KTH Engineering Sciences

Online Trajectory Planning and Observer Based Control

DAVID A. ANISI

Licentiate Thesis
Stockholm, Sweden 2006

Optimization and Systems Theory
Department of Mathematics
Royal Institute of Technology
SE-100 44 Stockholm, SWEDEN

TRITA MAT 06/OS/04
ISSN 1401-2294
ISBN 91-7178-469-1

Academic thesis, which with the approval of Royal Institute of Technology (KTH), will be presented for public review and in partial fulfillment of the requirements for a degree of Licentiate of Engineering in Applied Mathematics. The public review will be held on November 10, 2006 at 10.00 in Room 3721, Lindstedtsvägen 25, KTH, Stockholm, Sweden.

Print: Universitetsservice US AB

Every venture one's life may replete
Mathnavi's purpose is the Great Defeat.
Set afire, burning with cleansing heat,
On the anvil, egos ply and beat.
This book, if you open, read, entreat
Your life, a mendicant's, in the street.

- Rumi

To my beloved family

Abstract

The main body of this thesis consists of four appended papers. The first two consider different aspects of the trajectory planning problem, while the last two deal with observer design for mobile robotic and Euler-Lagrange systems respectively.

The first paper addresses the problem of designing a real time, high performance trajectory planner for aerial vehicles. The main contribution is two-fold. Firstly, by augmenting a novel safety maneuver at the end of the planned trajectory, this paper extends previous results by having provable safety properties in a 3D setting. Secondly, assuming initial feasibility, the planning method is shown to have finite time task completion. Moreover, in the second part of the paper, the problem of simultaneous arrival of multiple aerial vehicles is considered. By using a time-scale separation principle, one is able to adopt standard Laplacian control to this consensus problem, which is neither unconstrained, nor first order.

Direct methods for trajectory optimization are traditionally based on *a priori* temporal discretization and collocation methods. In the second paper, the problem of adaptive node distribution is formulated as a constrained optimization problem, which is to be included in the underlying nonlinear mathematical programming problem. The benefits of utilizing the suggested method for online trajectory optimization are illustrated by a missile guidance example.

In the third paper, the problem of active observer design for an important class of non-uniformly observable systems, namely mobile robotics systems, is considered. The set of feasible configurations and the set of output flow equivalent states are defined. It is shown that the inter-relation between these two sets may serve as the basis for design of active observers. The proposed observer design methodology is illustrated by considering a unicycle robot model, equipped with a set of range-measuring sensors.

Finally, in the fourth paper, a geometrically intrinsic observer for Euler-Lagrange systems is defined and analyzed. This observer is a generalization of the observer recently proposed by Aghannan and Rouchon. Their contractivity result is reproduced and complemented by a proof that the region of contraction is infinitely thin. However, assuming *a priori* bounds on the velocities, convergence of the observer is shown by means of Lyapunov's direct method in the case of configuration manifolds with constant curvature.

Keywords: Computational Optimal Control, Receding Horizon Control, Mission Uncertainty, Safety, Task Completion, Consensus Problem, Simultaneous Arrival, Adaptive Grid Methods, Missile Guidance, Nonlinear Observer Design, Active Observers, Non–uniformly Observable Systems, Mobile Robotic Systems, Intrinsic Observers, Differential Geometric Methods, Euler-Lagrange Systems, Contraction Analysis.

Acknowledgments

I would like to start by acknowledging my advisor, Professor Xiaoming Hu, for all his guidance, support and his open door policy. My views on scientific research have broadened considerably during our developing discussions and I certainly look forward to the continuation of my PhD studies.

The research that culminated in this thesis, has been conducted in cooperation with the Department of Autonomous Systems at the Swedish Defense Research Agency (FOI). The financial support is gratefully acknowledged. Of the people at FOI, I especially would like to thank my former Master's thesis supervisor, Johan Hamberg for being a great source of inspiration and knowledge. As my scientific mentor, I thank him for always believing in my ability to learn and truly hope that the future will allow us to continue our collaboration and discussions. My gratitude extends to my two other co-authors from FOI, John Robinson and Petter Ögren; not only for guidance in the direction of research, but also for providing and maintaining a friendly and creative research atmosphere. This thesis would surely not have looked the same without their involvement and commitment. This is also a great opportunity to recognize the fundamental role that Martin Hagström, the former head of the Department of Autonomous Systems at FOI, has played. This research project would not have been initialized without his personal influence. I am indebted for this initiative.

Needless to say, all faculty members and fellow PhD students at the Division of Optimization and Systems Theory have contributed to the completion of this thesis by providing an excellent working environment. In particular, I thank Associate Professor Ulf Jönsson wholeheartedly; for helping me out from day one (by suggesting interesting master's thesis opportunities at Georgia Tech.) until today, by generously reading through this thesis. I would further like to show my appreciation to Mats Werme, Nazbanoo Sedghi, Johan Karlsson and Anders Eriksson for their careful reading and constructive criticisms of parts of this thesis.

I am deeply thankful to my beloved mother who brought me up with endless love, perfect care and great expectations. I would also like to extend my deepest thanks to Darya (a.k.a. "bache meymoon") for her constant support and countless sacrifices. I am fortunate to have a sister like you. Another family member to acknowledge is my dear uncle; for being a distinguished source of admiration and respect. Capable to move a mountain, you gratify some people and astonish the rest.

Finally, I would like to thank Ghazal (a.k.a. "pishi") for her genuine support, mature understanding and amazing patience through the years. Thanks for reminding me that, as far as humans are concerned, best answers are to be found at the intersection of logic, intuition and volition. To me, you are that special breath of fresh air.

Stockholm, November 2006

David Alireza Anisi

Table of Contents

Introduction

Introduction

This thesis deals with two important problems in the field of control, namely trajectory planning and observer design. Here–below, both these terms, as well as their inter–relations, are explained in an introductory manner. Consequently, the more familiar reader may prefer to skip it at a first reading. The two subsequent sections, provide a more detailed treatment of these problems.

Trajectory planning is an instance of the *motion* planning problem, which is used as a collective term for both trajectory- and path planning. In its most basic form, motion planning is about finding a feasible path (or trajectory) connecting two given configurations, denoted p_i and p_f, for a single robot present in a static and known environment. A path is the image of a continuous function $\gamma : [0, 1] \to \mathcal{C}$, such that $\gamma(0) = p_i$ and $\gamma(1) = p_f$, which means that it connects the two given configurations. Here, $\mathcal{C}$ denotes the robot's so called configuration space [1]. A path should also fulfill the robot's configuration-level constraints, for instance, physical obstacles in the environment. A trajectory can then be defined by specifying the time evolution of a path [1]. Motion planning as described here, is a purely *geometric* problem that ignores the inherent dynamic limitations of the robot. Some authors therefore prefer to define a trajectory as designing time dependent configuration– *and* velocity functions that are consistent with the robot's dynamic constraints. This definition is adopted in this thesis.

There are at least two ways to decompose the motion planning problem. One alternative is to find a feasible trajectory through *refinement* [1, 2] (see Figure 1). Given the intermediate level objectives and constraints (*cf.* Figure 2), the path planner first provides a geometrically feasible path. It is then the task of the trajectory planner to convert this path to a trajectory that is consistent with the robot's dynamic constraints. This decomposition is however not necessary and methods that merge the two modules exist. The framework adopted in this thesis, namely that of Computational Optimal Control (see Section 1.1), serves as an example of this alternative approach that disregards the path planning part. This procedure can be interpreted in Figure 1 by overlooking the dashed path planner module.

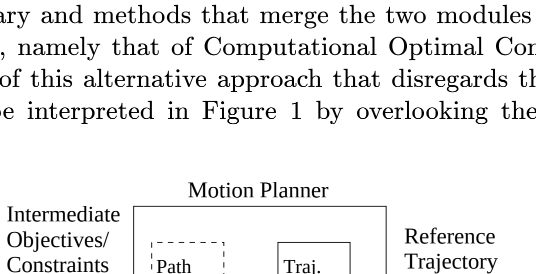


Figure 1: Possible decompositions of the motion planning problem.

Once we have a feasible reference trajectory at hand (denoted x_{ref} in Figure 2), we would like to use feedback control to make the robot follow the prescribed trajectory. Feedback control design techniques require information about at least some parts of the state vector. If all the state variables necessary for the control design can not be directly measured, which is a typical situation in complex systems, attention must be directed towards *estimating* them. This is achieved by designing an *observer*, whose task is to reconstruct missing state information while only using available measurements, y. Observers are defined more rigorously in Section 2.1.

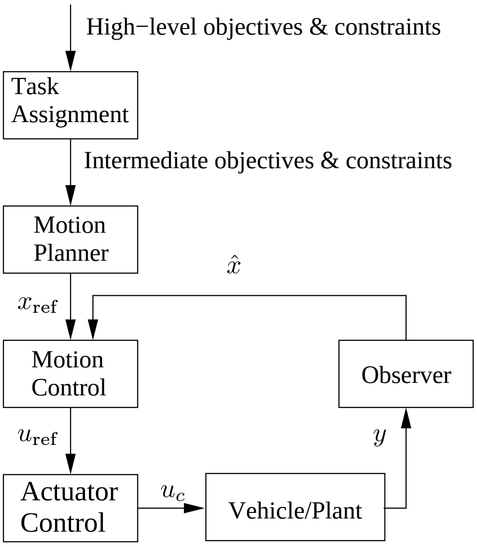


Figure 2: Possible structure of a modular control system.

Trajectory planning and observer design are both vital sub-problems in the creation of autonomous robots. They are however not the only problems that have to be addressed. Other important aspects include high–level task assignment (*i.e.* making strategic global plans), controller design, actuator design and sensor fusion. A possible, and rather classic, structure for decomposing the overall system architecture, has been depicted in Figure 2. Truly optimal design of autonomous robotic systems requires concurrent solution of all these sub-problems. Such an approach is however beyond our current understanding of these issues. As a natural step then, we limit ourselves to a modular and sub-optimal design scheme, where only parts of the interactions between the modules are counted for.

Kinodynamic Trajectory Planning

Trajectory planning arises as a natural and vital sub-problem of the noble ambition to design an autonomous system. Once implemented, it lifts the question of vehicle control to a higher level, where the input descriptions will specify the nature of the task to be carried out, rather than *how* to do it. In its most basic form, trajectory planning is about finding a feasible trajectory connecting two given states, for a single, fully actuated point, present in a static and known environment. Important extensions include the case of stochastic planning, temporal constraints, multi-vehicle planning, and trajectory planning from a given initial point to a terminal set. In this thesis, we are mainly interested in kinodynamic trajectory planning. The term kinodynamic planning was introduced by Canny *et al.* [3, 4] and refers to motion planning problems that have both kinematic (holonomic/nonholonomic), as well as dynamical constraints.

Another realistic issue, which is at the center of discussion in Papers A and B, involves imposing computational constraints on the trajectory planner. This requirement may originate from an assumption on having an imperfect world-model, *i.e.* severe information uncertainty with respect to the current objectives and constraints. In such a setting, the trajectory has to be *re-planned* in a fast and safe manner as time evolves, so that planning and execution phases can be interweaved. This is a non–trivial task, since it is known (see *e.g.* [3] and chapter 6.1 in [1]) that the solution time for the basic planning problem depends exponentially on the vehicle's degrees of freedom. This difficulty is reinforced by the fact that every extension of the basic planning problem, such as multiple robots and moving obstacles, adds new degrees of freedom to the problem. Therefore, in order to meet the online computational requirement, attention must be given to approximate solutions that are of low computational complexity.

The exposition of the planning approaches in this section is neither complete nor self-contained. The reader is referred to [1, 2, 5] for fuller discussions. The objective of this section is rather to explain a number of important issues for the main approach used in this thesis, namely computational optimal control. This is the subject of Section 1.1.

1.1 Computational Optimal Control

The paradigm of qualitative control design, which is associating a measure of the "utility" of a certain control action, has been a foundation of system engineering thinking. Optimal control is therefore regarded as one of the more appealing methodologies for trajectory planning. However, as captivating as the underlying theory might be, real-world applications have so far been scarce, particularly due to the high computational demand for solving nonlinear Optimal Control Problems (OCP). Consequently, attention has been paid to *approximation methods* and computationally efficient algorithms that compute solutions which are "near-optimal" in some sense. In this section, we treat a number of important issues for algorithmic solution of optimal control problems. Also, some major classes of computational methods are emphasized. A more comprehensive survey of computational methods for solving OCPs, as they appeared in the mid 90's, can be found in Chapter 2 of [6].

The discussion will be focused around the following optimal control problem:

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

where the state $x(t) \in \mathcal{X}$, the control $u(t) \in \mathcal{U}$, the constraints $d : \mathcal{X} \times \mathcal{U} \to \mathbb{R}^q$ and the terminal time, T, is a possibly *free* variable. Also, $\mathcal{X}$ and $\mathcal{U}$ are smooth manifolds of dimension n and p respectively. To further unburden the discussion, we make a standing assumption that all mappings are assumed to be sufficiently smooth, the state trajectory is uniquely defined and stays feasible at all time instances, that all stated minimization problems with respect to u are well–posed and that the minimum is attained.

Unless the objective function J, the system dynamics and the constraints are extremely simple, finding optimal controls *analytically* is in most cases prohibitive. Assuming that the considered OCP originates from a complex, real-world application, the existence of analytical solutions is thus deemed unlikely. Our objective is then to solve the OCP *numerically.*

For the actual design of the computational algorithm, the *infinite dimensional* problem of choosing the control function in a given space, has to be turned into a *finite dimensional* optimal parameter selection problem. This process of representing the continuous time functions by a finite number of parameters, is referred to as *transcription* and is typically achieved by either finite difference methods or finite sum of known basis functions [6, 7][1].

It is further conceptually important to differentiate between *direct* and *indirect* transcription methods (see Figure 1.1). These two categories will be dealt with in Section 1.1.1 and 1.1.2 respectively.

$$\begin{tikzcd}
\text{OCP} \arrow[r, shift left, "\text{direct transcription}"] \arrow[d, swap, "\begin{array}{r}\text{Hamiltonian}\\ \text{Adjoints}\\ \text{PMP/DP}\end{array}"] & \text{OCP}^{\text{N}} \arrow[l, shift left, "\text{consistency}"] \arrow[d, "\begin{array}{l}\text{Lagrangian}\\ \text{Lagrange multipliers}\\ \text{KKT}\end{array}"] \\
\text{OCP}^{\lambda} \arrow[r, shift left, "\text{indirect trancription}"] & \text{OCP}^{\text{N}\lambda} \arrow[l, shift left, "\text{consistency}"]
\end{tikzcd}$$

Figure 1.1: Direct and indirect transcription methods.

1.1.1 Indirect Transcription Methods

For a given OCP, indirect methods start off by introducing the Hamiltonian

$$\mathcal{H}(x,u,\lambda) = \lambda^T f(x,u) - \mathcal{L}(x,u),$$

[1] Certain choices for basis functions, blur the distinction between the two mentioned transcription methods (see *e.g.* [8, 9]).

where λ denotes the adjoint variables, and formulating the optimality conditions either according to the Pontryagin Maximum Principle (PMP) [10] or Dynamic Programming (DP) [11]. The PMP leads to an associated two point boundary value problem (TPBVP), while DP gives rise to the Hamilton-Jacobi-Bellman (HJB) partial differential equation. These infinite dimensional problems are denoted OCP^{λ} in Figure 1.1. Indirect transcription methods then proceed by approximating and numerically solving the TPBVP/HJB. Possible approaches for doing so include (see *e.g.* [12])

- (Multiple) shooting method
- Finite difference method
- Collocation method
- Galerkin method

In general, indirect methods are considered to produce more accurate results [13, 14]. In essence, direct methods which circumvent the PMP and DP by transcribing the OCP directly and using nonlinear programming techniques, have no way of fully utilizing the special structure of OCPs. Nevertheless, indirect methods are not typically used to solve problems having complex dynamics or constraint set. Nor are they suitable for problems where the underlying OCP is changeable in terms of the objective function, J, the final manifold, S_f and/or the constraint set, $d(x, u)$. This is most often due to:

- The ill-conditioned properties of the TPBVP[2]
- The occasionally tedious derivation of the optimality conditions via PMP/DP[3]
- The analytic intractability of solving the derived (nonlinear) optimality conditions
- The analytic intractability of solving the HJB partial differential equation
- The "curse of dimensionality", *i.e.* the inherent exponentially increasing computational complexity for solving HJB as the problem size increases [11]

Bearing in mind the assumption on information uncertainty made in this thesis, the principal interest will therefore be on *direct* transcription methods.

1.1.2 Direct Transcription Methods

The essential idea behind direct transcription methods is to use a finite number of basis functions to approximate the control manifold $\mathcal{U}$ and/or the state manifold $\mathcal{X}$. Which space, or spaces to parameterize and what basis functions to adopt, are some of the pivotal differences between existing direct transcription methods. Let us therefore discuss these two issues in greater detail.

[2] The associated TPBV can in fact be *singular*, in which case numerical solutions must be disregarded. This issue has been demonstrated in [15] by applying the PMP to a most simple OCP, namely the so called Dubins' problem [16].

[3] Symbolic mathematical packages have facilitated this procedure to a certain degree. Nevertheless, such luxuries cannot be enjoyed in a large class of interesting real-world applications, where look up tables dominate data presentation.

Space Parameterization

Regarding the choice of suitable space(s) to parameterize, there is no unambiguous and clear-cut answer. In the literature, parameterization of state [17], control [18, 19], and both the state and control [20–22], has been suggested. As a consequence of the regularity assumption on the control system, it follows that every control trajectory yields a unique state trajectory. In principle, it is therefore sufficient to only parameterize the control space, $\mathcal{U}$. Since the size of the transcribed optimization problem (denoted OCP^N in Figure 1.1) is proportional to the number of approximation parameters, only parameterizing $\mathcal{U}$ results in a comparatively small optimization problem, OCP^N. This approach is vindicated by the fact that optimization routines typically converge faster and more reliable on smaller problems. Nevertheless, it turns out that from a computational and implementation point of view, it might still be preferable to introduce parameterization variables in both $\mathcal{U}$ and $\mathcal{X}$. This is since in typical control applications, parts of the objective function J, as well as some of the constraints of the OCP, are state dependent. Examples include threat minimization objectives and various kinematic constraints, such as obstacle avoidance. In such cases, only parameterizing the control, leads to implicit constraint and gradient expressions in the transcribed optimization problem, OCP^N, which in turn may result in severely increased computational complexity. Finally, if only the state space, $\mathcal{X}$, is parameterized, the parameters have to be constrained as to be achievable by some feasible control. This method thus results in a differential inclusion formulation and is practically applicable for the limited class of problems where the following mapping from a point $x \in \mathcal{X}$ to a subset of the tangent space, $T_x\mathcal{X}$

$$S(x) = \{f(x,u) \in T_x\mathcal{X} : u \in \mathcal{U}, d(x,u) \leq 0\},$$

can be easily characterized or approximated. This set is termed holograph in [23]. Note that in order to solely involve the state parameters in the differential inclusion formulation, one must be able to also eliminate the control from the objective function J, as well. Interesting contributions, regarding the computational efficiency of the differential inclusion method, versus that of parameterizing both $\mathcal{U}$ and $\mathcal{X}$, can be found in [24–26].

To summarize this discussion, it is the author's belief that the choice of proper space(s) for parameterization should be made based on the particular OCP at hand. The leitmotiv should be to keep possible convexities of the objective function, and/or constraint set *intact*, even after the parameterization. Since, "the great watershed in optimization isn't between linearity and nonlinearity, but convexity and nonconvexity" [27], transforming these will most likely result in increased computational complexity (*cf.* [24, 28]). The statement that optimization routines converge faster and more reliable on smaller problems, holds true if two problems with similar structure and complexity (in terms of nonconvexity), but different number of variables are compared. The importance of problem structure versus that of number of variables is however an open and case specific question.

A parallel discussion, is that of using the flatness properties of a system for trajectory planning. Parameterizing the flat outputs – in addition to becoming a static problem – generally reduces the number of variables in OCP^N, but as noted in *e.g.* [29, 30], this parameterization might transform the objective function and constraints in a possibly complex (nonconvex) form, and consequently have negative influence on the convergence properties and/or the solution times[4].

[4] In [29], the evidence of the hypothesis that the solution time is an exponential decreasing function of the relative degree of the transformed system, are given by *numerical* experiments on a specific problem and can therefore not be considered as a firm affirmative proof.

Basis Functions

From the theory of approximation (see *e.g.* [31, 32]), by choosing the real parameters α_i, $i = 1, \cdots, n$ appropriately and n large enough, the finite sum

$$f_n(t) = \sum_{i=1}^{n} \alpha_i \phi_i(t),$$

can be made to approximate a well-behaved function, $f(t)$, to any degree of accuracy. Here, $\phi_i(t), i = 1, \cdots, n$ are known *basis functions* that span the approximating space. Different choices of basis functions, manage to approximate functions with different degrees of smoothness. As an example, consider the basis functions $\phi_i(t) = t^i$, *i.e.* approximating $f(t)$ in the space of n-th order polynomials. If $f(t)$ is a continuous function, uniform convergence of $f_n(t)$ to $f(t)$ follows by Weierstrass theorem, which states that the space of polynomials is dense in the space of continuous functions [33].

In principle, any preferred basis functions can be employed. The reader is referred to the introductory part of Ma's thesis [6], for a concise summary of different applicable approximation schemes. In practice however, piecewise polynomials [21], in particular cubic spline functions [20], belong to the classical choices. More recently, different orthonormal basis functions, *e.g.* Chebyshev polynomials [9,22,34] and Legendre polynomials [8,35], have been extensively considered for trajectory optimization problems.

Parameterizing $\mathcal{U}$ and/or $\mathcal{X}$ turns the infinite dimensional OCP into a finite dimensional optimal parameter selection problem, which can be seen as an *implicit* nonlinear mathematical programming problem (NLP). Implicit, since:

1. computing the integral cost,
2. finding the state trajectory solution consistent with the prescribed dynamics,
3. fulfilling the constraints $d(x, u) \leq 0$,

are all still infinite dimensional problems. Numerical procedures require further approximations. In most direct methods (see *e.g.* [7] and the references therein), this is achieved by *a priori* temporal discretization and approximation of the differential operator. The integral cost can then be approximately evaluated by any preferred quadrature rule (consult *e.g.* [12, 36]). In addition, the state and control constraints, $d(x, u) \leq 0$, are imposed at the temporal nodes and treated as regular constraints of the NLP. Finally, additional constraints are imposed on the NLP variables so that the generated state trajectory is consistent with the approximating differential operator.

From this discussion, one can realize that the accuracy of the obtained solution will generally depend on the temporal discretization scheme. As mentioned, *a priori* partition of the time interval into a prescribed number of sub-intervals, is the most intuitive, straightforward and widespread approach for this. It is however a well-established fact in numerical analysis (see *e.g.* [12, 37]), that a proper distribution of grid points is crucial for both the accuracy of the approximating solution, and the computational effort. The basic idea is that by concentrating the nodes and hence computational effort in those parts of the grid that require most attention (*e.g.* areas with sharp non-linearities and large solution variations), it becomes possible to gain accuracy whilst retaining computational efficiency. Since the solution is not known in advance, *a priori* node distribution has no way of paying attention to the particular problem at hand. To remedy this, *iterative* mesh refinement techniques have been suggested [38]. However, strategically adding new nodes to the current grid in an iterative manner, results in increased running times, which is counter productive for our online computational objectives. In Paper B, an adaptive temporal discretization method for trajectory optimization is suggested, where a *fixed* number of nodes are optimally distributed as to improve the accuracy of the approximation. The node distribution scheme is formulated as a constrained optimization problem, which is to be augmented to the underlying NLP and solved on the fly.

Observers for Nonlinear Systems

In complex real–life systems, it is typically some state variables which cannot be directly measured. If needed – for instance for feedback control design or monitoring purposes – one must aim at obtaining an *estimate* of these unknown state variables.

For a dynamical system, an *observer* is another dynamical system whose task is to reconstruct missing state information while only using available measurements. The input to the observer is the output of the original system (which may include its input), and the observer is expected to produce as output an estimate of some state function of the original system.

This section gives a brief and expository treatment of observers for nonlinear systems. One of the main objectives has been to relate the material to our viewpoint, and in extension, the relevant augmented papers (Paper C and D). The disposition is as follows. To start with, a concise and conceptually clear definition of an observer is given in Section 2.1. This definition is minimalistic in the sense that it specifies the minimal characteristics of an observer. For practical purposes, additional desired properties, such as domain of attraction, convergence rate, *etc.*, may be further specified. In particular, Section 2.2 is concerned with how to demonstrate the convergence properties of an observer. Other observer characteristics and classification thereof, are discussed more thoroughly in Section 2.3. One of the main paths for observer design, namely via nonlinear coordinate transformations, is discussed in Section 2.4. Finally, the main purpose of Section 2.5 is to pinpoint the nontrivial relationship between the concept of observability and observer existence for general nonlinear systems. This is an important point to make, not the least because of the treacherous similarities in the terminology.

2.1 Observer Definition

Consider the nonlinear control system

$$\Sigma : \begin{cases} \dot{x} &= \mathcal{F}(x,u) \qquad \text{(system dynamics)} \\ y &= h(x,u) \qquad \text{(system output)} \end{cases}$$

with state $x \in \mathcal{X}$, control $u \in \mathcal{U}$ and output $y \in \mathcal{Y}$. Here $\mathcal{X}$,$\mathcal{U}$ and $\mathcal{Y}$ are smooth manifolds of dimension n, p and m respectively. In the following, in addition to the measurements, the output y is supposed to include known control inputs. For the dynamical system Σ, an observer may be defined as follows.

Definition 2.1 (Observer)**.** *A dynamical system with state manifold $\mathcal{Z}$, input manifold $\mathcal{Y}$, together with a mapping $\hat{\mathcal{F}} : (\mathcal{Z} \times \mathcal{Y}) \to T\mathcal{Z}$ is an* observer *for the system Σ, if there exists a smooth mapping $\Psi : \mathcal{X} \to \mathcal{Z}$, such that the diagram shown in Figure 2.1 (the dashed arrow excluded), commutes. The observer gives a* full state reconstruction *if there in addition is a mapping $\Phi : (\mathcal{Z} \times \mathcal{Y}) \to \mathcal{X}$ such that the full diagram in Figure 2.1 is commutative (*cf. *[39] and [40]).*
Here, Ψ_* denotes the tangent mapping, π is projection upon a Cartesian factor, while τ denotes the projection of the tangent bundle.

$$\begin{tikzcd}
{} & T\mathcal{X} \arrow[r, "\Psi_*"] \arrow[dl, "\tau_{\mathcal{X}}"'] & T\mathcal{Z} \arrow[dr, "\tau_{\mathcal{Z}}"] & {} \\
\mathcal{X} & & & \mathcal{Z} \\
& \mathcal{X}\times\mathcal{U} \arrow[ul, "\pi_{\mathcal{X}}"] \arrow[uu, "\mathcal{F}"'] \arrow[r, "(\Psi\circ\pi_{\mathcal{X}}, h)"] & \mathcal{Z}\times\mathcal{Y} \arrow[uu, "\hat{\mathcal{F}}"] \arrow[ur, "\pi_{\mathcal{Z}}"'] \arrow[ull, dashed, bend left, "\Phi"]
\end{tikzcd}$$

Figure 2.1: Commutative diagram defining an observer.

According to Definition 2.1, the objective when designing a general observer, is to track $\Psi(x)$, rather than x itself. Note also, that the same observer dynamics $\hat{\mathcal{F}}$, may allow several *different* full state observer mappings, Φ, and that in general a full state observer

$$\hat{\Sigma} : \left\{ \begin{array}{rcl} \dot{z} & = & \hat{\mathcal{F}}(z, y) \\ \hat{x} & = & \Phi(z, y) \end{array} \right.$$

may *not* be put in the form $\dot{\hat{x}} = \Xi(\hat{x}, y)$.

As a consequence of this definition, an observer has the following basic property:

Property 2.1. *$z(t_0) = \Psi(x(t_0))$ at some time instance t_0, yields $z(t) = \Psi(x(t))$ for all $t \geq t_0$.*

Proposition 2.1. *An observer has Property 2.1 if and only if the diagram in Figure 2.1 is commutative.*

Proof. Assume we have Property 2.1, *i.e.*

$$z(t) = \Psi(x(t)), \quad \forall t \geq t_0.$$

Then taking the derivative with respect to time, yields

$$\dot{z} = \frac{\partial \Psi}{\partial x}\dot{x} = \frac{\partial \Psi}{\partial x}\mathcal{F}(x, u).$$

Comparing this with the expression for $\hat{\Sigma}$, we see that property 2.1 implies that

$$\frac{\partial \Psi}{\partial x}\mathcal{F}(x, u) = \hat{\mathcal{F}}(\Psi(x), h(x, u)), \tag{2.1}$$

which is exactly what the commutativity of the diagram in Figure (2.1) suggests.

In the other direction, assume that the diagram in Figure (2.1) commutes and that $z(t_0) = \Psi(x(t_0))$ for some $t_0 \in \mathbb{R}^+$. Solving the differential equation governing z, we have

$$z(t) - z(t_0) = \int_{t_0}^{t} \hat{\mathcal{F}}(z, y)\mathrm{d}\tau = \int_{t_0}^{t} \frac{\partial \Psi}{\partial x}\dot{x}\mathrm{d}\tau = \big[\Psi(x(\tau))\big]_{t_0}^{t} = \Psi(x(t)) - z(t_0),$$

the second and last equality following from the two assumptions made. □

Definition 2.1 thus provides a clear representation of Property 2.1 which is the minimal requirement that an observer has to fulfill. In particular, this definition does *not* impose any convergence requirements on the observer. This follows the line of thought in the pioneering work of Luenberger [41]. For practical purposes however, additional desired properties for an observer, such as domain of attraction, convergence rate, *etc.*, may be specified. This topic is discussed more thoroughly in Section 2.3. In particular, it is reasonable to require the additional property:

Property 2.2. *As time proceeds, the trajectories $z(t)$ and $\Psi(x(t))$ converges.*

This property, *i.e.* the convergence properties of an observer, may be demonstrated and defined in different ways. The convergence issue will be discussed in more detail in Section 2.2, where two approaches for showing convergence, namely Lyapunov-based methods and contraction analysis, are presented.

2.2 Convergence Analysis

2.2.1 Lyapunov-based methods

Lyapunov stability theory, and in particular Lyapunov's direct method, is a possible approach to determine the stability properties of a nonlinear system

$$\dot{w} = f(t, w), \qquad t \in \mathbb{R}^+, w \in \mathcal{W},$$

which may also represent controlled systems in closed loop form. As the subject is very well documented in the literature (consult *e.g.* [42–45]), the main emphasis here-within will be on using the Lyapunov theorems for showing observer convergence.

If one is able to find a (locally) positive definite, decrescent, continuously differentiable function, $V(t, w)$, whose total time derivative along the system dynamics, f,

$$\dot{V} = \frac{\partial V}{\partial t} + \frac{\partial V}{\partial w} f(t, w)$$

can be shown to be (locally) negative definite, then (local) uniformly asymptotic stability of the origin follows from Lyapunov theory (see *e.g.* Theorem 4.9 in [43]). Notice that in the case of time autonomous systems, the Lyapunov function may be taken as a time invariant functional. This result, is a strong, sufficient condition for stability and as such, incorporates a certain degree of conservatism. Despite the existence of converse theorems [46, 47], the main limitation of the Lyapunov based methods are that they are non-constructive, in the sense that they do not provide any systematic procedure for determining Lyapunov functions. Although natural Lyapunov candidates may be provided by Lyapunov-like "energy" functions, the choice of V is to a large extent a trial and error process that may be practically impossible for systems of high order.

In the observer design context of ours, we are interested in Lyapunov functions $V(z, \Psi(x)) \geq 0$ such that

$$V(z, \Psi(x)) = 0 \Longleftrightarrow z = \Psi(x).$$

In local coordinates, the objective is to determine the stability of the error dynamics, *i.e.* we wish to examine if the estimation error decays to zero. The more general choice is to set $w(t) = z(t) - \Psi(x(t))$, and consequently, $\mathcal{W} = \mathcal{Z}$. For full-state observers however, another possibility would be to consider the convergence of $\hat{x} = \Phi(z, y)$ to x in $\mathcal{X}$–space instead, *i.e.* setting $w(t) = \hat{x}(t) - x(t)$, and consequently, $\mathcal{W} = \mathcal{X}$. Because of its generality, we shall concentrate on the former case.

With $w(t) = z(t) - \Psi(x(t))$, the error dynamics becomes

$$\dot{w} = \dot{z} - \frac{\partial \Psi}{\partial x}\dot{x} = \hat{\mathcal{F}}(z, y) - \frac{\partial \Psi}{\partial x}\mathcal{F}(x, u) \triangleq f(t, w). \tag{2.2}$$

It is noteworthy that the commutativity of the diagram in Figure 2.1, makes the origin ($z(t) = \Psi(x(t))$) an equilibrium point of (2.2) (see also (2.1)). In general, error dynamics (2.2) is nonlinear. Further, it is a function of the true state x, which: first of all is unknown to us, and secondly is not a fixed quantity. These facts, clarify the need of techniques beyond the linear theory for analyzing the error dynamics (2.2). They also demonstrate the difficulty in making any statements regarding the convergence properties of the observer without any further specification of the functions involved.

2.2.2 Contraction Analysis

One way to determine the convergence properties of a dynamical system, such as the observer dynamics

$$\dot{z} = \hat{\mathcal{F}}(z, y),$$

is to use contraction analysis [48]. As a concept on a smooth Riemannian manifold, $\mathcal{Z}$, convergence of two neighboring trajectories is defined with respect to a given metric tensor, g. In essence, the dynamics $\hat{\mathcal{F}}$ is said to be a strict contraction with respect to the metric g, if for all inputs of $\hat{\Sigma}$,[1] $y \in \mathcal{Y}$, the symmetric part of its covariant derivative is negative definite. Since the Lie derivative of g with respect to the vector field $\hat{\mathcal{F}}$, denoted $\mathcal{L}_{\hat{\mathcal{F}}}g$, is proportional to the symmetric part of the covariant derivative of $\hat{\mathcal{F}}$, contraction may be characterized by negative definitiveness of $\mathcal{L}_{\hat{\mathcal{F}}}g$.

