td>17.191</td>
<td>102.976</td>
<td>164.020</td>
</tr>
</tbody>
</table>

As an additional verification step, a Monte Carlo analysis has been made on the reachable sets. 2000 random sample states are generated. Only the states for which an estimated arrival time larger than one second exists are considered. For each, a single, stochastically optimal trajectory is generated. Instead of producing a large number of global trajectories, the principle of dynamic programming is applied by first generating a set of candidate trajectory increments end then selecting the one that results in the best performance. On each time increment of 0.02 seconds, the dynamic system is solved for a set of 51 random permissible inputs. The candidate end state of this increment with the smallest value on the arrival time surface as estimated by the Level Set or Fast Marching method is accepted. The corresponding set of inputs is included in the set of trial inputs for the next iteration.

Figure 11 shows the resulting trajectories and initial states. The states for which the trajectory success-

---

*Computed on an EliteBook 8570w Mobile Workstation running Windows®7 Home Premium and MatLab R2013a. Logged with MatLab’s Profile toolbox

American Institute of Aeronautics and Astronautics
fully reaches the initial set within 110% of the estimated arrival time are colored green while the remaining initial states are red. A small number of trajectories escape from the region in which the estimated arrival time surface is properly defined and can no longer use it to optimize the control inputs. These trajectories should closely resemble the time-optimal paths. Figure 11a has a set of unverified nodes on the left edge of the reachable set. This concentration of unverified nodes could indicate an overestimate. It is expected that the Level Set method has incorrectly extended the lower-left corner of the trimmable set due to dissipation. Alternatively, the considered states can be particularly unforgiving for using sub-optimal inputs. The trajectories found from the unverified states pass close to the bottom-left corner. A truly optimal trajectory might have been able to reach this corner.

Table 3 gives an overview of these trajectories. The reach-rate indicates the fraction of initial states for which the trim set was reached successfully. The Success rate gives the fraction of these trajectories that reach the trim set within the time estimated by the arrival time surface. The remaining values give information on the arrival times obtained by the trajectories in comparison to the estimated arrival time. Interestingly to note is that on average the successful trajectories of the Monte Carlo simulation predict approximately 60% faster arrival times than the Level Set and safe Fast Marching method. Although only first-order approximations were used in the estimation process and although no design effort was put in optimizing the estimation accuracy, these results give a strong indication that the estimation techniques tend to be very conservative. It is therefore recommended to obtain further insight in techniques to improve accuracy of the arrival time estimations.

![Trajectories Diagrams](image1.png)

(a) backward LS  
(b) forward LS  
(c) backward sFM  
(d) forward sFM

Figure 11: Monte Carlo simulation with optimal trajectories as generated from random control inputs evaluated on the estimated arrival time surface.

Figure 13 shows the full reachable sets and flight envelopes for a perfect model as well as a model with 20% uncertainty. The sets are calculated with both the Level Set method and the Fast Marching method. The green surfaces in the center of each set represents the trimmable states for zero side-slip. The trimmed set requires higher velocities for larger roll angles, as can be expected. In the uncertain case the trim set as well as the reachable set become considerably smaller. The safe Fast Marching method is mostly conservative, also for the larger values of $\Phi$. 

16 of 22

American Institute of Aeronautics and Astronautics
Table 3: Statistical results from the Monte Carlo simulation with randomly generated optimal inputs.

<table>
<thead>
<tr>
<th>problem</th>
<th>Reach rate</th>
<th>Success rate</th>
<th>Minimum T</th>
<th>Maximum T</th>
<th>mean T</th>
<th>std</th>
</tr>
</thead>
<tbody>
<tr>
<td>LS forward</td>
<td>79.86%</td>
<td>96.93%</td>
<td>20.40%</td>
<td>108.00%</td>
<td>60.13%</td>
<td>21.69%</td>
</tr>
<tr>
<td>LS backward</td>
<td>98.18%</td>
<td>93.98%</td>
<td>19.20%</td>
<td>109.60%</td>
<td>60.68%</td>
<td>23.01%</td>
</tr>
<tr>
<td>sFM forward</td>
<td>99.05%</td>
<td>98.57%</td>
<td>17.60%</td>
<td>105.60%</td>
<td>56.64%</td>
<td>21.89%</td>
</tr>
<tr>
<td>sFM backward</td>
<td>99.80%</td>
<td>96.30%</td>
<td>14.80%</td>
<td>108.00%</td>
<td>57.51%</td>
<td>22.71%</td>
</tr>
</tbody>
</table>

