KTH Engineering Sciences

On Cooperative Surveillance, Online Trajectory Planning and Observer Based Control

DAVID A. ANISI

Doctoral Thesis
Stockholm, Sweden 2009

TRITA MAT 09/OS/03
ISSN 1401-2294
ISBN 978-91-7415-246-3

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

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 teknologie doktorsexamen,. The public review will be held on onsdagen den 1 april 2009 klockan 10.00 in rum E2, Lindstedtsvägen 3, Kungl Tekniska högskolan, Stockholm.

Print: Universitetsservice US AB

اسرار ازل را نه تو دانی و نه من
وین حل معما نه تو دانی و نه من
هست از پس پرده گفتگوی من و تو
چون پرده بیافتد نه تو مانی و نه من

The eternal secrets, neither you know nor I.
And the answer to the riddle, neither you know nor I.
Behind the veil, there is the talk of me and thee.
When the veil falls, neither you remain nor I.

- Omar Khayyam

To my beloved family

Abstract

The main body of this thesis consists of six appended papers. In the first two, different cooperative surveillance problems are considered. The second 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.

In Papers A and B, a combinatorial optimization based framework to cooperative surveillance missions using multiple Unmanned Ground Vehicles (UGVs) is proposed. In particular, Paper A considers the the Minimum Time UGV Surveillance Problem (MTUSP) while Paper B treats the Connectivity Constrained UGV Surveillance Problem (CUSP). The minimum time formulation is the following. Given a set of surveillance UGVs and a polyhedral area, find waypoint-paths for all UGVs such that every point of the area is visible from a point on a waypoint-path and such that the time for executing the search in parallel is minimized. The connectivity constrained formulation extends the MTUSP by additionally requiring the induced information graph to be kept *recurrently* connected at the time instants when the UGVs perform the surveillance mission. In these two papers, the $\mathcal{NP}$-hardness of both these problems are shown and decomposition techniques are proposed that allow us to find an approximative solution efficiently in an algorithmic manner.

Paper C addresses the problem of designing a real time, high performance trajectory planner for an aerial vehicle that uses information about terrain and enemy threats, to fly low and avoid radar exposure on the way to a given target. The high-level framework augments Receding Horizon Control (RHC) with a graph based terminal cost that captures the global characteristics of the environment. An important issue with RHC is to make sure that the greedy, short term optimization does not lead to long term problems, which in our case boils down to two things: not getting into situations where a collision is unavoidable, and making sure that the destination is actually reached. Hence, the main contribution of this paper is to present a trajectory planner with provable safety and task completion properties.

Direct methods for trajectory optimization are traditionally based on *a priori* temporal discretization and collocation methods. In Paper D, 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 Paper E, the problem of active observer design for an important class of non-uniformly observable systems, namely mobile robotic 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 Paper F, a geometrically intrinsic observer for Euler-Lagrange systems is defined and analyzed. This observer is a generalization of the observer proposed by Aghannan and Rouchon. Their contractivity result is reproduced and complemented by a proof that the region of contraction is infinitely thin. Moreover, 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: Surveillance Missions, Minimum-Time Surveillance, Unmanned Ground Vehicles, Connectivity Constraints, Combinatorial Optimization, Computational Optimal Control, Receding Horizon Control, Mission Uncertainty, Safety, Task Completion, 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 main 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. This gratitude naturally extends to my co-advisor and most frequent co-author, Petter Ögren, whos involvement and commitment have pervaded most results of this thesis.

The research that culminated in this thesis, has been partly conducted in cooperation with the Department of Autonomous Systems at the Swedish Defense Research Agency (FOI). 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 John Robinson; not only for guidance in the direction of research, but also for providing and maintaining a friendly and creative research atmosphere. 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. The initialization of this research project is due to 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 would like to show my appreciation to Mats Werme, Tove Gustavi, Johan Karlsson and Johan Thunberg for their careful reading and constructive criticism of parts of this thesis.

I whole-heartedly thank my oldest friends from the 165-area: Carl-Oskar, Jogge, Jonas, Henke; the original persian crew: Homan, Omid, Nahid, Reza; and the distinguished members of the ΔBK-club: Hossein, Payam, Amir and Ebi, for keeping my life well-balanced by providing perspectives from a world outside KTH.

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 thank Darya; for her constant support and countless sacrifices, and Anders; for his fun-loving attitude and high-spirit. I am fortunate to have a sister and brother-in-law like you. Another family member to acknowledge is my dear uncle; for being a distinguished source of admiration and respect.

Finally, I would like to gratefully acknowledge the financial support by the Department of Autonomous Systems at the Swedish Defence Research Agency (FOI), Swedish Governmental Agency for Innovation Systems (VINNOVA) and the Swedish Defence Material Administration (FMV) through the Technologies for Autonomous and Intelligent Systems (TAIS) project 297316-LB704859.

Stockholm, Mars 2009

David Alireza Anisi

Table of Contents

Introduction

Introduction

THIS thesis deals with three problems in the field of control, namely: concurrent task assignment and path planning; trajectory planning; and observer design. Here–below, all these subjects, as well as their inter–relation, are explained in an introductory manner. Consequently, the more familiar reader may prefer to skip it at a first reading. The three subsequent sections provide a more detailed treatment of these problems.

In order to provide the big picture first, Figure 1 depicts a possible structure for decomposing the overall system architecture.

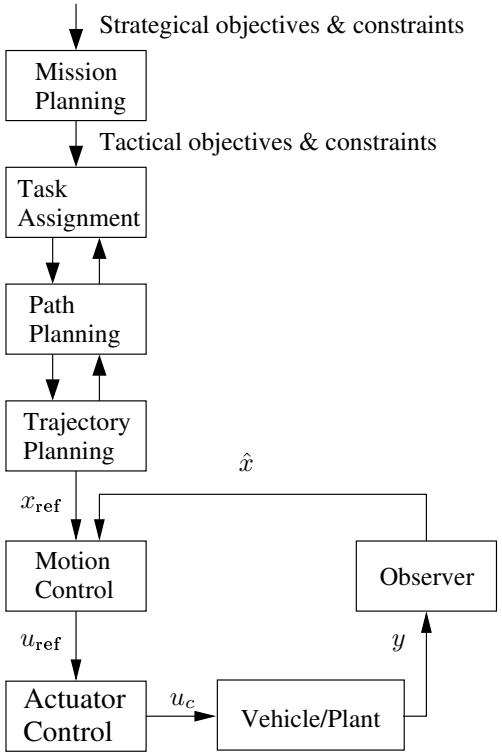
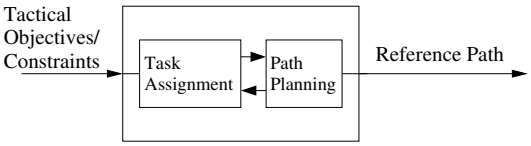


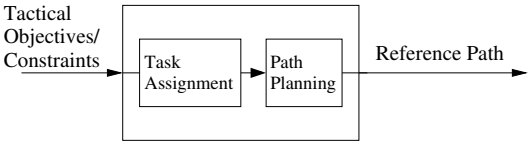
Figure 1: Possible structure of a modular design of an autonomous system.

The first problem area of this thesis, that is considered in Papers A and B, is concurrent task assignment and path planning. The term *concurrent* here refers to the fact that the task assignment- and path planning processes are occurring simultaneously in an interweaved fashion. This has been indicated in Figures 1 and 2(a) by arrows going in both directions between the task assignment- and path planning box. In contrast, as seen in Figure 2(b),

an alternative approach would be to assign the tasks and plan the paths *sequentially*. In this latter case, the path planner is not used during the task assignment process. In general, recognizing the coupling between these two processes leads to a better overall solution while neglecting it lowers the computational complexity.



(a) Concurrent task assignment and path planning.



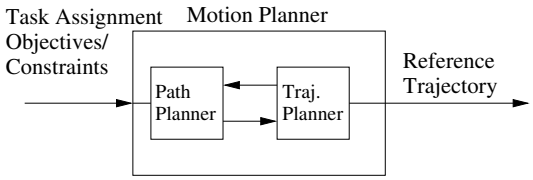
(b) Sequential task assignment and path planning.

Figure 2: Two possible interaction structures between the task assignment- and path planning processes.