To see this, let ρ_t denote the geodesic curve between two neighboring trajectories, $z_1(\cdot)$ and $z_2(\cdot)$ (see Figure 2.2). Let $\Upsilon^t_{\hat{\mathcal{F}}}\rho_0$ denote the evolution of the geodesic curve, ρ_0, under the dynamics $\hat{\mathcal{F}}$, at time t. Further, let $\tau : [0, 1] \to \mathcal{Z}$ be a parameterization of the curve $\Upsilon^t_{\hat{\mathcal{F}}}\rho_0$ such that $\tau(0) = z_1(t)$ and $\tau(1) = z_2(t)$. We have

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\Upsilon^t_{\hat{\mathcal{F}}}\rho_0} \mathrm{d}s = \int_{\Upsilon^t_{\hat{\mathcal{F}}}\rho_0} \frac{1}{2}(\mathcal{L}_{\hat{\mathcal{F}}}g)(\frac{\mathrm{d}\tau}{\mathrm{d}s}, \frac{\mathrm{d}\tau}{\mathrm{d}s})\mathrm{d}s,$$

so if $\mathcal{L}_{\hat{\mathcal{F}}}g$ is negative definite for every input $y \in \mathcal{Y}$ ($\mathcal{L}_{\hat{\mathcal{F}}}g < 0$), then

$$\int_{\rho_t} \mathrm{d}s \triangleq \inf \int_{z_1(t)}^{z_2(t)} \mathrm{d}s \leq \int_{\Upsilon^t_{\hat{\mathcal{F}}}\rho_0} \mathrm{d}s \leq \int_{\rho_0} \mathrm{d}s \triangleq \inf \int_{z_1(0)}^{z_2(0)} \mathrm{d}s,$$

that is, the Riemannian distance between any two trajectories tends to zero as time proceeds (*cf.* [48]).

Contraction solely implies that the Riemannian distance between neighboring trajectories tends to zero. In order to be able to conclude regarding observer convergence, one must also verify that the observer dynamics contains the actual plant trajectory as a particular solution. This issue is actually the essence of Property 2.1. In conjunction with contraction, Property 2.1 automatically yields Property 2.2, *i.e.* observer convergence. In other words, if the observer dynamics, $\hat{\mathcal{F}}$, is a strict contraction with respect to g and further turns the diagram of Figure 2.1 commutative, then the observer is convergent.

Remark 2.1. Contraction, as described here-above, is a property of the control system on $\mathcal{Z}$ alone. In particular, it does not involve neither the control input, nor output of the original system, Σ.

[1] Which naturally coincide with the outputs of Σ.

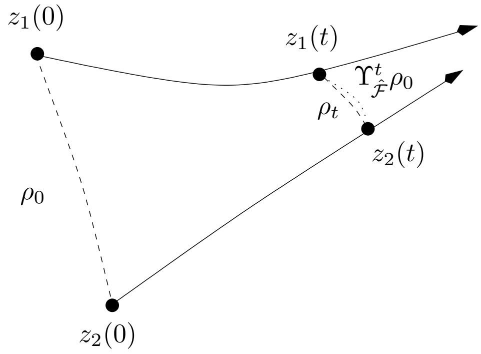


Figure 2.2: The length of the geodesic curve ρ_t, between two trajectories decreases if $\mathcal{L}_{\hat{\mathcal{F}}}g < 0$.

With this point of view, the control synthesis and observer design are decoupled processes and do not interact. To connect with the concept of *active observer design* of Paper C, requiring the symmetric part of the covariant derivative of $\hat{\mathcal{F}}$ to be negative definite for *all* inputs, $y \in \mathcal{Y}$, is an unnecessarily strong condition. As a convincing example, consider the case when the output map of Σ, $h : \mathcal{X} \times \mathcal{U} \to \mathcal{Y}$, is not onto. What can be done in the case when there are some y that do not turn $\hat{\mathcal{F}}$ contractive? Let the collection of y for which $\hat{\mathcal{F}}$ is contractive, be denoted by $\mathcal{Y}_c$. If the input to the original system, $u(\cdot)$, can be chosen such that $y(\cdot) \in \mathcal{Y}_c$, then the convergence of the observer is secured. The term active observer design refers exactly to this integrated fashion of control synthesis and observer design (*cf.* Paper C). The idea is to design the exciting control, while bearing in mind the convergence properties of the observer. This is an important issue since it is known that for nonlinear systems in general, the so called separation principle does not hold. That is, separate design of a stabilizing state feedback controller and a convergent observer, does not always result in a stabilizing output feedback controller. For a counter–example, consult [49].

The assumption that the observer dynamics is contractive, is however very restrictive and in many cases Property 2.2 has to be shown by means of Lyapunov–based methods. To this end, Lyapunov's direct method was briefly reviewed in the previous section 2.2.1.

2.3 Observer Properties and Classification

The advantages with representing an observer as in Definition 2.1, accrue particularly in the case of observer classification. This definition is minimalistic in the sense that it specifies the minimal characteristics of an observer, namely that the observer dynamics contains the actual plant trajectory as a particular solution. For practical purposes, additional properties may be further specified. This viewpoint, facilitates keeping the added properties separated and thereby achieving a transparent classification scheme for observers. In this section, a number of such properties are discussed. Moreover, some of the different observer types occurring in the literature are classified and related to these properties. The material presented in this section is of independent interest since observer nomenclature is not standardized and is to a large extend "author dependent". The list of observer properties and definitions in this section, by no means cover all characteristics that could be associated with observers. It rather provides the foundation for further elaboration and extensions.

As one of the key properties, *convergence* of observers has at least three aspects associated with it: the domain of attraction, the rate of convergence and its relation to the choice of system input. These are the first three properties to be discussed below.

Domain of attraction:

A most natural question, involves the extension of the domain of attraction, *i.e.* the set of initial points for which the observer converges. This characteristic basically tells us how far from the true state the initial estimation can be made without jeopardizing the convergence properties of the observer. In other words, this property puts a constraint on the distance between $\Psi(x(t_0))$ and $z(t_0)$ in $\mathcal{Z}$–space. There are at least three different restrictions to be discussed, namely:

- Global domain of attraction: There is no restriction, *i.e.* the observer converges for all $z(t_0) \in \mathcal{Z}$. Using the terminology of [48], having a global domain of attraction means that $\mathcal{Z}$ is a region of contraction.
- Local domain of attraction: For all $x(t_0) \in \mathcal{X}$, there is a $\varepsilon > 0$ such that the observer is convergent for all $z(t_0) \in \mathbb{B}(\Psi(x(t_0)), \varepsilon)$. Here, $\mathbb{B}(\Psi(x(t_0)), \varepsilon)$ denotes the ε–ball, centered at $\Psi(x(t_0))$.
- Semi–global domain of attraction: This refers to the cases in which it is possible to design observers that converge on every compact subset of $\mathcal{Z}$. For instance, if for all $\varepsilon > 0$, one can possibly tune the observer design parameters so that convergence is guaranteed for all $z(t_0) \in \mathbb{B}(\Psi(x(t_0)), \varepsilon)$.

Remark 2.2. Unless $\Psi(x(t_0))$ is *a priori* known, which is seldom the case, global domain of attraction is the only fully implementable version listed above. Having only a local domain of attraction, renders observer initialization practically a trial and error procedure (since ε is unknown and might potentially be very small). The semi–global version of the domain of attraction property could be of interest, for instance, when there are *a priori* known bounds on the systems domain of operation[2] (in $\mathcal{X}$–space). If compact, then by the smoothness (and hence continuity) of Ψ, this also bounds $\Psi(x(t_0))$ in $\mathcal{Z}$–space.

Rate of convergence:

This property concerns the rate with which the estimation error decays to zero. As the notion of "convergence rate" might be familiar to most readers, the most frequently discussed ones are simply listed.

- Asymptotic convergence rate
- Exponential convergence rate
- Finite–time convergence

It is noteworthy that finite–time convergence is not possible to obtain with locally Lipschitz vector fields. Some other issues related to the rate of convergence are to be found in Definition 2.2 and Remark 2.5.

Input dependent convergence:

For forced nonlinear systems in general, convergence of the observer is an *input dependent* property. As a particular example, which will be at the center of discussion in Paper C, consider the case of a mobile robot equipped with exteroceptive sensors. For simplicity of discussion, it is assumed that the system is globally observable, *i.e.* for any two initial states, $x_1, x_2 \in \mathcal{X}$, there exist a control trajectory, $u(\cdot) \in \mathcal{U}$, that distinguishes the outputs, $y_1(\cdot) = h(X(x_1, u(\cdot)), u(\cdot))$ and $y_2(\cdot) = h(X(x_2, u(\cdot)), u(\cdot))$.

[2] The operation domain of a system is yet another observer property to be discussed later in this section.

Here, $X(x_1, u(\cdot))$ denotes a solution of Σ using the control input, $u(\cdot)$ with x_1 as the initial condition. Despite this strong assumption, there might be control inputs for which the observer does not converge. As in the example of a unicycle robot model, the zero input corresponds to the mobile platform standing still and therefore not collecting any new data. As the output remains constant and no new information arrives to the observer, the estimation procedure can not proceed, hence observer convergence is not possible.

With this in mind, it is possible to distinguish the following observer properties (*cf.* [50]).

- Uniform convergence: This most demanding version of this property requires convergence of the observer for *all* control inputs, $u \in \mathcal{U}$.
- Semi–uniform convergence: A more moderate version would be to require observer convergence whenever the input is restricted to given compact subsets of $\mathcal{U}$, for instance, dictated by ways of norm restrictions. The bounding set is however allowed to be arbitrarily large, possibly with the observer design parameters varying accordingly.
- Non–uniform convergence: This refers to the case when there are control inputs for which the observer does not converge.

Remark 2.3. The attentive reader might here discern the connection of input dependent convergence with Remark 2.1 and the succeeding discussion on active observer design (*cf.* Paper C).

The two last characteristics to be discussed, involve different restrictions on the observer- and system trajectories respectively.

Domain of operation of the observer:

Another distinguishable property of an observer, concerns whether or not the observer state, $z(t)$, is restricted to remain close to $\Psi(x(t))$. This is an extremely important concept, not the least for safety critical output feedback controllers. In such systems, special attention has to be paid to the "peaking phenomenon" of the estimation error [51], *i.e.* the potentially large mismatch between $\Psi(x(t))$ and $z(t)$, and in extension, between the output feedback controllers, $u(y(t), \Psi(x(t)))$ and $u(y(t), z(t))$. Similar restricted concepts have already been defined in the case of accessibility, controllability, stabilizability and observability (see *e.g.* [51, 52] and [53] page 11).

The *global* case implies unrestricted observer trajectories, *i.e.* the mismatch between $z(t)$ and $\Psi(x(t))$ may vary arbitrarily. An observer having *semi–global* domain of attraction, refers to the case when $z(t)$ can be made to stay in *a chosen neighborhood* of $\Psi(x(t))$. The semi–global version is of utmost practical interest, since it puts an upper bound on the maximum estimation error. Finally, the most restricted version, termed *local*, restricts the generated observer state, $z(t)$, to stay in *any prescribed neighborhood* of $\Psi(x(t))$. Here, we assume that the observer is initiated at a point inside the given neighborhood. Thus it is possible to dictate the estimation accuracy of an observer with a local domain of operation, which is a very strong requirement.

Domain of operation of the system:

Yet another characteristic worth mentioning, is the region of the state space in which the system is operating. This issue involves whether or not we have restrictions on the unmeasured states of the system and exists in a global, semi–global and local version (*cf.* [54]). If the state, $x(t)$, may vary arbitrarily on the state space, $\mathcal{X}$, without jeopardizing the convergence properties of the observer, then the corresponding observer is termed *global.* The *semi-global* version refers to those cases when it is possible to design a convergent observer, once the unmeasured states are restricted to a given compact subset of the state space. The size of this region is however allowed to be made arbitrarily large, possibly with the observer design parameters varying accordingly. Finally, the *local* version refers to those cases when it is possible to design an observer that converges only if the unmeasured states are restricted to a neighborhood of a given state, x_0.

Remark 2.4. The intrinsic observer of Paper D, serves as an illustrative example of the importance of the system's domain of operation in observer design and convergence analysis.

From this discussion, it should be clear that the terminology for observers and observer design is not a trivial matter. For instance, a "local observer" might refer to several distinct properties. Therefore, one should always strive to adopt a descriptive nomenclature and keep the properties and the spaces they live in separated. To this end, Table 2.1 collects the properties listed in this section.

Observer property	Restricted versions	Property space
Domain of attraction	Global, semi–global, local	$\mathcal{Z}$
Rate of convergence	Asymp., Exp., Finite–time	$\mathcal{Z}$ or $\mathcal{X}$
Input dependent conv.	Unif., semi–unif., non–unif.	$\mathcal{U}$
Observer's domain of oper.	Global, semi–global, local	$\mathcal{Z}$
System's domain of oper.	Global, semi–global, local	$\mathcal{X}$

Table 2.1: By keeping the observer properties and their associated spaces separated, it is possible to set up a transparent classification scheme and adopt a descriptive nomenclature.

We proceed by relating some of the observer types occurring in the literature to the concepts and properties listed so far.

Definition 2.2 (Asymptotic, Exponential, and Finite–time Observer)**.** *An observer whose estimation error has an asymptotic (exponential) rate of decay, is called an* asymptotic (exponential) observer*. Finite-time observers provide a correct estimation of* $\Psi(x(t))$ *within finite–time.*

Remark 2.5. Despite the fact that neither asymptotic, nor exponential observers are able to reconstruct $\Psi(x(t))$ within finite–time, they are the most frequently existing observers in the literature. One explanation of this might be that finite–time convergent observers require non–smooth vector fields, $\hat{\mathcal{F}}$.

Definition 2.3 (Identity Observer)**.** *The special case when* Ψ *equals the identity map and* $\mathcal{Z} = \mathcal{X}$*, is often referred to as an* identity observer*.*

Definition 2.4 (Smooth and Continuous Observers [54])**.** *Referring to diagram 2.1, if* Ψ^{-1} *is a smooth map (*i.e. Ψ *is a diffeomorphism), the observer is referred to as a* smooth observer*, while* Ψ^{-1} *being merely continuous (*i.e. Ψ *being a semi-diffeomorphism), yields a* continuous observer*.*

A continuous observer might be of interest whenever the smooth exponential observer falls short, which occurs exactly when the Taylor linearization of system Σ is undetectable [55]. For an example, explicitly constructed to show the non-existence of a smooth observer and point out the potential of continuous observers, consult [56].

Definition 2.5 (Reduced Order, Full Order and Expanded Order Observers [54,56])**.** *These three classes can be distinguished according to the dimension of $\mathcal{Z}$, which defines the order of the observer. An observer is termed* reduced order *if its order is less than n (the dimension of $\mathcal{X}$). It is called* full order *if it is of order n and the full-state estimate does not depend directly on y,* i.e. $\hat{x} = \Phi(z)$*. Finally, an observer of order greater than n is called* expanded order*.*

Definition 2.6 (Uniform and Non–Uniform Observer [50])**.** *An observer whose convergence properties does not depend on the input to the original system, Σ, is called a* uniform observer*. Else, it is termed* non–uniform observer*.*

Remark 2.6. If one considers disturbances as unknown (unmeasured) inputs to the original system, *robust observers* [57] may be seen to equal the concept of uniform observers.

2.4 Coordinate Transformations and Linear Error Dynamics

The problem of existence and synthesis of observers for linear systems is fully understood [41]. It may then seem natural that for the design of observers for nonlinear systems, significant amounts of research have been conducted with the aim of finding special coordinate systems – but also conditions for the existence of them – in which one can adopt techniques from linear systems theory. In fact, one of the main paths for observer design for nonlinear systems, goes via nonlinear coordinate transformation. The idea is to turn the original nonlinear system into some specific "observer form", utilizing *e.g.* diffeomorphism or immersion [58–64]. In these new coordinates, classical methods from linear systems theory are employed to complete the observer design procedure.

In two seminal papers [59,60] – which treat the case of unforced single output and MIMO systems respectively – the idea of using state transformations in order to turn the nonlinear system into a linear one up to output injection was settled. In a first step, the authors seek diffeomorphisms,

$$\tilde{x} = T_1(x) \quad \text{and} \quad \tilde{y} = T_2(y),$$

such that the original nonlinear system

$$\Sigma : \left\{ \begin{array}{rcl} \dot{x} & = & \mathcal{F}(x,u), \\ y & = & h(x), \end{array} \right.$$

is transformed into

$$\tilde{\Sigma} : \left\{ \begin{array}{rcl} \dot{\tilde{x}} & = & A\tilde{x} - g(y), \\ \tilde{y} & = & C\tilde{x}, \end{array} \right.$$

with (A, C) an observable pair. System $\tilde{\Sigma}$ is an observable linear system up to output injection which is known to admit Luenberger type of observers. Notice that both these papers require the output map to be linear as well. The observer design procedure is then completed in a second step by adopting a Luenberger style observer

$$\hat{\Sigma} : \left\{ \begin{array}{rcl} \dot{z} & = & Az - g(y) + L(\tilde{y} - Cz) \\ \hat{x} & = & T_1^{-1}(z), \end{array} \right.$$

where L is the constant observer gain. To see the main advantage with the proposed transformation, one must have a look at the error dynamics;

$$\frac{\mathrm{d}}{\mathrm{d}t}(z - \tilde{x}) = \dot{z} - \dot{\tilde{x}} = Az - g(y) + L(\tilde{y} - Cz) - A\tilde{x} + g(y) = (A - LC)(z - \tilde{x}). \tag{2.3}$$

Since (A, C) is an observable pair, pole placement can be used to obtain exponential convergence (with arbitrary rate) for the *linear* error dynamics (2.3).

More recently, Kazantzis and Kravaris [65] have proposed to rather seek a diffeomorphism, $\tilde{x} = T(x)$, that transforms the original nonlinear system to a system having linear dynamics up to output injection (the output map may however be nonlinear)

$$\tilde{\Sigma}' : \left\{ \begin{array}{rcl} \dot{\tilde{x}} & = & A\tilde{x} - g(y) \\ y & = & h(T^{-1}(\tilde{x})) \end{array} \right.$$

where the matrix A is Hurwitz. It is then possible to propose an observer for the transformed system, $\tilde{\Sigma}'$, namely the dynamic system

$$\hat{\Sigma}' : \left\{ \begin{array}{rcl} \dot{z} & = & Az - g(y) \\ \hat{x} & = & T^{-1}(z), \end{array} \right.$$

with the associated linear stable error dynamics

$$\frac{\mathrm{d}}{\mathrm{d}t}(z - \tilde{x}) = A(z - \tilde{x}),$$

will serve as a smooth observer having exponential error decay. In terms of the original set of coordinates, the following dynamic system

$$\dot{\hat{x}} = \mathcal{F}(\hat{x}, u) + L(\hat{x})\big[g(y) - g(h(\hat{x}))\big], \tag{2.4}$$

with nonlinear gain

$$L(\hat{x}) = \big[\frac{\partial T}{\partial \hat{x}}(\hat{x})\big]^{-1},$$

is a full-state observer for the original system, Σ. Note that it is not in general possible to put a full state observer (such as $\tilde{\Sigma}'$) in the form $\dot{\hat{x}} = \Xi(\hat{x}, u, y)$ as is done in (2.4).

These described approaches have some drawbacks:

- The structural requirements are extremely stringent so that large classes of nonlinear systems are excluded. In particular, it is noteworthy that both Σ and $\tilde{\Sigma}'$ exclude the class of non–uniformly observable systems.

- Finding the right state transformation may be difficult since it involves solving a system of first-order partial differential equations. In order to overcome this and make practical use of the described approaches, different approximation schemes may be adopted. In [65] and [66][3] approximation schemes based on simple series expansions are proposed.

- The observer has only local properties, more specifically where the coordinate transformation is valid. Global statements require that the collection of local transformations are consistent, *i.e.* form an atlas.

[3] Mind the important erratum to this paper [67].

2.5 Observability and the Existence of Observers

The main purpose of this section is to pinpoint the nontrivial relationship between the concept of observability and observer existence for general nonlinear systems. This is an important point to make, not the least because of the treacherous similarities in the terminology. By using two examples, it will be shown that for nonlinear systems in general, and non-uniformly observable systems in particular, observability may not imply, nor is implied by the existence of an observer. A more suitable concept from the observer design point of view, is that of detectability [50, 56]. This is in accordance with linear systems theory.

Example 2.1 (observability $\not\Rightarrow$ observer)**.** Consider

$$\Sigma_1 \left\{ \begin{array}{rcl} \dot{x}_1 & = & -x_1 + u x_2^3 \\ \dot{x}_2 & = & x_2 + x_1^2 \\ y & = & x_1 \end{array} \right.$$

It is easy to show that Σ_1 is (non-uniformly) observable by noting that $u_0(\cdot) = 1$ distinguishes all initial points. To see this, assume there are two initial states

$$x_1 = [x_{11}\ x_{12}]^T \quad \text{and} \quad x_2 = [x_{21}\ x_{22}]^T$$

such that

$$y_0(x_1, t) \equiv y_0(x_2, t), \quad \forall t \geq 0. \tag{2.5}$$

Here $y_0(x_i, t)$ denotes the output trajectory when the system is initiated at x_i, $i = 1, 2$ and driven by the input function $u_0(\cdot) = 1$. From (2.5) we immediately obtain $x_{11} = x_{21}$, for the special choice of $t = 0$. Equation (2.5) also implies

$$\dot{y}_0(x_1, t) \equiv \dot{y}_0(x_2, t) \quad \forall t \geq 0$$

which yields $x_{12} = x_{22}$, for the special choice of $t = 0$. Hence we conclude that $x_1 = x_2$, *i.e.* $u_0(\cdot) = 1$ distinguishes all initial points. However, it has been shown in [68] (see also [56]) that there does not exist any smooth observer with asymptotically stable error dynamics for Σ_1, due to the positive eigenvalue associated with x_2.

Example 2.2 (observability $\not\Leftarrow$ observer)**.** Consider

$$\Sigma_2 \left\{ \begin{array}{rcl} \dot{x}_1 & = & u \\ \dot{x}_2 & = & x_1 + x_2^2 \\ \dot{x}_3 & = & -x_3 + x_2 \\ y & = & x_2 \end{array} \right.$$

which is not observable since x_3 is neither measured nor affects the dynamics of x_1 or x_2. Still it can be shown by considering the error dynamics, that for proper choice of α and β

$$\hat{\Sigma}_2 \left\{ \begin{array}{rcl} \dot{\hat{x}}_1 & = & u + \alpha(y - \hat{x}_2) \\ \dot{\hat{x}}_2 & = & \hat{x}_1 + y^2 + \beta(y - \hat{x}_2) \\ \dot{\hat{x}}_3 & = & -\hat{x}_3 + y, \end{array} \right.$$

acts as an input-independent observer for Σ. This is because the dynamics for x_3 is stable in itself, once the other two states have been driven to zero.

Reader's Guide

This thesis consists of four independent papers presented in reversed chronological order. This section offers a brief and descriptive summary of the appended papers. The first two papers consider different aspects of the trajectory planning problem.

Paper A: *Online Trajectory Planning for Aerial Vehicles: a Safe Approach with Guaranteed Task Completion*, coauthored with J. Robinson and P. Ögren.

In this paper, online trajectory planning for aerial vehicles subject to simultaneous kinematic and dynamic constraints is considered. The trajectory planning problem is formulated as a somewhat modified Optimal Control Problem (OCP). An underlying assumption however, is that due to imperfect information, the kinematic constraints, as well as the location of the target set and possible threats, might change during the course of flight. Also, assuming that the problem originates from a complex, real-world application, the existence of analytical solutions is disregarded; thus seeking fast computational algorithms for solving the OCP.

In order to cope with the real-time objectives, the line of thought presented in this paper merges that of point-wise satisficing control [69,70] and Receding Horizon Control (RHC) [71]. Here-within, the global characteristics of the environment and mission objectives are captured in a functional, calculated off-line and passed to the online receding horizon controller as a terminal cost. It should however be mentioned that the possibility of updating the "off-line" computed terminal cost should not be overlooked. As pointed out in [72], the term "off-line" is rather to be interpreted as, at a much slower sampling rate than the control loop. As new information about the environment or mission objectives is gathered when the mission unfolds, it can be processed and fed back regularly to the vehicle through an updated terminal cost, as discussed in *e.g.* [73].

It is known that in the absence of particular precautions when approximating the tail of the integral cost with a terminal cost, neither closed-loop stability (or task completion[1]), nor safety can be assured.

These issues have been the leitmotivs of Paper A. Its main contributions are two-fold; by augmenting a so called safety maneuver at the end of the planned trajectory, this paper extends previous results by addressing provable safety properties in a 3D setting. In addition, assuming initial feasibility, the planning method presented is shown to have finite time task completion. As a subsidiary consequence, it is noteworthy that introducing the safety maneuver, also makes it possible to cope with *hard* real-time systems as well as various optimization routine failures including non–convergence and abnormal termination.

[1] Standard RHC is tailored for steady-state control or asymptotic stabilization to the origin in the Lyapunov sense. The notion of *task completion* considered in this paper is a *different problem*, namely it aims at controlling the plant into a target set which is not necessarily control invariant or contain any equilibrium points.

Moreover, a quantitative comparison between the two competing objectives of optimality and computational tractability is made. Some other key characteristics of the trajectory planner, such as ability to minimize threat exposure and robustness, are highlighted through simulations.

The second part of this paper is devoted to the problem of simultaneous arrival of multiple aerial vehicles. This is an application of the wider class of consensus problems. Standard results in this field involve consensus problems described by unconstrained, scalar (or fully actuated) first order linear systems (see *e.g.* [74–76]). In our case however, the relation between the control (*i.e.* the vehicle acceleration), and the consensus quantity (which is the Estimated Time to Arrival, ETA) is neither unconstrained nor first order. However, by using a time-scale separation principle, we are able to achieve consensus by adopting standard Laplacian control.

Paper A is based on the following publications:

A1: D.A. Anisi, J. Robinson and P. Ögren, *Safe Receding Horizon Control of an Aerial Vehicle*, IEEE Conference on Decision and Control, San Diego, CA, Dec., 2006.

A2: D.A. Anisi, J. Robinson and P. Ögren, *Online Trajectory Planning for Aerial Vehicles: a Safe Approach with Guaranteed Task Completion*, AIAA Guidance, Navigation and Control Conference and Exhibit, Keystone, Colorado, Aug. 2006.

A3: D.A. Anisi, J. Hamberg and X. Hu, *Nearly Time-Optimal Paths for a Ground Vehicle*, Journal of Control Theory and Applications, Nov. 2003.

Paper B: *Adaptive Node Distribution for Online Trajectory Planning.*

For the actual design of the computational algorithm for solving an Optimal Control Problem (OCP) – like the one considered in Paper A – the infinite-dimensional problem of choosing a control function in a given space, has to be *transcribed* into a finite dimensional parameter selection problem, or a nonlinear mathematical programming problem (NLP).

Traditional transcription methods are based on *a priori* partition of the time interval into a prescribed number of subintervals whose endpoints are called *nodes*. Generally, trajectory optimization run-times are critically depending on the number of variables in the NLP. These in turn, are proportional to the number of nodes in the temporal discretization. Therefore, it is extremely important to keep the number of nodes as low as possible when aiming at constructing computationally efficient methods for trajectory optimization.

It is a well-established fact in numerical analysis, that a proper distribution of grid points is crucial for both the accuracy of the approximating solution, and the computational effort (see *e.g.* [12, 37]). The basic idea is that by concentrating the nodes and hence computational effort in those parts of the grid that require most attention, *e.g.* areas with sharp non-linearities and large solution variations, it becomes possible to gain accuracy whilst retaining computational efficiency.

Inspired by this, Paper B advocates that in any computationally efficient method for trajectory optimization, node distribution should be a part of the optimization process. More precisely, once the number of nodes in the temporal discretization has been decided (depending on *e.g.* computational resources), the question of optimal node distribution is raised. Based on two existing frameworks for adaptive grid generation, namely equidistribution principle and functional minimization, node distribution is formulated as a constrained optimization problem, which is to be augmented with the underlying NLP. Although adaptive grid methods - which mainly concern node distribution in the *spatial* domain - have been an active field for the last couple of decades, to the best of the author's knowledge, utilizing them for adaptive node distribution (in the *temporal* domain) and online trajectory optimization has not been considered elsewhere. The benefits of utilizing the suggested adaptive node distribution method for online trajectory optimization, are illustrated by a missile guidance example.

This paper corresponds to the following publications:

B1: D.A. Anisi, *Adaptive Node Distribution for Online Trajectory Planning*, Congress of the International Council of the Aeronautical Sciences (ICAS), Hamburg, Germany, Sep. 2006.

B2: D.A. Anisi, *Online Trajectory Planning Using Adaptive Temporal Discretization*, Swedish Workshop on Autonomous Robotics (SWAR), Stockholm, Sweden, Sep. 2005.

The last two papers of this thesis deal with observer design for mobile robotic and Euler-Lagrange systems, respectively. A short description of these papers follows.

Paper C: *Active Observers for Mobile Robotic Systems*, coauthored with X. Hu.

Feedback control design techniques require knowledge about at least some parts of the state vector. If all the state variables necessary for the control system can not be directly measured, which is a typical situation in complex systems, attention must be directed towards obtaining an *estimate* of the unknown state variables. Most current methodologies for observer design, such as observers with linearizable error dynamics [59, 60, 65] and high gain observers [77, 78], lead to the design of an exponential observer. As a necessary condition for the existence of a smooth exponential observer, the linearized pair must be detectable [55]. In fact, most of the existing nonlinear observer design methods are only applicable to uniformly observable nonlinear systems. Study for observer design of non-uniformly observable systems is still lacking, except for bilinear systems. This is witnessed in [79], where it is pointed out that one of the key questions in nonlinear control is "how to design a nonlinear observer for nonlinear systems whose linearization is neither observable nor detectable".

An important class of non-uniformly observable systems comes from applications in mobile robotics. A mobile robot typically operates in an environment (work-space) with obstacles, and is equipped with exteroceptive sensors to aid localization. For such systems, due to environmental restrictions and the way the sensors function, the exciting control has to be chosen in a deliberate manner, *i.e.* an *active observer* has to be designed.

Paper C considers the problem of active observer design for mobile robotic systems and proposes an alternative design methodology. Moreover, it extends the observability concept to the field of mobile robotics by proposing a new concept called *small-time observability.*

The main ingredients of the proposed methodology include:

- The set of feasible configurations
- The set of output flow equivalent states

In this paper, it is shown that the inter-relation between these two sets may serve as the basis for design of active observers. More precisely, the main theoretical result states that if the exciting control is chosen such that the intersection of the two sets is a singleton, then the system is small-time observable.

In order to give a conceptually clear description of the main ingredients and steps required in the construction, a design study is presented. There-within, an active observer is designed for a unicycle robot model, equipped with a set of range-measuring sensors. Finally, by means of Lyapunov's direct method, it is shown that the designed observer has locally bounded error and that this bound can be made arbitrary small by tuning the observer gains.

An earlier version of paper C has appeared as

C1: D.A. Anisi and X. Hu, *Observability and Active Observers for Mobile Robotic Systems*, International Symposium on Mathematical Theory of Networks and Systems (MTNS), Kyoto, Japan, Jul. 2006.

Paper D: *Riemannian Observers for Euler-Lagrange Systems*, coauthored with J. Hamberg.

In the last paper of this thesis, a *geometrically intrinsic* observer for a class of nonlinear systems is defined and analyzed. The subclass considered is that of Euler-Lagrange systems, where the output of the system is assumed to be the generalized position and force, and the goal is to reconstruct the generalized velocities. An often practiced solution to the problem of reconstructing the velocity variables is to numerically differentiate the known position measurements. This approach however, fails to perform for high and fast varying velocities, but naturally also when noise has made the position measurements havoc.

It is known that the Euler-Lagrange equations are *intrinsic* and may be written in a coordinate-free way (see *e.g.* [80]). It is then natural to keep this coordinate independence in the observer design as well. The Riemannian geometric point of view has influenced part of control theory, *e.g.* optimal control and control design. However, the impact on observer design, has been modest. Suppressing unnecessary coordinates in the observer design has several prominent features. Beside the obvious advantage of having one universal observer for all coordinate systems, the minimum quantities for defining an observer become evident. These issues are two of the principal interests of the work presented in this paper.

The presented observer is a generalization of the one recently proposed by Aghannan and Rouchon [81]. There, the authors successfully adopt contraction analysis [48], to address convergence of an intrinsic observer for Euler-Lagrange systems with position measurements. In this paper, their contractivity result is reproduced and complemented by a proof that the region of contractivity is infinitely thin. In addition, the results of [81] are extended by using Lyapunov theory to show convergence in the constant curvature case, whenever we have *a priori* given bounds on the generalized velocities. In the case of physical (*e.g.* mechanical or electrical) Euler-Lagrange systems, this assumption is a realistic one.

Finally, the convergence properties of the observer are illustrated by an example where the configuration manifold is the three-dimensional sphere, S^3.

A more compressed version of this paper has been previously published as

D1: D.A. Anisi and J. Hamberg, *Riemannian Observers for Euler-Lagrange Systems*, IFAC World Congress, Prague, Czech Republic, July 2005.

3.1 Remark on Notation

In the four independent papers that follow, the notation is introduced separately in each paper. The reader is urged to mind notational collision.

Main Contributions and Limitations

The main contributions of this thesis are:

+ Paper A extends previous results regarding Receding Horizon Control (RHC) of autonomous vehicles, by addressing safety and task completion properties in a 3D setting.
+ Paper B presents an adaptive node distribution scheme for online trajectory planning.
+ A concise and conceptually clear definition of an observer is given. This definition underlies the results of Paper C and D.
+ Based on this definition, several distinguished observer properties are listed and a classification scheme for observers is proposed.
+ Paper C proposes an alternative methodology for designing active observers for mobile robotic systems.
+ Paper C also proposes a new observability concept called *small-time observability* and provides sufficient condition for it.
+ Paper D defines and analyzes a geometrically intrinsic observer for Euler-Lagrange systems with position measurements.

The main limitations of the results of this thesis are:

– The main results of Paper A and B should be interweaved. That is, investigation should be pursued about the possibility of increasing the accuracy of the safe trajectory planner of Paper A, by using the adaptive node distribution scheme proposed in Paper B.

– In Paper C, the important question of the relation between the given environmental map and the global convergence properties of the proposed observer, should be more extensively studied.

– Using the approach from Paper D on more general spaces is prohibitive. Approximation schemes are called for.

– The observer of Paper D should be combined with an intrinsic formulation of state-feedback control. This idea has been elaborated upon in [82, 83] in the special case when the manifold is a Lie group and the kinetic energy is left invariant.