Figure 12: Arrival time iso-lines and difference surface for the forwards and backward reachable sets
Figure 13: Full nominal and uncertain reachable sets and envelope for arrival time of 5 s (red) and trimmed initial sets (green)
VII. Conclusion & recommendation

A simple thought experiment has shown that it is not possible to guarantee convergence of an arbitrary initial set. Similarly, it has been shown that a set found by recursively propagating a reachable set over a time-variant event cannot be used for the backward reachable set due to causality and does not produce a reachable set suitable for flight envelope estimation.

It was argued that permitting sub-optimal solutions by simplifying the optimization problem will result in more conservative minimal arrival times but should not affect the accuracy of the steady set. This simplification could save considerable computation time in the optimization aspect of the problem solving, which is of particular interest when considering more complex control problems. The simplification may be extended to the update schemes as well. Since no application of this idea has been tested, it is still open for further investigation.

It was further proven that under the assumption of a time-invariant system and a maintainable initial set the reachable tube can be found from the minimal arrival time. This time can be found by the stationary boundary formulation. The schemes for this method are not subjected to the CFL stability criteria and do not require augmented dissipation in order to find a weak solution.

A variety of Fast Marching methods have been tested for applicability on a set of simple test problems. Although it was hypothesized that the classical Fast Marching method may be able to correctly identify the steady reachable set it was found that this is not the case. A physical explanation for this inability was found for both the Eulerian and the semi-Lagrangian approach. The safe Fast marching method has been demonstrated to find a correct but incomplete reachable sets. An iterative upwind method was shown to require special modifications to prevent over-estimated reachable sets. The performance of these methods is compared to the Level Set method on a variety of test systems and a simplified aircraft model. The behavior of the safe Fast Marching method was demonstrated on a simplified aircraft model for the slow aircraft dynamics. The method required fewer calculations despite of an inefficient optimization scheme. The estimated arrival times differ from those calculated with the level set method. This difference is partially attributed to the low resolution of attempted control input samples. The results of the Eikonal equation demonstrate that the SPM method can outperform the Level Set method in terms of accuracy. The method is expected to perform better in anisotropic problems when a larger stencil size is used, as suggested for the ordered upwind method. Since the stencil size can give considerable overhead in larger dimensions it is not recommended to use this method to find solutions in highly anisotropic regions. Combined with a depth-first trajectories it may however serve as a method to quickly identify a first accurate region of the flight envelope that can be extended afterwards with another scheme. An overview on the applicability of the solving schemes is given in table 6.

In a search for more localized solving schemes, it was proven that a set of depth-first trajectories can be combined with the propagating front methods to quickly reach out to state-regions of immediate interest. This technique can be applied to both the Level Set method and the Minimal time method. Although this method may alter the computed arrival times when applied to the Level Set method it can be guaranteed to correctly identify the reachable states. This approach has been demonstrated with the safe Fast Marching scheme on a mass-spring-damper system where it was found to aid the solution finding of the safe OUM. During implementation it was realized that this technique is better suited for the minimal time method as the notion of arrival time can be maintained in the initialization. Iterative minimal arrival schemes should able to converge to the true minimal arrival time even when the selected trajectories are not optimal. The advantages of this technique are earlier verification of important regions, faster convergence and better handling of anisotropy for the safe OUMs.

Based on the experience gained in this study, the following recommendations can be made:

- Continue the exploration of Minimal arrival time calculation methods. In particular, investigate the fast iterative method (FIM) and FSM as they should result in competitive computational efficiency. The observations made with the mass-spring-damper problem suggest that a modification may be required to prevent the occurrence of over-estimation when approaching the steady reachable set. The
FSM may be improved by extending it to a narrow band method.