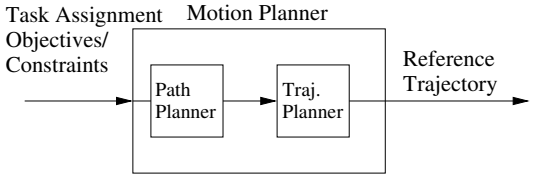
The second problem area, namely trajectory planning, is an instance of the *motion* planning problem, which is used as a collective term including both trajectory- and path planning (*cf.* Figure 3). In its most basic form, motion planning is about finding a feasible trajectory connecting two given configurations, denoted p_i and p_f, for a single robot in a static and known environment. A path is the image of a continuous function $\gamma : [0, 1] \rightarrow \mathcal{C}$, 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 time dependent configuration– *and* velocity functions, that are also consistent with the robot's dynamic constraints. This definition is adopted in this thesis.

In a similar fashion as in the task assignment and path planning problem, there are two ways to decompose the motion planning problem. The first alternative is *sequential* motion planning, *i.e.*, finding a feasible trajectory through *refinement* [1,2] (see Figure 3(b)). Given the task assignment objectives and constraints, 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 there exists methods that perform the path- and trajectory processes *concurrently* (see Figure 3(a)). The framework adopted in Papers C and D, namely that of Computational Optimal Control, serves as an example of this alternative approach that merges the path- and trajectory planning part (see Section 2.1).

Once we have a feasible reference trajectory at hand (denoted x_{ref} in Figure 1), 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



(a) Concurrent path- and trajectory planning.



(b) Sequential path- and trajectory planning.

Figure 3: Possible decompositions of the motion planning problem.

state information while only using available measurements, y. Observers are defined more rigorously in Section 3.1 and observer design, which constitutes the third problem area of this thesis, has been considered in Papers E and F.

An important point to notice is that although concurrent task assignment and path planning, trajectory planning and observer design are all vital sub-problems in the creation of autonomous systems, they are not the only problems that have to be addressed. Other important aspects include high–level mission planning (*i.e.*, making strategic global plans), controller design, actuator design and sensor fusion. Truly optimal design of autonomous robotic systems requires concurrent solution of all these sub-problems. Such an approach is however beyond our current reach. 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 taken into consideration.

Cooperative Surveillance using Multiple Unmanned Ground Vehicles

In both civil and military applications, surveillance is performed in order to assist in the prevention, detection and monitoring of intrusion, theft or other safety-related incidents. Application areas and facilities that require such supervision are numerous and include airport facilities, military installations, territorial borders, storage buildings, harbors, power plants, banks, factories and offices. Todays surveillance and security solutions are based on a combination of

- human guards (manned gates, airport screeners, store detectives),
- electronic systems (cameras, intrusion alarms, fire detection),
- physical security (fences, gates), and
- software (reporting, verification, logging).

In the ideal case, surveillance should be performed in a continuous manner and cover the entire facility, although in practice, financial and head-count constraints limit it to only encompass the most important and critical areas. Recent scientific and technological developments is however taking us towards more autonomous and mobile solutions. The market for semi-autonomous sentry vehicles is in fact already established and growing. As for today, there are a few tailor-made safety and security vehicles on the market, but so far, they possess a quite limited capabilities and thus functionality [3]. From a performance standpoint, the potential benefits with adopting an Unmanned Ground Vehicle (UGV) for security and surveillance applications are numerous and include:

1. Cost savings.
2. Removal of humans from direct exposure to potentially harmful situations.
3. Elimination, or at least reduction, of the risk of "inside jobs".
4. More effective performance of many security and surveillance routines as autonomous systems - unlike humans - do not get bored and thereby inattentive during long working hours.

In comparison with stationary surveillance cameras, a more mobile solution has several advantages, most apparently that of *flexibility*, in the sense of being able to cope with application areas that are changing in time. For instance, using stationary cameras, a great deal of camera redundancy would be required to fully monitor a harbor area where the container setup is varying on a daily basis. However, using surveillance UGVs, these changes can be easily incorporated in the planning scheme.

Here-below, in order to concretize a specific problem instance and thereby put the reader in the right frame of mind, a fictitious motivating example is described and discussed in

more detail. Following that, we start Section 1.2 by arguing that the considered example is in fact an instance of a larger problem class, namely *concurrent task assignment and path planning* for multi vehicle systems. The remaining part of Section 1.2 is therefore devoted to formally defining this problem class. Finally, Section 1.3 provides an overview of related research.

1.1 Motivating Example: Constrained rounds of patrol

Traditionally, patrolling rounds are performed by humans. To obtain maximal security, these rounds should be performed in a continuous manner. However, having a full-time guard 24 hours a day, 7 days a week will cost more than 3 full-time employees. This fact has prevented EsCoTer, a (fictious) company in Stockholm, from obtaining this high level of security. EsCoTer is an importer and distributer of Asian scooters, ATVs and dirt-bikes into the Swedish market. As such, it has a warehouse that has repeatedly been an object of interest for intruders and burglars. Faced with this problem, the owner of the company has therefore looked for alternative solutions to complement the traditional way of patrolling. The most flexible and cost effective offer so far has been delivered by a security company called Sentry Inc. and involves using a small group of semi-autonomous vehicles for performing these patrolling rounds.

The most basic solution Sentry Inc. provides is to engage a group of sentry UGVs that cooperatively visit a set of known and predefined sites in a regular and repetitive manner. On their way between the sites, each UGV should survey its surroundings using its on-board sensors. Possible onboard sensors include laser scanners, IR-cameras and chemical sensors with which one can detect, *e.g.*, intruders, fire, gas leaks or even abnormal radioactivity. As a more refined solution for more challenging scenarios, Sentry Inc. provides a solution that can handle patrolling rounds which are *constrained* to fulfill different conditions. According to the specifications, possible constraints that could be elaborated upon include, but are not limited to:

Temporal and/or spatial visiting constraints: It might for instance be desirable to assure that sensitive sites of high priority are visited at least once during given time intervals. This imposes a *temporal* constraint on the solution. Sensors with limited field of view provide a prototype example of *spatial* constraints.

Line of sight constraints: In addition to visiting the sites, the threat situation may call for monitoring of the UGVs themselves. It is therefore of interest to have the capability to perform the patrolling rounds while mutually keeping the line of sight between the UGVs clear.

Non-predictability constraints: Performing the rounds in a regular manner, makes it easy for potentially hostile forces to plan their actions and circumvent this line of defense. Therefore, it is desirable to introduce some degree of non-predictability in the patrolling rounds.

Verifiability constraints: These are introduced to attain a quality certification, *i.e.*, assure a certain levels on key features. This might for instance involve guarantees that the non-deterministic rounds will not neglect any site completely.

The owner of EsCoTer realizes that the more enhanced alternative suits the needs of his company better and therefore signs a two year contract with Sentry Inc. within a matter of weeks.

1.2 Concurrent task assignment and path planning for surveillance and security applications

In this section, we argue that the example of Section 1.1 is an instances of a larger problem class, namely Concurrent Task Assignment and Path Planning (C–TAPP). To see this, notice that visiting any of the sites can be seen as a task and the goal is to plan the paths for all the UGVs such that these tasks are performed in an optimal manner while fulfilling the constraints. Next, to set a common ground for the subsequent sections, a more formal definition of C–TAPP is given.

Problem 1.1 (Concurrent Task Assignment and Path Planning (C–TAPP)). *Given N UGVs and M tasks that they have to perform, concurrently assign the tasks and find paths for all the vehicles in a cooperative manner. The task assignment and the generated paths are to fulfill all the constraints imposed on them while minimizing a given cost function.*

Notice that this problem formulation allows some of the N vehicles to remain idle in their initial position. This could be of great strategical interest since the inactive vehicles can be used for performing other missions in parallel.

Remark 1.1. Obviously one of the keywords in Problem 1.1 is *cooperation*. Surprisingly, occurrence of concrete definitions of the meaning of this term within the multi-vehicle field are sparse in the literature. For now, we define *cooperation* from an optimization perspective: "Cooperation emerges from the objective of minimizing the given cost function"[1]. In Section 1.3 however, we follow [4] providing a list of some alternative definitions of cooperation.