4.1 Work Division

The first three papers are mainly due to the first author. The coauthors have here provided invaluable inputs by pointing out unclear arguments and suggesting improvements. However, the lion's share of Paper D is profoundly based on the arsenal of differential geometric tools of Johan Hamberg. The first author made contributions mostly in the introductory part, the formulation of the given proofs and the section on Euler-Lagrange systems. He also served as a critical reviewer of the extensive index gymnastics.

References

[1] Latombe, J.-C., *Robot motion planning*, Kluwer Academic Publishers, 1991.

[2] LaValle, S. M., *Planning Algorithms*, Cambridge University Press, Cambridge, U.K., 2006, Also available at http://planning.cs.uiuc.edu/.

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

[4] Donald, B., Xavier, P., Canny, J., and Reif, J., "Kinodynamic motion planning," *J. Assoc. Comput. Mach.*, Vol. 40, No. 5, 1993, pp. 1048–1066.

[5] Laumond, J.-P., editor, *Robot motion planning and control*, LAAS-CNRS, Toulouse, France, Aug. 1997.

[6] Ma, B., *An improved algorithm for solving constrained optimal control problems*, Ph.D. thesis, Institute for Systems Research, University of Maryland, 1994.

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

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

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

[10] Pontryagin, L. S., Boltyanskii, V. G., Gamkrelidze, R. V., and Mishchenko, E. F., *The mathematical theory of optimal processes*, Translated from the Russian by K. N. Trirogoff; edited by L. W. Neustadt, Interscience Publishers John Wiley & Sons, Inc. New York-London, 1962.

[11] Bellman, R. E., *Dynamic programming*, Princeton Univeristy Press, Princeton, N. J., 1957.

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

[13] Von Stryk, O. and Bulirsch, R., "Direct and indirect methods for trajectory optimization," *Ann. Oper. Res.*, Vol. 37, No. 1-4, 1992, pp. 357–373, Nonlinear methods in economic dynamics and optimal control (Vienna, 1990).

[14] Collis, S. S. and Heinkenschloss, M., "Analysis of the streamline upwind/Petrov Galerkin method applied to the solution of optimal control problems," Tech. rep., CAAM TR02-01, Rice University, Mar. 2002.

[15] Anisi, D. A., Hamberg, J., and Hu, X., "Nearly time-optimal paths for a ground vehicle," *Journal of Control Theory and Applications*, Vol. 1, No. 1, 2003, pp. 2–8.

[16] Dubins, L., “On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents,” *American Journal of Mathematics*, Vol. 79, 1957, pp. 497–516.

[17] Sirisena, H. R. and Chou, F. S., “State parameterization approach to the solution of optimal control problems,” *Optimal Control Appl. Methods*, Vol. 2, No. 3, 1981, pp. 289–298.

[18] Sirisena, H., “Computation of optimal controls using a piecewise polynomial parameterization,” *IEEE Transactions on Automatic Control*, Vol. 18, No. 4, Aug. 1973, pp. 409–411.

[19] Goh, C. J. and Teo, K. L., “Control parametrization: A unified approach to optimal control problems with general constraints,” *Automatica*, Vol. 24, No. 1, Jan. 1988, pp. 3–18.

[20] Neuman, C. and Sen, A., “A suboptimal control algorithm for constrained problems using cubic splines,” *Automatica*, Vol. 9, No. 5, Sep. 1973, pp. 601–613.

[21] Hargraves, C. and Paris, S., “Direct trajectory optimization using nonlinear programming and collocation,” *Journal of Guidance, Control, and Dynamics*, Vol. 10, No. 4, 1987, pp. 338–342.

[22] Vlassenbroeck, J. and Van Dooren, R., “A Chebyshev technique for solving nonlinear optimal control problems,” *IEEE Trans. Automat. Control*, Vol. 33, No. 4, 1988, pp. 333–340.

[23] Seywald, H., “Trajectory optimization based on differential inclusion,” *Journal of Guidance, Control and Dynamics*, Vol. 17, No. 3, 1994, pp. 480–487.

[24] Kumar, R. R. and Seywald, H., “Should controls be eliminated while solving optimal control problems via direct methods?” *Journal of Guidance, Control and Dynamics*, Vol. 19, No. 2, 1996, pp. 418–423.

[25] Conway, B. A. and Larson, K. M., “Collocation versus differential inclusion in direct optimization,” *Journal of Guidance, Control and Dynamics*, Vol. 21, No. 5, 1998, pp. 780–785.

[26] Fahroo, F. and Ross, M. I., “A second look at approximating differential inclusions,” *Journal of Guidance, Control and Dynamics*, Vol. 24, No. 1, 2001, pp. 131–133.

[27] Rockafellar, R. T., “Lagrange Multipliers and Optimality,” *SIAM Review*, Vol. 35, No. 2, 1993, pp. 183–238.

[28] Sirisena, H. R. and Chou, F. S., “An efficient algorithm for solving optimal control problems with linear terminal constraints,” *IEEE Trans. Automatic Control*, Vol. AC-21, No. 2, 1976, pp. 275–277.

[29] Petit, N., Milam, M. B., and Murray, R. M., “Inversion based constrained trajectory pptimization,” *Proc. of IFAC Symposium on Nonlinear Control Systems Design*, 2001.

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

[31] Watson, G. A., *Approximation theory and numerical methods*, John Wiley & Sons Ltd., Chichester, 1980, A Wiley-Interscience Publication.

[32] Powell, M. J. D., *Approximation theory and methods*, Cambridge University Press, Cambridge, 1981.

[33] Rudin, W., *Principles of mathematical analysis*, McGraw-Hill Book Co., New York, 3rd ed., 1976, International Series in Pure and Applied Mathematics.

[34] Elnagar, G. N. and Kazemi, M. A., "Pseudospectral Chebyshev optimal control of constrained nonlinear dynamical systems," *Comput. Optim. Appl.*, Vol. 11, No. 2, 1998, pp. 195–217.

[35] Fahroo, F. and Ross, M., "Legendre pseudospectral approximations of optimal control problems," *New trends in nonlinear dynamics and control, and their applications*, Vol. 295 of *Lecture Notes in Control and Inform. Sci.*, Springer, Berlin, 2003, pp. 327–342.

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

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

[38] Betts, J. T. and Huffman, W. P., "Mesh refinement in direct transcription methods for optimal control," *Optimal Control Appl. Methods*, Vol. 19, No. 1, 1998, pp. 1–21.

[39] Van der Schaft, A., "On nonlinear observers," *IEEE Trans. Automat. Control*, Vol. AC-30, No. 12, Dec. 1985, pp. 1254–1256.

[40] Thau, F., "Observing the state of non-linear dynamic systems," *International Journal of Control*, Vol. 17, 1973, pp. 471–479.

[41] Luenberger, D. G., "An introduction to observers," *IEEE Trans. Automat. Control*, Vol. AC-16, No. 6, 1971, pp. 596–602.

[42] Vidyasagar, M., *Nonlinear systems analysis*, Vol. 42 of *Classics in Applied Mathematics*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2002, Reprint of the second (1993) edition.

[43] Khalil, H. K., *Nonlinear systems*, Prentice-Hall, Inc., Upper Saddle River, NJ, 3rd ed., 2002.

[44] Sastry, S., *Nonlinear systems*, Vol. 10 of *Interdisciplinary Applied Mathematics*, Springer-Verlag, New York, 1999, Analysis, stability, and control.

[45] Hahn, W., *Stability of motion*, Translated from the German manuscript by Arne P. Baartz. Die Grundlehren der mathematischen Wissenschaften, Band 138, Springer-Verlag New York, Inc., New York, 1967.

[46] Kurzweil, J., "On the inversion of Lyapunov's second theorem on stability of motion," *Ann. Math. Soc. Transl.*, Vol. Ser. 2, No. 24, 1956, pp. 19–77.

[47] Massera, J. L., "Contributions to stability theory," *Ann. of Math. (2)*, Vol. 64, 1956, pp. 182–206.

[48] Lohmiller, W. and Slotine, J., "On contraction analysis for non-linear systems," *Automatica J. IFAC*, Vol. 34, No. 6, 1998, pp. 683–696.

[49] Mazenc, F., Praly, L., and Dayawansa, W. P., "Global stabilization by output feedback: examples and counterexamples," *Systems Control Lett.*, Vol. 23, No. 2, 1994, pp. 119–125.

[50] Besançon, G., "A viewpoint on observability and observer design for nonlinear systems," *New directions in nonlinear observer design (Geiranger Fjord, 1999)*, Vol. 244 of *Lecture Notes in Control and Inform. Sci.*, Springer, London, 1999, pp. 3–22.

[51] Sussmann, H. J. and Kokotović, P. V., "The peaking phenomenon and the global stabilization of nonlinear systems," *IEEE Trans. Automat. Control*, Vol. 36, No. 4, 1991, pp. 424–440.

[52] Hermann, R. and Krener, A. J., "Nonlinear controllability and observability," *IEEE Trans. Automat. Control*, Vol. AC-22, No. 5, 1977, pp. 728–740.

[53] Anisi, D. A., *Optimal motion control of a ground vehicle*, Master's thesis, Royal Institute of Technology (KTH), Stockholm, Sweden, 2003.

[54] Xia, X. and Zeitz, M., "On nonlinear continuous observers," *International Journal of Control*, Vol. 66, No. 6, 1997, pp. 943–954.

[55] Xia, X. and Gao, W., "On exponential observers for nonlinear systems," *Systems and Control Letters*, Vol. 11, 1988, pp. 319–325.

[56] Krener, A. J., "Nonlinear stabilizability and detectability," *Systems and networks: mathematical theory and applications, Vol. I (Regensburg, 1993)*, Vol. 77 of *Math. Res.*, Akademie-Verlag, Berlin, 1994, pp. 231–250.

[57] Marino, R., Santosuosso, G. L., and Tomei, P., "Robust adaptive observers for nonlinear systems with bounded disturbances," *IEEE Trans. Automat. Control*, Vol. 46, No. 6, 2001, pp. 967–972.

[58] Bestle, D. and Zeitz, M., "Canonical form observer design for non-Linear time-variable systems," *Int. J. Control*, Vol. 38, No. 2, 1983, pp. 419–431.

[59] Krener, A. J. and Isidori, A., "Linearization by output injection and nonlinear observers," *Systems & Control Letters*, Vol. 3, No. 1, 1983, pp. 47–52.

[60] Krener, A. J. and Respondek, W., "Nonlinear observers with linearizable error dynamics," *SIAM J. Control Optim.*, Vol. 23, No. 2, 1985, pp. 197–216.

[61] Hammouri, H. and Gauthier, J.-P., "Bilinearization up to output injection," *Systems & Control Letters*, Vol. 11, No. 2, 1988, pp. 139–149.

[62] Levine, J. and Marino, R., "Nonlinear system immersion, observers and finite-dimensional filters," *Systems & Control Letters*, Vol. 7, No. 2, 1986, pp. 133–142.

[63] Back, J. and Seo, J. H., "Immersion of non-linear systems into linear systems up to output injection: characteristic equation approach," *Internat. J. Control*, Vol. 77, No. 8, 2004, pp. 723–734.

[64] Keller, H., "Nonlinear observer design by transformation into a generalized observer canonical form," *Internat. J. Control*, Vol. 46, No. 6, 1987, pp. 1915–1930.

[65] Kazantzis, N. and Kravaris, C., "Nonlinear observer design using Lyapunov's auxiliary theorem," *Systems & Control Letters*, Vol. 34, No. 5, 1998, pp. 241–247.

[66] Krener, A. J. and Xiao, M., "Nonlinear observer design in the Siegel domain," *SIAM J. Control Optim.*, Vol. 41, No. 3, 2002, pp. 932–953 (electronic).

[67] Krener, A. J. and Xiao, M., "Erratum: "Nonlinear observer design in the Siegel domain" [SIAM J. Control Optim. **41** (2002), no. 3, 932–953 (electronic);]," *SIAM J. Control Optim.*, Vol. 43, No. 1, 2004, pp. 377–378 (electronic).

[68] Hu, X., "On state observers for nonlinear systems," *Systems & Control Letters*, Vol. 17, No. 6, 1991, pp. 465–473.

[69] Goodrich, M. A., Stirling, W. C., and Frost, R. L., "A theory of satisficing decisions and control," *IEEE Transactions on Systems, Man and Cybernetics, Part A*, Vol. 28, No. 6, Nov. 1998, pp. 763–779.

[70] Curtis, J. W. and Beard, R. W., "Satisficing: a new approach to constructive nonlinear control," *IEEE Trans. Automat. Control*, Vol. 49, No. 7, 2004, pp. 1090–1102.

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

[72] Mettler, B. and Bachelder, E., "Combining on- and offline optimization techniques for efficient autonomous vehicle's trajectory Planning," *Proc. of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, San Francisco, CA, USA, Aug. 2005.

[73] Ferguson, D. and Stentz, A., "The Delayed D* algorithm for efficient path replanning," *Proceedings of the IEEE International Conference on Robotics and Automation*, April 2005.

[74] Olfati-Saber, R., Fax, J. A., and Murray, R. M., "Consensus and cooperation in networked multi-agent systems," Proceedings of the IEEE, Invited Paper, Jan. 2007.

[75] Jadbabaie, A., Lin, J., and Morse, A. S., "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Trans. Automat. Control*, Vol. 48, No. 6, 2003, pp. 988–1001.

[76] Fax, J. A. and Murray, R. M., "Information flow and cooperative control of vehicle formations," *IEEE Trans. Automat. Control*, Vol. 49, No. 9, 2004, pp. 1465–1476.

[77] Tornambé, A., "Use of asymptotic observers having high-gains in the state and parameter estimation," *Proc. of the* 28^{th} *IEEE Conference on Decision and Control*, Dec. 1989, pp. 1791–1794.

[78] Khalil, H. K., "High-gain observers in nonlinear feedback control," *Lecture Notes in Control and Information Sciences*, edited by H. Nijmeijer and T. Fossen, Vol. 244, Springer Verlag, 1999, pp. 249–268.

[79] Lin, W., Baillieul, J., and Bloch, A., "Call for papers for the special issue on new directions in nonlinear control," *IEEE Trans. Automat. Control*, Vol. 47, No. 3, 2002, pp. 543–544.

[80] Hamberg, J., "Controlled lagrangians, symmetries and conditions for strong matching," *Lagrangian and Hamiltonian methods for nonlinear control*, edited by N. Leonard and R. Ortega, Elsevier, 2000, pp. 62–67.

[81] Aghannan, N. and Rouchon, P., "An intrinsic observer for a class of Lagrangian systems," *IEEE Trans. Automat. Control*, Vol. 48, No. 6, 2003, pp. 936–945.

[82] Maithripala, D. H. S., Dayawansa, W. P., and Berg, J. M., "Intrinsic observer-based stabilization for simple mechanical systems on Lie groups," *SIAM J. Control Optim.*, Vol. 44, No. 5, 2005, pp. 1691–1711 (electronic).

[83] Maithripala, D. H. S., Berg, J. M., and Dayawansa, W. P., "Almost-global tracking of simple mechanical systems on a general class of Lie groups," *IEEE Trans. Automat. Control*, Vol. 51, No. 2, 2006, pp. 216–225.

Paper A

Online Trajectory Planning for Aerial Vehicles: a Safe Approach with Guaranteed Task Completion

Online Trajectory Planning for Aerial Vehicles: a Safe Approach with Guaranteed Task Completion

David A. Anisi, John W.C. Robinson and Petter Ögren

Abstract

On-line trajectory optimization in the three dimensional space is the main topic of the paper at hand. The high-level framework augments on-line receding horizon control with an off-line computed terminal cost that captures the global characteristics of the environment, as well as any possible mission objectives. The first part of the paper is devoted to the single vehicle case while the second part considers the problem of simultaneous arrival of multiple aerial vehicles. The main contribution of the first part is two-fold. Firstly, by augmenting a so called safety maneuver at the end of the planned trajectory, this paper extends previous results by addressing provable safety properties in a 3D setting. Secondly, assuming initial feasibility, the planning method presented is shown to have finite time task completion. Moreover, a quantitative comparison between the two competing objectives of optimality and computational tractability is made. Finally, some other key characteristics of the trajectory planner, such as ability to minimize threat exposure and robustness, are highlighted through simulations. As for the simultaneous arrival problem considered in the second part, by using a time-scale separation principle, we are able to adopt standard Laplacian control to a consensus problem which is neither unconstrained, nor first order.

Keywords: On-line trajectory optimization, Computational Optimal Control, Mission Uncertainty, Trajectory Re-planning, Safety, Task Completion, Consensus Problem, Simultaneous Arrival.

A.1 Introduction

In this paper, on-line trajectory planning for an aerial vehicle subject to simultaneous kinematic and dynamic constraints is considered. The trajectory planning problem is formulated as a somewhat modified Optimal Control Problem (OCP), where optimality can be dictated by a mixture of conditions and penalties relating to time- and energy efficiency, threat avoidance, stealth behavior and various end-point constraints imposed by the target-seeker and/or terrain. An underlying assumption however, is that due to imperfect information, the kinematic constraints, as well as the location of the target set and possible threats, might change during the course of flight. Consequently, we can not use the family of techniques that rely on off-line generation of a trajectory database for on-line interrogation [1–3]. Also, assuming the problem originates from a complex real-world application, the existence of analytical solutions is unlikely; thus we seek fast computational algorithms for solving the OCP.

Background and Solution Foundation

In late 80's, by extending their "free path encoding method" [4], Canny and Reid demonstrated the $\mathcal{NP}$ - hardness of finding a shortest kinodynamic path for a point moving amidst polyhedral obstacles in a three dimensional environment [5]. Therefore, in order to meet the on-line computational requirement, attention must be paid to *approximation methods*, *i.e.* computationally efficient algorithms that compute kinodynamically feasible trajectories that are "near-optimal" in some sense.

One way to pursue, is to re-evaluate our notion of optimality. If finding a truly optimal solution is intractable, one might instead consider finding *pareto-optimal* solutions, where the computational load is considered as one of the objective functions to be minimized (*cf.* [6]). Pareto-optimality is a concept of multiple objective decision making [7] and, intuitively, a pareto-optimal solution is one in which no other solution can improve one objective without a simultaneous deterioration of at least one of the others. It should however be noted that in our case, not every pareto-optimal solution will be regarded as feasible. If just augmenting the objective function from the original OCP with a quantification of the computational load, one pareto-optimal solution would be not to optimize the trajectory at all. To see this, notice that the computational load would then be zero and thus any effort to improve the quality of the trajectory would require increased computational load. In particular, such a trivially pareto-optimal trajectory, would typically *not* end in the target set.

To set this right, it must be noted that although formulated as an OCP, finding a provably collision free path that is guaranteed to end in the target set must be given higher *priority* than the optimality properties thereof. This diversification, or ranking, of our objectives is quite natural since the optimal control formulation can be considered as a *tool* for choosing one single input in the set of controls that fulfill our minimum requirements[1], which in our particular case will be to generate collision free paths that lead us to the target set. A collision free vehicle path is called *safe*, while reaching the target set will hence-forth be referred to as *task completion.* As will be shown, the controller design of this paper have both provable safety properties, as well as a guaranteed finite time task completion.

The line of thought presented in this paper merges that of point-wise satisficing control [9, 10] and Receding Horizon Control (RHC) or Model Predictive Control (MPC) [11]. In RHC/MPC, the doubtful viability of long term optimization under uncertain conditions is adhered, so that instead of solving the OCP on the full interval, one repeatedly solves it on the interval $[t_c, t_c + T_p]$ instead. Here t_c denotes the current time instance and T_p is the planning horizon. Upon applying the first control element, measuring the obtained state and moving the time interval forward in time, the optimization step is iteratively performed. This closes the loop and obtains a certain robustness against modelling errors or disturbances. Unfortunately, it is known that in the absence of particular precautions,

[1] Any solution meeting the minimum requirements is called *satisficing* [8]. For a more control-oriented presentation of the concept, consult [9, 10].

closed-loop stability[2] cannot be assured. In principle however, this issue is relatively easily tackled by introducing stability terms or constraints. In the literature (see *e.g.* [11]), several approaches to ensure stability of RHC/MPC schemes can be found, one of the most appealing approaches being that of utilizing a constrained control Lyapunov function (CLF) as terminal cost [12, 13]. As a matter of fact, it has been shown in [14], that OCP, RHC as well as the set of CLF-based continuous control designs (such as Sontag's universal formula [15], Freeman and Kokotović's min-norm formula [16], but also the satisficing control methods [9,10]) can be viewed in a unified manner; namely that OCP and CLF-based methods are the two limiting cases of RHC when the planning horizon goes to infinity and zero respectively (see Figure A.1). This enables us to adopt a *horizon independent* point of view, where the length of the planning horizon, T_p, is determined based on accuracy demands and computational resources, while inherently more global properties (such as stability or task completion) are handled by the monotonicity properties of the composite cost.

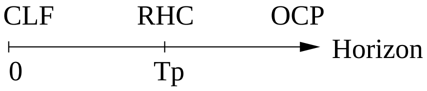


Figure A.1: A unifying view on OCP, RHC and the CLF-based control methods.

A.1.1 Related work

Task completion has been previously considered by Richards and How [17, 18]. By augmenting the system with a binary "target state", that indicates whether the target set is reached or not, the authors end up with a hybrid system at hand. Task completion is then guaranteed by imposing a hard terminal equality constraint on the target state which restricts the trajectory candidates to those that end up in the terminal set at the end of the planning horizon. Although intuitively appealing, this is indeed a very restrictive and computationally demanding constraint that beside the introducement of binary states, require needlessly long planning horizons. In addition, early termination of the optimization routine may cause violation of the equality constraint and consequently jeopardize the task completion objective. The alternative solution outlined in this paper, argues that requiring monotonicity of the composite cost is sufficient to obtain the desired task completion property. This decouples the length of the planning horizon from task completion and thus allows us to choose the planning horizon only taking computational resources and real-time constraints into account.

Regarding safety concerns, reference should be made to the recent works of Ögren and Leonard [19], Schouwenaars *et. al* [20] and Kuwata and How [21,22], who consider safe RHC of autonomous vehicles. These approaches are however tailored for a 2D setting. The last two references further utilize visibility graphs for environmental representation. Although visibility graph based methods are effective in urban terrain (where the buildings provide neat blocks and corner points to exploit), they are not suitable for the mountain terrain data used in this paper. In addition to regarding a 3D problem, our work shows that it is possible to address both safety and task completion issues in a unified manner without introducing any integer variables. It also differs from the mentioned papers by the fact that

[2] Standard RHC/MPC is tailored for steady-state control or asymptotic stabilization to the origin in the Lyapunov sense. The notion of *task completion* considered in this paper is a *different problem*, namely it aims at controlling the plant into a target set which is not necessarily control invariant or contain any equilibrium points.

in our case, safety also renders task completion possible. This is simply due to an elaborate choice of the so called "set of safe states", in which the safety maneuver is to end.

The general framework utilized in this paper for path planning in the three dimensional space, is reminiscent of that presented in [23]. In both papers, the global characteristics of the environment and mission objectives are captured in a functional, calculated off-line and passed to the on-line receding horizon controller as a terminal cost. However, safety and task completion concerns are the pivotal differences between these two papers. This is also a convenient point at which to mention that the possibility of updating the "off-line" computed terminal cost should not be overlooked. As pointed out in [23], the term "off-line" is rather to be interpreted as, at a much slower sampling rate than the control loop, *i.e.* in the order of tens of seconds. As new information about the environment or mission objectives is gathered when the mission unfolds, it can be processed and fed back regularly to the vehicle through an updated terminal cost, as discussed in *e.g.* [24].

Simultaneous arrival of multiple vehicles is an application of the wider class of consensus problems. Standard results in this field involve consensus problems described by unconstrained, scalar (or fully actuated) first order linear systems (see *e.g.* [25–27]). In our case however, the relation between the control (*i.e.* the vehicle acceleration), and the consensus quantity (which is the Estimated Time to Arrival, ETA) is neither unconstrained nor first order. However, by using a time-scale separation principle, we are able to achieve consensus by using standard Laplacian control. In a non receding horizon setting, synchronization of timing-critical missions has been considered in a number of papers, including [28, 29].

This paper is organized as follows; Section A.2 introduces the trajectory optimization problem as well as some basic terminology used in this paper. Section A.3 presents the problem formulation and clarifies the relation between the original OCP and the transcription thereof into a finite-dimensional nonlinear mathematical programming problem (NLP). Subsequently, the fundamental role of the safety maneuver and the implication of it on task completion are discussed in Section A.4. Details on environment representation and terminal cost computation can be found in Section A.5. In Section A.6, a quantitative comparison between the two competing objectives of optimality and computational tractability is made for the problem at hand. Section A.7 presents a small selection of the simulations made with the proposed trajectory planning algorithm while Section A.8 extends these results to the multi-vehicle case. Finally, Section A.9 concludes this paper with some expository remarks.

A.2 Preliminaries

In this section, we review some standard background material and present the used terminology. Consequently, the more familiar reader may prefer to skip it at a first reading. Consider the following trajectory optimization or optimal control problem (OCP):

$$\begin{aligned} \underset{u}{\text{minimize}} \quad & \int_{t_c}^{T} \mathcal{L}(x,u)\mathrm{d}t \\ \text{s.t.} \quad & \dot{x} = f(x) + g(x)u \\ & d(x,u) \leq 0 \\ & x(t_c) = x_c \in \mathbb{R}^n \\ & x(T) \in S_f \subseteq \mathbb{R}^n, \end{aligned} \tag{A.1}$$

where the state $x(t) \in \mathcal{X} \subseteq \mathbb{R}^n$, the control $u(t) \in \mathcal{U} \subseteq \mathbb{R}^m$, the constraints $d : \mathcal{X} \times \mathcal{U} \to \mathbb{R}^p$, the terminal time T is a possibly *free* variable, and t_c can be read as the "current" or "considered" time. All mappings are assumed to be sufficiently smooth and the dynamical system complete. To further unburden the discussion, we make a standing assumption throughout the paper, that all stated minimization problems with respect to u are well–posed and that the minimum is attained.

Optimal Control Problem

We consider henceforth only *feasible* state trajectories x, *i.e.* state histories such that all conditions in (A.1) are met for all times t and the closed target set S_f is reached in finite time $T \in [0, \infty)$. Define $\bar{\mathcal{U}}$ as the set of control functions such that u remains in $\mathcal{U}$ at all times and a feasible state trajectory x is generated. Let the *value function* $J : \mathcal{X} \times \bar{\mathcal{U}} \mapsto [0, \infty)$ be defined by

$$J(x(t_c), u) = \int_{t_c}^{T} \mathcal{L}(x(t), u(t))\, \mathrm{d}t, \quad t_c \in [0, T],$$

that is the cost to go from $x(t_c)$ to S_f using a control u. We assume that an optimum to (A.1) always exists and define accordingly the *optimal value function* $J^* : \mathcal{X} \mapsto [0, \infty)$ as

$$J^*(x(t_c)) = \min_{u \in \bar{\mathcal{U}}} \int_{t_c}^{T} \mathcal{L}(x(t), u(t))\, \mathrm{d}t, \quad t_c \in [0, T],$$

which thus represents the optimal cost to go from $x(t_c)$ to S_f. Without loss of generality we may assume that $\mathcal{L}(x(t_c), u(t_c)) = 0$ if and only if $x(t_c) \in S_f$ so that $J^*(x(t_c)) = 0$ is equivalent to $x(t_c) \in S_f$.

Receding Horizon Control

In the receding horizon setting, we consider a slightly modified OCP where the *planning horizon* $T_p \in [0, T - t_c]$ is given relative to some $t_c \in [0, T]$, and the *discarded optimal tail cost*, defined as

$$\min_{u \in \bar{\mathcal{U}}} \int_{t_c + T_p}^{T} \mathcal{L}(x(t), u(t))\, \mathrm{d}t,$$

is upper bounded by some *terminal cost*, $\Psi : \mathcal{X} \mapsto [0, \infty)$. In other words, we have that

$$\Psi(x(t_c + T_p)) \geq \min_{u \in \bar{\mathcal{U}}} \int_{t_c + T_p}^{T} \mathcal{L}(x(t), u(t))\, \mathrm{d}t.$$

The terminal cost, $\Psi(x(t_c + T_p))$, is meant to act as a more easily computed (but conservative) approximation to the optimal cost to go from $x(t_c + T_p)$ to S_f. With the aid of the terminal cost, the *composite cost* $\tilde{J} : \mathcal{X} \times \bar{\mathcal{U}} \mapsto [0, \infty)$, is defined as

$$\tilde{J}(x(t_c), u) \int_{t_c}^{t_c + T_p} \mathcal{L}(x(t), u(t))\, \mathrm{d}t + \Psi(x(t_c + T_p)).$$

Since the terminal cost acts as an upper bound on the discarded optimal tail cost we have

$$\begin{aligned} \tilde{J}(x(t_c), u) &= \int_{t_c}^{t_c + T_p} \mathcal{L}(x(t), u(t))\, \mathrm{d}t + \Psi(x(t_c + T_p)) \\ &\geq \int_{t_c}^{t_c + T_p} \mathcal{L}(x(t), u(t))\, \mathrm{d}t + \min_{u \in \bar{\mathcal{U}}} \int_{t_c + T_p}^{T} \mathcal{L}(x(t), u(t))\, \mathrm{d}t \\ &\geq \min_{u \in \bar{\mathcal{U}}} \int_{t_c}^{T} \mathcal{L}(x(t), u(t))\, \mathrm{d}t = J^*(x(t_c)), \end{aligned}$$

that is, the composite cost provides an upper bound on the optimal value function. Now, *if* the control u can be chosen such that $\tilde{J}$ is monotonically decreasing along a feasible trajectory, then $\tilde{J}$ acts as a Lyapunov–like function for the receding horizon controlled system. This will, together with a sandwiching argument, guarantee task completion for the original OCP, *i.e.* $J^*(x(T)) = 0$ for some $T < \infty$.
Here, one of the key issues that motivate the present work reveals itself. Namely, since Ψ is merely an approximation to the optimal cost to go, it will in general *not* be possible to find a control u that both satisfies all the constraints in (A.1), and cause a decrease in $\tilde{J}$ (unless Ψ is chosen as a constrained CLF, which will not be assumed in this paper). Using the terminology of [9, 10], the set of satisficing controls might be *empty*. In these cases, a reserve plan (namely using the augmented safety maneuver) turns out to be a useful tool for ensuring both safety and task completion.

Ideally, $\Psi(x(t_c + T_p))$ should be chosen to equal the optimal value function,

$$J^*(x(t_c + T_p)) = \min_{u \in \mathcal{U}} \int_{t_c + T_p}^{T} \mathcal{L}(x, u) \mathrm{d}t,$$

which can be found by solving the Hamilton-Jacobi-Bellman partial differential equation (HJBE); a highly nontrivial task. If one would be able to solve HJBE for J^* and set the terminal cost equal to it, then the RHC scheme would coincide with that of solving the OCP on the full time interval, $[t_c,\ T]$.

A.3 Problem Formulation

We are now ready to state our trajectory planning problem. The aerial vehicle is modeled as a unit-mass point in $\mathbb{R}^3$ with bounded velocity and acceleration. The equations of motion are those of Newtonian mechanics so that the control inputs are the applied forces or accelerations (by virtue of the unit-mass assumption). Let $p = [x\ y\ z]^T \in \mathbb{R}^3$ denote the position of the vehicle in its work-space, $\mathbb{R}^3$, while $v = \dot{p} = [v_x\ v_y\ v_z]^T$ denotes its velocity. Let further

$$\mathcal{X} = \{(p, v) : p \in \mathbb{R}^3, v \in \mathbb{R}^3\},$$

denote the state-space. A state $s = (p, v) \in \mathcal{X}$, is a position-velocity-tuple and hence $\mathcal{X}$ is isomorphic to $\mathbb{R}^6$. Designing the aerial vehicle controller, we would like to meet the following objectives:

- Avoid ground collision
- Arrive at the target set
- Compute controls in real time
- Allow for information updates
- Use small control effort
- Achieve a short time of flight
- Achieve low threat and radar exposure

Ideally, we would like to formally guarantee the first two items, satisfy the following two and minimize an objective function composed of the last three. Our primary objective is however to find a provably collision free trajectory from a specified initial vehicle position and velocity, *i.e.* from a given state $s_i = (p_i, v_i) \in \mathcal{X}$, that reaches a final vehicle position with an *arbitrary* final velocity. Hence, the target set can be written as

$$S_f = \{(p, v) \in \mathcal{X} : p = p_f\},$$

where p_f is given. With this diversification, or ranking, of our objectives, we are naturally lead to consider approximative and near-optimal solutions since solving the problem to optimality turns out to be computationally intractable, contradicting our third listed objective.

Remark A.1. Note that S_f will in general not be an invariant set. Standard RHC/MPC is tailored for steady-state control or asymptotic stabilization to the origin in the Lyapunov sense. The notion of *task completion* considered in this paper is a *different problem*, namely controlling the plant into a target set which is not necessarily control invariant or contain any equilibrium points.

Problem Transcription

For the actual design of the receding horizon controller, the OCP (A.1), which is an infinite-dimensional problem of choosing a control function in a given space, has to be *transcribed* into a finite-dimensional parameter selection problem. To this end, uniform temporal discretization is performed and the differential operator is approximated with a zero-order sampled dynamic model. Although it is possible to adopt higher order quadrature rules [30], a straightforward Riemannian sum is used to approximate the integral cost. The objective will be to minimize a combination of the L_2-norm of the applied control and the task completion time. Other types of mission objectives, such as low threat exposure, may be incorporated by the terminal cost. On top of the kinematic constraints, the dynamic constraints we consider are hard L_∞-norm constraints on both applied acceleration and vehicle velocity. Then, at each time-step, $k \in \mathbb{N} = \{1, 2, \dots\}$, upon measuring the current state, (p_c, v_c), the receding horizon control action, $a^*_{k,1}$ is computed. Here, the *control sequence* $a^*_k = [a^*_{k,1}, \cdots, a^*_{k,N}] \in \mathbb{R}^{3N}$, denotes a solution to the finite-dimensional optimal control problem[3];

$$
\begin{aligned}
\underset{a_k}{\text{minimize}} \quad & \sum_{i=1}^{N-1} h(\|a_{k,i}\|_2^2 + \beta) + \Psi(p_{k,N}) && && \text{(A.2)}\\
\text{s.t.} \quad & p_{k,i+1} = p_{k,i} + h\, v_{k,i} && i = 1, \cdots, N-1 \\
& v_{k,i+1} = v_{k,i} + h\, a_{k,i} && i = 1, \cdots, N-1 \\
& d(p_{k,i}) \leq 0 && i = 1, \cdots, N \\
& \|v_{k,i}\|_\infty \leq v_{\max} && i = 1, \cdots, N \\
& \|a_{k,i}\|_\infty \leq a_{\max} && i = 1, \cdots, N \\
& p_{k,1} = p_c \\
& v_{k,1} = v_c \\
& a_{k,N} \in S_\varepsilon(s_{k,N}),
\end{aligned}
$$

where $h > 0$ is the sampling interval and the design parameter $\beta > 0$ determines the relative importance between time-optimality and energy efficiency. In addition, the set $S_\varepsilon(s_{k,N})$ is defined as follows.