- Of all the single-pass methods only the safe upwind method is believed suitable for producing correct reachable sets in anisotropic problems. The SFM method may have applications for problems where an incomplete but otherwise accurate reachable set has to be quickly available.

- It is expected that the Level Set method, but in particular the boundary value scheme can result in significant computational savings when combining the methods with a depth-first method. Particular applications and implementation methods could be explored further.

- The idea to accept sub-optimal solutions by simplifying the optimization problem at the loss of accurate arrival times has not been assessed beyond the theoretical concept and remains open for further investigation. No indications were found that would restrict the applicability.

- The narrow band Level Set method is still a good competitor to many of the alternative methods explored in this study and should not be forsaken in further research.

- The Roe-Fix scheme may be extended to also handle non-smooth value surfaces and ambiguities that are independent of the flow field.

- The performed demonstrations have shown indications of excessive under-estimation of arrival times for all approximation schemes. Some investigation may be in order to obtain further insight in techniques to improve accuracy of the arrival time estimations.

The recommendations are summarized in Table 4.

Finally it is stressed that in order to use any form of model-based moving front theory it is required that the model is accurate in all regions where the front is propagated. Although a more localized front propagation technique may make this requirement easier to achieve, it is stressed that the current adaptive model estimation techniques are only accurate near the states where measurements have been made. In order to safely use any of the proposed techniques, the model estimation method has to be able to predict the occurrence of bifurcation in unexplored states. Unless a suitable approximation technique is found, the techniques discussed in this study will be of limited use on on-line applications.

Table 4: Recommendations on considered simplification methods.

- - = not possible, - = not recommended, + = acceptable with loss of information, ++ = recommended

<table>
<thead>
<tr>
<th>Method</th>
<th>Moving front</th>
<th>Minimal time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Split to problems of lower dimension</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Convergence of estimated set</td>
<td>- -</td>
<td>- -</td>
</tr>
<tr>
<td>Recursion on time-invariant system</td>
<td>- -</td>
<td>- -</td>
</tr>
<tr>
<td>Localized front propagation</td>
<td>- -</td>
<td>-</td>
</tr>
<tr>
<td>Collaboration with trajectories</td>
<td>+</td>
<td>++</td>
</tr>
<tr>
<td>Acceptance of sub-optimal inputs</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Use of formulation</td>
<td>++</td>
<td>+</td>
</tr>
<tr>
<td>Use of classic Fast Marching method</td>
<td>++</td>
<td>-</td>
</tr>
<tr>
<td>Use of Narrow band method</td>
<td>++</td>
<td>++</td>
</tr>
</tbody>
</table>

References


<table>
<thead>
<tr>
<th>Scheme</th>
<th>Complexity</th>
<th>CFL</th>
<th>Upwind</th>
<th>Iterative</th>
<th>Anisotropic</th>
<th>Speed</th>
<th>Suitable</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSM</td>
<td>$O(N^{n+1})$</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>NB-LSM</td>
<td>$O(kN^n)$</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>ITM</td>
<td>$O(N^{n+1})$</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>FSM</td>
<td>$O(N^{n+1})$</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>$&lt; ITM$</td>
<td>Maybe</td>
<td>Maybe</td>
</tr>
<tr>
<td>NB-FSM</td>
<td>$O(kN^n)$</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Maybe</td>
<td>Maybe</td>
</tr>
<tr>
<td>E FMM</td>
<td>$O(N^n \ln N)$</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>SL FMM</td>
<td>$O(N^n \ln N)$</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>E OUM</td>
<td>$O(kN^n \ln N)$</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>SL OUM</td>
<td>$O(kN^n \ln N)$</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>s OUM</td>
<td>$O(kN^n \ln N)$</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>BFM</td>
<td>$O(kN^n \ln N)$</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>$&gt; ITM$</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>PFM</td>
<td>$O(kN^n \ln N)$</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>$&gt; ITM$</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>FIM</td>
<td>$O(kN^n \ln N)$</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>$\approx FSM$</td>
<td>Maybe</td>
<td>Maybe</td>
</tr>
</tbody>
</table>