Before presenting various choices of objective functions and constraints in C–TAPP, we make a small digression to put it into perspective by presenting our view on how C–TAPP enters the overall system architecture. As discussed earlier in the introduction, in this work, an overall *modular* design of the system architecture is assumed. Figure 1.1 depicts a way of decomposing the overall problem of designing a multi-UGV system, where only parts of the interactions between the modules are counted for. The principal interest of Problem 1.1 lies on the task assignment and path planning module of Figure 1.1. Hence, this formulation neglects several other crucial subproblems, such as, trajectory planning, actuator control, observer design, sensor fusion, communication technology issues and sensor detection. We assume these modules are available to us, so that for instance the vehicles are assumed to know their positions, either through direct measurements from on-board sensors, or through a suitably deigned observer.

Having put the C–TAPP problem into perspective, we proceed by listing some relevant choices of objective function and constraints in C–TAPP. This list particularly emphasizes C–TAPP problems for surveillance and security applications.

1. **Possible objective functions:**
   a) Minimize the total time for completing the tasks[2].
   b) Minimize the distance traveled while performing the tasks.
   c) Minimize the maximal or accumulative threat encountered during the mission.
   d) Minimize a combination of task completion time and number of vehicles used.

[1] *By this we are also able to distinguish* cooperation *from* coordination, *which can be thought of as an implication of the constraints of an optimization problem.*

[2] Assumption on constant vehicle speed gives an equal work-load formulation.

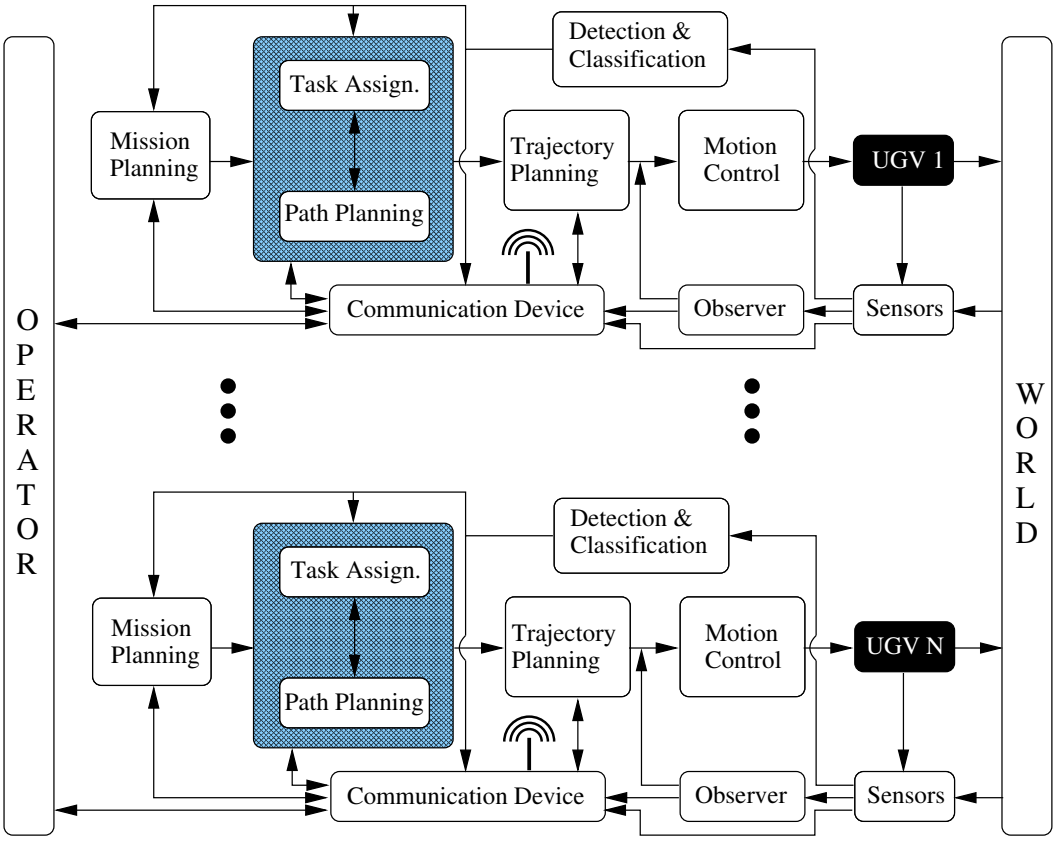


Figure 1.1: Possible modular structure of a multi-UGV system.

e) Minimize a combination of the previous objectives.

f) Provide soft ordering by associating revenues to all tasks and maximizing the total revenue (*cf.* [5–7]).

g) Maximize total utility as defined by the difference between total cost and total revenue [5, 8].

Remark 1.2. Problem 1.1 is a generalization of the Traveling Salesmen Problem (TSP) and is therefore also $\mathcal{NP}$-hard. Consequently, we can not expect to solve all problem instances to optimality within a reasonable amount of time. In practice, some heuristic algorithm may be used for solving Problem 1.1. A possible constraint on the solution would then be:

- to obtain ε-optimal solutions, *i.e.*, solutions whose costs are within an ε factor from the optimal one.

2. **Possible task assignment constraints:**

a) No task should be neglected.

b) Every task should be assigned to one and only one UGV.

c) Number of tasks assigned to each vehicle is bounded from above.

d) A task, m, must have at least $n_m > 1$ UGVs assigned to it.

e) Different UGVs have different capabilities so that not all vehicles can perform all tasks.

f) Temporal constraints such as:

- time-windows for the completion of certain tasks,
- ordering, *e.g.*, that task m_i has to be performed before task m_j.

g) Non-predictability.

3. **Possible path planning constraints:**

a) Line of sight/communication maintenance constraints.

b) Spatial constraints such as:

- upper bound on the total path length for some of the vehicles (fuel constraint).
- given/free initial and/or final positions for some of the vehicles.
- collision free paths.

In addition to these, one might consider other spatial constraints imposed by such tasks which cannot be solved if:

- the distance to the task location is larger than a given threshold,
- the task location is approached from certain directions.

Both these examples are highly relevant for camera surveillance scenarios.

c) Dynamic feasibility (*i.e.*, the needs of the trajectory planner is addressed, *cf.* Figure 1.1).

d) Non-predictability.

1.3 C–TAPP Related Research

A vast amount of research and a huge number of publications have been devoted to problem formulations more or less related to the C–TAPP problem, as defined in Section 1.2. In this section a broad overview of the research that is currently ongoing in this field is provided. The exposition is neither complete nor self-contained, hence appropriate external references are provided in order to allow the interested reader to probe more deeply into the subject[3].

In essence, C–TAPP related research has so far been quite informal, concept-oriented and primarily focused on:

1. Specification of particular problem instances. Often, this is done with some real-world application in mind.

2. Presentation of some heuristic, empirical or *ad hoc* solution method, *e.g.*, a proper coordination and cooperation architectures and different problem decomposition techniques.

3. Validation of the proposed solution method through simulations or experiments in a proof-of-concept fashion.

[3]Short reviews of some of the referenced papers can be found in [3].

These three steps are the foundation of the overwhelming majority of the papers in this field. The first step involves specification of the objective function to be minimized and relevant constraints imposed on the task- and path planner. As previously mentioned, this step is normally inspired from a particular application-domain. As an illustrative example, reference [9] considers a scenario where a group of N vehicles are required to visit M known target locations within a hostile environment with P static threats. The objective function consists of a combination of risk minimization, balancing the workloads between the vehicles and minimizing the mission completion time. As for the task- and path-wise constraints, the authors require all M targets to be visited, while avoiding collisions and flying within predefined length limits (fuel constraint). To make the problem more realistic, the authors may further impose timing and ordering constraints on the tasks as well as an upper limit on the number of targets that can be assigned to each vehicle.