Definition A.1 (Target approaching controls)**.** *The set of target approaching controls, $S_\varepsilon(s_{k,i})$, is defined to be the subset of control values in $\mathcal{U}$, such that*

$$\Big[\Psi(p_{k,i} + hv_{k,i}) - \Psi(p_{k,i})\Big] + h\mathcal{L}(s_{k,i}, a_{k,i}) \leq -\varepsilon.$$

[3] Mind the difference between the time step index, $k \in \mathbb{N}$, and the index used in the RHC sequence, $i \in \{1, \cdots, N\}$.

Remark A.2. An utmost important question, involves the non–emptiness of the set of target approaching controls. Under what conditions can we assure that S_ε is nonempty and what can be done in the case of infeasibility? It can be shown (*cf.* [10]) that if Ψ is chosen as a constrained CLF and the stage cost, $\mathcal{L}$, is convex in the control variable, then S_ε is a convex, and hence connected set. The parameter ε, determines the size of this set and if fulfilling a simple inequality constraint, always turns S_ε nonempty. However, because of the apparent difficulties with choosing Ψ as a CLF, this work will recognize the existence of cases when S_ε turns out to be empty. The augmented safety maneuver, which will be introduced properly in Section A.4, provides a solution to this infeasibility problem.

The motivation behind the nomenclature used for S_ε, will be clear from the proposition to follow.
Let $a_k = [a_{k,1}, \cdots, a_{k,N}] \in \mathbb{R}^{3N}$ denote the control sequence used at time step k. Let further

$$a_{k+1} = \overleftarrow{T}(a_k, \star) = [a_{k,2}, \cdots, a_{k,N}, \star] \in \mathbb{R}^{3N},$$

denote the control sequence used at time step $k+1$. Here, $\overleftarrow{T}$ denotes a left shift operator and the star symbol, $\star$, denotes an arbitrary control element. These control sequences, give rise to the trajectories

$$\begin{cases} p_k = [p_{k,1}, \cdots, p_{k,N}] \\ v_k = [v_{k,1}, \cdots, v_{k,N}] \end{cases} \quad \text{and}$$

$$\begin{cases} p_{k+1} = \overleftarrow{T}(p_k, p_{k,N} + hv_{k,N}) = [p_{k,2}, \cdots, p_{k,N}, p_{k,N} + hv_{k,N}] \\ v_{k+1} = \overleftarrow{T}(v_k, v_{k,N} + ha_{k,N}) = [v_{k,2}, \cdots, v_{k,N}, v_{k,N} + ha_{k,N}] \end{cases}$$

respectively.

Proposition A.1 (Target set approaching). *If choosing $a_{k,N} \in S_\varepsilon(s_{k,N})$, then*

$$\tilde{J}(p_{k+1,1}, a_{k+1}) < \tilde{J}(p_{k,1}, a_k),$$

that is, the composite cost at the next time step has been reduced, hence the target set is approaching.

Proof.

$$\begin{aligned}
\tilde{J}(p_{k+1,1}, a_{k+1}) &= \sum_{i=1}^{N-1} h\mathcal{L}(s_{k+1,i}, a_{k+1,i}) + \Psi(p_{k+1,N}) \\
&= \sum_{i=2}^{N} h\mathcal{L}(s_{k,i}, a_{k,i}) + \Psi(p_{k,N}) - \Psi(p_{k,N}) + \Psi(p_{k,N} + hv_{k,N}) \\
&\quad + h\mathcal{L}(s_{k,1}, a_{k,1}) - h\mathcal{L}(s_{k,1}, a_{k,1}) \\
&= \sum_{i=1}^{N-1} h\mathcal{L}(s_{k,i}, a_{k,i}) + \Psi(p_{k,N}) - \Psi(p_{k,N}) + \Psi(p_{k,N} + hv_{k,N}) \\
&\quad + h\mathcal{L}(s_{k,N}, a_{k,N}) - h\mathcal{L}(s_{k,1}, a_{k,1}) \\
&\leq \tilde{J}(p_{k,1}, a_k) - \varepsilon - h\mathcal{L}(s_{k,1}, a_{k,1}) < \tilde{J}(p_{k,1}, a_k). \quad \square
\end{aligned}$$

Remark A.3. Although insignificant for the proof of this particular proposition, recursive use of this result (in order to achieve target set *reaching*) requires

$$\star = a_{k+1,N} \in S_\varepsilon(s_{k+1,N}).$$

For the sake of completeness and to make the effect of the above described transcription clear, if we define a parameter vector,

$$\xi^k = [p_{k,1}, \cdots, p_{k,N},\ v_{k,1}, \cdots, v_{k,N},\ a_{k,1}, \cdots, a_{k,N}]^T \in \mathbb{R}^{9N},$$

an objective function,

$$\mathcal{F}(\xi^k) = \sum_{i=1}^{N-1} h(\|a_{k,i}\|_2^2 + \beta) + \Psi(p_{k,N}),$$

and constraint functions, $\mathcal{G}^1(\xi^k)$ and $\mathcal{G}^2(\xi^k)$, according to

$$\begin{aligned} \mathcal{G}^1(\xi^k) &= \begin{bmatrix} p_{k,i+1} - p_{k,i} - h\ v_{k,i} \\ v_{k,i+1} - v_{k,i} - h\ a_{k,i} \\ p_{k,1} - p_c \\ v_{k,1} - v_c \end{bmatrix} \\ \mathcal{G}^2(\xi^k) &= \begin{bmatrix} d(p_{k,i}) \\ \|v_{k,i}\|_\infty - v_{\max} \\ \|a_{k,i}\|_\infty - a_{\max} \\ h(\|a_{k,N}\|_2^2 + \beta) + [\Psi(p_{k,N} + h\ v_{k,N}) - \Psi(p_{k,N})] + \varepsilon \end{bmatrix}, \end{aligned}$$

then (A.2) can be written as

$$\begin{aligned} \underset{\xi^k}{\text{minimize}} \quad & \mathcal{F}(\xi^k) \\ \text{s.t.} \quad & \mathcal{G}^1(\xi^k) = 0 \\ & \mathcal{G}^2(\xi^k) \leq 0 \end{aligned} \tag{A.3}$$

which is the constrained NLP that needs to be solved on-line at each time-step, k.

A.4 The Safety Maneuver and Task Completion

This section provides an alternative whenever the NLP solver fails to provide a solution to (A.2). This infeasibility may have different sources, including;

- The set of target approaching controls being empty (as remarked upon in connection with Definition A.1)
- Since the terminal cost is most often calculated from a graph representation of the environment, it might lead to paths that turn out to be dynamically infeasible in the future
- Various optimization routine failures including non–convergence and abnormal termination

Remark A.4. Optimization routine failures will always be present regardless the choice of Ψ, and may therefore not be neglected in any case.

Previously, in a 2D setting, Schouwenaars *et. al* [20] and Kuwata and How [21, 22], have considered safe RHC of autonomous vehicles. By constraining the computed path at each time-step to end on either a right, or a left turning collision free circle, where the vehicle can safely remain for an indefinite period of time (or at least until it runs out of fuel), the first mentioned authors account for safety. The authors of the latter mentioned papers make use of three circles to smoothen out all the corners of the straight line segments in the visibility graph, modify the cost map and thereby incorporate vehicle dynamics in the terminal cost. To the contrary, our work considers path planning in 3D space and further differs from previous results, by the fact that in our case, safety also renders task completion possible. This is simply due to an elaborate choice of the so called set of safe states, in which the safety maneuver is to end. We argue that a safe state should be chosen such that there are more options left than just aimlessly lingering in a loiter pattern. The choice made in this paper (vertically aligned vehicle), always leaves you with the possibility of continuing upward and thus, upon a full turn above the target point, p_f, renders task completion possible (see Figure A.2). The remaining of this section is devoted to show how this simple observation underlies both the safety and task completions results, presented as Proposition A.2 and A.3 respectively.

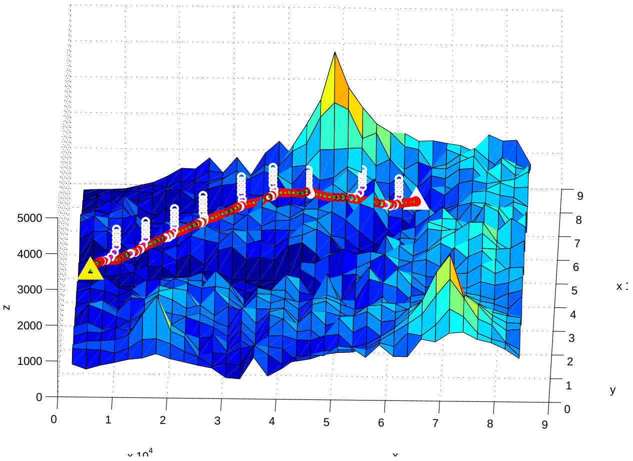


Figure A.2: The safety maneuver connects the planned trajectory with the set of safe states from where the existence of a safe path to the target set is known to exist. Here, to simplify exposition, the augmented safety maneuver is shown at every tenth time step.

Definition A.2 (Safe path). *A path,* $p_j = [x_j\ y_j\ z_j]^T, j \in \mathcal{I} \subset \mathbb{N}$, *is called safe if and only if its vertical elements,* z_j, *are all located above the terrain surface,* i.e.

$$d(p_j) \triangleq H(x_j, y_j) + h_{min} - z_j \leq 0, \quad \forall\, j \in \mathcal{I}.$$

Here, $H(x_j, y_j)$ *denotes the altitude of the mountain at the point* (x_j, y_j) *(as interrogated from a more detailed map than the one used for spatial decomposition), and* $h_{min} > 0$ *denotes the minimum clearance distance.*

Definition A.3 (Set of safe states). *A state,* $s = (p, v) \in \mathcal{X}$, *is called* safe *if there exists a safe, as well as dynamically feasible path linking* p *to the target set,* S_f. *For the particular choice of mountainous terrain, a distinguishable subset of the safe states is*

$$S_s = \{(p, v) \in \mathcal{X} :\ d(p) \leq 0,\ e_z^T v - \|v\| \sin \bar{\alpha} \geq 0\},$$

that is, the states where the vehicle is flying with a flight path angle larger than the maximal terrain inclination, $\bar{\alpha}$.

The way to ensure collision avoidance, lies in the construction of a so called "safety maneuver",

$$\begin{aligned}\breve{p}_k &= [\breve{p}_{k,1}, \cdots, \breve{p}_{k,\bar{N}}] \in \mathbb{R}^{3\bar{N}} \\ \breve{v}_k &= [\breve{v}_{k,1}, \cdots, \breve{v}_{k,\bar{N}}] \in \mathbb{R}^{3\bar{N}} \\ \breve{a}_k &= [\breve{a}_{k,1}, \cdots, \breve{a}_{k,\bar{N}}] \in \mathbb{R}^{3\bar{N}},\end{aligned}$$

that connects the optimized trajectory,

$$\begin{aligned}p^*_k &= [p^*_{k,1}, \cdots, p^*_{k,N}] \in \mathbb{R}^{3N} \\ v^*_k &= [v^*_{k,1}, \cdots, v^*_{k,N}] \in \mathbb{R}^{3N} \\ a^*_k &= [a^*_{k,1}, \cdots, a^*_{k,N}] \in \mathbb{R}^{3N},\end{aligned}$$

to S_s. In other words, we require $(\breve{p}_{k,\bar{N}}, \breve{v}_{k,\bar{N}}) \in S_s$. Here, $\bar{N}$ denotes the maximum number of steps needed to perform the safety maneuver. How to determine $\bar{N}$ will be discussed in more detail here-below. This approach can be thought of as increasing the number of nodes in the temporal discretization from N to $N + \bar{N}$. However, in order to be computationally efficient, the augmented part, *i.e.* the components of the safety maneuver, are predetermined and do not take part in the optimization process. The freedom in predetermining the safety maneuver is limited by the following set of constraints

$$\begin{aligned}&\breve{p}_{k,i+1} = \breve{p}_{k,i} + h\ \breve{v}_{k,i} && i = 1, \cdots, \bar{N} - 1 \\ &\breve{v}_{k,i+1} = \breve{v}_{k,i} + h\ \breve{a}_{k,i} && i = 1, \cdots, \bar{N} - 1 \\ &\breve{p}_{k,i}(z) \geq H(\breve{p}_{k,i}(x), \breve{p}_{k,i}(y)) + h_{\min} && i = 1, \cdots, \bar{N} \\ &\|\breve{v}_{k,i}\|_\infty \leq v_{\max} && i = 1, \cdots, \bar{N} \\ &\|\breve{a}_{k,i}\|_\infty \leq a_{\max} && i = 1, \cdots, \bar{N} \\ &(\breve{p}_{k,1}, \breve{v}_{k,1}) = (p^*_{k,N} + h\ v^*_{k,N}, v^*_{k,N} + h\ a^*_{k,N}) && \\ &(\breve{p}_{k,\bar{N}}, \breve{v}_{k,\bar{N}}) \in S_s, && \qquad (A.4)\end{aligned}$$

which restrict the safety maneuver to be a kinodynamically feasible trajectory for the aerial vehicle. In the following, the term *concatenated solution* refers to the augmentation of the optimized solution, (p^*_k, v^*_k, a^*_k), with the safety maneuver, $(\breve{p}_k, \breve{v}_k, \breve{a}_k)$.

Proposition A.2 (Safety). *Assume the existence of a feasible concatenated solution at time $k = 1$. If the safety maneuver is chosen such that it fulfills the constraints* (A.4)*, then there exists a safe path for all future time steps, $k \in \mathbb{N} \setminus 1$.*

Proof. We reason by mathematical induction. The initialization step is trivially fulfilled due to the assumption made in the proposition formulation. Next, assume the existence of a feasible concatenated solution at time step k. In particular, this assumption implies

$$(\breve{p}_{k,\bar{N}}, \breve{v}_{k,\bar{N}}) \in S_s. \qquad (A.5)$$

Remains to show the existence of a feasible concatenated solution at time step, $k + 1$. To this end, using the left shift operator, $\overleftarrow{T}$, we set

$$\begin{aligned}p_{k+1} &\triangleq \overleftarrow{T}(p^*_k, \breve{p}_{k,1}) &&= [p^*_{k,2}, \cdots, p^*_{k,N}, \breve{p}_{k,1}] \\ v_{k+1} &\triangleq \overleftarrow{T}(v^*_k, \breve{v}_{k,1}) &&= [v^*_{k,2}, \cdots, v^*_{k,N}, \breve{v}_{k,1}] \\ a_{k+1} &\triangleq \overleftarrow{T}(a^*_k, \breve{a}_{k,1}) &&= [a^*_{k,2}, \cdots, a^*_{k,N}, \breve{a}_{k,1}] \\ \breve{p}_{k+1} &\triangleq \overleftarrow{T}(\breve{p}_k, \breve{p}_{k,\bar{N}} + h\breve{v}_{k,\bar{N}}) &&= [\breve{p}_{k,2}, \cdots, \breve{p}_{k,\bar{N}}, \breve{p}_{k,\bar{N}} + h\breve{v}_{k,\bar{N}}] \\ \breve{v}_{k+1} &\triangleq \overleftarrow{T}(\breve{v}_k, \breve{v}_{k,\bar{N}}) &&= [\breve{v}_{k,2}, \cdots, \breve{v}_{k,\bar{N}}, \breve{v}_{k,\bar{N}}] \\ \breve{a}_{k+1} &\triangleq \overleftarrow{T}(\breve{a}_k, \bar{0}) &&= [\breve{a}_{k,2}, \cdots, \breve{a}_{k,\bar{N}}, \bar{0}].\end{aligned}$$

From the definition of $\breve{v}_{k+1}$ and Equation A.5 it follows that

$$e_z^T \breve{v}_{k+1,\bar{N}} \geq \|\breve{v}_{k+1,\bar{N}}\| \sin \bar{\alpha}. \tag{A.6}$$

If we let α denote the angle between the velocity vector $\breve{v}_{k,\bar{N}}$ and the horizontal plane (*i.e.* the flight path angle), the maximal terrain inclination implies

$$\begin{aligned} |H(\breve{p}_{k+1,\bar{N}}) - H(\breve{p}_{k,\bar{N}})| &\leq \tan \bar{\alpha} \cos \alpha \ \|\breve{p}_{k+1,\bar{N}} - \breve{p}_{k,\bar{N}}\| \\ &\leq \sin \bar{\alpha} \ \|\breve{p}_{k+1,\bar{N}} - \breve{p}_{k,\bar{N}}\|, \end{aligned}$$

which can be interpreted as a Lipschitz condition on the terrain surface and used to show that

$$\begin{aligned} &H(\breve{p}_{k+1,\bar{N}}) + h_{\min} \leq \\ &|H(\breve{p}_{k+1,\bar{N}}) - H(\breve{p}_{k,\bar{N}})| + |H(\breve{p}_{k,\bar{N}}) + h_{\min}| \leq \\ &\sin \bar{\alpha} \ \|\breve{p}_{k+1,\bar{N}} - \breve{p}_{k,\bar{N}}\| + H(\breve{p}_{k,\bar{N}}) + h_{\min} \leq \\ &\sin \alpha \ \|h \ \breve{v}_{k,\bar{N}}\| + \breve{p}_{k,\bar{N}}(z) = \breve{p}_{k+1,\bar{N}}(z). \end{aligned} \tag{A.7}$$

Combining Equations (A.6) and (A.7) yields

$$(\breve{p}_{k+1,\bar{N}}, \breve{v}_{k+1,\bar{N}}) \in S_s.$$

It is then straightforward to verify that $(\breve{p}_{k+1}, \breve{v}_{k+1}, \breve{a}_{k+1})$ fulfills the remaining constraints of (A.4) and thus, augmented with $(p_{k+1}, v_{k+1}, a_{k+1})$ serves as a feasible concatenated solution at time step $k+1$. □

Remark A.5. Mind the intentional digression from the notation used earlier; solution $(p_{k+1}, v_{k+1}, a_{k+1})$ is defined by the left shift operator, $\overleftarrow{T}$, and does not necessarily equal $(p^*_{k+1}, v^*_{k+1}, a^*_{k+1})$ - the output obtained from solving the NLP. This is an important point to make, since the essence of Proposition A.2 is that if the NLP output is missing or do not satisfy the feasibility constraints of (A.3) and (A.4) at the time step $k+1$, then one always have the choice of taking

$$(p^*_{k+1}, v^*_{k+1}, a^*_{k+1}) = (p_{k+1}, v_{k+1}, a_{k+1}),$$

that is, sticking with the demonstrably feasible solution obtained from the previous time step, k. In this view, the introduction of the safety maneuver also makes it possible to cope with *hard* real-time constraints; namely it provides an applicable option even when the NLP solver has not converged, or terminates abnormally.

Proposition A.3 (Task completion). *Assuming the existence of a feasible concatenated solution at time $k = 1$, the trajectory planner will generate safe paths that end in the target set, S_f. Hence, task completion is guaranteed.*

Proof. Let $\bar{k}$ denote the largest time step for which there exist control inputs $a_{m,N} \in S_{\varepsilon_m}(s_{m,N})$, giving rise to feasible concatenated solutions for all $m \leq \bar{k}$. Here, ε_m denotes the parameter used in time step m. From the initial assumption made, we have $\bar{k} \geq 1$. Now, if

$$\sum_{m=1}^{\bar{k}} \varepsilon_m + h\bar{k}\beta \geq \tilde{J}(p_{1,1}, a_1),$$

task completion follows directly by a recursive call on Proposition A.1 (*cf.* Remark A.3), the non-negativity of $\tilde{J}$ and a sandwiching argument. Else, by setting

$$(p^*_{\bar{k}+l}, v^*_{\bar{k}+l}, a^*_{\bar{k}+l}) = (p_{\bar{k}+l}, v_{\bar{k}+l}, a_{\bar{k}+l})$$

for $l = 1, \cdots, \bar{N}$, and iteratively adopting the argument used in the proof of Proposition A.2, we get

$$(p_{\bar{k}+\bar{N}+N,1}, v_{\bar{k}+\bar{N}+N,1}) \in S_s.$$

Task completion is then guaranteed due to Definition A.3; namely the existence of a kinodynamically feasible path connecting the points of S_s to the target set. □

Since $\bar{k}$ is ultimately determined by the terminal cost, Proposition A.3 also reveals the importance of choosing Ψ with a small degree of conservatism.

A.5 Environment Representation and Terminal Cost Computation

For environmental representation, real terrain elevation data over the Cascade mountains, WA, have been used (see Figure A.3). The dataset used is a subset extracted from the one appearing in Reference [31][4]. The full-resolution elevation image, is made up of $16,385 \times 16,385$ nodes at 10 meters horizontal spacing. The vertical resolution is 0.1 meters. This dataset occupies roughly 5 GB on disk and is therefore impractical to work with. However, as will be seen from the simulation results but also pointed out in [23], the environment should be decomposed in a manner that is consistent with the maneuvering capabilities of the vehicle. Therefore, this high level of accuracy is not needed to capture the global characteristics of the environment by the terminal cost, Ψ. The lower-resolution maps used in the simulations have therefore been sub-sampled at every 16th and 256th instance, resulting in a inter-pixel spacing of 160 and 2560 meters respectively. In the vertical direction, there are five horizontal layers with 600 meters in between. The vertical positions of each node depend on the altitude of the terrain at that particular point of the map. The non-uniform grid built this way, can be seen as stretching out the layers of a uniform grid on the terrain surface.

Upon this spatial-decomposition procedure, we end up with a grid having only $33 \times 33 \times 5$ nodes. Given the target node, $n_f = S_f$, the terminal cost at any node, $\Psi(n_i)$, is an approximation of the cost to go from n_i to n_f. Naturally, we have that $\Psi(n_f) = 0$. There is a cost of c_{ij} to be associated with a transition between node i and j. In the nominal case, this cost will be taken proportional to the Euclidean distance [5]. However, if there exist threat zones or other preferences regarding the path of the vehicle, these costs are readily modified to account for them as well. For instance, the transition cost is to be set to a higher value within the detection area of a known SAM-site or radar. The actual values of $\Psi(n_i)$ are calculated off-line using an implementation based on Dijkstra's algorithm, which returns the cheapest path (as defined by the costs, c_{ij}) from node n_f to all other nodes. The value of the terminal cost at an arbitrary position in the grid is then found using interpolation between the nodes surrounding that point. The interpolation routine used, can be shown to be consistent and free from local minimums inside each cube in the grid.

[4] This freely available data can also be found at http://duff.geology.washington.edu/data/raster/tenmeter/onebytwo10/.

[5] Beside its computational simplicity, this choice is motivated by its utility for the simultaneous arrival problem considered in Section A.8.

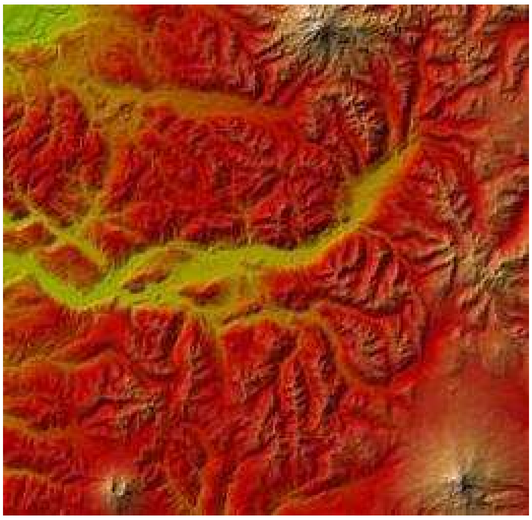


Figure A.3: The terrain elevation map used in the simulations represents an area of more than 82km ×82km taken from the Cascade range, WA. It contains the summit of Mt. Rainier, Mt. Adams and Mt. St. Helen's.

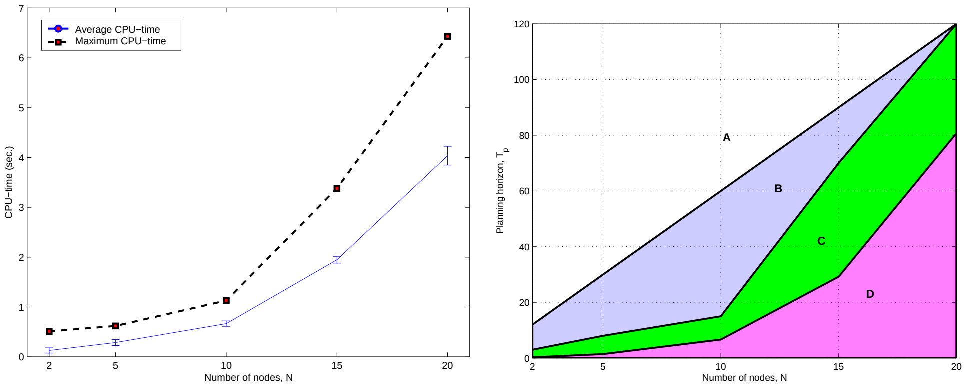
Finally, as pointed out in Section A.1, the term "off-line" here is rather to be interpreted as at a much slower sampling rate than the control loop, *i.e.* in the order of tens of seconds. As new information about the environment or mission objectives is gathered as the mission unfolds, it can be processed and fed back regularly to the vehicle through an updated terminal cost (*cf.* [23, 24]). With the non-optimized MATLAB code used, it takes on average 15.8 seconds to both built the graph representing the environment and calculate the terminal cost. Modifying an existing graph (in order to incorporate mission objectives), takes only 1.5 seconds on average. All computations have been performed on a shared Linux cluster, using one of its four 2.80 GHz Intel Xeon processors.

A.6 Optimality and Computational Load

In this section, we first flesh out the relation between the prediction horizon, T_p, and the number of nodes used for the temporal discretization, N. Then, we empirically verify the intuition that a longer prediction horizon (paired with a suitable N), generates solutions closer to optimum. As expected, the price for this reduction in the objective function has to be paid in terms of increased computational load. It is then important to set up a quantitative comparison between these two competing objectives. As a particular example, it is shown that by accepting a deterioration of less than 3% in the objective function, it is possible to reduce the run-times with more than 38%.

Prediction Horizon and Temporal Discretization

Generally, trajectory optimization run-times are critically depending on the number of variables in the underlying NLP. These in turn, are proportional to the number of nodes in the temporal discretization, N. How the solution time varies as a function of N is depending on the particular NLP solver used. Figure A.4(a) illustrates the average, and maximum run-times of NPOPT; the solver used for all simulations here-within. NPOPT is an updated version of NPSOL; a sequential quadratic programming (SQP) based method for solving NLPs [32]. It it worth mentioning, that the average and maximum have been taken both over a number of planning horizons (typically 10 different values) and iterations (typically $100-150$ iterations per planning horizon). The error bars in Figure A.4(a) represent deviation from the mean value when different planning horizons are chosen. From the raw data, it can be extracted that there is no dependency between these deviations and the length of the planning horizon, T_p. Hence these relatively small error bars, rather stem from the naturally existing fluctuations in solution times of NLPs. It then follows that the choice of N, is to a large extent restricted by the real-time computational requirement.



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

(b) Area A and D: set of infeasible planning horizons. B: feasible but non-optimal planning horizons. C: strip of appropriate planning horizons.

Figure A.4:

Once N has been chosen depending on computational resources[6], a guideline for choosing an appropriate planning horizon is as follows. A lower bound for it is achieved by the requirement that the sampling time,

$$h = \frac{T_p}{N-1},$$

must be large enough to nest the average run-time. Since constraint fulfillment can only be imposed at the sampling instances, a hard upper bound for T_p can be found by choosing a highest allowable inter-node spacing. By inter-node spacing, d, we mean

$$d = \max_{k\in\{2,\ldots,N\}} \|p_k - p_{k-1}\| \leq d_{\max}, \tag{A.8}$$

that is, the maximum distance (in the workspace) between two subsequent sampling instances. However, simulations indicate the existence of a vaguer upper bound, above which deterioration in the objective function arises. All these three bounds are depicted in Figure A.4(b). It is to be interpreted as follows. For a given N, all the planning horizons

[6] The horizon independent approach of this paper, allows us to choose N only taking computational resources and real-time objectives into account; this without jeopardizing task completion/stability. Other works, where the length of the planning horizon is involved in the task completion/stability proof, do not enjoy this characteristic.

lying in area A (above the first mentioned upper bound) are infeasible since they violate the highest allowable inter-node spacing constraint (A.8) and are hence to be disregarded. Although feasible, the planning horizons of strip B are non-optimal in the sense that there always exists a smaller T_p in C that gives a lower value on the objective function. Moving on to strip C, it constitutes the set of applicable planning horizons. It is in this sector that the choice of T_p should be made. Finally, all planning horizons in area D (below the lower bound) give rise to sampling times smaller than the average run-time. The optimization routine will then typically not have enough time to converge. In this sense, all T_p in D are to be considered as infeasible.

A Quantitative Comparison Method

In general, a longer prediction horizon - paired with a suitable N as extracted from Figure A.4(b) - generates solutions closer to optimum. As seen from Figure A.5 and Table A.1, longer prediction horizon gives rise to shorter trajectories that in addition require less control effort, *i.e.* have lower value on the objective function. Referring to Figure A.5, as T_p increases, the planner becomes more predictive and "cuts corners", resulting in shorter trajectories. Regarding the objective function, it can be noted from the fourth column of Table A.1, that the control effort is also inversely proportional to the horizon length. However, as evident from the last two columns of Table A.1, this brings a dramatic increase in average and maximum run-times.

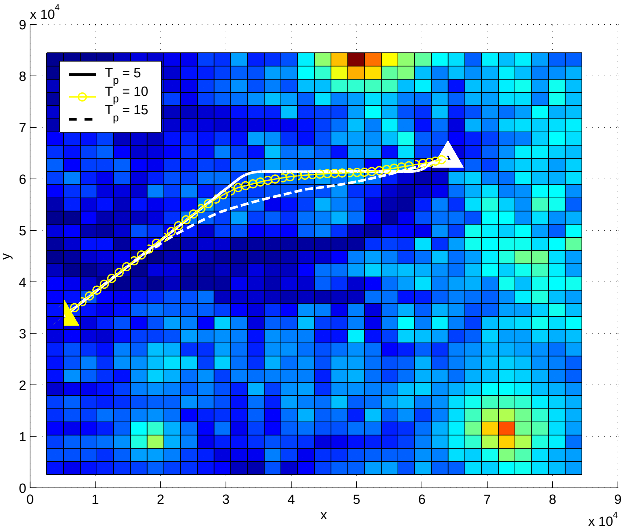


Figure A.5: The impact of varying the planning horizon length, $T_p \in \{5, 10, 15\}$. As T_p increases, the planner becomes more predictive and "cuts corners".

When going from one planning horizon to another, two important issues are those of a possible decrease in the objective function (relative optimality) and the increased run-time associated with it (relative computational load). Table A.2 puts these two aspects in perspective. There, comparison can be made between the relative increase in the objective function and the relative reduction in run-time, when going from one planning horizon, T_{p_1} to a shorter one, T_{p_2}. This allows us to quantify the "select-ability" and "reject-ability" properties of this choice. As a particular example, it can be seen from the last row of

T_p	N	Path length	$\|a\|_2$	Mean CPU-time	Max CPU-time
5 s	4	71 080 m	6.07	0.17 s	0.37 s
10 s	7	70 382 m	4.37	0.35 s	0.74 s
15 s	10	69 919 m	3.72	0.62 s	1.07 s
20 s	13	69 551 m	3.62	1.01 s	1.51 s

Table A.1: Longer prediction horizon generates solutions with lower value on the objective function. The price of that being the evident increase in computational load.

$T_{p_1} \rightarrow T_{p_2}$	Rel. optimality	Rel. comp. load (mean)	Rel. comp. load (max)
10 → 5 s	+38.9%	−52.3%	−50.0%
15 → 10 s	+17.4%	−43.2%	−30.84%
20 → 15 s	+ 2.9%	−38.6%	−29.14%

Table A.2: The impact of the length of T_p on the optimality and computational load. By accepting a deterioration of less than 3% in the objective function, it is possible to reduce the run-times with more than 38%.

Table A.2, that by accepting a deterioration of less than 3% in the objective function, it is possible to reduce the run-times with more than 38% by reducing T_p from twenty to fifteen seconds.

A.7 Simulations

In this section, a small selection of the simulations made with the proposed trajectory planning algorithm is presented. The intention is to merely illustrate two specific properties of it; ability to minimize threat exposure, robustness and balancing the two competing objectives of time-optimality and energy efficiency.

Threat Exposure

As earlier depicted, it is the terminal cost, $\Psi(p)$, that captures the global characteristics of both the environment and the mission objectives. This is readily done by varying the costs, c_{ij}, in the graph representation of the environment. Figure A.6 shows the effect of switching on a radar having a detection radius of 10km. The position of the radar is marked with a black triangle, while yellow circles are used to map out the volume where the vehicle is visible to the radar. The path with circles, shows the outcome of the trajectory planner when the radar is not accounted for. Unaware of its existence, the generated path passes right through the detection area of the radar. The other path, namely the one marked with squares, shows the outcome when the terminal cost incorporates the radar. The threat exposure is now minimized by flying at a much lower altitude, utilizing the protection provided by the terrain and thereby avoiding radar detection.