As for the second step, the literature includes a wide variety of techniques and ideas. This is also usually where the main research focus lies. A classical approach to solve challenging combinatorial optimization problems such as C–TAPP is based on clustering [8, 10]. The two main ideas here are to either cluster-first-route-second or the other way around. In order to improve solution quality, the clustering and routing phases can be iteratively repeated – at the expense of computational load. As an example, reference [9] approximates an exact MILP formulation of C–TAPP in four different ways using the clustering ideas presented here-above. These approximations have lower computational complexity and are therefore better suited for online purposes. The approach taken by Maddula *et al.* [11] illustrates another distinguished way of tackling the C–TAPP problem. In a first phase, an initial assignment is constructed. In a second phase, this initial assignment is refined using four target exchange operators that are defined in the paper. The same idea is elaborated upon in several other papers encountered in this domain. Heuristic ways of improving an initially feasible solution include

- Tabu search [6, 12] (which are known to perform well on various routing problems [10, 13, 14]),
- stochastic hill-climbing [15],
- ant colony optimization [16],
- genetic algorithms [17].

As mentioned earlier, there is a natural way of decomposing the C–TAPP problem into two subproblems, namely the optimal task assignment- and the optimal path planning problem. Unless the objective function in the task assignment problem is *path independent*, this modular scheme is bound to produce sub-optimal solutions (*cf.* Figure 2(b)). Having a path independent objective function is hardly the case in most realistic surveillance and security applications. Consequently, the ideal case from this thesis' point of view, is to solve these two subproblems *concurrently*. This approach has been depicted in Figures 1.1 and 2(a).

Since the field of cooperative multi-vehicle systems is a relatively young research domain, some important aspects of C–TAPP have been largely untreated in the literature. In particular, the following two aspects deserve much more attention from the research community.

1. More theoretical aspects and frameworks for formal analysis [5, 18],
2. Evaluative and comparative studies [13, 19].

As indicated in Remark 1.1, occurrence of concrete definitions of the meaning of the key term "cooperation" are sparse in the multi-vehicle literature. Next, we follow [4] and provide a list of some alternative definitions of this term. Explicit definitions of cooperation include:

1. "joint collaborative behavior that is directed toward some goal in which there is a common interest or reward",

2. "a form of interaction, usually based on communication",

3. "[joining] together for doing something that creates a progressive result such as increasing performance or saving time".

This last definition is probably the one closest to the definition provided in Remark 1.1: "Cooperation emerges from the objective of minimizing the given cost index". This definition originates from an optimization perspective. Also, as mentioned earlier, this point of view allows us to distinguish *cooperation* from *coordination*, which can be thought of as something emerging from the constraints of an optimization problem.

In the literature, there exists a body of work that aims at providing a suitable classification scheme and taxonomy for the field of cooperative multi-robotics (see, *e.g.*, [4, 5, 18, 20, 21]). These papers also provide excellent surveys of the literature at different times. Next, a handful of selected topics from these important papers will be discussed.

In [20] the authors present a taxonomy that classifies cooperative teams. In addition, a rather comprehensive survey of existing work as it appeared in the mid 90's is provided. Seven important aspects are mentioned and include collective size, the systems communication and computational capabilities. A summary of the proposed taxonomic axis can be found in Table 1.1. It can be noted that the communication issue constitutes a relatively large fraction of the classification dimensions. In order to illustrate the usefulness of

Taxonomic Axis	Description
Collective Size	The number of robots in the group
Collective Reconfigurability	Rate for spatial re-organization
Collective Composition	Group being homogeneous or heterogeneous
Communication Range	Upper limit on the inter-robot distance such that communication is still possible
Communication Topology	Describes possible inter-robot communication
Communication Bandwidth	Amount of information that can be transmitted
Processing Ability	Each units model of computation

Table 1.1: Summary of the taxonomic axis as they appear in [20].

the suggested taxonomy, [20] sorts the surveyed papers according to their position in the taxonomy.

Another important work that provides natural dimensions along which multi-robot systems can be separated is [4]. In this paper, the authors identify five important "research axis", or taxonomic axis, that can be used when comparing different system designs.

Group Architecture: this axis can be described as the "infrastructure upon which collective behaviors are implemented". Concepts such as group differentiation (homo-

geneity/ heterogeneity)[4], control type (centralization/ decentralization) and communication structure fall into this category.

Resource Conflict: strategy for resolving possible group conflicts, *e.g.*, the collision avoidance problem in mobile robotics or the multi-access problem in computer networks.

Origin of Cooperation: (biologically/socially inspired) mechanisms that motivate and achieve cooperation in systems where this has not been "explicitly engineered" into the system.

Learning: strategies for finding correct values for design parameters, *e.g.*, reinforcement learning, genetic algorithms or neural networks.

Geometric Problems: issues tied to the embedding of the system in a two- or three-dimensional world. Examples include multi-robot path planing and moving to formation.

Also in [4], the authors provide a survey of existing work and discuss some open research problems, technological constraints and the influence of other academic disciplines that have shaped the field of cooperative robotics. The reader is urged to consult [4] for a fuller discussion. It must be emphasized however that the task assignment problem is largely overlooked in [4]. From the C–TAPP's point of view, the task decomposition and allocation aspects certainly requires axes on their own.

A possible classification of different coordination schemes is that of explicit *vs.* implicit coordination [18, 22][5]. A multi-vehicle team may coordinate *explicitly* using communication or negotiations. An example of one such mechanism is market-based coordination [8], where individual vehicles competitively bid for the tasks to be performed. This auction-based approach is based on some given bidding rule [24]. However, multi-vehicle teams may also cooperate *implicitly*. In this case, communication is mediated through inter-vehicle and vehicle-world interactions. This type of communication is called *stigmergic* in the biological literature [25]. As an example, a box-pushing application is considered in [26] that achieves cooperation without communication. This is possible since the object being manipulated also functions as a channel of communication that is shared by all the robots. The relative merit of these two coordination schemes remains an open question. According to [18] however, it is in general easier to perform a formal analysis on explicit approaches. They are also considered to produce more accurate and near-optimal solutions. On the downside, explicit coordination schemes are not as flexible, robust and – due to the inherent computation and communication complexity – scalable as the implicit approach.

Another feature than can be used for classifying different architectures is whether the system is centralized or not [4]. In centralized systems, the decisions regarding cooperation and coordination are made at one single central control unit. Decentralized systems on the other hand, are characterized by the lack of such a unit. Instead, robots rely solely on locally available and processed knowledge. As far as pros and cons are considered, decentralized systems are generally considered to be inherently more reliable, robust and scalable [8, 18]. In reality however, there is a continuum of possible system designs that span the spectrum between the two extreme cases. Market-based approaches serve as a typical example that resides in the middle of the spectrum.

A challenging and highly relevant extension to the C–TAPP problem is to explicitly recognize the presence of *uncertainty*. In the face of measurement noise, parametric uncertainty, modeling errors and other disturbances, the deterministic nature of Problem 1.1

[4]Which corresponds to the "collective composition" axis of [20].

[5]This can also be referred to as intentional *vs.* emergent coordination [21, 23].

falls short. One approach in the literature to handle this issue is to pose the C–TAPP problem within a stochastic or robust optimization framework. To reasonably limit its scope, the main focus of this chapter will however be on a more implicit approach to handle the uncertainty issue, namely requiring fast solutions to the C–TAPP problem. The low computational time allows us to perform re-planning online as new information about the environment or mission objectives is gathered. This information can then be processed and fed back regularly to the C–TAPP planner. This way, feedback is incorporated and a certain degree of robustness is obtained. The reader should however be aware of the important work and significant progresses that have been made in explicitly incorporating uncertainty in the problem formulation. Stochastic or robust versions of problems related to C–TAPP have been considered in, *e.g.*, [7, 15, 27–29].

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.* [30,31] 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 C and D, 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.*, [30] 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 approximative 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, 32] 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 2.1.

2.1 Computational Optimal Control

The paradigm of qualitative control design, which is associating a measure of the "utility" with 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 impact have so far been moderate, 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 [33].

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

$$\underset{u}{\text{minimize}} \quad J = \int_0^T \mathcal{L}(x,u)\mathrm{d}t + \Psi(x(T)) \qquad \text{(OCP)}$$

$$\begin{aligned} \text{such that} \qquad \dot{x} &= f(x,u) \\ d(x,u) &\leq 0 \\ x(0) &= x_i \\ x(T) &\in S_f \end{aligned}$$

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 of the OCP are simple enough, finding optimal controls *analytically* is not a viable approach. 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 [33,34][1].

It is further conceptually important to differentiate between *direct* and *indirect* transcription methods (see Figure 2.1). These two categories will be dealt with in Section 2.1.1 and 2.1.2 respectively.

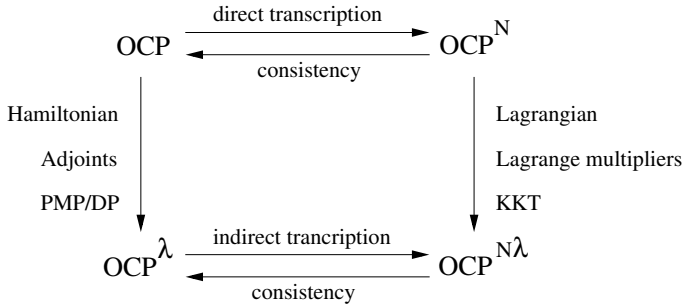


Figure 2.1: Direct and indirect transcription methods.

[1] Certain choices for basis functions, blur the distinction between the two mentioned transcription methods (see, *e.g.*, [35, 36]).

2.1.1 Indirect Transcription Methods

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

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

where λ denotes the adjoint variables, and formulating the optimality conditions either according to the Pontryagin Maximum Principle (PMP) [37] or Dynamic Programming (DP) [38]. 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 2.1. Indirect transcription methods then proceed by approximating and numerically solving the TPBVP/HJB. Possible approaches for doing so include, (multiple) shooting method, finite difference method, collocation method and Galerkin method [39].

In general, indirect methods are considered to produce more accurate results [40, 41]. 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 (*cf.* Footnote 1). Nevertheless, indirect methods are not typically used to solve problems having complex dynamics or constraint set. This is most often due to the:

- Ill-conditioned properties of the TPBVP[2].
- Occasionally tedious derivation of the optimality conditions via PMP/DP[3].
- Analytic intractability of solving the derived (nonlinear) optimality conditions.
- Analytic intractability of solving the HJB partial differential equation.
- "Curse of dimensionality", *i.e.*, the inherent exponentially increasing computational complexity for solving HJB as the problem size increases [38].

Notice also that for problems where the underlying OCP is changeable in terms of the objective function, J, the final manifold, S_f and/or the constraints, $d(x, u)$, the optimality conditions must be restated. This in fact makes indirect methods less suitable for this class of problems. Bearing in mind the assumption on information uncertainty made in this thesis, the principal interest will therefore be on *direct* transcription methods.

2.1.2 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 [42, 43] by applying the PMP to a most simple OCP, namely the so called Dubins' problem [44].

[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 [45], control [46,47], and both the state and control [48–50], 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 2.1) 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 reliably 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

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

which is a mapping from a point $x \in \mathcal{X}$ to a subset of the tangent space, $T_x\mathcal{X}$, can be easily characterized or approximated. The set $S(x)$ is termed holograph in [51]. 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 [52–54].

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 non-convexity" [55], transforming these will most likely result in increased computational complexity (*cf.* [52,56]). The statement that optimization routines converge faster and more reliably on smaller problems, holds true if two problems with similar structure and complexity (in terms of non-convexity), 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.*, [57,58], 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 [57], 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.*, [59, 60]), 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 in any reasonable function space. Here, $\{\phi_i(t) : i \in \mathbb{N}\}$, is a family of known *basis functions* that span the function space in question. 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 $f(t)$ by polynomials on a (compact) interval, I. 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 [61].

In principle, any preferred basis functions can be employed. The reader is referred to the introductory part of Ma's thesis [33], for a concise summary of different applicable approximation schemes. In practice however, piecewise polynomials [49], in particular cubic spline functions [48], belong to the classical choices. More recently, different orthonormal basis functions, *e.g.*, Chebyshev polynomials [36, 50, 62] and Legendre polynomials [35, 63], 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 computing the integral cost, finding the state trajectory solution consistent with the prescribed dynamics and 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.*, [34] 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.*, [39, 64]). 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.*, [39, 65]), 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 [66]. 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 D, 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 E and F). The disposition is as follows. To start with, a concise and conceptually clear definition of an observer is given in Section 3.1. 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 3.2 is concerned with how to demonstrate the convergence properties of an observer. Other observer characteristics and classification thereof, are discussed more thoroughly in Section 3.3. One of the main paths for observer design, namely via nonlinear coordinate transformations, is discussed in Section 3.4. Finally, the main purpose of Section 3.5 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.

3.1 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 3.1 (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 3.1 (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 3.1 is commutative (*cf. *[67] and [68]).*
Here, Ψ_* denotes the tangent mapping, π is projection upon a Cartesian factor, while τ denotes the projection of the tangent bundle.

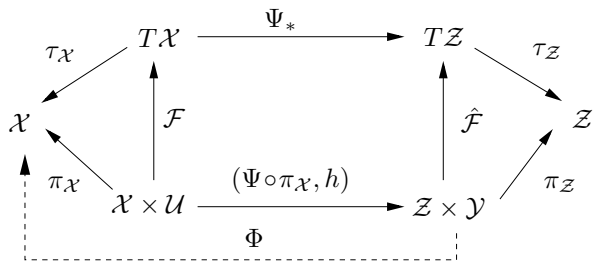


Figure 3.1: Commutative diagram defining an observer.

According to Definition 3.1, 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

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

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

Proposition 3.1. *An observer has Property 3.1 if and only if the diagram in Figure 3.1 is commutative.*

Proof. Assume we have Property 3.1, *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 3.1 implies that

$$\frac{\partial \Psi}{\partial x}\mathcal{F}(x,u) = \hat{\mathcal{F}}(\Psi(x), h(x,u)), \tag{3.1}$$

which is exactly what the commutativity of the diagram in Figure (3.1) suggests.

In the other direction, assume that the diagram in Figure (3.1) 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

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

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

Definition 3.1 thus provides a clear representation of Property 3.1 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 [69]. 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 3.3. In particular, it is reasonable to require the following additional property.

Property 3.2. *As time proceeds, the trajectories $z(t)$ and $\Psi(x(t))$ converge.*

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 3.2, where two approaches for showing convergence, namely Lyapunov-based methods and contraction analysis, are presented.

3.2 Convergence Analysis

3.2.1 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.*, [70–73]), 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 [71]). 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 [74, 75], 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

$$\dot{w} = \dot{z} - \frac{\partial \Psi}{\partial x}\dot{x} = \hat{\mathcal{F}}(z, y) - \frac{\partial \Psi}{\partial x}\mathcal{F}(x, u) \triangleq f(t, w). \tag{3.2}$$

It is noteworthy that the commutativity of the diagram in Figure 3.1, makes the origin, $z(t) = \Psi(x(t))$, an equilibrium point of (3.2) (see also (3.1)). In general, error dynamics (3.2) is nonlinear and is further 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 (3.2). They also demonstrate the difficulty in making any statements regarding the convergence properties of the observer without any further specification of the functions involved.

3.2.2 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 [76]. 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 at time t between two neighboring trajectories, $z_1(\cdot)$ and $z_2(\cdot)$ (see Figure 3.2). 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 ($\mathcal{L}_{\hat{\mathcal{F}}} g < 0$) for every input $y \in \mathcal{Y}$, then

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

that is, the Riemannian distance between any two trajectories tends to zero as time proceeds (*cf.* [76]).

Contraction solely implies that the Riemannian distance between neighboring trajectories tends to zero. In order to draw conclusions 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 3.1. In conjunction with contraction, Property 3.1 automatically yields Property 3.2, *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 3.1 commutative, then the observer is convergent.

Remark 3.1. Contraction, as described 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.

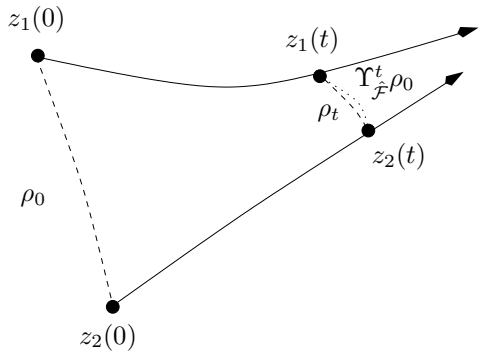


Figure 3.2: The length of the geodesic curve ρ_t, between two trajectories decreases if $\mathcal{L}_{\hat{\mathcal{F}}} g < 0$.