Robustness

One of the most prominent characteristics of adopting a RHC scheme is that, by reducing the computational effort drastically, it gives us the possibility to repeatedly solve the NLP

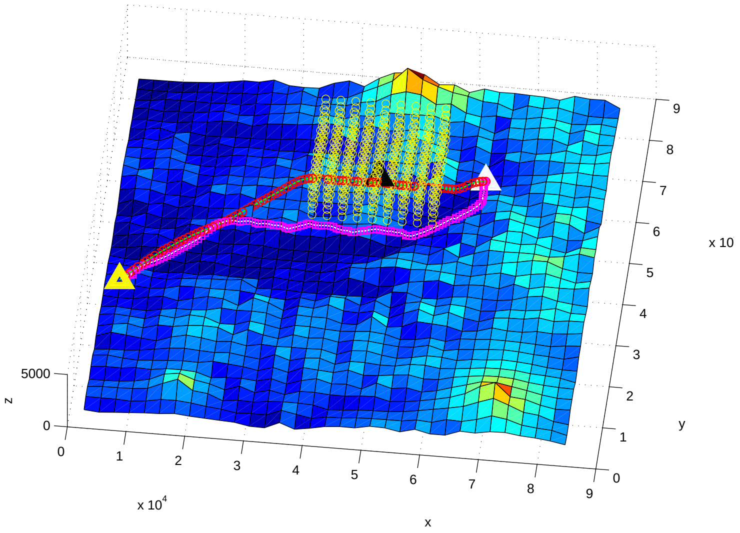


Figure A.6: The effect of threat exposure on the generated path.

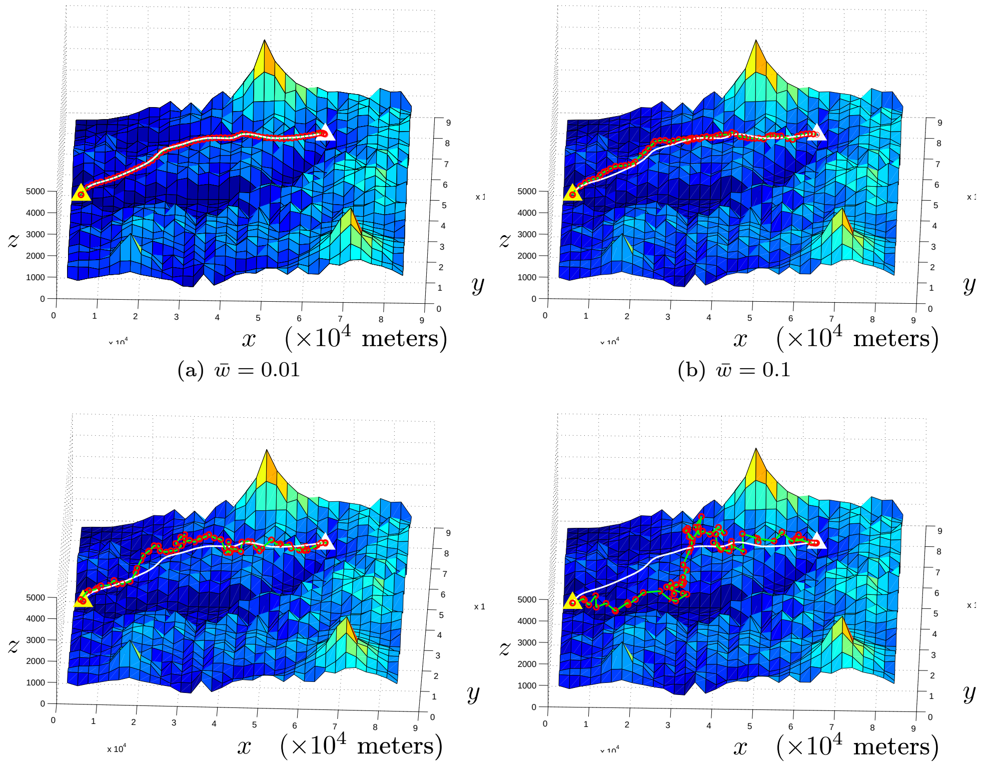
on-line with the current state as a new initial value. This way, feedback is incorporated and a certain degree of robustness is obtained. Next, to put the robustness properties of the trajectory planner to test, the existence of parametric uncertainty, measurement noise and other disturbances (such as wind gust or plant-model mismatch) is introduced. To this end, the nominal update equation, $p_{k+1,1} = p_{k,2}$, is modified to

$$p_{k+1,1} = p_{k,2} + w,$$

where w is a uniformly distributed noise parameter, $U(-\bar{w}, \bar{w})$. This modification implies that we, at the next time instance, will not move exactly to the nominal position we aimed for but rather to a random point in its vicinity. In what follows, in order to isolate the effect of the noise parameter, we set $T_p = 10$, $N = 6$, and study the generated paths and objective function (control effort) as $\bar{w}$ varies in the interval $[0 \quad 0.5]$. Figure A.7 shows the generated paths corresponding to four increasing values on the noise parameter. As can be seen in the third column of Table A.3, the disturbance corresponds to a displacement of more than 40%. Table A.3 also offers additional insight into the simulations made. In addition to the particular values of $\bar{w}$ that have been chosen, the actual effect of it, expressed as maximum offset (in meters) from the nominal position, $p_{k,2}$, can be read from the second column. The relative disturbance, *i.e.* the ratio between the maximum offset and the maximum inter-node spacing, d (see Equation (A.8)), serves as a comparative measure of the size of the disturbance. Finally, as seen in the fourth column, the control effort increases as a function of $\bar{w}$.

Time-Optimality vs. Energy Efficiency

Next, attention is paid to the role of the penalty parameter, β, which determines the relative importance between time-optimality and energy efficiency. Low values on β indicates a relatively high penalty on control usage (which yields energy efficient trajectories), while β taking on a high value reflects the objective of time optimal control. Figure A.8 shows two trajectories generated with two distinct values on β, namely $\beta_L = 1$ (Low penalty) and



(a) $\bar{w} = 0.01$

(b) $\bar{w} = 0.1$

(c) $\bar{w} = 0.3$

(d) $\bar{w} = 0.5$

Figure A.7: The effect of the noise parameter, w, on the generated paths. For the sake of reference, the nominal path ($\bar{w} = 0$), has been sketched with a white solid line. As $\bar{w}$ increases, the offset from the nominal path becomes more evident, but is repeatedly suppressed by the planner.

$\bar{w}$	Max. offset	Relative disturbance	$\lVert a \rVert_2$
0.0	–	–	4.88
0.01	± 37 m	2.5%	4.98
0.05	± 184 m	10.8%	7.94
0.1	± 367 m	20.3%	15.12
0.2	± 734 m	29.6%	17.32
0.3	±1101 m	36.0%	30.48
0.5	±1835 m	42.4%	48.73

Table A.3: Impact of the added disturbance.

$\beta_H = 10^4$ (High penalty). Table A.4 offers additional insight into the simulations made. In particular, the simulations verify the intuition that β_L produces more energy effective paths, while β_H results in more aggressive maneuvers and faster task completion.

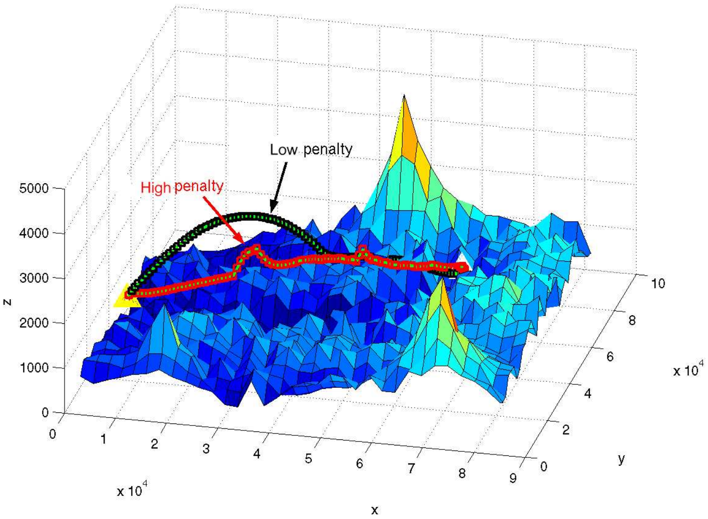


Figure A.8: The effect of varying the penalty parameter, β.

Penalty parameter	Time of flight	Energy consumption
$\beta_L = 1$	$T_L = 362.5$ s	$\|u\|_L = 0.26$
$\beta_H = 10^4$	$T_H = 227.5$ s	$\|u\|_H = 1.13$

Table A.4: The trade-off between the two competing objectives of time-optimality and energy efficiency can be made by varying the penalty parameter, β.

A.8 Simultaneous Arrival of Multiple Aerial-Vehicles

In this section we extend the trajectory optimization problem considered in Section A.3 to the multi-vehicle case. In a centralized setting, such generalization is straightforward with inter-vehicle collision avoidance and increased computational effort as the main hurdles. The first issue can be handled explicitly by the trajectory planner by imposing anti-collision constraints such as safety zones around each vehicle. To hold the computational burden in check, parallel and distributed computing techniques might be considered. A decentralized RHC scheme to generate collision free trajectories have been proposed in [33] where the vehicles, based on local information, *sequentially* plan their individual trajectories. Further, in the venue of formation stabilization of a group of vehicles that are only coupled through the cost function, [34] proposes a distributed implementation of RHC. In this paper, we outset from these works and pay attention to cooperative decision making and consensus seeking in the multi-vehicle network. In particular, we are interested in the problem of agreeing on a simultaneous arrival time. Task synchronization gives major strategical advantages for a large class of mission scenarios, including the case when jamming the enemy radar.

In what follows, in Section A.8.1, we present the framework that will capture this consensus problem of ours and review some established results. Standard results in this field involve consensus problems described by unconstrained, scalar (or fully actuated) first order linear systems (see *e.g.* [25–27]). Extensions in the direction of time dependent interaction topology and communication delays have also been made [35–37]. In our case however, the relation between the control (*i.e.* the vehicle acceleration), and the consensus quantity (which is the Estimated Time to Arrival, ETA) is neither unconstrained nor first order. To remedy this, our design study is presented, where we propose a time-scale separation principle (Section A.8.2) which gives rise to simple equations of motion for the ETA. The interaction between the consensus planner and the trajectory planner is discussed in Section A.8.3 while simulation results are presented in Section A.8.4.

A.8.1 Consensus Problems on Weighted Graphs

Consider a weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A})$ representing the interaction topology of the network of vehicles. Here $\mathcal{V}$ denotes the set of vertices, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges, while $\mathcal{A}$ is a weighted adjacency matrix, $\mathcal{A} = [a_{lm}]$, with non-negative elements, a_{lm}. Each vertex $\mathrm{v}_m \in \mathcal{V}$ is an aerial-vehicle. Let $n = |\mathcal{V}|$ denote the cardinality of the vertex set, *i.e.* the total number of vehicles in the group, so that $m \in \mathbb{N}_n \triangleq \{1, \cdots, n\}$. There is an edge $e_{lm} = (\mathrm{v}_l, \mathrm{v}_m)$ between any two vertices v_l and v_m, if and only if the two vehicles can communicate. The elements of the adjacency matrix $\mathcal{A}$ are associated with the edges of the graph through $e_{lm} \in \mathcal{E} \iff a_{lm} > 0$. In this work, we assume the information flow, and hence the graph $\mathcal{G}$, to be undirected and connected, *i.e.* there is a path connecting any two vertices of the graph. If we let $\mathcal{N}_m = \{l \in \mathbb{N}_n : e_{lm} \in \mathcal{E}\}$ denote the set of neighbors of vehicle m, assuming $\mathcal{G}$ to be connected implies that $\mathcal{N}_m \neq \emptyset$, for all $m \in \mathbb{N}_n$.

Let $\eta_m(t) \in \mathbb{R}$ denote the *consensus state* of vehicle m at time t. In our case η_m represents the Estimated Time to Arrival (ETA) for vehicle m. Simultaneous arrival then means reaching consensus regarding the ETA *i.e.*

$$\eta(t) = [\eta_1(t), \cdots, \eta_n(t)] \to \eta_d \mathbf{1}, \quad \text{as} \quad t \to \infty,$$

with $\mathbf{1} = [1, \cdots, 1] \in \mathbb{R}^n$ and the *consensus value* $\eta_d \in \mathbb{R}$. Assuming the simple first-order dynamics[7]

$$\dot{\eta}_m = u_m, \quad \forall m \in \mathbb{N}_n, \tag{A.9}$$

it has been shown (see *e.g.* [26, 35]), that the following distributed control law (or *protocol*)

$$u_m = \sum_{l \in \mathcal{N}_m} a_{lm}(\eta_l - \eta_m) \tag{A.10}$$

will globally exponentially reach consensus with the speed of the algebraic connectivity of the graph (see *e.g.* Theorem 8 in [35]). Protocol (A.10) can be referred to as the “Laplacian protocol”. The reason for this is apparent when considering the closed-loop network dynamics

$$\dot{\eta}(t) = \Delta\eta(t) = -L\eta(t), \tag{A.11}$$

where Δ denotes the discrete Laplace operator and L is the Laplacian matrix of the graph $\mathcal{G}$, defined as

$$L = \mathrm{diag}(d_1, \cdots, d_n) - \mathcal{A}.$$

Here d_m denotes the valency (*i.e.* the in/out degree) of vertex m.

[7] Technically, we do not have this dynamics at hand, but in a discrete setting, we can use a time-scale separation principle to still motivate the use of this (see Section A.8.2 for details).

A.8.2 The Time-scale Separation Principle

Standard results in the field of cooperative control and decision making involve consensus problems described by unconstrained, scalar (or fully actuated) first order linear systems (see *e.g.* [25–27]). As previously mentioned, for any vehicle in the group, the relation between the actual control inputs (*i.e.* the vehicle acceleration, a_m) and the consensus state (which is the ETA, η_m) is neither unconstrained nor first order. In this section however, by utilizing a time-scale separation principle and introducing the artificial consensus planner control inputs, u_m, we are able to motivate the use of the unconstrained first order dynamics (A.9).

To set stage for the discussion, it might be fruitful to make a minor digression to the principle of guidance and control. Traditionally, the problem of steering a vehicle to its target is broken down into (at least) two subproblems: the problem of *trajectory optimization* (or guidance) and the problem of *auto-pilot design* (or control). This can be viewed as a control system having two degree of freedom; an inner loop (the auto-pilot) and an outer loop (the trajectory optimizer) (see Figure A.9). The pivotal idea that justifies this separation is the multiple time-scales on which guidance and control occurs on. The discrete-time guidance problem considered in this paper occurs on a relatively slow time-scale (in the order of seconds). The inner control loop on the other hand takes place on a much faster time-scale (in the order of milliseconds) and is most often based on a dynamic model of the vehicle. It is noteworthy that by virtue of this separation, only *suboptimal* solutions can in general be found, but the advantage is that the details of the (nonlinear) dynamics of the vehicle only enters into the trajectory optimization part as relatively simple conditions on the reference trajectory. This separation thus justifies the simple linear dynamics used to describe the aerial vehicle in (A.2).

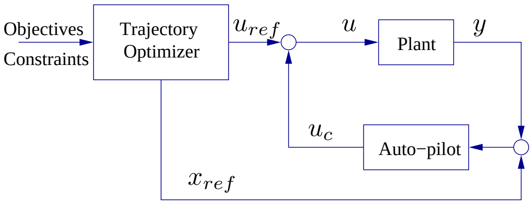


Figure A.9: The two level separation of the control system is possible due to the multiple time-scales of guidance and control.

As for the consensus problem of ours, we imitate the above described idea of time-scale separation principle and design the consensus planner on a fast time-scale. More precisely, between any two subsequent time-steps in the discrete RHC setting (*i.e.* between time-steps k and $k+1$), we adopt a consensus problem on the fast time-scale, where each vehicle, v_m, shares its consensus state η_m with its neighbors, $\mathcal{N}_m$. A schematic block-diagram representation of the multi-vehicle network can be seen in Figure A.10. By possibly scaling the Laplace protocol, its exponential convergence rate implies practical convergence of the consensus states at the end of the k^{th} time-step. Subsequently, the consensus planner sends back the consensus value (*i.e.* the desired ETA for all vehicles), denoted η_d in Figure A.10. The trajectory planner is then to take η_d into account when planning the trajectory in the next time-step. How to do this, as well as how to calculate η_m, is discussed in Section A.8.3.

By virtue of this time-scale separation, the freedom of design of the consensus planner box is obtained. To this end, the artificial consensus planner control inputs, u_m, are intro-

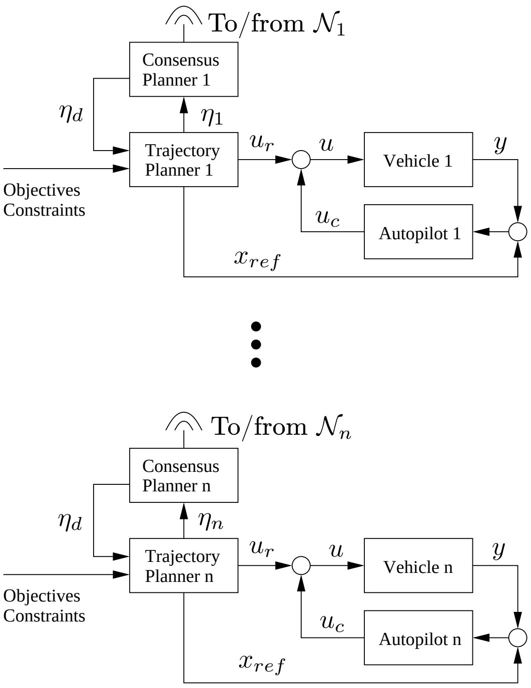


Figure A.10: Block-diagram representing consensus planners and the information flow in the multi-vehicle network.

duced. Further, the dynamics of the state of the consensus planner (*i.e.* η_m) may have the linear first order dynamics (A.9).

Remark A.6. As pointed out in [38], all the previously mentioned works on consensus problems (*i.e.* References [25–29, 35–37]) assume there is a global clock that synchronizes the state updates. This ruins the truly decentralized touch of the approaches. However, based on the time-scale separation principle depicted above, we argue that by introducing the clock as another *actuated* consensus state in the consensus planner (on the fast time-scale), the vehicles can reach consensus regarding the global clock in the same way as with the ETA.

A.8.3 Consensus Planner Interaction

As depicted in Figure A.10, the consensus planner of the m^{th} vehicle receives the ETA, η_m, at the beginning of each time-step. At the end of the time-step, the consensus planner returns the consensus value, η_d, to the trajectory planner. Two important questions regarding this interaction are:

1. How the trajectory planner should estimate the time to arrival (*i.e.* calculate η_m) based on the output from the finite-dimensional optimization problem (A.3).
2. How the trajectory planner should take η_d into account when planning the trajectory in the next time-step.

These two issues are further discussed here-below.

Estimating the Time to Arrival

As mentioned in Section A.3.A.5 the terminal cost, $\Psi(p)$, conveys information about the distance from p to the target point, p_f. Our main objective for making this choice has been its computational simplicity. The consensus problem currently considered, reveals yet another advantage for making this choice namely, it facilitates arrival time estimation. Thus the simultaneous arrival problem currently considered serves as an example for a case when it might be beneficial not choosing the terminal cost, Ψ, close to the optimal value function, J^*.

Regarding the estimation of the arrival time based on the output of (A.3), the following is proposed:

$$\eta_{m,k} = T_p + \frac{\Psi(p_{k,N,m})}{v_{\max}}, \tag{A.12}$$

where $\eta_{m,k}$ denotes the ETA of the m^{th} vehicle at the k^{th} time-step, while $p_{k,N,m}$ is the final position of ditto. This choice is naturally not unique. In particular, one could choose the denominator in (A.12) differently, for instance as the average (or maximum) of the velocities in the planned trajectory, *i.e.*

$$\frac{1}{N}\sum_{i=1}^{N} v_{k,i,m}, \quad (\text{or } \max_i v_{k,i,m}).$$

The current choice (A.12) is however motivated by the fact that, if not updated in the far future, Ψ is an overestimation of the distance to go, why an apparent overestimation of the velocity in the denominator might be advantageous. In simulations, (A.12) has turned out to be a successful choice.

Consensus Value Incorporation

Next we consider the impact of the consensus planner on the trajectory optimizer, *i.e.* raise the question of how the trajectory planner should take η_d into account when planning the trajectory in the next time-step, $k+1$. In practice, the trajectory planner can satisfy the objective of the consensus planner by either imposing a hard constraint

$$T_p + \frac{\Psi(p_{k+1,N,m})}{v_{\max}} + h - \eta_d = 0, \tag{A.13}$$

or by penalizing deviations from the consensus value, η_d. For instance the following penalty term

$$\left[T_p + \frac{\Psi(p_{k+1,N,m})}{v_{\max}} + h - \eta_d\right]^q, \quad q \text{ positive even integer},$$

could be included in the objective function. In the simulations presented in Section A.8.4, the former approach has been taken. The main reason, is its deterministic nature.

This is a convenient point at which to emphasize that since Ψ is merely an approximation of the distance to go (an overestimation if Ψ is not updated in the far future), one must expect and prepare for the case when the ETA turn out to be false at later time instances. To cope with this issue, as well as with the limited vehicle capability to adjust its ETA (in order to fulfill (A.13)), an "outlier detection rule" is adopted. The main idea is that if the consensus value is beyond the range which a vehicle can adjust its arrival time, it should classify itself as an "outlier" and as such, passively follow as the mission unfolds. Note however, that there is no need for an outlier to permanently resign from the mission as η_d might change non-uniformly in the far future. In addition, in order to keep the graph connected - but without affecting the consensus value - an outlier should communicate the average value of the consensus states of its neighbors as its own consensus state. How the outlier classification rule should be designed is an ambiguous question but, based on a scenario involving many vehicles with a few distinct outliers, the following rule

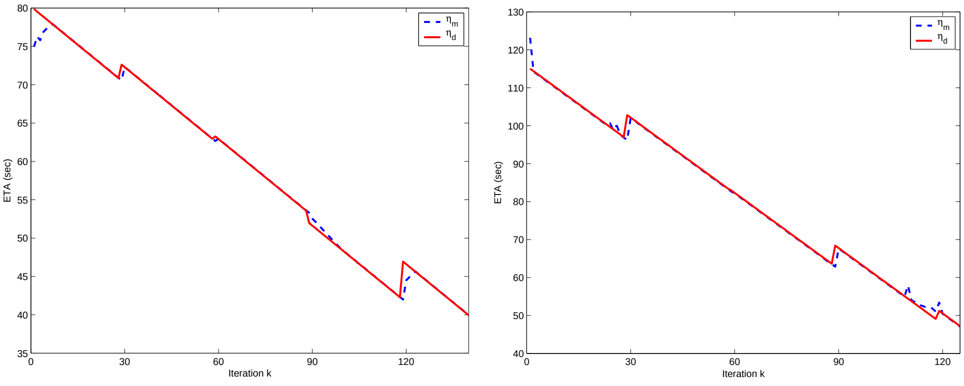
$$\frac{\eta_d - h}{\eta_{k,m}} \in [1-\delta, 1+\delta] \tag{A.14}$$

with $\delta = 0.3$ has been used in the simulations to follow.

A.8.4 Simulations

Since the convergence properties of the consensus value, η_d, can be handled separately, the principal objective for the simulations presented in this section, is to concentrate on the interaction between the consensus planner and the trajectory planner for one single vehicle. The following setup has been used to highlight this aspect and to incorporate the impact of the other vehicles. In each iteration, a fictitious consensus planner receives η_m for the vehicle and sends back a modified consensus value, η_d. In the nominal case, in order to capture time-evolution, η_d will just decrease with a constant. Every 30$^{\text{th}}$ time-step however, η_d is set to random number that deviates from η_m by a given percentage at most. To connect to the outlier detection rule (A.14), a maximum deviation of 30% has been used in the simulations.

Although the simulations only involve the trajectory planner of one single vehicle, it must be emphasized that this setup enables us to capture several aspects of the impact of the other vehicles in the group as well. To start with, a deviating η_d may reflect the most ordinary case of different vehicles having different ETAs. Alternatively, this deviation may stem from the important case when the terminal cost, Ψ, turns out to be false for some of the vehicles. Finally, the occasionally occurring case when the optimization routine used in a vehicle fails to satisfy the hard equality constraint (A.13) may result in a deviating η_d. This latter case may occur even though η_d has passed the outlier detection rule (A.14) and has been observed in the simulations, as can be seen in Figure A.11.



(a) (b)

Figure A.11: Two simulations showing the ETA of a vehicle (dashed line) and the consensus value η_d (solid line) which incorporates the impact of the other vehicles in the group.

The simulations indicate that, in most cases, the trajectory planner manages to fulfill the will of the consensus planner in one single time step. However, in some cases when there is a large change in the set point, or when the optimization routine fail to satisfy the equality constraint (A.13) (most often since the limit for the maximum number of major iterations is reached), it might take more than one time-step for the trajectory planner to converge to the η_d dictated by the consensus planner.

A.9 Conclusion

In this paper, results regarding trajectory optimization for aerial vehicles in the three dimensional space has been presented. In particular, properties such as *safety*, *completion* and *simultaneous arrival* were in focus. Other prominent characteristics of the presented method, such as ability to minimize threat exposure and robustness, were highlighted through simulations.

As for the safety concerns, the alternative outlined in this paper, extends previous results by possessing provable safety properties in a 3D setting. In addition, in our case, safety also renders task completion possible. This is simply due to an elaborate choice of the set of safe states, in which the augmented safety maneuver is to end. As a subsidiary consequence, introducing the safety maneuver also makes it possible to cope with hard real-time systems.

Because of the computational burden it introduces, task completion is here not achieved by merely adjusting the length of the planning horizon. Instead, it is argued that requiring monotonicity of the composite cost is sufficient for approaching the target set. Decoupling the length of the planning horizon from our task completing objectives, enables us to determine it solely on the basis of accuracy demands and computational resources. To guide this selection, the interaction between optimality and computational load was empirically examined in Section A.6, where a quantitative comparison between these two competing objectives where made.

Finally, by using a time-scale separation principle, we were able to adopt standard Laplacian control to a consensus problem which is neither unconstrained nor first order.

A.10 References

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

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

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

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

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

[6] Pirjanian, P., "Multiple objective behavior-based control," *Robotics and Autonomous Systems*, Vol. 31, No. 1–2, Apr. 2000, pp. 53–60.

[7] Chankong, V. and Haimes, Y. Y., *Multiobjective decision making*, Vol. 8 of *North-Holland Series in System Science and Engineering*, North-Holland Publishing Co., New York, 1983.

[8] Simon, H. A., *The new science of management decision*, The Ford Distinguished Lectures, Vol. III, Harper & Brothers Publishers, New York, 1960.

[9] Goodrich, M. A., Stirling, W. C., and Frost, R. L., "A theory of satisficing decisions and control," *IEEE Transactions on Systems, Man and Cybernetics, Part A*, Vol. 28, No. 6, Nov. 1998, pp. 763–779.

[10] Curtis, J. W. and Beard, R. W., "Satisficing: a new approach to constructive nonlinear control," *IEEE Trans. Automat. Control*, Vol. 49, No. 7, 2004, pp. 1090–1102.

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

[12] Primbs, J. A. and Nevistić, V., "A new approach to stability analysis for constrained finite receding horizon control without end constraints," *IEEE Trans. Automat. Control*, Vol. 45, No. 8, Aug. 2000, pp. 1507–1512.

[13] Jadbabaie, A., Yu, J., and Hauser, J., "Unconstrained receding-horizon control of nonlinear systems," *IEEE Trans. Automat. Control*, Vol. 46, No. 5, May 2001, pp. 776–783.

[14] Primbs, J., Nevistic, V., and Doyle, J., "A receding horizon generalization of pointwise min-norm controllers," *IEEE Trans. Automat. Control*, Vol. 45, No. 5, May 2000, pp. 898–909.

[15] Sontag, E. D., "A "universal" construction of Artstein's theorem on nonlinear stabilization," *Systems & Control Letters*, Vol. 13, No. 2, 1989, pp. 117–123.

[16] Freeman, R. A. and Kokotovic, P. V., "Optimal nonlinear controllers for feedback linearizable systems," *Proc. of the IEEE American Control Conference*, Vol. 4, Seattle, WA, USA, Jun. 1995, pp. 2722–2726.

[17] Richards, A. and How, J. P., "Model predictive control of vehicle maneuvers with guaranteed completion time and robust feasibility," *Proc. of the IEEE American Control Conference*, Vol. 5, Jun. 2003, pp. 4034–4040.

[18] Richards, A. and How, J. P., "Robust variable horizon model predictive control for vehicle maneuvering," *Internat. J. Robust Nonlinear Control*, Vol. 16, No. 7, 2006, pp. 333–351.

[19] Ögren, P. and Leonard, N. E., "A convergent dynamic window approach to obstacle avoidance," *IEEE Trans. on Robotics*, Vol. 21, No. 2, 2005, pp. 188–195.

[20] Schouwenaars, T., How, J., and Feron, E., "Receding horizon path planning with implicit safety guarantees," *Proc. of the IEEE American Control Conference*, Vol. 6, 2004, pp. 5576–5581.

[21] Kuwata, Y. and How, J. P., "Stable trajectory design for highly constrained environments using receding horizon control," *Proc. of the IEEE American Control Conference*, Vol. 1, 2004, pp. 902–907.

[22] Bellingham, J., Kuwata, Y., and How, J. P., "Stable receding horizon trajectory control for complex environments," *Proc. of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, Austin, Texas, Aug. 2003.

[23] Mettler, B. and Bachelder, E., "Combining on- and offline optimization techniques for efficient autonomous vehicle's trajectory Planning," *Proc. of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, San Francisco, CA, USA, Aug. 2005.

[24] Ferguson, D. and Stentz, A., "The Delayed D* algorithm for efficient path replanning," *Proceedings of the IEEE International Conference on Robotics and Automation*, April 2005.

[25] Olfati-Saber, R., Fax, J. A., and Murray, R. M., "Consensus and cooperation in networked multi-agent systems," Proceedings of the IEEE, Invited Paper, Jan. 2007.

[26] Jadbabaie, A., Lin, J., and Morse, A. S., "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Trans. Automat. Control*, Vol. 48, No. 6, 2003, pp. 988–1001.

[27] Fax, J. A. and Murray, R. M., "Information flow and cooperative control of vehicle formations," *IEEE Trans. Automat. Control*, Vol. 49, No. 9, 2004, pp. 1465–1476.

[28] McLain, T. W., Chandler, P. R., Rasmussen, S., and Pachter, M., "Cooperative control of UAV rendezvous," *Proc. of the IEEE American Control Conference*, Vol. 3, Arlington, VA, jun. 2001, pp. 2309–2314.

[29] McLain, T. W. and Beard, R. W., "Cooperative path planning for timing-critical missions," *Proc. of the IEEE American Control Conference*, Vol. 1, Jun. 2003, pp. 296–301.

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

[31] Lindström, P. and Pascucci, V., "Visualization of large terrains made easy," *Proc. of the IEEE Visualization, VIS '01*, San Diego, CA, USA, Oct. 2001, pp. 363–574, [Online] http://www.cc.gatech.edu/projects/large_models/ps.html.

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

[33] Kuwata, Y., Richards, A., Schouwenaars, T., and How, J., "Decentralized robust receding horizon control for multi-vehicle guidance," *Proc. of the IEEE American Control Conference*, Minneapolis, Minnesota, Jun. 2006.

[34] Dunbar, W. B. and Murray, R. M., "Distributed receding horizon control for multivehicle formation stabilization," *Automatica J. IFAC*, Vol. 42, No. 4, 2006, pp. 549–558.

[35] Olfati-Saber, R. and Murray, R. M., "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Trans. Automat. Control*, Vol. 49, No. 9, 2004, pp. 1520–1533.

[36] Moreau, L., "Stability of multiagent systems with time-dependent communication links," *IEEE Trans. Automat. Control*, Vol. 50, No. 2, 2005, pp. 169–182.

[37] Ren, W. and Beard, R. W., "Consensus seeking in multiagent systems under dynamically changing interaction topologies," *IEEE Trans. Automat. Control*, Vol. 50, No. 5, 2005, pp. 655–661.

[38] Fang, L. and Antsaklis, P. J., "Information consensus of asynchronous discrete-time multi-agent systems," *Proc. of the IEEE American Control Conference*, Vol. 3, Jun. 2005, pp. 1883–1888.

Paper B

Adaptive Node Distribution for Online Trajectory Planning

Adaptive Node Distribution for Online Trajectory Planning

David A. Anisi

Abstract

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

Keywords: Computational Optimal Control, On-line Trajectory Planning, Adaptive Grid Methods, Missile Guidance.

B.1 Introduction

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

It is a well-established fact in numerical analysis, that a proper distribution of grid points is crucial for both the accuracy of the approximating solution, and the computational effort (see *e.g.* [3, 4]). In general, grid adaption is carried out by some combination of re-distribution (strategically moving the nodes), refinement (adding/deleting nodes), or employing higher order numerical schemes in regions where the local accuracy needs to be improved (consult *e.g.* [5]). In most cases however, there exist a trade-off between accuracy and efficiency in terms of computational effort. In this paper, the focus is on improving accuracy for a given efficiency requirement. More precisely, once the number of nodes in the temporal discretization has been decided (depending on *e.g.* computational resources), the question of optimal node distribution is raised. Although adaptive grid methods - which mainly concern node distribution in the *spatial* domain - have been an active field for the last couple of decades, to the best of our knowledge, utilizing them for adaptive node distribution (in the *temporal* domain) and on-line trajectory optimization has not been considered elsewhere.

This paper is organized as follows. In Section B.2 some background material regarding computational methods for solving optimal control problems is presented. Subsequently in Section B.3, we advocate that in any computationally efficient method, node distribution should be a part of the optimization process and show that the receding horizon control (RHC) method can be considered as an outcome of such a paradigm. In Section B.4, the benefits of utilizing the suggested method are confirmed by a missile guidance example. Finally, this paper is concluded in Section B.5 with some expository remarks.

B.2 Computational Optimal Control

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

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

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

Problem Transcription

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

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

$$\begin{tikzcd}
\mathrm{OCP} \arrow[r, shift left, "\text{direct transcription}"] \arrow[d, swap, "\substack{\text{Hamiltonian}\\ \text{Adjoints}\\ \text{PMP}}"] & \mathrm{OCP}^{\mathrm{N}} \arrow[l, shift left, "\text{consistency}"] \arrow[d, "\substack{\text{Lagrangian}\\ \text{Lagrange multipliers}\\ \text{KKT}}"] \\
\mathrm{OCP}^{\lambda} \arrow[r, shift left, "\text{indirect trancription}"] & \mathrm{OCP}^{\mathrm{N}\lambda} \arrow[l, shift left, "\text{consistency}"]
\end{tikzcd}$$

Figure B.1: Direct and indirect transcription methods.

Although indirect methods are considered to produce more accurate results, they are not typically used to solve problems having complex dynamics or constraint set. Neither are they suitable for problems where the underlying OCP is considered to be changeable in terms of the final manifold, S_f and/or the constraint set, $g(x, u)$. This is mainly due to the possibly ill-conditioned properties of the TPBVP, but also the occasionally tedious derivation of the necessary conditions via PMP. Bearing in mind the type of problems considered in this paper, the focus will therefore be on direct transcription methods.