With this point of view, the control synthesis and observer design are decoupled processes and do not interact. To set the stage for the concept of *active observer design* of Paper E, 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 to this integrated fashion of control synthesis and observer design (*cf.* Paper E). 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 [77].

The assumption that the observer dynamics is contractive, is however very restrictive and in many cases Property 3.2 has to be shown by means of Lyapunov–based methods.

3.3 Observer Properties and Classification

The advantages of representing an observer as in Definition 3.1, 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 dependence on 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 [76], 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 $d > 0$, one can possibly tune the observer design parameters so that convergence is guaranteed for all $z(t_0) \in \mathbb{B}(\Psi(x(t_0)), d)$.

Remark 3.2. 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 $\mathcal{X}$). If compact, then by the smoothness (and hence continuity) of Ψ, this also bounds $\Psi(x(t_0))$ in $\mathcal{Z}$.

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 3.2 and Remark 3.5.

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 E, 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.* [78]).

- *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 is 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 3.3. The attentive reader might here discern the connection of input dependent convergence with Remark 3.1 and the succeeding discussion on active observer design (*cf.* Paper E).

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 [79], *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.*, [79, 80] and [43] 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 operation, 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.* [81]). 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 3.4. The intrinsic observer of Paper F, 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 3.1 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 3.1: 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 3.2 (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 3.5. 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 3.3 (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 3.4 (Smooth and Continuous Observers [81]). *Referring to diagram 3.1, 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 [82]. For an example, explicitly constructed to show the non-existence of a smooth observer and point out the potential of continuous observers, consult [83].

Definition 3.5 (Reduced Order, Full Order and Expanded Order Observers [81,83]). *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 3.6 (Uniform and Non–Uniform Observer [78]). *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 3.6. If one considers disturbances as unknown (unmeasured) inputs to the original system, *robust observers* [84] may be seen to equal the concept of uniform observers.

3.4 Coordinate Transformations and Linear Error Dynamics

The problem of existence and synthesis of observers for linear systems is fully understood [69]. 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 [85–91]. In these new coordinates, classical methods from linear systems theory are employed to complete the observer design procedure.

In two seminal papers [86,87] – 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 permanently established. 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

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

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

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

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;

$$\frac{\mathrm{d}}{\mathrm{d}t}(z-\tilde{x}) = \dot{z} - \dot{\tilde{x}} = Az - g(y,u) + L(\tilde{y} - Cz) - A\tilde{x} + g(y,u) = (A-LC)(z-\tilde{x}). \quad (3.3)$$

Since (A, C) is an observable pair, pole placement can be used to obtain exponential convergence (with arbitrary rate) for the *linear* error dynamics (3.3).

More recently, Kazantzis and Kravaris [92] 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)

$$\tilde{\Sigma}' : \begin{cases} \dot{\tilde{x}} &= A\tilde{x} - g(y,u) \\ y &= h(T^{-1}(\tilde{x})) \end{cases}$$

where the matrix A is Hurwitz and $g(y,u)$ is locally analytic around the origin with $g(0,0) = 0$. It is then possible to propose a full-state observer for the original system, Σ, namely the dynamic system

$$\hat{\Sigma}' : \begin{cases} \dot{z} &= Az - g(y,u) \\ \hat{\tilde{x}} &= \mathbb{I}(z), \end{cases}$$

with the associated linear stable error dynamics

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

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

$$\dot{\hat{x}} = \mathcal{F}(\hat{x},u) + L(\hat{x})\big[g(y,u) - g(h(\hat{x}),u)\big], \quad (3.4)$$

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.

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 $\tilde{\Sigma}$ 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 [92] and [93][3] approximation schemes based on 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 [94].

3.5 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 [78, 83]. This is in accordance with linear systems theory.

Example 3.1 (observability $\not\Rightarrow$ observer). Consider

$$\Sigma_1 \begin{cases} \dot{x}_1 &= -x_1 + ux_2^3 \\ \dot{x}_2 &= x_2 + x_1^2 \\ y &= x_1 \end{cases}$$

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

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

$$y_0(x_1, t) \equiv y_0(x_2, t), \quad \forall t \geq 0. \tag{3.5}$$

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 (3.5) we immediately obtain $x_{11} = x_{21}$, for the special choice of $t = 0$. Equation (3.5) 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 [95] (see also [83]) 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 3.2 (observability $\not\Leftarrow$ observer). Consider

$$\Sigma_2 \begin{cases} \dot{x}_1 &= u \\ \dot{x}_2 &= x_1 + x_2^2 \\ \dot{x}_3 &= -x_3 + x_2 \\ y &= x_2 \end{cases}$$

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

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

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.

This concludes this part of the thesis. In the next section, the reader will be guided through each of the six appended papers.

Reader's Guide

This section offers a brief and descriptive summary of the six papers appended in this thesis. Papers A and B deal with different cooperative surveillance missions using multiple UGVs and in particular focus on the minimum time and connectivity constrained formulations which are treated within an optimization based framework. The subsequent two papers (Papers C and D) consider different aspects of the online trajectory planning problem. Finally, Papers E and F deal with observer design for mobile robotic and Euler-Lagrange systems, respectively. Next, a short description of all these papers follows.

Paper A: *Minimum Time Surveillance using Multiple Unmanned Ground Vehicles*, coauthored with P. Ögren.

In Paper A an optimization based framework for dealing with cooperative surveillance missions involving multiple UGVs is proposed. More precisely, this paper focuses on the so called Minimum Time UGV Surveillance Problem (MTUSP). Informally, MTUSP can be described as follows. Given a set of surveillance UGVs and an area to be surveyed, find waypoint-paths for all UGVs such that every point of the area is visible from a point on a waypoint-path and such that the time for executing the search in parallel is minimized. Here, the field of view of the sensors are assumed to have limited coverage range and be occluded by the obstacles.

In the case when maximum sensor range is the only limitation on the sensors' field of view - which is relevant in application domains such as vacuum cleaning and demining [96, 97] - many papers study the minimum time coverage problem. However, when occlusion is also taken into account, as in the MTUSP formulation, we have found no paper addressing the minimum time objective. The main contribution of this paper is to formulate such a problem, show $\mathcal{NP}$-hardness of it and then propose decomposition techniques that allow us to find an approximative solution efficiently in an algorithmic manner.

The MTUSP solution method presented in this paper has been implemented both in MATLAB and in C^{++}. The later implementation runs as a part of a demonstration testbed developed within the Technologies for Autonomous and Intelligent Systems (TAIS) project[1] and has been depicted in Figure A.9. The performance of the proposed solution algorithm will be further evaluated in real-world experiments that are to be carried out in mid April 2009 in cooperation with Rotundus AB, the manufacturer of the surveillance UGV Groundbot (see Figure A.1).

Paper A is based on:

A1: D.A. Anisi, P. Ögren and X. Hu, *Cooperative Minimum Time Surveillance with Multiple Ground Vehicles*, Submitted to the IEEE Transactions on Automatic Control, Jan., 2009.

[1] Project 297316-LB704859.

A2: D.A. Anisi, P. Ögren and X. Hu, *Cooperative Surveillance Missions with Multiple UGVs*, IEEE Conference on Decision and Control (CDC), Cancun, Mexico, Dec., 2008.

A3: D.A. Anisi and P. Ögren, *Minimum Time Multi-UGV Surveillance*, In Optimization and Cooperative Control Strategies, Lecture Notes in Control and Information Sciences, Springer-Verlag, 2008.

A4: J. Thunberg, D.A. Anisi and P. Ögren, *A Comparative Study of Task Assignment and Path Planning Methods for Multi-UGV missions*, In Optimization and Cooperative Control Strategies, Lecture Notes in Control and Information Sciences, Springer-Verlag, 2008.

A5: D.A. Anisi and J. Thunberg *Survey of Patrolling Algorithms for Surveillance UGVs*, Scientific Report, Swedish Defence Research Agency (FOI), FOI-R–2266–SE, Apr., 2007.

Paper B: *Connectivity Constrained Surveillance using Multiple Unmanned Ground Vehicles*, coauthored with P. Ögren and X. Hu.

The Connectivity Constrained UGV Surveillance Problem (CUSP) considered in Paper B can be described as follows. Given a set of surveillance UGVs and a user defined area to be covered, find waypoint-paths for all UGVs such that every point of the area can be seen from a point on a waypoint-path, the induced information graph is kept *recurrently* connected at the time instants when the UGVs perform the surveillance mission, and the time for cooperatively executing the search in parallel is minimized. Hence, the CUSP formulation is an extension of the MTUSP considered in Paper A where connectivity constraints where not taken into account.

In this formulation, the field of view of the onboard sensors are assumed to be occluded by the obstacles and limited by a maximal sensor range. Also, connectivity constraints of both line-of-sight and limited sensor range types are considered.

Paper B provides a formal definition of CUSP, shows that this optimization problem is $\mathcal{NP}$-hard and subsequently, presents decomposition techniques that allow us to find an approximative solution efficiently in an algorithmic manner. In this context, we also introduce and utilize the notion of *recurrent connectivity* of a graph, which is a significantly more flexible connectivity constraint than, *e.g.*, the 1-hop connectivity constraint. The main motivation for introducing this weaker notion of connectivity is security and surveillance applications where the sentry vehicles may have to split temporary in order to complete the given mission efficiently but are required to establish contact recurrently in order to exchange information or to make sure that all units are intact and well-functioning. From a theoretical standpoint, recurrent connectivity is shown to be sufficient for exponential convergence of consensus filters for the collected sensor data.

Paper B is based on:

B1: D.A. Anisi, P. Ögren and X. Hu, *Cooperative Minimum Time Surveillance with Multiple Ground Vehicles*, Submitted to the IEEE Transactions on Automatic Control, Jan., 2009.

B2: D.A. Anisi, P. Ögren and X. Hu, *Cooperative Surveillance Missions with Multiple UGVs*, IEEE Conference on Decision and Control (CDC), Cancun, Mexico, Dec., 2008.

B3: D.A. Anisi, P. Ögren and X. Hu, *Communication Constrained Multi-UGV Surveillance*, IFAC World Congress, Seoul, Korea, July, 2008.

B4: D.A. Anisi, T. Lindskog and P. Ögren, *Algorithms for the Connectivity Constrained UGV Surveillance Problem*, European Control Conference (ECC), Budapest, Hungary, Aug., 2009.

Paper C: *Online Trajectory Planning for Aerial Vehicles: Safety and 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 main objective is to use information about terrain and enemy threats to fly low and avoid radar exposure on the way to a given target. An underlying assumption however, is that due to imperfect information, the kinematic constraints, as well as the location of the target and possible threats, might change during the course of flight. Hence, the trajectory planner should be able to incorporate information updates. 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 approximately solving the trajectory optimization problem.

In order to cope with the real-time objectives, the high-level framework utilized in this work combines the well known approach of Receding Horizon Control (RHC) with a sporadically updated terminal cost that captures the global characteristics of the environment and mission objectives. The terminal cost is calculated off-line and passed to the online receding horizon planner. It should however be mentioned that the possibility of updating the "off-line" computed terminal cost should not be overlooked. As pointed out in [98], the term "off-line" is rather to be interpreted as, at a much slower sampling rate than the trajectory planning 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 online planner through an updated terminal cost, as discussed in [99].

An important issue with RHC is to make sure that the greedy, short term optimization does not lead to long term problems. In the vehicle control domain - since the terminal cost is most often calculated from a graph representation of the environment and as such neglects the vehicle's dynamic constraints - this issue often boils down to two things: not getting into situations where a collision is unavoidable, and making sure that the destination is actually reached.

These issues have been the leitmotivs of Paper C. 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 and existence of a maximal terrain inclination, the planning method presented is shown to have finite time task completion. This is due to a combination of ideas which include a novel safety maneuver combined with a task completing trajectory and a conditional plan-changing strategy, where, starting from a feasible solution, a new plan for the remaining part of the mission is only accepted if it gives an incremental decrease to the terminal cost. 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.

Paper C is based on:

C1: D.A. Anisi, J. Robinson and P. Ögren, *Online Trajectory Planning for Aerial Vehicles: Safety and Task Completion*, Submitted to the AIAA Journal of Guidance, Control, and Dynamics, Mar., 2009.

C2: D.A. Anisi, P. Ögren and J. Robinson, *Safe Receding Horizon Control of an Aerial Vehicle*, IEEE Conference on Decision and Control, San Diego, CA, Dec., 2006.

C3: 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.

C4: 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 D: *Adaptive Node Distribution for Online Trajectory Planning.*

For the design of a computational algorithm for solving a trajectory optimization problem – like the one considered in Paper C – 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).

Direct transcription methods are traditionally 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 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.*, [39, 65]). 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 D 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 is based on:

D1: D.A. Anisi, *Adaptive Node Distribution for Online Trajectory Planning*, Congress of the International Council of the Aeronautical Sciences (ICAS), Hamburg, Germany, Sep., 2006.

D2: D.A. Anisi, *Online Trajectory Planning Using Adaptive Temporal Discretization*, Swedish Workshop on Autonomous Robotics (SWAR), Stockholm, Sweden, Sep., 2005.

Paper E: *Nonlinear Observability and 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 [86, 87, 92] and high gain observers [100, 101], 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 [82]. 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 [102], 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 E 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. Namely, 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.

This paper is based on:

E1: D.A. Anisi and X. Hu, *Nonlinear Observability and Active Observers for Mobile Robotic Systems*, Submitted to Automatica, Jan., 2009.

E2: 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 F: *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.*, [103]). 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 proposed by Aghannan and Rouchon [104]. There, the authors successfully adopt contraction analysis [76], 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 [104] 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

F1: D.A. Anisi and J. Hamberg, *Riemannian Observers for Euler-Lagrange Systems*, IFAC World Congress, Prague, Czech Republic, July 2005.

4.1 Remark on Notation

In the six 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:

+ Papers A and B present an optimization based framework for solving various multi-UGV surveillance missions. In particular, both the minimum time- and connectivity constrained UGV surveillance problems are formulated, their $\mathcal{NP}$-hardness are shown and decomposition techniques are presented that allow us to find an approximative solution efficiently in an algorithmic manner.

+ The concept of a *maximal* convex cover is introduced and utilized in Papers A and B. This is a generalization of a convex partitioning and enables improved minimum time solutions.

+ Paper B also introduces the notion of *recurrent* connectivity of a graph, which is further shown to be sufficient for convergence of consensus filters for the collected sensor data.

+ Paper C extends previous results regarding Receding Horizon Control (RHC) of autonomous vehicles, by addressing safety and task completion properties in a 3D setting.

+ Paper D 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 E and F.

+ Based on this definition, several distinguished observer properties are listed and a classification scheme for observers is proposed.

+ Paper E proposes an alternative methodology for designing active observers for mobile robotic systems.

+ Paper E also proposes a new observability concept called *small-time observability* and provides sufficient condition for it.

+ Paper F 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 performance of the solution algorithms proposed in Papers A and B have to be further evaluated, both in real-world experiments[1] and in comparison with the globally optimal solutions.

[1] Real-world experiments are to be conducted in mid April 2009 in cooperation with Rotundus AB, the manufacturer of the surveillance UGV Groundbot (see Figure A.1).

- Papers A and B could also be extended to recognize the presence of uncertainty more explicitly. This could for instance be handled within a stochastic or robust optimization framework.
- The main results of Paper C and D should be interweaved. That is, investigation should be pursued about the possibility of increasing the accuracy of the safe trajectory planner of Paper C, by using the adaptive node distribution scheme proposed in Paper D.
- In Paper E, 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 of Paper F on more general spaces is prohibitive. Approximation schemes are called for.
- The observer of Paper F should be combined with an intrinsic formulation of state-feedback control. This idea has been elaborated upon in [105,106] in the special case when the manifold is a Lie group and the kinetic energy is left invariant.

5.1 Work Division

Regarding Papers A and B, the problem formulations and the proposed decomposition techniques for solving them efficiently have been developed in close cooperation with Petter Ögren. The subsequent three papers are mainly due to the first author. The coauthors have here provided invaluable inputs by pointing out unclear arguments and suggesting improvements. The lion's share of Paper F 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] Anisi, D. A. and Thunberg, J., *Survey of patrolling algorithms for surveillance UGVs*, Scientific Report, Swedish Defence Research Agency (FOI), FOI-R–2266–SE, Apr. 2007.