In most direct methods (see *e.g.* [12] and the references therein), transcription is achieved by *a priori* partition of the time interval into a prescribed number of subintervals whose endpoints are called *nodes*. The NLP variables may then be taken as the value of the controls and the states at these nodes. The integral cost functional and the constraint set are discretized similarly and approximated by any preferred quadrature rule (consult *e.g.* [3, 14]). Finally, additional constraints are imposed on the NLP variables so that the state equations are fulfilled at the so called collocation points.

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

B.3 Adaptive Node Distribution

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

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

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

Generally, trajectory optimization run-times are critically depending on the number of variables in the NLP. These in turn, are proportional to the number of nodes in the temporal discretization, hence-forth denoted N. How the solution time varies as a function of N is depending on the structure of the considered problem, adopted solution method and not the least: the particular NLP solver used. Figure B.2 illustrates the average, and maximum run-times of NPOPT; the solver used for all simulations here-within. NPOPT is an updated version of NPSOL; a sequential quadratic programming (SQP) based method for solving NLPs [18]. It is worth mentioning, that the average and maximum have been taken both over a number of planning horizons (typically 10 different values) and iterations (typically $100 - 150$ iterations per planning horizon). This in order to isolate the relation between the number of nodes and the solution run-times.

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

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

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

The equidistribution principle (EP) requires a chosen positive definite monitor function (or weight), w, to be equidistributed over all subintervals. Mathematically, the EP can be

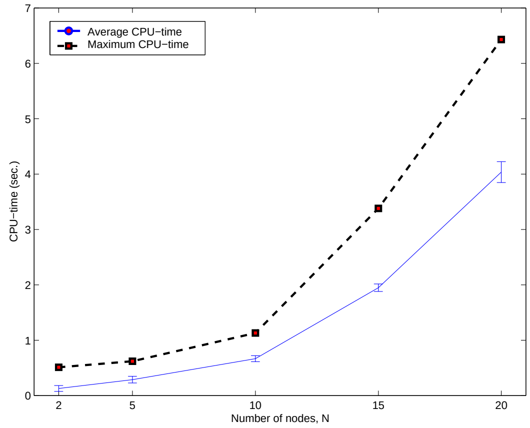


Figure B.2: The increasing average and maximum run-times of NPOPT as a function of N. Computations are performed on a shared Linux cluster, using one of its four 2.80 GHz Intel Xeon processors.

expressed in various equivalent forms, *e.g.*:

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

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

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

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

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

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

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

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

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

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

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

Example B.2. The linear constraint

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

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

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

B.4 Design Study: Missile Guidance

Traditionally, the problem of steering a missile to its target is broken down into (at least) two subproblems: the problem of *trajectory optimization* and the problem of *auto-pilot design*. This can be viewed as a control system having two degree of freedom; an inner loop (the auto-pilot) and an outer loop (the trajectory optimizer) (see Figure B.3). The trajectory optimizer provides a feasible feed-forward control and reference trajectory that is optimal in some specified sense with respect to *e.g.* time to intercept or intercept velocity, and subject to constraints on *e.g.* terminal aspect angle (given by warhead efficiency and target vulnerability) or path segment location (dictated by tactical considerations). It is then the task of the auto-pilot to perform the trajectory following.

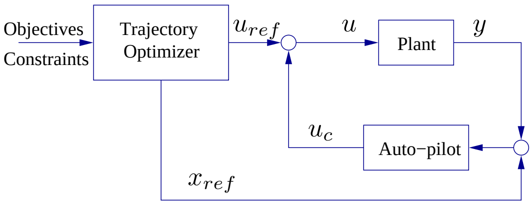


Figure B.3: Two level separation of the missile guidance problem.

By virtue of this separation, only *suboptimal* solutions can in general be found, but the advantage is that the details of the dynamics of the missile only enters into the trajectory optimization part of the problem as (relatively simple) conditions on the reference trajectory. In this work, the existence of an auto-pilot is assumed, so that the focus will solely be on the trajectory optimization part.

By means of standard approximation procedures in flight-community (see *e.g.* [20, 21]), the six-degree-of-freedom (6DoF) equations of motion of the missile in $\mathbb{R}^3$, can be reduced to 3DoF planar movement in two orthogonal subspaces, namely the pitch-, and yaw-plane. Since the 3DoF equations of motions in these planes are similar and decoupled, in what follows, just the pitch-plane dynamics will be considered.

The 3DoF equations of motion in the pitch plane consider the rotation of a body-fixed coordinate frame, (X_b, Z_b) about an Earth-fixed inertial frame, (X_e, Z_e) (see Figure B.4).

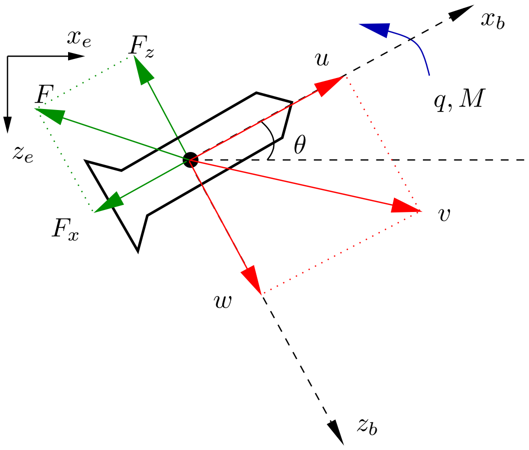


Figure B.4: Missile system variables.

The governing dynamic equations are

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

where u and w are the X_b and Z_b components of the velocity vector, x_e and z_e denote the position of the missile in the inertial frame (X_e, Z_e), q is the pitch angular rate, θ denotes the pitch angle, m is the missile mass, g is the gravitational force, while I_y denotes the pitching moment of inertia. The system inputs are the applied pitch moment, M, together with the aerodynamic forces, F_x, F_z, acting along the X_b and Z_b axis respectively. During the simulations we adopt the constants given in Reference [22] and set $m = 204.02\ kg$, $g = 9.8\ m/s^2$ and $I_y = 247.437\ kg\,m^2$.

Referring to Figure B.5, the first simulation shows the terminal guidance part of a missile trajectory optimization problem. The missile starts off horizontally from $(0, 10)$

aiming at a target in $(700, 0)$ with terminal aspect angle $-\frac{\pi}{2}$. Figure B.5 depicts the reference trajectories with the missile velocities (in the inertial frame) indicated by small line segments. In the adaptive case, an EP based on the arclength monitor function is used.

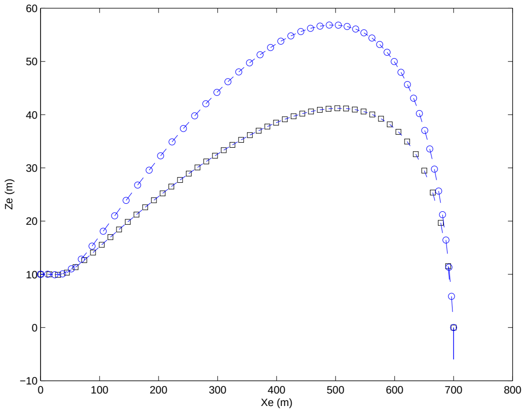


Figure B.5: Reference trajectories: static (○) and adaptive (□).

Seeing beyond the unequal axis scales, the nodes have been distributed more evenly over different path segments. In fact, there are 7 nodes/100 m path segment in the adaptive case, while the same figure varies between $5 - 13$ in the static case.

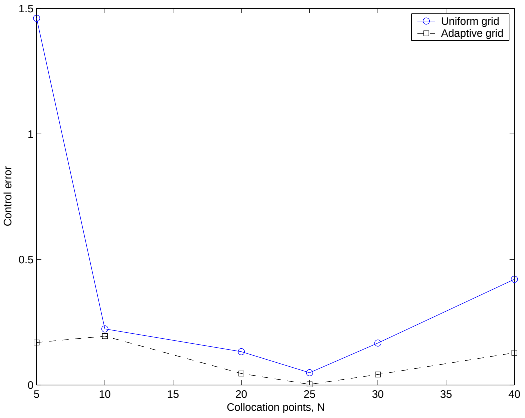


Figure B.6: The accuracy of the approximating control.

Figure B.6 shows the optimal control approximation error as a function of N. It can be noted that, for a given N, the extra degree of freedom provided by distributing the nodes is used constructively to improve accuracy. This illustrates the soundness of the proposed approach. Moreover, Figure B.6 reveals the nonuniform convergence rate of the approximation error which - in our particular case - is seen to be minimized for $N = 25$. The reason for this is the pronounced nonlinearity of the considered NLP together with the fact that the used optimization routine (NPOPT) is a local optimizer, *i.e.* does not guarantee convergence to a *global* minimum. It is therefore not possible to expect that a higher value on N should always yield a better trajectory approximation.

As previously mentioned, in general, there is a trade-off between accuracy and efficiency in terms of computational effort. Once we have observed that re-distributing the nodes improves the accuracy of the approximation, one might wonder how this effects the computation time. Figure B.7 shows the average CPU-time used in the simulations for different values on N. It can be noted that adopting the proposed adaptive node distribution scheme, does *not* bring any increase in the average computational time. We believe that the nonlinearity of the original set of equations describing the motion of the missile, is one of the main reasons for this.

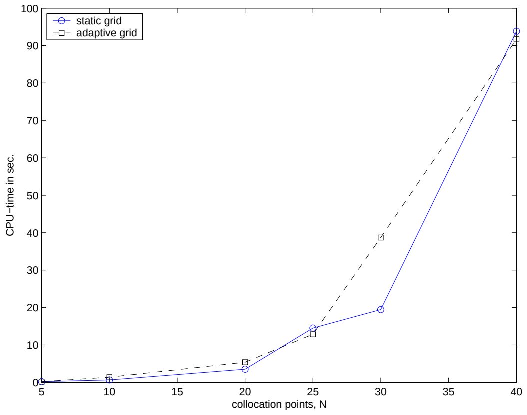


Figure B.7: The average CPU-time for the uniform- and adaptive grid generation scheme as a function of N.

B.5 Concluding Remarks

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

It is argued that in any computationally efficient method, node distribution should be a part of the optimization process. This, in order to minimize the discretization error and gain accuracy, *without* bringing any drastic increase in the computational effort. Here-within, re-distributing the nodes have been formulated as a constrained optimization problem; to be included in the underlying NLP.

The missile guidance problem considered, showed that the extra degree of freedom provided by distributing the nodes is used constructively to improve accuracy. These advantages accrue particularly in the case when having a nonlinear dynamic system at hand. The reason for this being that having the node positions as variables in the underlying NLP, turns a linear system into a bilinear one, which may then give rise to an undesirable increase in computational complexity.

B.6 References

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Paper C

Active Observers for Mobile Robotic Systems

Active Observers for Mobile Robotic Systems

David A. Anisi and Xiaoming Hu

Abstract

An important class of non-uniformly observable systems comes from applications in mobile robotics. In this paper, the problem of active observer design for such systems is considered. The set of feasible configurations and the set of output flow equivalent states are defined. It is shown that the inter-relation between these two sets may serve as the basis for design of active observers. The proposed observer design methodology is illustrated by considering a unicycle robot model, equipped with a set of range-measuring sensors.

Keywords: Nonlinear Observer Design, Active Observers, Non–uniformly Observable Systems, Mobile Robotic Systems.

C.1 Introduction

Since 1970's there has been an extensive study on the design of observers for nonlinear control systems, [1–7]. It is known that for such systems, observability does not only depend on the initial conditions, but also on the exciting control. Most current methods, such as observers with linearizable error dynamics [3] and high gain observers [6, 7], lead to the design of an exponential observer. As a necessary condition for the existence of a smooth exponential observer, the linearized pair must be detectable [5]. In fact, most of the existing nonlinear observer design methods are only applicable to uniformly observable nonlinear systems. This is witnessed in [8], where it is pointed out that one of the key questions in nonlinear control is "how to design a nonlinear observer for nonlinear systems whose linearization is neither observable nor detectable".

An important class of non-uniformly observable systems comes from applications in mobile robotics. For such systems, due to environmental restrictions and the way the sensors function, constraints have to be put on the control. This thus presents an interesting issue: how to design an exciting control to maximize the rate of convergence for an observer, namely how to design an *active observer*. Maximizing "observability" has been an important issue in the field of active perception in robotics and computer vision [9]. However, study from the systems and control point of view in terms of observer design still lacks, [10].

This paper considers the problem of active observer design for mobile robotic systems and an alternative design method is presented. The disposition is as follows; In Section C.2, a brief review on nonlinear observability and observers is given. This would set stage for our study on observability and active observer design for mobile robotic systems in Section C.3. To illustrate the concepts introduced in Section C.3.1, a case study is given in Section C.4. The simulation results thereof are presented in Section C.5 and finally, some concluding remarks are made in Section C.6.

C.2 Preliminaries

Consider the nonlinear control system

$$\Sigma : \left\{ \begin{array}{lcll} \dot{x} & = & \mathcal{F}(x,u) & \text{(system dynamics)} \\ y & = & h(x) & \text{(system output)} \end{array} \right.$$

with state $x \in \mathcal{X}$, control $u \in \mathcal{U}$ and output $y \in \mathcal{Y}$. Here $\mathcal{X}$,$\mathcal{U}$ and $\mathcal{Y}$ are smooth manifolds of dimension n, p and m respectively. All mappings in this paper, are assumed to be smooth. If Σ is complete, the composed mapping from $u(\cdot)$ to $y(\cdot)$ is referred to as the input-output map of Σ at x_0 [11]:

$$\mathcal{IO}^{\Sigma}_{x_0} : u(\cdot) \mapsto y(\cdot).$$

The most common definitions of the observability properties of Σ then boil down to the injectivity properties of $\mathcal{IO}^{\Sigma}_{x_0}$ with respect to the initial condition, x_0. Consider two states, x_1 and x_2, being equivalent (denoted $x_1 \sim x_2$) if and only if they have the same input-output map for all admissible inputs, *i.e.*

$$x_1 \sim x_2 \Longleftrightarrow \mathcal{IO}^{\Sigma}_{x_1}(u(\cdot)) = \mathcal{IO}^{\Sigma}_{x_2}(u(\cdot)), \quad \forall u(\cdot) \in \mathcal{U}.$$

Further, let $\mathbb{I}(x_0)$ denote the *equivalence class* of x_0, *i.e.* let $\mathbb{I}(x_0) = \{x \in \mathcal{X} \colon x \sim x_0\}$. Based on this, we arrive at the following two definitions [12, 13].

Definition C.1 (Indistinguishability)**.** *Two states, x_1 and x_2 are said to be* indistinguishable *if and only if they are equivalent.*

Definition C.2 (Observability)**.** *Σ is said to be* observable at x_0 *if $\mathbb{I}(x_0) = \{x_0\}$. It is further said to be observable if $\mathbb{I}(x) = \{x\}$ for all $x \in \mathcal{X}$.*

It is notable that the equivalence relation on $\mathcal{X}$, and hence observability, is a *global concept* in two senses:

Property C.1. All *states in $\mathcal{X}$ are to be distinguished from each other.*

Property C.2. *The generated trajectories are unrestricted.*

Also, observability is an *infinite-horizon concept*, since:

Property C.3. *There is no upper bound on the time-interval that has to be considered in order to distinguish points.*

Consequently it is possible to introduce various restrictions, or relaxations on Definition C.2. Some of these modifications are considered below.[1]

Given a system Σ and an open set $\Omega \subseteq \mathcal{X}$, the restriction Σ_Ω refers to a control system with state space Ω, defined by the restriction of $\mathcal{F}$ and h to $\Omega \times \mathcal{U}_\Omega$ and Ω respectively. Here $\mathcal{U}_\Omega$ denotes the subset of all admissible inputs that generates trajectories that lie in Ω.

[1] The observability nomenclature is not standardized. In this article, the terms used by Hermann and Krener in [12] and Respondek in [13] are merged.

Definition C.3 (Ω-indistinguishability)**.** *Two initial states, $x_1, x_2 \in \Omega$ are said to be* Ω-indistinguishable *if*

$$\mathcal{IO}_{x_1}^{\Sigma_\Omega}(u(\cdot)) = \mathcal{IO}_{x_2}^{\Sigma_\Omega}(u(\cdot)), \;\; \forall u(\cdot) \in \mathcal{U}_\Omega.$$

This relation will be denoted $x_1{}_{\widetilde{\Omega}}\, x_2$ and $\mathbb{I}_\Omega(x)$.

Definition C.4 (Strong observability)**.** *The system Σ is said to be strongly observable at x_0 if for* every *open neighborhood Ω of x_0, $\mathbb{I}_\Omega(x_0) = \{x_0\}$. Σ is called* strongly observable *if it is strongly observable for all $x \in \mathcal{X}$.*

Note that strong observability implies observability since $\mathbb{I}_\Omega(x) = \{x\}$ for all $\Omega \subseteq \mathcal{X}$ gives $\mathbb{I}(x) = \{x\}$ for the special choice of $\Omega = \mathcal{X}$.

Definition C.5 (Weak observability)**.** *The system Σ is called weakly observable at x_0 if there exist a neighborhood of x_0, $N(x_0)$, such that $\mathbb{I}(x_0) \cap N(x_0) = \{x_0\}$. Σ is* weakly observable *if it is weakly observable at every $x \in \mathcal{X}$.*

Definition C.6 (Instant observability)**.** *The system Σ is said to be instantaneously observable at x_0 if there exist a neighborhood $N(x_0)$, such that for every open neighborhood Ω of x_0 contained in N, $\mathbb{I}_\Omega(x_0) = \{x_0\}$. Σ is called* instantaneously observable *if it is so at every $x_0 \in \mathcal{X}$.*

For the dynamical system, Σ, an observer may be defined as follows (*cf.* [1, 4, 14]).

Definition C.7 (Observer)**.** *A dynamical system with state manifold $\mathcal{Z}$, input manifold $\mathcal{U} \times \mathcal{Y}$, together with a mapping $\hat{\mathcal{F}} : (\mathcal{Z} \times \mathcal{U} \times \mathcal{Y}) \to T\mathcal{Z}$ is an* observer *for the system Σ, if there exists a smooth mapping $\Psi : \mathcal{X} \to \mathcal{Z}$, such that the diagram shown in Figure C.1, commutes and the error trajectory $x(t) - \hat{x}(t)$ converges to zero as $t \to \infty$.*

$$\begin{tikzcd}
 & T\mathcal{X} \arrow[r, "\Psi_*"] \arrow[dl, "\tau_{\mathcal{X}}"'] & T\mathcal{Z} \arrow[dr, "\tau_{\mathcal{Z}}"] & \\
\mathcal{X} & & & \mathcal{Z} \\
 & \mathcal{X} \times \mathcal{U} \arrow[ul, "\pi_{\mathcal{X}}"] \arrow[uu, "\mathcal{F}"'] \arrow[r, "(\Psi\circ\pi_{\mathcal{X}}, \pi_{\mathcal{U}}, h\circ\pi_{\mathcal{X}})"] & \mathcal{Z} \times \mathcal{U} \times \mathcal{Y} \arrow[uu, "\hat{\mathcal{F}}"] \arrow[ur, "\pi_{\mathcal{Z}}"'] \arrow[ull, dashed, bend left, "\Phi"] &
\end{tikzcd}$$

Figure C.1: Commutative diagram defining an observer.

In diagram C.1, Ψ_* denotes the tangent mapping, π is projection upon a Cartesian factor, while τ denotes the projection of the tangent bundle.

According to Definition C.7, the objective when designing a general observer, is to track $\Psi(x)$, rather than x itself. Note that the same observer dynamics, $\hat{\mathcal{F}}$, may allow several *different* full observer mappings, Φ, and that in general, a full state observer

$$\hat{\Sigma} : \left\{ \begin{array}{lcl} \dot{z} & = & \hat{\mathcal{F}}(z, u, y) \\ \hat{x} & = & \Phi(z, u, y) \end{array} \right.$$

may *not* be put in the form $\dot{\hat{x}} = \Xi(\hat{x}, u, y)$.

C.3 Mobile Robotic Systems

One distinguishing feature of mobile robots is the use of *exteroceptive* sensors for sensing the environment and aid localization. The output of Σ is next extended to more explicitly incorporate exteroceptive sensor readings. Bearing in mind the particular applications encountered in the robotics community, it seems convenient to split the state vector, $x \in \mathcal{X}$, into two parts; one defining the state of the platform in its *work-space*, $\mathcal{W}$, and the other only in its *configuration-space*, $\mathcal{C}$, so that $x = (x_w, x_c) \in \mathcal{W} \times \mathcal{C} = \mathcal{X}$. The work-space of the robot, $\mathcal{W}$, is assumed to be a smooth and connected manifold of dimension $n_w \in \{1, 2, 3\}$. However, the configuration-space, $\mathcal{C}$, might have arbitrary dimension, n_c, and includes typically the description of the internal states of the platform.

Consider control-affine dynamic systems of form:

$$\Sigma_{rob} : \begin{cases} \dot{x}_w &= f_w(x) + g_w(x)u \\ \dot{x}_c &= f_c(x) + g_c(x)u \\ y &= \tilde{h}(x, s_\theta(x)) \\ q &= \theta(s_\theta), \end{cases}$$

where $x_w \in \mathcal{W}$, $x_c \in \mathcal{C}$, $u \in \mathcal{U}$ and $y \in \mathcal{Y}$. We use $s_\theta(x)$ to indicate the interaction of the sensors with the environment but also to emphasize the dependence of the output on the *environmental map*, θ. In this paper, the case where the components of the environment (*e.g.* surrounding terrain, obstacles or walls) can be modeled as a single, connected, $(n_w - 1)$-dimensional smooth manifold (hyper-surface) in $\mathcal{W}$ will be in focus. It is further assumed that this hyper-surface can be parameterized as

$$q = \theta(s_\theta), \qquad s_\theta \in \mathcal{S} \subseteq \mathbb{R}^{(n_w - 1)},$$

where θ is known. This last assumption relates to one of the fundamental problems in robotics, namely the simultaneous localization and map building problem (SLAM), where one tries to reconstruct the environmental map, θ, and the full state vector, x, at the same time. By assuming the map to be given, we focus on a subproblem in SLAM, namely the re-localization problem where the state vector, x, is to be reconstructed based on a combination of exteroceptive and introceptive sensor-readings.

Example C.1. Consider a nonholonomic vehicle equipped with a range sensor mounted along its direction of orientation, ϕ. It moves inside an elliptic field, with half-axes c_1 and c_2, centered at the origin of $\mathcal{W}$. Then the hyper-surface

$$\left(\frac{q_1}{c_1}\right)^2 + \left(\frac{q_2}{c_2}\right)^2 - 1 = 0,$$

models the surrounding in $\mathbb{R}^2$. It can be parameterized by the angle $\tau \in S^1$, so that

$$q = \begin{bmatrix} q_1 \\ q_2 \end{bmatrix} = \begin{bmatrix} c_1 \cos(\tau) \\ c_2 \sin(\tau) \end{bmatrix} = \theta(s_\theta),$$

with $s_\theta = [\tau \; c_1 \; c_2]$. The control system can be modeled as

$$\begin{aligned} \dot{x}_1 &= u_1 \cos(\phi) \\ \dot{x}_2 &= u_1 \sin(\phi) \\ \dot{\phi} &= u_2 \\ y &= (c_1 \cos(\tau) - x_1) \cos(\phi) + (c_2 \sin(\tau) - x_2) \sin(\phi), \end{aligned}$$

where $(x_1, x_2) \in \mathbb{R}^2$ is the position of the reference point on the robot, $\phi \in S^1$ denotes its orientation and the two control inputs, u_1 and u_2 are the robot's linear- and lateral velocities respectively. In addition, τ as a function of the state is implicitly defined by

$$\frac{c_2 \sin(\tau) - x_2}{c_1 \cos(\tau) - x_1} = \tan(\phi).$$

Example C.2. Consider a nonholonomic mobile robot equipped with a centrally mounted video camera. The environment consists of the goal flag and the start flag. The task for the robot is to map the environment while localizing itself in the map. Naturally, one of the easiest ways to construct a coordinate system is to set the goal flag as the origin and set the start flag on the x_1-axis, *i.e.* with coordinates $(d_0, 0)$. If we assume that on the image plane what we can identify is the distance of the vertical line feature to the center and the focal length of the camera is one, then the output of the system can be expressed as

$$\begin{array}{rcl} y_1 & = & \tan(\phi - \mathrm{atan}(-x_2, d_0 - x_1)) \\ y_2 & = & \tan(\phi - \mathrm{atan}(-x_2, -x_1)). \end{array}$$

C.3.1 Observability and Active Observers

As pointed out in Section C.2, observability is an *infinite-horizon* concept (Property C.3). To adapt this for the area in mind, the following is suggested.

Definition C.8 (Small-time observability)**.** *A nonlinear system, Σ_{rob}, is said to be* small-time observable *at x_1, if for any $x_2 \in \mathcal{X}$ and $T > 0$, there exists a control, $u(\cdot) \in \mathcal{U}$ and $t_0 \leq T$, such that*

$$\tilde{h}(x(t_0, x_1, u(\cdot)), s_\theta) \neq \tilde{h}(x(t_0, x_2, u(\cdot)), s_\theta).$$

It is further said to be small-time observable if it is so at every $x_1 \in \mathcal{X}$.

Remark C.1. Although not made explicit, modified versions of Definition C.8 (*i.e.* weakly/strongly small time observability) can be obtained in apparent manners.

To stress the distinction between the newly introduced definition and those of Section C.2, recall that Ω-distinguishability, the underlying concept of Definition C.4, only involves separation of points in the restricting Ω. In extension, the term "instantaneously" in Definition C.6 has to be interpreted in two senses; namely that a point can be *instantly distinguished* from its *instant neighbors*. Therefore, there is no natural setting for solely modifying Property C.3, without necessarily modifying Property C.1 and/or C.2. In contrast, small-time observability requires instant distinction of x_1 from *all* other states $x_2 \in \mathcal{X}$, or in the case of weakly small-time observability, instant distinction of x_1 from all x_2 in some open neighborhood of x_1. Hence, they *only* restrict the time-interval that have to be considered in order to find deviating output.

Given the environmental map $\theta(s_\theta)$, the sensor measurements are considered as a mapping, $\tilde{h} : \mathcal{X} \to \mathcal{Y}$. For a given measurement, $y \in \mathcal{Y}$, the inverse image of y under $\tilde{h}$ is the set of all $x \in \mathcal{X}$ such that $\tilde{h}(x, s_\theta) = y$. In general, $\mathcal{X}$ and $\mathcal{Y}$ do not have the same cardinal number so that a measurement might correspond to more that one state in $\mathcal{X}$.

Definition C.9 (Set of feasible states)**.** *The* set of feasible states *with respect to y, denoted $\mathcal{FS}_y$, is defined as the inverse image of y under $\tilde{h}$ in the state-space,* i.e.

$$\mathcal{FS}_y = \{x \in \mathcal{X} : \tilde{h}(x, s_\theta) = y\}.$$

To introduce a measure of how well a certain point in the state-space matches a given measurement, a functional or *value-function* is needed:

Definition C.10 (Value-function). *A non-negative functional,*

$$V_y \colon \mathcal{X} \to \mathbb{R}^+,$$

such that,

$$x \in \mathcal{FS}_y \iff V_y(x) = 0,$$

is called a value-function.

It is notable that Definition C.10 is well-suited for scenarios where one might have noisy measurements. In such cases, the feasible states may consist of all x, such that $V_y(x) \leq \varepsilon$, for some $\varepsilon \in \mathbb{R}^+$.

By utilizing the value-function, it is possible to drive the state estimation within the set of feasible states, $\mathcal{FS}_y$. This will be shown in greater detail in Section C.4. Next, we focus on the problem of localizing the actual state *within* this set. In order to distinguish the states in $\mathcal{FS}_y$, it is necessary that the system output do not remain constant, *i.e.* the exciting control has to be designed such that $\dot{y} \neq 0$. For each point $x_0 \in \mathcal{X}$, it is possible to associate another set to it consisting of all points that have the same output flow.

Definition C.11 (Set of output flow equivalent states). *Given any admissible control, $u(\cdot) \in \mathcal{U}$, for each state $x_0 \in \mathcal{X}$, the set of states that are* output flow equivalent to x_0 under $u(\cdot)$, *denoted $\mathcal{OF}^u_{x_0}$, is defined as all states $x_1 \in \mathcal{X}$, such that there exists $T > 0$ such that for all $t \in [0 \quad T]$,*

$$\tilde{h}(x(t, x_1, u(\cdot)), s_\theta) - \tilde{h}(x_1, s_\theta) \equiv \tilde{h}(x(t, x_0, u(\cdot)), s_\theta) - \tilde{h}(x_0, s_\theta).$$

By means of the two sets defined in this section, it is possible to put constraints on the exciting control.

Proposition C.1. *Given $x_0 \in \mathcal{X}$, if there exists an exciting control, $u_0(\cdot) \in \mathcal{U}$, and a neighborhood, $N(x_0)$ such that*

$$\mathcal{FS}_y \cap \mathcal{OF}^{u_0}_{x_0} \cap N(x_0) = \{x_0\}, \tag{C.1}$$

then the system is weakly small-time observable at x_0.

Proof. We prove by contradiction. Suppose the system is *not* weakly small-time observable at x_0, *i.e.*

$$\begin{aligned} &\exists\, x_1 \in N(x_0)\backslash x_0 \text{ and } T > 0 : \forall t \in [0 \quad T] \text{ and } \forall u(\cdot) \in \mathcal{U}, \\ &\tilde{h}(x(t, x_1, u(\cdot)), s_\theta) \equiv \tilde{h}(x(t, x_0, u(\cdot)), s_\theta). \end{aligned} \tag{C.2}$$

For the special choice of $t = 0$, Equation (C.2) gives

$$\tilde{h}(x_1, s_\theta) = \tilde{h}(x_0, s_\theta) = y, \tag{C.3}$$

meaning that $x_1 \in \mathcal{FS}_y$. Consider then the special choice of $u(\cdot) = u_0(\cdot)$, which together with Equation (C.3) and Definition C.11 implies that $x_1 \in \mathcal{OF}^{u_0}_{x_0}$. Hence we have shown that assuming (C.2) implies the existence of x_1 such that

$$\begin{aligned} &(x_1 \in N(x_0)\backslash x_0) \wedge (x_1 \in \mathcal{FS}_y) \wedge (x_1 \in \mathcal{OF}^{u_0}_{x_0}), \quad \Leftrightarrow \\ &\mathcal{FS}_y \cap \mathcal{OF}^{u_0}_{x_0} \cap N(x_0) \neq \{x_0\}. \quad \square \end{aligned}$$

Constraint (C.1) serves as the basis for design of active observers.

Proposition C.1 also provides an intuitive interpretation of the concept of small-time observability. Namely, if from the knowledge of the measurement, $y(t)$ (which defines $\mathcal{FS}_y$) and its derivative flow at time t (which defines $\mathcal{OF}_x^u$) one can uniquely determine the state, $x(t)$, then a dynamical system is said to be small-time observability.

C.4 Observer Design Study

In this section, we revisit the robot model from Example C.1. The sensor readings however will differ. It is now assumed that the robot is equipped with l range-measuring sensors, oriented at angles $\alpha_i, i = 1, \cdots, l$ with respect to ϕ. Referring to Figure C.2, sensor i measures distance ρ_i against some smooth closed curve, $\theta : S^1 \to \mathbb{R}^2$, that models the terrain. Each sensor is directed along a ray making an angle of $\phi + \alpha_i$ with the x_1-axis. Thus the outputs for the system are

$$y_i = \rho_i, \quad i = 1, \cdots, l.$$

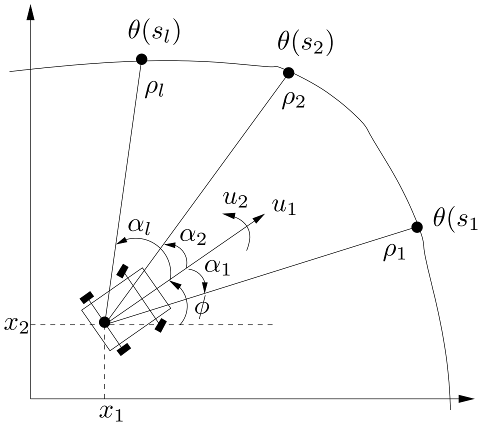


Figure C.2: The unicycle robot equipped with l range-measuring sensors.

The goal is to reconstruct the full state, x, based on sensor readings $\rho = (\rho_1, \cdots, \rho_l)$. As remarked in conjunction with Definition C.7, the objective when designing a general observer is to track $z = \Psi(x)$, rather than x itself. To this end, it is noted that the problem of reconstructing x in this particular problem, is equivalent to the reconstruction of vehicle orientation, ϕ, and the parameter values, $s_i \in S^1$, $i = 1, \cdots, l$, corresponding to the points on the curve measured against. It is so since the relative orientation angle of each sensor, α_i, is known. Setting the observer state

$$z = \Psi(x) = [s_1, \cdots, s_l, \phi]^T \in \mathcal{Z},$$

the following geometrical relationship between x and z holds:

$$\theta(s_i) = [x_1, x_2]^T + \rho_i R_{(\phi+\alpha_i)} e_1, \;\; i = 1, \cdots, l, \tag{C.4}$$

where $e_1 = [1 \quad 0]^T$ and R_α denotes the rotation matrix,

$$R_\alpha = \begin{bmatrix} \cos(\alpha) & \sin(\alpha) \\ -\sin(\alpha) & \cos(\alpha) \end{bmatrix}^T.$$

Aiming at constructing an appropriate value-function that can aid the observer design, for each sensor i, define a mapping $v_i : \mathcal{Z} \to \mathbb{R}^2$ according to

$$v_i(z) = \theta(s_i) - \rho_i R_{(\phi+\alpha_i)} e_1.$$

Intuitively, $v_i(z)$ points out where measurement i indicates that the vehicle is located in $\mathbb{R}^2$. Next, define $v_{ij} : \mathcal{Z} \to \mathbb{R}^2$ as

$$v_{ij}(z) = v_i(z) - v_j(z),$$

which indicates the difference between the vehicle location estimated by measurements ρ_i and ρ_j. Finally, the value-function is defined as

$$V_\rho(z) = \sum_{i=1}^{l-1} \sum_{j>i} v_{ij}(z)^T v_{ij}(z).$$

The non-negative value-function, $V_\rho(z)$, serves as a measure of how well z, matches a set of measurements, ρ. To see this, it is noticed that $V_\rho(z) = 0$ implies that in the observer state, z (which naturally corresponds to a state, $x \in \mathcal{X}$, by relation (C.4)), the vehicle precisely measures the distances ρ_i against the points $\theta(s_i), i = 1, \cdots, l$. In the other direction, clearly if ρ are the measured distances and z is the actual observer state, then $v_i(z) = [x_1, x_2]^T, \forall i$, and hence $V_\rho(z) = 0$. This allows us to specify the set of feasible states by means of the value-function, as discussed earlier.

As for the set of output flow equivalent states, from (C.4) we obtain

$$0 = \theta'(s_i)^T R_\phi (u_1 e_2 + \dot{\rho}_i R_{\alpha_i} e_2 - \rho_i u_2 R_{\alpha_i} e_1) \triangleq Q_i,$$

by first differentiating with respect to time and then multiplying by $\theta'(s_i)^T R_{\frac{\pi}{2}}$ from the left. Then, the mapping defined by

$$Q = [Q_1, \cdots, Q_l]^T = 0,$$

characterizes the set of output flow equivalent states for this system. Under suitable assumptions on the exciting control, the sensor orientations and the environmental map (see [15] for details), it can be shown that this set and the set of feasible states together fulfill the condition of Proposition C.1, which implies that we are bound to have weakly small-time observability.

Putting it all together, the following observer dynamics is proposed for this particular problem:

$$\dot{z} = -k_V \left[\frac{\partial V_\rho}{\partial z}\right]^T - k_Q \left[\frac{\partial Q}{\partial z}\right]^T Q, \tag{C.5}$$

where $k_V, k_Q > 0$ are suitably chosen observer gains.

To complete the observer design, the full observer mapping, Φ, is to be decided (*cf.* Figure C.1). By relation (C.4), any parameter value, s_i, together with ϕ, suffice for reconstructing x. Thus there are several choices for Φ. However, in the case of faulty measurements, different parameter values might give inconsistent state estimation, why for instance a simple vector average can be chosen.

The following proposition addresses the convergence properties of the proposed observer.

Proposition C.2. *Assume the control and it's first derivative are both bounded. Then the estimation error of the proposed observer* (C.5) *is locally bounded and can be made arbitrarily small by tuning the observer parameters, k_V and k_Q.*

Proof. Set $k = k_V = k_Q$. Consider then the time dependent Lyapunov function candidate

$$V(z,t) = V_\rho + \frac{1}{2} Q^T Q \in \mathbb{R}^+,$$

which is locally positive definite around the true trajectory. The time dependency originates from the collective contribution of u, θ, ρ and their rates of change. Due to the differentiability assumptions (on u and the smoothness of the other mappings), V is also continuously differentiable and decrescent. The total derivate of V along the observer dynamics (C.5) becomes

$$\dot{V} = \frac{\partial V}{\partial z}\dot{z} + \frac{\partial V}{\partial t} = \Big[\frac{\partial V_\rho}{\partial z} + Q^T \frac{\partial Q}{\partial z}\Big]\dot{z} + K(t) = -k \| \frac{\partial V_\rho}{\partial z} + Q^T \frac{\partial Q}{\partial z} \|^2 + K(t).$$

Here, $K(t)$ denotes the collective contribution of u, θ, ρ and their rates of change. By the differentiability assumptions, this term is bounded. Since the condition of Proposition C.1 is fulfilled, the first term is locally negative definite. Therefore, the size of the design parameter k, bounds and tunes the estimation error. □

C.5 Simulations

In this section we consider the case when the robot is equipped with two range-measuring sensors ($l = 2$) and moves inside the same elliptic field as considered in Example C.1. In what follows, x and z will denote the true states while $\hat{x}$ and $\hat{z}$ will denote the estimations of them. All true states will be plotted with blue/dashed lines, while estimations will be graphed in red/solid. The robot starts off from $x(0) = [1,\ -1,\ \frac{\pi}{2}]^T$, which corresponds to $z(0) = [\frac{23\pi}{4},\ \frac{242\pi}{1101},\ \frac{\pi}{2}]^T$ in the $\mathcal{Z}-$space. The observer is initialized at $\hat{z}(0)$, a *randomly* generated point in the vicinity of $z(0)$. The observer gains are set to $k_V = 5, k_Q = 1$.

Figure C.3 shows the trajectory of the components of $z(t)$ (in dashed/blue) and $\hat{z}(t)$ (in solid/red). This, together with Figure C.4, where the relative errors have been plotted, shows the convergence of the observer in the $\mathcal{Z}$-space.

Of more practical importance however, is the convergence of $\hat{x}(t)$ to $x(t)$ in the state-space, $\mathcal{X}$. Figures C.5 and C.6 show the observation and relative errors as measured after mapping $\hat{z}$ into $\hat{x}$ by means of the full observer mapping, Φ.

Noisy Measurements

Next, attention is paid to the case when the presence of measurement noise is recognized. The noise parameter has been chosen such that the relative measurement errors amount to approximately 5%. Referring to Figure C.7, it can be noted how the observer rejects the disturbance and tracks the true observer state quite well even in the presence of measurement errors. This statement is verified when considering the time history of $x(t)$ and $\hat{x}(t)$ in the state-space (Figure C.8). In cases when (C.4) is inconsistent for $i = 1$ and 2, a simple vector average has served as the estimated position. One desirable property of this choice is that a true measurement from one sensor can be used constructively to compensate for the faulty measurement of the other one. Thus we notice in Figure C.8 that, in the presence of measurement noise, the position estimation is much better than the estimation of the orientation angle, ϕ.

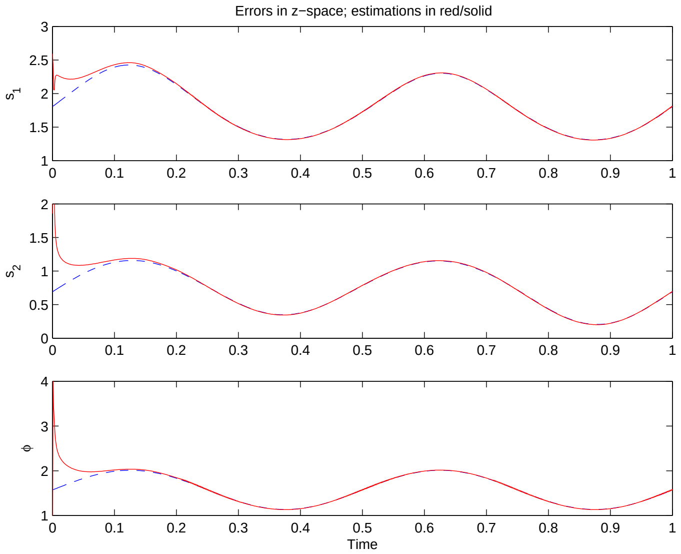


Figure C.3: Observation error in $\mathcal{Z}$-space.

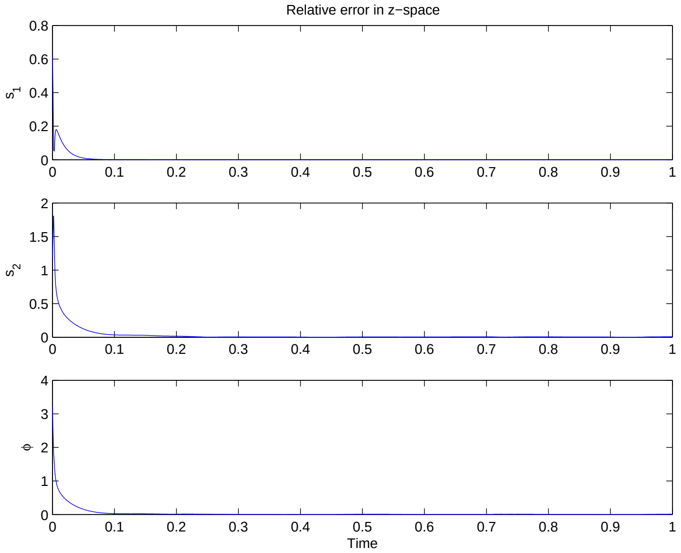


Figure C.4: Relative error in $\mathcal{Z}$-space.

C.6 Concluding Remarks

In this paper, the extension of the concepts of observability and observer design to the field of mobile robotics is considered. Such systems have several distinguishing features. Firstly, mobile robots are typically non-uniformly observable systems so that the observer gains, as well as its convergence properties will depend on the system input. In addition, because of the interaction of the exteroceptive sensors with the environment, the convergence of the observer typically will also depend on the environment. Therefore, in order to succeed in

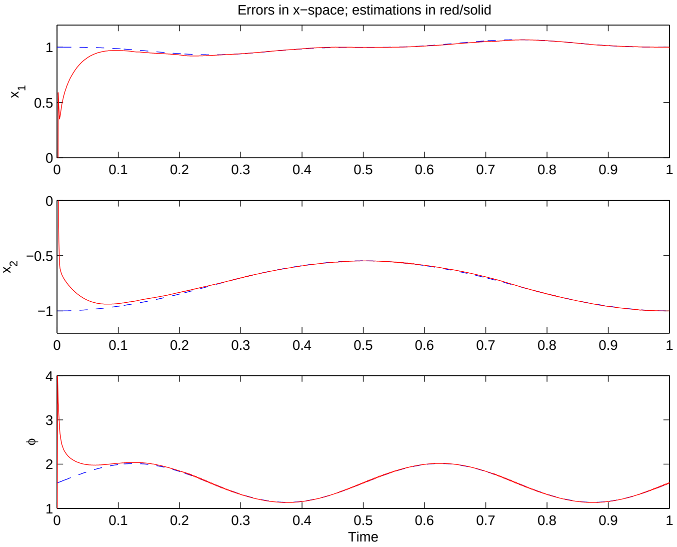


Figure C.5: Observation error in $\mathcal{X}$-space.

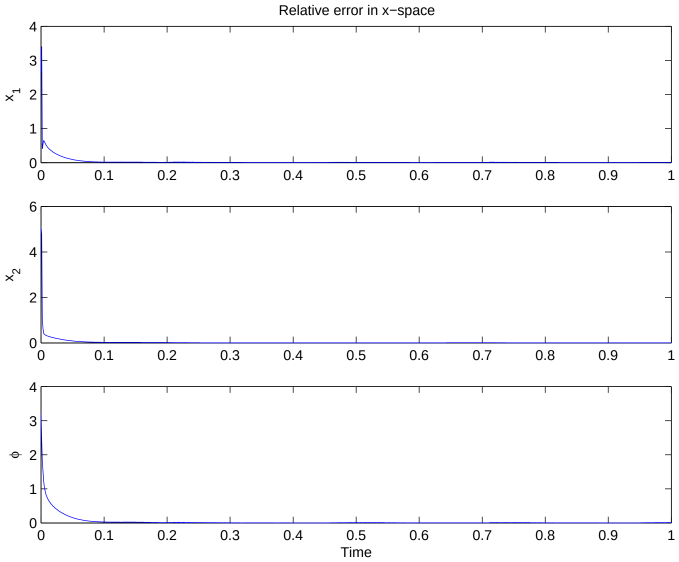


Figure C.6: Relative error in $\mathcal{X}$-space.

reconstructing the state, the exciting control has to be chosen in a deliberate manner, *i.e.* an *active observer* has to be designed. Finally, since most existing observer design techniques are only applicable to uniformly observable nonlinear systems, alternative approaches that aid the observer design are needed. The set of feasible configurations, its relation with the value-function, the set of output flow equivalent states, and the inter-relation between these two sets, provide such a setting. The design study presented here-within, serves to illustrate the use of these concepts in the observer design process.

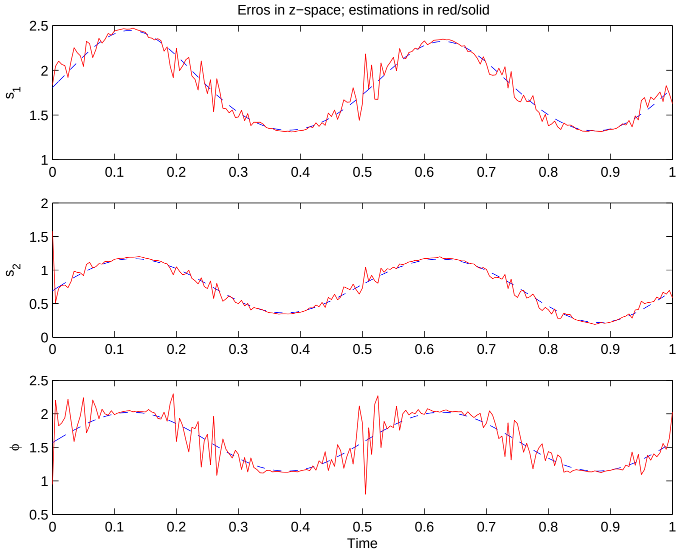


Figure C.7: Observation error in $\mathcal{Z}$-space with noise.

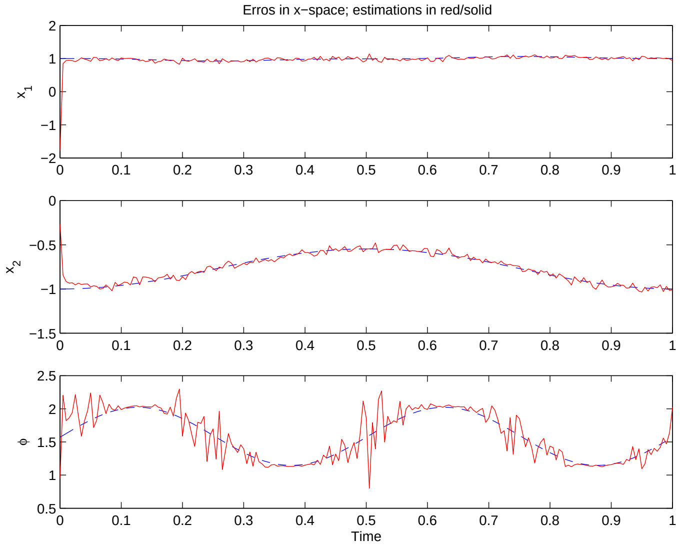


Figure C.8: Observation error in $\mathcal{X}$-space with noise.

C.7 References

[1] Thau, F., "Observing the state of non-linear dynamic systems," *International Journal of Control*, Vol. 17, 1973, pp. 471–479.

[2] Kou, R., S., Elliott, D. L., and Tarn, T. J., "Exponential observers for nonlinear dynamic systems," *Information and Control*, Vol. 29, No. 3, 1975, pp. 204–216.

[3] Krener, A. J. and Respondek, W., "Nonlinear observers with linearizable error dynamics," *SIAM J. Control Optim.*, Vol. 23, No. 2, 1985, pp. 197–216.

[4] Van der Schaft, A., "On nonlinear observers," *IEEE Trans. Automat. Control*, Vol. AC-30, No. 12, Dec. 1985, pp. 1254–1256.

[5] Xia, X. and Gao, W., "On exponential observers for nonlinear systems," *Systems and Control Letters*, Vol. 11, 1988, pp. 319–325.

[6] Tornambé, A., "Use of asymptotic observers having high-gains in the state and parameter estimation," *Proc. of the* 28^{th} *IEEE Conference on Decision and Control*, Dec. 1989, pp. 1791–1794.

[7] Khalil, H. K., "High-gain observers in nonlinear feedback control," *Lecture Notes in Control and Information Sciences*, edited by H. Nijmeijer and T. Fossen, Vol. 244, Springer Verlag, 1999, pp. 249–268.

[8] Lin, W., Baillieul, J., and Bloch, A., "Call for papers for the special issue on new directions in nonlinear control," *IEEE Trans. Automat. Control*, Vol. 47, No. 3, 2002, pp. 543–544.

[9] Eklundh, J., Uhlin, T., Nordlund, P., and Maki, A., "Active vision and seeing robots," *The* 7^{th} *Symp. on Robotics Res.*, edited by G. Giralt and G. Hirzinger, Springer Verlag, Berlin, 1996, pp. 543–544.

[10] Hu, X. and Ersson, T., "Active state estimation of nonlinear systems," *Automatica*, Vol. 40, No. 12, Dec. 2004, pp. 2075–2082.

[11] Sussmann, H. J., "Single-input observability of continuous-time systems," *Math. Systems Theory*, Vol. 12, No. 4, 1979, pp. 371–393.

[12] Hermann, R. and Krener, A. J., "Nonlinear controllability and observability," *IEEE Trans. Automat. Control*, Vol. AC-22, No. 5, 1977, pp. 728–740.

[13] Respondek, W., "Introduction to geometric nonlinear control; linearization, observability, decoupling," *Mathematical control theory, Part 1, 2 (Trieste, 2001)*, ICTP Lect. Notes, VIII, Abdus Salam Int. Cent. Theoret. Phys., Trieste, 2002, pp. 169–222 (electronic).

[14] Anisi, D. A. and Hamberg, J., "Riemannian observers for Euler-Lagrange systems," *Proc. of the* 16^{th} *IFAC World Congress*, Prague, Czech Republic, July 2005.

[15] Cederwall, S. and Hu, X., "Active nonlinear observers for mobile systems," *Proc. of the* 43^{rd} *IEEE Conference on Decision and Control*, Bahamas, Dec. 2004, pp. 3898–3902.

Paper D

Riemannian Observers for Euler-Lagrange Systems

Riemannian Observers for Euler-Lagrange Systems

David A. Anisi and Johan Hamberg

Abstract

In this paper, a *geometrically intrinsic* observer for Euler-Lagrange systems is defined and analyzed. This observer is an generalization of the observer recently proposed by Aghannan and Rouchon. Their contractivity result is reproduced and complemented by a proof that the region of contraction is infinitely thin. However, assuming *a priori* bounds on the velocities, convergence of the observer is shown by means of Lyapunov's direct method in the case of configuration manifolds with constant curvature. The convergence properties of the observer are illustrated by an example where the configuration manifold is the three-dimensional sphere, S^3.

Keywords: Nonlinear Observers, Intrinsic Observers, Differential Geometric Methods, Euler-Lagrange Systems, Contraction Analysis, Nonlinear Systems Theory.

D.1 Introduction

Feedback control design techniques require knowledge about at least some parts of the state vector. If all the state variables necessary for the control system can not be directly measured, which is a typical situation in complex systems, we must aim at obtaining an *estimate* of the unknown state variables. For a dynamical system, an *observer* is another dynamical system whose task is to reconstruct missing state information, while only using available measurements. The input to the observer is the output of the original system (which may include its input), and the observer is expected to produce as output an estimate of some state function of the original system.

Consider the nonlinear dynamical system

$$\Sigma : \begin{cases} \dot{z} & = & \mathcal{F}(z,u) \\ y & = & h(z,u) \end{cases}$$

with state $z \in \mathcal{Z}$, control $u \in \mathcal{U}$ and output $y \in \mathcal{Y}$. Here, $\mathcal{Z}$,$\mathcal{U}$ and $\mathcal{Y}$ are smooth manifolds. All mappings in this paper, are assumed to be smooth.

Definition D.1 (Observer). *A dynamical system with state manifold $\mathcal{W}$, input manifold $\mathcal{Y}$, together with a mapping $\hat{\mathcal{F}} : (\mathcal{W} \times \mathcal{Y}) \to T\mathcal{W}$ is an* observer *for the system Σ, if there exists a smooth mapping $\Psi : \mathcal{Z} \to \mathcal{W}$, such that the diagram shown in Figure D.1 (the dashed arrow excluded), commutes. The observer gives a* full state reconstruction *if there is a mapping $Z : (\mathcal{W} \times \mathcal{Y}) \to \mathcal{Z}$ such that the full diagram in Figure D.1 is commutative (*cf. *[1] and [2]).*

$$\begin{tikzcd}
 & T\mathcal{Z} \arrow[dl, "\tau_{\mathcal{Z}}"'] \arrow[r, "\Psi_*"] & T\mathcal{W} \arrow[dr, "\tau_{\mathcal{W}}"] & \\
\mathcal{Z} & & & \mathcal{W} \\
 & \mathcal{Z} \times \mathcal{U} \arrow[ul, "\pi_{\mathcal{Z}}"] \arrow[u, "\mathcal{F}"'] \arrow[r, "(\Psi \circ \pi_{\mathcal{Z}}, h)"] & \mathcal{W} \times \mathcal{Y} \arrow[u, "\hat{\mathcal{F}}"] \arrow[ur, "\pi_{\mathcal{W}}"'] \arrow[ull, dashed, bend left, "Z"] &
\end{tikzcd}$$

Figure D.1: Commutative diagram defining an observer.

In diagram D.1, Ψ_* denotes the tangent mapping, π is projection upon a Cartesian factor, while τ denotes the projection of the tangent bundle.

According to definition D.1, the objective when designing a general observer, is to track $\Psi(z)$, rather than z itself. The special case when Ψ equals the identity mapping and $\mathcal{W} = \mathcal{Z}$, is often referred to as an *identity observer.* Also, note that the same observer dynamics, $\hat{\mathcal{F}}$, may allow several *different* full observer mappings, Z, and that in general, a full state observer

$$\hat{\Sigma} : \begin{cases} \dot{w} &= \hat{\mathcal{F}}(w, y) \\ \hat{z} &= Z(w, y) \end{cases}$$

may *not* be put in the form $\dot{\hat{z}} = \Xi(\hat{z}, y)$.

As a consequence of this definition, an observer has the following property:

Property D.1. *$w(t_0) = \Psi(z(t_0))$ at some time instance t_0, yields $w(t) = \Psi(z(t))$ for all $t \geq t_0$.*

It is also reasonable to require the additional property:

Property D.2. *As time proceeds, the trajectories $w(t)$ and $\Psi(z(t))$ converges*[1] *for every input.*

This second property, *i.e.* the convergence properties of the observer, may be demonstrated in different ways. If G is a Riemannian metric on $\mathcal{W}$, whose Lie derivative along the vector field $\hat{\mathcal{F}}$, is negative for every input, y, ($\mathcal{L}_{\hat{\mathcal{F}}} G < 0$), then the Riemannian distance between any two trajectories tends to zero (*cf.* [3]). This is a property of the control system $\mathcal{W}$ alone. In conjunction with property D.1, this implies property D.2. More precisely, we have that

$$\frac{\mathrm{d}}{\mathrm{d}t} \int_{\Upsilon^t_{\hat{\mathcal{F}}} \rho_0} \mathrm{d}s = \int_{\Upsilon^t_{\hat{\mathcal{F}}} \rho_0} \frac{1}{2} (\mathcal{L}_{\hat{\mathcal{F}}} G)(\frac{\mathrm{d}\tau}{\mathrm{d}s}, \frac{\mathrm{d}\tau}{\mathrm{d}s}) \mathrm{d}s,$$

[1] *"Convergence" in some metric sense, or – for relatively compact trajectories – in a purely topological sense.*

so if $\mathcal{L}_{\hat{\mathcal{F}}}G < 0$, then

$$\inf \int_{w_1(t)}^{w_2(t)} \mathrm{d}s \leq \int_{\Upsilon^t_{\hat{\mathcal{F}}}\rho_0} \mathrm{d}s \leq \int_{\rho_0} \mathrm{d}s \triangleq \inf \int_{w_1(0)}^{w_2(0)} \mathrm{d}s.$$

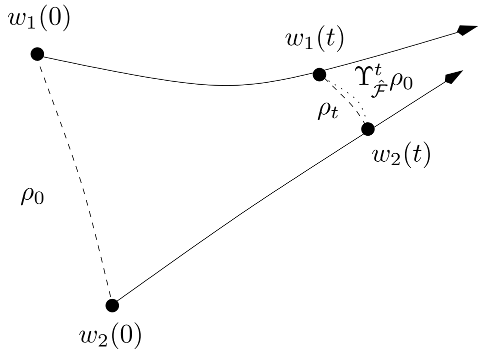


Figure D.2: The length of the geodesic curve ρ_t, between two trajectories decreases if $\mathcal{L}_{\hat{\mathcal{F}}}G < 0$.

However, the assumption that the observer dynamics is contractive, is very restrictive and in most cases, property D.2 has to be shown by means of direct Lyapunov methods.

In this paper, we study the observer design problem for a class of nonlinear systems, *viz.* Euler-Lagrange systems, where the output of the system is assumed to be the generalized position and force, and the goal is to reconstruct the generalized velocities. An often practiced solution to the problem of reconstructing the velocity variables is to numerically differentiate the known position measurements. This approach however, fails to perform for high and fast varying velocities, but naturally also when the position measurements are corrupted by noise.

The Euler-Lagrange equations are *intrinsic* and may be written in a coordinate-free way [4]. It is natural to keep this coordinate independence in the observer design as well. The Riemannian geometric point of view has influenced part of control theory, *e.g.* optimal control and control design. However, the impact on observer design, has been modest.

In [5], the authors successfully adopt the formerly mentioned contraction analysis approach to address convergence of an intrinsic observer for Euler-Lagrange systems with position measurements. These results have been specialized to the case of left invariant systems on Lie groups in [6]. In the present paper, we extend the results of [5] by using Lyapunov theory to show convergence in the constant curvature case, whenever we have *a priori* given bounds on the state variables. In the case of physical (mechanical or electrical) Euler-Lagrange systems, this assumption is a realistic one.

The organization of this paper is as follows. Section D.2 is devoted to introducing some preliminary concepts of tangent bundle geometry (Section D.2.1) and Euler-Lagrange systems (Section D.2.2). The design of the observer is the subject of Section D.3, while Section D.4 is devoted to the convergence properties of it. Finally, these properties are illustrated in Section D.5, where we present some simulation results.

D.2 Preliminaries

D.2.1 Tangent Bundle Geometry

This paper assumes a previous knowledge of classical tensor analysis as well as familiarity with coordinate-free concepts like tangent bundle, Lie derivatives and affine connections (consult *e.g.* [7] or [8]). Throughout the paper, Einstein summation convention is used (*i.e.* we sum over all indices appearing repeatedly), partial derivatives are indicated with a comma, $U_{,i} = \frac{\partial U}{\partial x^i}$, while covariant derivatives of contravariant tensors are indicated with a bar, $F^i_{\ |j} = F^i_{\ ,j} + \Gamma^i_{kj} F^k$. If g_{ij} are the components of a Riemannian metric, g^{ij} denotes the components of the dual ("inverse") metric, and the components of the Levi-Civita connections (the Christoffel symbols) are given by $\Gamma^i_{jk}(x) = \frac{1}{2} g^{il}(g_{lj,k} + g_{kl,j} - g_{jk,l})$. By $\operatorname{grad} U$, we mean the vector field $g^{ij} U_{,j} \frac{\partial}{\partial x^i}$. The curvature tensor, R, is defined by

$$R(X,Y)Z = (\nabla_X \nabla_Y - \nabla_Y \nabla_X - \nabla_{[X,Y]})Z.$$

With the index ordering conventions from [7], the type $(1,3)$ tensor R has components

$$R_{j\ kl}^{\ i} \frac{\partial}{\partial x^i} = R(\frac{\partial}{\partial x^l}, \frac{\partial}{\partial x^k}) \frac{\partial}{\partial x^j},$$

so that

$$R_{m\ jk}^{\ i} = \Gamma^i_{mj,k} - \Gamma^i_{mk,j} + \Gamma^i_{nk} \Gamma^n_{mj} - \Gamma^i_{nj} \Gamma^n_{mk},$$

and the the Ricci identity,

$$Y^i_{\ |j|k} - Y^i_{\ |k|j} = R_{m\ jk}^{\ i} Y^m, \tag{D.1}$$

holds[2].

We now review some less well-known constructions, namely lifting geometrical structures on a manifold $\mathcal{X}$ to geometrical structures on its tangent bundle, $T\mathcal{X}$ (*cf.* [9]). These operations will be helpful while studying the convergence properties of the observer through contraction analysis (section D.4.3). Let x be local coordinates on $\mathcal{X}$ and (x, v) the corresponding induced coordinates on $T\mathcal{X}$.

- The *vertical lift* of a vector field $Y = Y^i \frac{\partial}{\partial x^i}$ on $\mathcal{X}$, is the vector field on $T\mathcal{X}$ given by $Y^{\mathbb{V}} = Y^i \frac{\partial}{\partial v^i}$.
- The *horizontal lift* of Y depends on the choice of a connection and is the vector field on $T\mathcal{X}$ given by $Y^{\mathbb{H}} = Y^i (\frac{\partial}{\partial x^i} - \Gamma^m_{li} v^l \frac{\partial}{\partial v^m})$.
- The *geodesic spray* is a vector field $\overset{\nabla}{Z}$ on $T\mathcal{X}$, uniquely constructed from a connection ∇ on $\mathcal{X}$ as $\overset{\nabla}{Z} = v^i \frac{\partial}{\partial x^i} - \Gamma^i_{jk} v^j v^k \frac{\partial}{\partial v^i}$.

If ϕ is a differential form on $\mathcal{X}$, $\tau^* \phi$ denotes its pullback to $T\mathcal{X}$. A differential 1-form ϕ on $\mathcal{X}$, also defines a scalar function $\mathbb{I}(\phi)$ on $T\mathcal{X}$ given by $\mathbb{I}(\mathrm{d}x^i) = v^i$ (This notation is not standard. The letter $\mathbb{I}$ stands for *identification*, since a covector ϕ, in a sense, already *is* a function on the tangent vectors). The $\mathbb{I}$ construction extends to higher order tensors.

Expressions for the bracket between these lifted vector fields are listed in Table D.1. These expressions will be used for the component–wise contraction analysis in Section D.4.3.

[2] It holds whenever the connection is torsion-free, which the Levi-Civita connections is.

$[\cdot,\cdot]$	$Y^{\mathbb{H}}$	$Y^{\mathbb{V}}$	$\overset{\triangledown}{Z}$
$X^{\mathbb{H}}$	$[X,Y]^{\mathbb{H}}$	$-(\nabla_X Y)^{\mathbb{V}}$	$\mathbb{I}((R(\cdot,X)-\nabla X))^{\mathbb{V}}$
$X^{\mathbb{V}}$	$-[Y^{\mathbb{H}},X^{\mathbb{V}}]$	0	$X^{\mathbb{H}}-\mathbb{I}(\nabla X)^{\mathbb{V}}$
$\overset{\triangledown}{Z}$	$-[Y^{\mathbb{H}},\overset{\triangledown}{Z}]$	$-[Y^{\mathbb{V}},\overset{\triangledown}{Z}]$	0

Table D.1: Brackets of the lifted vector fields.

Given a Riemannian metric, g, on $\mathcal{X}$, there is a family of natural metrics on $T\mathcal{X}$ given by

$$G(X^{\mathbb{H}}+Y^{\mathbb{V}},X^{\mathbb{H}}+Y^{\mathbb{V}})=\tau^*(ag(X,X)+bg(Y,Y)+2cg(X,Y)),$$

where a,b and c are constants, or in general, functions of $g_{ij}v^iv^j$. The case $a=b=1$ and $c=0$, was studied in [10]. The generalized Sasaki metric reads

$$G=\begin{bmatrix}\mathrm{d}x^i\\ \mathrm{D}v^i\end{bmatrix}^T\otimes\begin{bmatrix}a & c\\ c & b\end{bmatrix}g_{ij}\begin{bmatrix}\mathrm{d}x^j\\ \mathrm{D}v^j\end{bmatrix},$$

where $\mathrm{D}v^i=\mathrm{d}v^i+\Gamma^i_{jk}v^j\mathrm{d}x^k$. Here, $[\mathrm{d}x^i,\mathrm{D}v^i]$ is the coframe dual to the frame $[\frac{\partial}{\partial x^i}^{\mathbb{H}},\frac{\partial}{\partial x^i}^{\mathbb{V}}]$.

At the origin of a geodesic normal coordinate system, the Lie derivatives of the coframe, equal

$$\begin{aligned}
\mathcal{L}_{Y^{\mathbb{V}}}\begin{bmatrix}\mathrm{d}x^i\\ \mathrm{D}v^i\end{bmatrix} &= \left[\begin{array}{c|c}0 & 0\\ \hline Y^i_{\ |j} & 0\end{array}\right]\begin{bmatrix}\mathrm{d}x^j\\ \mathrm{D}v^j\end{bmatrix} \qquad \text{(D.2)}\\
\mathcal{L}_{Y^{\mathbb{H}}}\begin{bmatrix}\mathrm{d}x^i\\ \mathrm{D}v^i\end{bmatrix} &= \left[\begin{array}{c|c}Y^i_{\ |j} & 0\\ \hline R_{k\ jl}^{\ i}\ v^k\ Y^l & 0\end{array}\right]\begin{bmatrix}\mathrm{d}x^j\\ \mathrm{D}v^j\end{bmatrix}\\
\mathcal{L}_{\overset{\triangledown}{Z}}\begin{bmatrix}\mathrm{d}x^i\\ \mathrm{D}v^i\end{bmatrix} &= \left[\begin{array}{c|c}0 & \delta^i_j\\ \hline R_{k\ jl}^{\ i}\ v^k\ v^l & 0\end{array}\right]\begin{bmatrix}\mathrm{d}x^j\\ \mathrm{D}v^j\end{bmatrix}\\
\mathcal{L}_{\mathbb{I}(R(\cdot,Y))^{\mathbb{V}}}\begin{bmatrix}\mathrm{d}x^i\\ \mathrm{D}v^i\end{bmatrix} &= \left[\begin{array}{c|c}0 & 0\\ \hline (R_{m\ kl}^{\ i}Y^l)_{|j}\ v^k\ v^m & (R_{j\ kl}^{\ i}+R_{k\ jl}^{\ i})Y^lv^k\end{array}\right]\begin{bmatrix}\mathrm{d}x^j\\ \mathrm{D}v^j\end{bmatrix}.
\end{aligned}$$

D.2.2 Euler-Lagrange Systems

Consider a system with generalized position coordinates, x, and generalized velocities, v, for which we are able to define kinetic- and potential energy. For such systems, the following scalar function,

$$L(x,v)\triangleq T(x,v)-U(x),$$

where $T(x,v)$ and $U(x)$ denote the kinetic- and potential energy respectively, defines the *Lagrangian* of the system. In this work, we focus our attention on systems whose kinetic energy function is of the form

$$T(x,v)=\frac{1}{2}g_{ij}(x)v^iv^j,$$

where g_{ij} is a Riemannian metric on the configuration manifold, $\mathcal{X}$. The Euler-Lagrange differential equations, which we assume govern the motion of the considered system, define a dynamical system on the state space, $\mathcal{Z} = T\mathcal{X}$, the tangent bundle of the configuration manifold, $\mathcal{X}$, and are given by

$$\dot{x}^i = v^i, \quad \frac{\mathrm{d}}{\mathrm{d}t}(\frac{\partial L}{\partial v^i}) - \frac{\partial L}{\partial x^i} = g_{ij}F^j, \qquad i = 1, \ldots, n,$$

where F^j are the external forces acting on the system, which *may* be interpreted as the input. We further assume that we have direct measurements on the position variables and forces. Combining this with the expression for the Lagrangian, the system can be written, in local coordinates, as

$$\Sigma_0 : \left\{ \begin{array}{rcll} \dot{x}^i & = & v^i, & i = 1, \ldots, n \\ \dot{v}^i & = & -\Gamma^i_{jk}(x)v^j v^k - g^{ij}U_{,j} + F^i & \\ y & = & h(x, v, F) = (x, F). & \end{array} \right.$$

In terms of the absolute time-derivative, $\mathrm{D}_t v^i = \frac{\mathrm{d}v^i}{\mathrm{d}t} + \Gamma^i_{jk} v^j \frac{\mathrm{d}x^k}{\mathrm{d}t}$, system Σ_0 can equivalently be written as

$$\boxed{\Sigma : \left\{ \begin{array}{rcl} \dot{x}^i & = & v^i \\ \mathrm{D}_t v^i & = & -g^{ij}U_{,j} + F^i \\ y & = & (x, F). \end{array} \right.}$$

Using the introduced lifting operations, the dynamics of system Σ, is given by the vector field

$$\boxed{\mathcal{F} = \overset{\nabla}{Z} - (\operatorname{grad} U)^{\mathbb{V}} + F^{\mathbb{V}}}$$

D.3 Observer Design

For the class of systems, Σ, described in Section D.2.2, we now introduce a full state identity observer, $\hat{\Sigma}$.

Referring to Figure D.3, we let (ξ, η) denote the state of the observer, $S(x, \xi) = \frac{1}{2}\mathrm{dist}(x, \xi)^2$, $S_\beta = \frac{\partial S}{\partial \xi^\beta}$ and $R^\alpha = \mathbb{I}(R(\cdot, \operatorname{grad} S))^\alpha = R_\beta{}^\alpha{}_{\gamma\iota}\eta^\beta S^\gamma \eta^\iota$, where $R_\beta{}^\alpha{}_{\gamma\iota}$ is the curvature tensor. In addition, Φ^α denotes the *parallel transport* of F^i along the geodesic curve, ρ, from x to ξ. The parallel transport operator, K^i_α, has the following properties, which are easily verified in Fermi coordinates:

$$K^i_{\alpha|\beta} S^\beta = 0. \tag{D.3}$$

$$K^i_\alpha S^\alpha = -S^i \tag{D.4}$$

$$K^i_\alpha K^j_\beta g^{\alpha\beta} = g^{ij} \tag{D.5}$$

Upon introducing this notation, the following observer dynamics is suggested for Σ:

$$\begin{array}{rcll} \dot{\xi}^\alpha & = & \eta^\alpha - A g^{\alpha\beta} S_\beta, & \alpha = 1, \ldots, n \\ \mathrm{D}_t \eta^\alpha & = & -B g^{\alpha\beta} S_\beta - g^{\alpha\beta} U_{,\beta} + \Phi^\alpha + C R^\alpha, & \end{array} \tag{D.6}$$

where A, B and C are observer gains, possibly depending on S and $|\eta|_g$. Note that when $\xi = x$, then $S_\beta = 0$ and $K^i_\alpha = \delta^i_\alpha$ (the Kronecker delta), hence (D.6) satisfies the diagram property of definition D.1. As observer output mapping, Z, we may for instance use $Z_1 =$

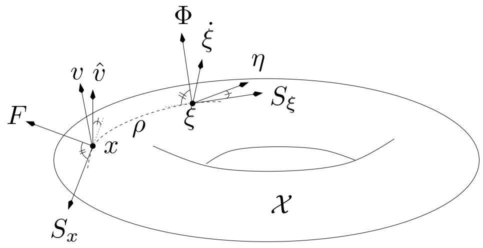


Figure D.3: The system- and observer variables, are denoted by Latin and Greek letters respectively.

$(\xi^\alpha, \eta^\alpha)$, or $Z_2 = (x^i, K^i_\alpha \eta^\alpha)$. Choosing the latter approach[3], the velocities, v^i, are estimated as

$$\hat{v}^i = K^i_\alpha \eta^\alpha. \tag{D.7}$$

Thus, putting (D.6) and (D.7) together, the following observer, $\hat{\Sigma}$, is suggested for Σ

$$\boxed{\hat{\Sigma} : \begin{cases} \dot{\xi}^\alpha &= \eta^\alpha - A g^{\alpha\beta} S_\beta \\ \mathrm{D}_t \eta^\alpha &= -B g^{\alpha\beta} S_\beta - g^{\alpha\beta} U_{,\beta} + \Phi^\alpha + C R^\alpha \\ \hat{v}^i &= K^i_\alpha \eta^\alpha \end{cases}}$$

Using the introduced lifting operators, the dynamics of the observer is governed by the vector field

$$\boxed{\hat{\mathcal{F}} \;=\; \overset{\nabla}{Z} - A(\mathrm{grad}\, S)^{\mathbb{H}} - B(\mathrm{grad}\, S)^{\mathbb{V}} - (\mathrm{grad}\, U)^{\mathbb{V}} + \Phi^{\mathbb{V}} + C\mathcal{R}}$$

where $\mathcal{R} = R^\alpha \frac{\partial}{\partial v^\alpha}$. It is notable that in the case of flat metric, $\hat{\Sigma}$ reduces to the well-known Luenberger observer.

The observer $\hat{\Sigma}$, is essentially the same as the one introduced in [5], see also [6]. We here allow the observer gains to vary and have a choice of moving force terms between U and F, which are treated differently in our observer. This latter freedom will however not be exploited in the present paper. In Section D.4.3, we follow [5], by choosing $C = 1$ and the output mapping Z_1, while in Section D.4.4 we use a general C and Z_2.

D.4 Convergence Analysis

In this section, convergence issues are treated by means of contraction analysis (Section D.4.3) and, in the case of constant curvature, by means of a conventional Lyapunov method (Section D.4.4). To this end however, we devote Section D.4.1 and D.4.2 to deriving expressions for the variation of some quantities along a geodesic.

D.4.1 Transport Equations

Letting $S^\alpha = g^{\alpha\beta} S_\beta$, the Hamilton-Jacobi equation $\sigma_{|\alpha}\sigma_{|\beta} g^{\alpha\beta} = 1$, for $\sigma = \sqrt{2S}$, implies

$$S_\alpha S^\alpha - 2S = 0, \tag{D.8}$$

[3] In the face of noisy measurements, it might be advantageous to consider Z_1 instead.

Taking the covariant derivative of (D.8), utilizing the fact that the Levi-Civita connection is torsion-free ($S_{\alpha|\beta} = S_{\beta|\alpha}$) and raising the first index, we have

$$S^{\beta}{}_{|\alpha} S^{\alpha} - S^{\beta} = 0. \tag{D.9}$$

Then, combining (D.3), (D.4), (D.8) and (D.9), it follows that

$$S^{i}{}_{|\beta} S^{\beta} - S^{i} = 0. \tag{D.10}$$

The covariant derivative of (D.9) gives

$$S^{\beta}{}_{|\alpha|\gamma} S^{\alpha} + S^{\beta}{}_{|\alpha} S^{\alpha}{}_{|\gamma} - S^{\beta}{}_{|\gamma} = (S^{\beta}{}_{|\alpha|\gamma} - S^{\beta}{}_{|\gamma|\alpha}) S^{\alpha} + S^{\beta}{}_{|\gamma|\alpha} S^{\alpha} + S^{\beta}{}_{|\alpha} S^{\alpha}{}_{|\gamma} - S^{\beta}{}_{|\gamma} = 0.$$

Utilizing Ricci's identity, $S^{\beta}{}_{|\alpha|\gamma} - S^{\beta}{}_{|\gamma|\alpha} = R_{\iota}{}^{\beta}{}_{\alpha\gamma} S^{\iota}$, we get

$$S^{\beta}{}_{|\gamma|\alpha} S^{\alpha} = S^{\beta}{}_{|\gamma} - R_{\iota}{}^{\beta}{}_{\alpha\gamma} S^{\iota} S^{\alpha} - S^{\beta}{}_{|\alpha} S^{\alpha}{}_{|\gamma}. \tag{D.11}$$

In a similar fashion, we obtain

$$K^{i}_{\beta|\gamma|\alpha} S^{\alpha} = R_{\beta}{}^{\epsilon}{}_{\alpha\gamma} K^{i}_{\epsilon} S^{\alpha} + K^{i}_{\beta|\alpha} S^{\alpha}{}_{|\gamma}. \tag{D.12}$$

It should be possible to derive Grönwall-like estimates of $S^{\beta}{}_{|\gamma}$ and $K^{i}_{\beta|\gamma}$ from (D.11) and (D.12). In the present paper, however, we focus on spaces of constant curvature.

D.4.2 Constant Curvature

In the case when $\mathcal{X}$ has constant curvature, *i.e.* when

$$R_{\iota}{}^{\beta}{}_{\alpha\gamma} = \kappa(\delta^{\beta}_{\gamma} g_{\iota\alpha} - \delta^{\beta}_{\alpha} g_{\iota\gamma}), \tag{D.13}$$

equation (D.11) may be explicitly solved for $S^{\beta}{}_{|\gamma}$ by means of the Ansatz

$$S^{\beta}{}_{|\gamma} = \Upsilon_1(S)\delta^{\beta}_{\gamma} + \Upsilon_2(S) S^{\beta} S_{\gamma}. \tag{D.14}$$

Multiplying with S^{γ} from the right and using (D.9) immediately yields that $\Upsilon_1 + 2S\Upsilon_2 = 1$. Substituting this back into (D.11), it reads

$$(2S\Upsilon_1' + 2S\kappa + \Upsilon_1^2 - \Upsilon_1)(\delta^{\beta}_{\gamma} - \frac{1}{2S} S^{\beta} S_{\gamma}) = 0,$$

from which we obtain

$$\Upsilon_1(S) = \begin{cases} \sqrt{2\kappa S} \cot \sqrt{2\kappa S} & \text{if } \kappa > 0 \\ 1 & \text{if } \kappa = 0 \\ \sqrt{2|\kappa| S} \coth \sqrt{2|\kappa| S} & \text{if } \kappa < 0, \end{cases}$$

and as stated earlier,

$$\Upsilon_2(S) = \frac{1 - \Upsilon_1(S)}{2S}.$$

Remark D.1. The formulas when $\kappa \leq 0$, are the analytical continuation of the formula when $\kappa > 0$. In the sequel, only the $\kappa > 0$ form is given.

Considering the parallel transport operator, we make the Ansatz

$$K^i_\alpha = \Upsilon_3(S){S^i}_{|\alpha} + \Upsilon_4(S)S^iS_\alpha. \tag{D.15}$$

Multiplying (D.15) with S^α from the right and using (D.4), and (D.10), we obtain $\Upsilon_3 + 2S\Upsilon_4 = -1$. Substituting this back in the Ansatz (D.15) and utilizing property (D.3), we get

$$K^i_{\alpha|\beta}S^\beta\big(2S\Upsilon_3' + \Upsilon_3 - \Upsilon_3\Upsilon_1\big)({S^i}_{|\alpha} - \frac{1}{2S}S^iS_\alpha) = 0.$$

Solving for $\Upsilon_3(S)$, we arrive at

$$\Upsilon_3(s) = -\frac{\sin\sqrt{2\kappa S}}{\sqrt{2\kappa S}},$$

and consequently,

$$\Upsilon_4(s) = -\frac{1}{2S}(1 + \Upsilon_3(S)) = -\kappa\frac{\sqrt{2\kappa S} - \sin\sqrt{2\kappa S}}{(\sqrt{2\kappa S})^3}$$

which yields the final expression for the parallel transport

$$K^i_\alpha = -\frac{\sin\sqrt{2\kappa S}}{\sqrt{2\kappa S}}{S^i}_{|\alpha} - \kappa\frac{\sqrt{2\kappa S} - \sin\sqrt{2\kappa S}}{(\sqrt{2\kappa S})^3}S^iS_\alpha.$$

Noticing Remark D.1, we differentiate (D.15) with respect to ξ^β and obtain

$${K^i}_{\alpha|\beta} = \Upsilon_3'(S)S_\beta{S^i}_{|\alpha} + \Upsilon_3(S){S^i}_{|\alpha|\beta} + \Upsilon_4'(S)S_\beta S^iS_\alpha + \Upsilon_4(S){S^i}_{|\beta}S_\alpha + \Upsilon_4(S)S^iS_{\alpha|\beta}.$$

Regarding the first and fourth term, from (D.15), we have

$${S^i}_{|\alpha} = \frac{1}{\Upsilon_3(S)}\big[K^i_\alpha + \Upsilon_4(S)s^iS_\alpha\big].$$

The third term is satisfactory, while $S_{\alpha|\beta}$ in the fifth term can be calculated via (D.14) and equals

$$S_{\alpha|\beta} = \Upsilon_1(S)g_{\alpha\beta} + \Upsilon_2(S)S_\alpha S_\beta. \tag{D.16}$$

Paying attention to the second term, since we have

$${S^i}_{|\alpha|\beta} = g^{ij}S_{\alpha|\beta|j}, \tag{D.17}$$

we differentiate (D.16) with respect to x^j, yielding

$$S_{\alpha|\beta|j} = \Upsilon_1'(S)S_jg_{\alpha\beta} + \Upsilon_2'(S)S_jS_\alpha S_\beta + \Upsilon_2(S)S_{\alpha|j}S_\beta + \Upsilon_2(S)S_\alpha S_{\beta|j}.$$

Substituting this back into (D.17), we get

$${S^i}_{|\alpha|\beta} = \Upsilon_1'(S)S^ig_{\alpha\beta} + \Upsilon_2'(S)S^iS_\alpha S_\beta + \Upsilon_2(S){S^i}_{|\alpha}S_\beta + \Upsilon_2(S)S_\alpha{S^i}_{|\beta}.$$

Combining the obtained expressions here–above, with those of $\Upsilon_1, \Upsilon_2, \Upsilon_3$ and Υ_4 presented earlier, gives us the final expression

$$\left\{\begin{array}{lcl} K^i_{\alpha|\beta} & = & \Upsilon(S)(S^ig_{\alpha\beta} + K^i_\beta S_\alpha), \\ \Upsilon(S) & = & \kappa\frac{\tan\frac{1}{2}\sqrt{2\kappa S}}{\sqrt{2\kappa S}}. \end{array}\right. \tag{D.18}$$

By manipulating (D.18), with the roles of x and ξ reversed, we also obtain

$$K^j_{\alpha|\beta}K^\alpha_mK^\beta_k = K^j_{\alpha|k}K^\alpha_m = \Upsilon(S)(S^jg_{km} - \delta^j_mS_k). \tag{D.19}$$

D.4.3 Contraction Theory

Let A, B and C be constants. Then the computations, when examining whether $\mathcal{L}_{\hat{\mathcal{F}}}G$ is negative definite or not, can be done component-wise, that is

$$\mathcal{L}_{\hat{\mathcal{F}}}G = \mathcal{L}_{\overset{\triangledown}{Z}}G - A\mathcal{L}_{(\mathrm{grad}\,S)^{\mathbb{H}}}G - B\mathcal{L}_{(\mathrm{grad}\,S)^{\mathbb{V}}}G - \mathcal{L}_{(\mathrm{grad}\,U)^{\mathbb{V}}}G + \mathcal{L}_{\Phi^{\mathbb{V}}}G + C\mathcal{L}_{\mathcal{R}}G.$$

Using the formulas (D.2), we arrive at

$$\mathcal{L}_{\hat{\mathcal{F}}}G = \begin{bmatrix} \mathrm{d}\xi^{\alpha} \\ \mathrm{D}\eta^{\alpha} \end{bmatrix}^T \otimes \mathcal{M} \begin{bmatrix} \mathrm{d}\xi^{\beta} \\ \mathrm{D}\eta^{\beta} \end{bmatrix}, \tag{D.20}$$

where the matrix

$$\mathcal{M} = \begin{pmatrix} a & c \\ c & b \end{pmatrix} \begin{pmatrix} M_{\alpha\beta} & N_{\alpha\beta} \\ P_{\alpha\beta} & Q_{\alpha\beta} \end{pmatrix} + \begin{pmatrix} M_{\beta\alpha} & P_{\beta\alpha} \\ N_{\beta\alpha} & Q_{\beta\alpha} \end{pmatrix} \begin{pmatrix} a & c \\ c & b \end{pmatrix},$$

has components given by

$$\begin{array}{rcl} M_{\alpha\beta} & = & -AS_{\alpha|\beta} \\ N_{\alpha\beta} & = & g_{\alpha\beta} \\ Q_{\alpha\beta} & = & C(R_{\beta\alpha\gamma\iota} + R_{\gamma\alpha\beta\iota})S^{\iota}\eta^{\gamma} \\ P_{\alpha\beta} & = & Y_{\alpha\beta\gamma\iota}\eta^{\gamma}\eta^{\iota} + AR_{\gamma\alpha\beta\iota}\eta^{\gamma}S^{\iota} - BS_{\alpha|\beta} - U_{\alpha|\beta} + g_{mn}F^m K^n_{\alpha|\beta}, \end{array}$$

with $Y_{\alpha\beta\gamma\iota} = (R_{\gamma\alpha\beta\iota} - C(R_{\gamma\alpha\epsilon\iota}S^{\epsilon})_{|\beta})$.

In the case when we set $C = 1$ and $S = 0$, we have, $S_{\alpha|\beta} = g_{\alpha\beta}$, $S^{\iota} = 0$ and $K^n_{\alpha|\beta} = 0$, and $\mathcal{M}$ becomes

$$\mathcal{M} = \begin{bmatrix} -2(aA + cB)g_{\alpha\beta} - 2cU_{\alpha|\beta} & D_{\alpha\beta} \\ D_{\alpha\beta} & 2cg_{\alpha\beta} \end{bmatrix},$$

where $D_{\alpha\beta} = (a - bB - cA)g_{\alpha\beta} - bU_{\alpha|\beta}$. From this it is possible to derive conditions for contractivity. When $U \equiv 0$[4], the observer dynamics is contractive for suitable a, b and c. This is in accordance with the results in [5]. However, whenever $S > 0$ and $Y_{\alpha\beta\gamma\iota}\eta^{\gamma}\eta^{\iota} \neq 0$ for some η, then

$$\mathcal{M} = \begin{bmatrix} 2c & b \\ b & 0 \end{bmatrix} Y_{\alpha\beta\gamma\iota}\eta^{\gamma}\eta^{\iota} + \mathcal{O}(\eta). \tag{D.21}$$

Based on these calculations, the following result can be formulated.

Theorem D.1. *The contracting neighborhood of the set $S = 0$ shown in [5], is* infinitely thin *as $|\eta|_g \to \infty$.*

Proof. We outset from (D.21) and show that for η large enough, the matrix preceding $Y_{\alpha\beta\gamma\iota}\eta^{\gamma}\eta^{\iota}$ and hence $\mathcal{L}_{\hat{\mathcal{F}}}G$ is *indefinite*. To this end, note that

$$\begin{bmatrix} a & c \\ c & b \end{bmatrix} > 0 \Longrightarrow b > 0.$$

Consider the determinant of the matrix of interest:

$$\begin{vmatrix} 2c & b \\ b & 0 \end{vmatrix} = -b^2 < 0.$$

[4] It is always possible to move terms between U and F.

It then follows that
$$\begin{bmatrix} a & c \\ c & b \end{bmatrix} > 0 \Longrightarrow \begin{bmatrix} 2c & b \\ b & 0 \end{bmatrix} \text{ is indefinite.}$$

Therefore, as $|\eta|_g \to \infty$, $\mathcal{L}_{\hat{\mathcal{F}}} G$ becomes indefinite. □

D.4.4 Lyapunov Approach

We now investigate the convergence of $\hat{\Sigma}$, in the case of constant curvature. We also put $U \equiv 0$ and let B be a constant. For the Lyapunov function candidate

$$V(x, v, \xi, \hat{v}) = \frac{1}{2} g_{ij} \Delta v^i \Delta v^j + BS(x, \xi),$$

with $\Delta v^i = (v^i - \hat{v}^i)$, the total derivative becomes

$$\dot{V} = g_{ij} \Delta v^i (\mathrm{D}_t v^j - \mathrm{D}_t \hat{v}^j) + BS_i \dot{x}^i + BS_\alpha \dot{\xi}^\alpha, \tag{D.22}$$

along the system dynamics of Σ and $\hat{\Sigma}$. Now,

$$\begin{aligned} \mathrm{D}_t \hat{v}^j &= \mathrm{D}_t (K^j_\alpha \eta^\alpha) = K^j_{\alpha|k} \dot{x}^k \eta^\alpha + K^j_{\alpha|\beta} \dot{\xi}^\beta \eta^\alpha + K^j_\alpha \mathrm{D}_t \eta^\alpha \\ &= K^j_{\alpha|k} v^k \eta^\alpha + K^j_{\alpha|\beta} \dot{\xi}^\beta \eta^\alpha + K^j_\alpha \big[- B g^{\alpha\beta} S_\beta + \Phi^\alpha + C \mathcal{R}^\alpha \big] \end{aligned}$$

where we have used the system dynamics of Σ and $\hat{\Sigma}$. From D.4 we obtain

$$-BK^j_\alpha \underbrace{g^{\alpha\beta} S_\beta}_{S^\alpha} = BS^j.$$

Also, noting that $K^j_\alpha \Phi^\alpha = F^j$, we can continue the calculations as

$$\mathrm{D}_t \hat{v}^j = K^j_{\alpha|k} v^k \eta^\alpha + K^j_{\alpha|\beta} \dot{\xi}^\beta \eta^\alpha + BS^j + F^j + C K^j_\alpha R^\alpha. \tag{D.23}$$

Concentrating on the last term in (D.23), from (D.13) and (D.19) it follows that

$$K^j_\alpha R^\alpha = \kappa \Upsilon^{-1}(S) K^j_{\alpha|k} K^\alpha_m \hat{v}^k \hat{v}^m.$$

Next, we pay attention to the second term in (D.23) and notice that

$$K^j_{\alpha|\beta} \dot{\xi}^\beta \eta^\alpha = K^j_{\alpha|\beta} \big[\eta^\beta - A g^{\beta\gamma} S_\gamma \big] \eta^\alpha = K^j_{\alpha|\beta} \eta^\beta \eta^\alpha = K^j_{\alpha|\beta} K^\beta_k \hat{v}^k K^\alpha_m \hat{v}^m = K^j_{\alpha|k} K^\alpha_m \hat{v}^k \hat{v}^m,$$

where the equalities origin from the dynamics of $\hat{\Sigma}$, (D.3), $\eta^\alpha = K^\alpha_m \hat{v}^m$ and (D.19) respectively. Then by substituting $\eta^\alpha = K^\alpha_m \hat{v}^m$ in the first term as well, (D.23) can be seen to equal

$$\mathrm{D}_t \hat{v}^j = K^j_{\alpha|k} K^\alpha_m \hat{v}^m \big[v^k + \hat{v}^k + C \kappa \Upsilon^{-1}(S) \hat{v}^k \big] + BS^j + F^j.$$

With $C = -2\kappa^{-1} \Upsilon(S)$, we arrive at the final expression

$$\mathrm{D}_t \hat{v}^j = K^j_{\alpha|k} K^\alpha_m \hat{v}^m \Delta v^k + BS^j + F^j. \tag{D.24}$$

Consider next the last terms in the total derivative (D.22). From (D.4), in conjunction with dynamics of $\hat{\Sigma}$ and the Hamilton-Jacobi equation (D.8), it follows that

$$S_\alpha \dot{\xi}^\alpha = -K^i_\alpha S_i (\eta^\alpha - A g^{\alpha\beta} S_\beta) = -S_i \hat{v}^i - A g^{\alpha\beta} S_\alpha S_\beta = -S_i \hat{v}^i - A S^\beta S_\beta = -S_i \hat{v}^i - 2AS.$$

Substituting these modified terms back into (D.22) yields

$$\begin{aligned}\dot{V} &= g_{ij}\Delta v^i\big[\cancel{F^j} - K^j_{\alpha|k}K^\alpha_m\hat{v}^m\Delta v^k - BS^j - \cancel{F^j}\big] + BS_i v^i + B\big[-S_i\hat{v}^i - 2AS\big]\\ &= -g_{ij}\Delta v^i K^j_{\alpha|k}K^\alpha_m\hat{v}^m\Delta v^k - Bg_{ij}\Delta v^i S^j + BS_i\Delta v^i - 2ABS\\ &= -g_{ij}\Delta v^i K^j_{\alpha|k}K^\alpha_m\hat{v}^m\Delta v^k - \cancel{BS_i\Delta v^i} + \cancel{BS_i\Delta v^i} - 2ABS\\ &= -g_{ij}\Delta v^i K^j_{\alpha|k}K^\alpha_m\hat{v}^m\Delta v^k - 2ABS.\end{aligned}$$

By using (D.19), we arrive at

$$\begin{aligned}\dot{V} &= -g_{ij}\Delta v^i\Upsilon(S)\big[S^j g_{km} - \delta^j_m S_k\big]\hat{v}^m\Delta v^k - 2ABS\\ &= -\Upsilon(S)\big[g_{km}S_i - g_{im}S_k\big]\hat{v}^m\Delta v^i\Delta v^k - 2ABS. \qquad \text{(D.25)}\end{aligned}$$

Theorem D.2. *If it is known that* $\sup_t |v(t)|_g \leq v_{\max}$ *, the injectivity radius of the manifold is greater than* ρ *everywhere,* $A > \sqrt{2}B^{-1}S^{-\frac{1}{2}}|\Upsilon(S)|(v_{\max} + |\eta|_g)^2|\eta|_g$*,* $B > (\frac{v_{\max}}{\rho})^2$ *and* $C = -2\kappa^{-1}\Upsilon(S)$*, then the observer* $\hat{\Sigma}$ *initiated at* $\xi(0) = x(0)$*,* $\eta(0) = 0$ *converges.*

Proof. Let the design parameter B be fixed and determined with some objective in mind[5]. As will be illustrated, it is sufficient to choose A properly in order to turn $\dot{V}$ negative definite. To this end, consider

$$-\Upsilon(S)\big[g_{km}S_i - g_{im}S_k\big]\hat{v}^m\Delta v^i\Delta v^k \leq |\Upsilon(S)|\Big[|g_{km}S_i\hat{v}^m\Delta v^i\Delta v^k| + |g_{im}S_k\hat{v}^m\Delta v^i\Delta v^k|\Big].$$

From (D.8) and the assumption $\sup_t |v(t)|_g \leq v_{\max}$, it follows that

$$|g_{km}S_i\hat{v}^m\Delta v^i\Delta v^k| \leq \sqrt{2S}|\eta|_g(v_{\max} + |\eta|_g)^2,$$

where the conservative worst case scenario is assumed. The same bound is obtained for

$$|g_{im}S_k\hat{v}^m\Delta v^i\Delta v^k|.$$

Therefore, the following inequality holds

$$\dot{V} \leq 2|\Upsilon(S)|\sqrt{2S}|\eta|_g(v_{\max} + |\eta|_g)^2 - 2ABS.$$

Then straightforward calculations show that choosing

$$A > \sqrt{2}B^{-1}S^{-\frac{1}{2}}|\Upsilon(S)|(v_{\max} + |\eta|_g)^2|\eta|_g,$$

yields $\dot{V} < 0$. Hence, asymptotic convergence of the estimation error to zero, follows from Lyapunov's direct method.

Let us then consider the choice of B. The chosen initialization point, yields $V(0) = \frac{1}{2}g_{ij}v^i(0)v^j(0) \leq \frac{1}{2}v^2_{\max}$. At any arbitrary time instance t, it then follows that

$$V(t) = \frac{1}{2}g_{ij}\Delta v^i\Delta v^j + BS \leq \frac{1}{2}v^2_{\max},$$

which implies $BS \leq \frac{1}{2}v^2_{\max}$. Since $S = \frac{1}{2}d(x,\xi)^2$, by requiring $B > (\frac{v_{\max}}{\rho})^2$ we obtain

$$d(x,\xi) \leq \frac{v_{\max}}{\sqrt{B}} < \rho,$$

which guarantees that $d(x,\xi)$ stays smaller than the injectivity radius at all time instances. □

[5] In our case, the objective will be to ensure that $d(x,\xi)$ is strictly less than the injectivity radius at all times. This is important in order to keep the derivatives of $S = \frac{1}{2}d(x,\xi)^2$ well-defined.

D.5 Example

Let $\mathcal{X}$ be the unit 3-sphere parameterized by $x_1, x_2 \in [0, \pi]$ and $x_3 \in [0, 2\pi]$. This is a space of constant curvature $\kappa = 1$. The metric is given by

$$g_{\cdot,\cdot} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \sin^2 x_1 & 0 \\ 0 & 0 & \sin^2 x_1 \sin^2 x_2 \end{bmatrix}$$

which implicitly gives the distance function, S, as

$$\begin{aligned} &\cos\sqrt{2S} = \cos x_1 \cos \xi_1 + \\ &\sin x_1 \sin \xi_1 [\cos x_2 \cos \xi_2 + \cos(x_3 - \xi_3) \sin x_2 \sin \xi_2]. \end{aligned}$$

The exterior forces, F, are given by $-\operatorname{grad} W$, where $W = \sin x_1 \cos x_2 \cos x_3$ and $U \equiv 0$. We define an observer $\hat{\Sigma}$ by the choices $A = 3\frac{1+S}{\sqrt{S+10^{-7}}}$, $B = 3$ and $C = -1$. Figure D.4 show the convergence of the observer when the initial data are

$$\begin{cases} x_1(0) = \xi_1(0) = 1 \\ x_2(0) = \xi_2(0) = 0.7 \\ x_3(0) = \xi_3(0) = 2 \end{cases} \begin{cases} v_1(0) = 2.25 \\ v_2(0) = 1.25 \\ v_3(0) = 4 \end{cases} \begin{cases} \hat{v}_1(0) = 0 \\ \hat{v}_2(0) = 0 \\ \hat{v}_3(0) = 0. \end{cases}$$

Similar simulation results have also been obtained in the cases of the hyperbolic plane (constant negative curvature) and the inverted pendulum on a cart (zero curvature).

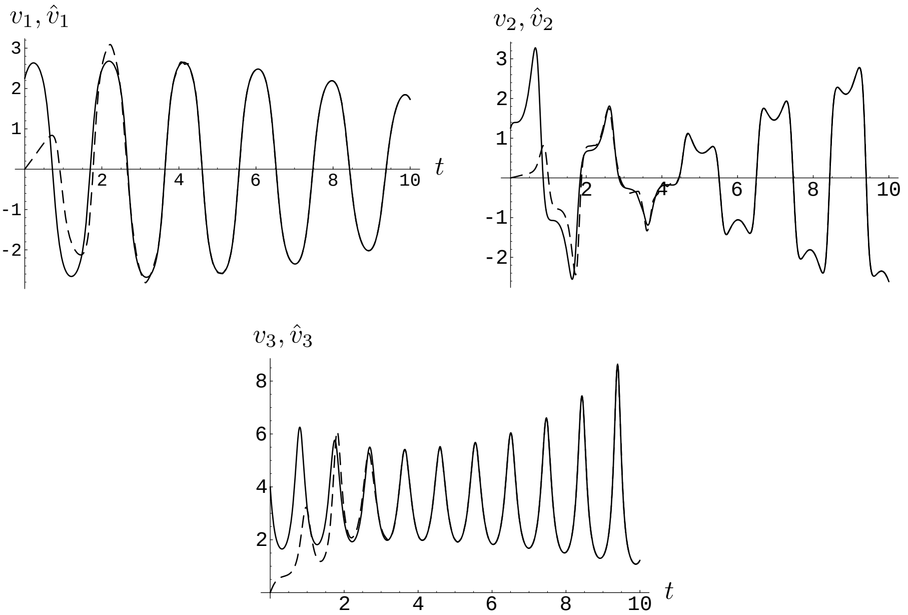


Figure D.4: The solid line refers to the original system, while the dashed line represents the observer.

D.6 Concluding Remarks

The observer presented in this paper, requires the explicit computation of the distance function, S, as well as the parallel transport operator, K, which is prohibitive unless the configuration manifold is extremely simple, *e.g.* manifolds of constant curvature, Lie groups (*cf.* [6]) *etc.* For more general spaces, schemes of approximation are called for (*cf.* [5]).

D.7 References

[1] Van der Schaft, A., "On nonlinear observers," *IEEE Trans. Automat. Control*, Vol. AC-30, No. 12, Dec. 1985, pp. 1254–1256.

[2] Thau, F., "Observing the state of non-linear dynamic systems," *International Journal of Control*, Vol. 17, 1973, pp. 471–479.

[3] Lohmiller, W. and Slotine, J., "On contraction analysis for non-linear systems," *Automatica J. IFAC*, Vol. 34, No. 6, 1998, pp. 683–696.

[4] Hamberg, J., "Controlled lagrangians, symmetries and conditions for strong matching," *Lagrangian and Hamiltonian methods for nonlinear control*, edited by N. Leonard and R. Ortega, Elsevier, 2000, pp. 62–67.

[5] Aghannan, N. and Rouchon, P., "An intrinsic observer for a class of Lagrangian systems," *IEEE Trans. Automat. Control*, Vol. 48, No. 6, 2003, pp. 936–945.

[6] Maithripala, D., Berg, J., and Dayawansa, W., "An intrinsic observer for a class of simple mechanical systems on a Lie group," *Proc. of the IEEE American Control Conference*, 2004, pp. 1546–1551.

[7] Lovelock, D. and Rund, H., *Tensors, differential forms, and variational principles*, Dover Publications, New York, 2nd ed., 1989.

[8] Abraham, R. and Marsden, J. E., *Foundations of mechanics*, Benjamin/Cummings Publishing Co. Inc. Advanced Book Program, Reading, Mass., 1978, Second edition, revised and enlarged, With the assistance of Tudor Raţiu and Richard Cushman.

[9] Yano, K. and Ishihara, S., *Tangent and cotangent bundles: differential geometry*, Marcel Dekker Inc., New York, 1973, Pure and Applied Mathematics, No. 16.

[10] Sasaki, S., "On the differential geometry of tangent bundles of Riemannian manifolds," *Tôhoku Math. J. (2)*, Vol. 10, 1958, pp. 338–354.