[4] Cao, Y., Fukunaga, A., and Kahng, A., "Cooperative Mobile Robotics: Antecedents and Directions," *Autonomous Robots*, Vol. 4, No. 1, 1997, pp. 7–27.

[5] Gerkey, B. and Mataric, M., "A Formal Analysis and Taxonomy of Task Allocation in Multi-Robot Systems," *The International Journal of Robotics Research*, Vol. 23, No. 9, 2004, pp. 939.

[6] Ögren, P., Wirkander, S.-L., Stefansson, A., and Pelo, J., "Formulation and Solution of the UAV Paparazzi Problem," *AIAA Guidance, Navigation, and Control Conference and Exhibit; Keystone, CO, USA*, 21–24 August 2006.

[7] Bertuccelli, L., Alighanbari, M., and How, J., "Robust planning for coupled cooperative UAV missions," *Proc. of the* 43^{rd} *IEEE Conference on Decision and Control (CDC)*, Vol. 3, Paradise Island, Bahamas, Dec. 2004.

[8] Dias, M., Zlot, R., Kalra, N., and Stentz, A., "Market-based multirobot coordination: a survey and analysis," *Proceedings of the IEEE*, Vol. 94, No. 7, 2006, pp. 1257–1270.

[9] Kim, Y., Gu, D.-W., and Postlethwaite, I., "Real-time optimal mission scheduling and flight path selection," *IEEE Transactions on Automatic Control*, Vol. 52, No. 6, 2007, pp. 1119–1123.

[10] Laporte, G., "The vehicle routing problem: An overview of exact and approximate algorithms," *European Journal of Operational Research*, Vol. 59, No. 3, Jun. 1992, pp. 345–358.

[11] Maddula, T., Minai, A., and Polycarpou, M., "Multi-Target assignment and path planning for groups of UAVs," *Recent Developments in Cooperative Control and Optimization*, edited by S. Butenko, R. Murphey, and P. Pardalos, Kluwer Academic Publishers, 2003.

[12] Anisi, D. A. and Ögren, P., "Minimum Time Multi-UGV Surveillance," *Optimization and Cooperative Control Strategies*, edited by M. Hirsch, C. Commander, P. Pardalos, and R. Murphey, Lecture Notes in Control and Information Sciences, Springer Verlag, 2008.

[13] Laporte, G. et al., *Classical and Modern Heuristics for the Vehicle Routing Problem*, Blackwell Synergy, 1999.

[14] Cordeau, J.-F., Gendeau, M., Laporte, G., Potvin, J.-Y., and Semet, F., "A guide to vehicle routing heuristics," *Journal of Operational Research Society*, Vol. 53, No. 5, May 2002, pp. 512–522.

[15] Gerkey, B., Thrun, S., and Gordon, G., "Parallel Stochastic Hill-Climbing with Small Teams," *Multi-Robot Systems: From Swarms to Intelligent Automata*, Vol. 3, 2005, pp. 65–77.

[16] Gambardella, L., Taillard, É., and Agazzi, G., "MACS-VRPTW: a multiple ant colony system for vehicle routing problems with time windows," *Mcgraw-Hill'S Advanced Topics In Computer Science Series*, 1999, pp. 63–76.

[17] John, M., Panton, D., and White, K., "Mission Planning for Regional Surveillance," *Annals of Operations Research*, Vol. 108, 2001, pp. 157–173.

[18] Jones, C., Shell, D., Mataric, M., and Gerkey, B., "Principled Approaches to the Design of Multi-Robot Systems," *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2004.

[19] Kim, Y., Gu, D.-W., and Postlethwaite, I., "A Comprehensive Study on Flight Path Selection Algorithms," *Proc. of the IEE Seminar on Target Tracking: Algorithms and Applications*, 2006, pp. 77–90.

[20] Dudek, G., Jenkin, M., Milios, E., and Wilkes, D., "A taxonomy for multi-agent robotics," *Autonomous Robots*, Vol. 3, No. 4, 1996, pp. 375–397.

[21] Dutta, I., Bogobowicz, A., and Gu, J., "Collective robotics-a survey of control and communication techniques," *Proc. of the International Conference on Intelligent Mechatronics and Automation*, 2004, pp. 505–510.

[22] Espinosa, A., Lerch, J., and Kraut, R., "Explicit vs. Implicit Coordination Mechanisms and Task Dependencies: One Size Does Not Fit All," *Team Cognition: Understanding the factors that drive processes and performance. American Psychological Association*, 2004.

[23] Gerkey, B. and Mataric, M., "Sold!: auction methods for multirobot coordination," *IEEE Transactions on Robotics and Automation*, Vol. 18, No. 5, 2002, pp. 758–768.

[24] Tovey, C., Lagoudakis, M., Jain, S., and Koenig, S., "The generation of bidding rules for auction-based robot coordination," *Proc. of the 3rd International Multi-Robot Systems Workshop*, 2005, pp. 14–16.

[25] Beckers, R., Holland, O., and Deneubourg, J., "From local actions to global tasks: Stigmergy and collective robotics," *Artificial Life IV*, Vol. 181, 1994, pp. 189.

[26] Donald, B., Jennings, J., and Rus, D., "Analyzing teams of cooperating mobile robots," *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, 1994, pp. 1896–1903.

[27] Alighanbari, M., Bertuccelli, L., and How, J., "Filter-Embedded UAV Task Assignment Algorithms for Dynamic Environments," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2004, pp. 1–15.

[28] Alighanbari, M. and How, J., "Decentralized Task Assignment for Unmanned Aerial Vehicles," *Proc. of the* 44^{th} *IEEE Conference on Decision and Control and European Control Conference, CDC-ECC'05*, 2005, pp. 5668–5673.

[29] Frazzoli, E. and Bullo, F., "Decentralized algorithms for vehicle routing in a stochastic time-varying environment," *Proc. of the* 43^{rd} *IEEE Conference on Decision and Control (CDC)*, Paradise Island, Bahamas, Dec. 2004.

[30] 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.

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

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

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

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

[35] 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.

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

[37] 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.

[38] Bellman, R. E., *Dynamic programming*, Princeton University Press, Princeton, N. J., 1957.

[39] 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.

[40] 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).

[41] 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.

[42] 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.

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

[44] 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.

[45] 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.

[46] 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.

[47] 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.

[48] 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.

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

[50] 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.

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

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

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

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

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

[56] 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.

[57] Petit, N., Milam, M. B., and Murray, R. M., "Inversion based constrained trajectory optimization," *Proc. of IFAC Symposium on Nonlinear Control Systems Design*, 2001.

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

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

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

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

[62] 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.

[63] 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.

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

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

[66] 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.

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

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

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

[70] 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.

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

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

[73] 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.

[74] 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.

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

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

[77] 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.

[78] 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.

[79] 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.

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

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

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

[83] 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.

[84] 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.

[85] 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.

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

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

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

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

[90] 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.

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

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

[93] 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).

[94] 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).

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

[96] Acar, E., Choset, H., Zhang, Y., and Schervish, M., "Path Planning for Robotic Demining: Robust Sensor-based Coverage of Unstructured Environments and Probabilistic Methods." *International Journal of Robotics Research*, Vol. 22, No. 7, 2003, pp. 441–466.

[97] Guo, Y., Parker, L., and Madhavan, R., "Towards Collaborative Robots for Infrastructure Security Applications," *Proceedings of The 2004 International Symposium on Collaborative Technologies and Systems*, 2004, pp. 235–240.

[98] 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.

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

[100] 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 (CDC)*, Dec. 1989, pp. 1791–1794.

[101] 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.

[102] 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.

[103] 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.

[104] 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.

[105] 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).

[106] 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

Minimum Time Surveillance Using Multiple UGVs

Minimum Time Surveillance using Multiple Unmanned Ground Vehicles

David A. Anisi and Petter Ögren

Abstract

This work addresses the problem of cooperative surveillance using Unmanned Ground Vehicles (UGVs) such that a user defined area is covered by the UGVs' sensors in minimum time. Here, the field of view of the onboard sensors are assumed to be occluded by the obstacles and limited by maximal sensor range. We first formulate the problem, and show that it is in fact $\mathcal{NP}$-hard. We then propose a solution algorithm that decomposes the problem into three subproblems. The first is to find a maximal convex covering the search area. Most results on static coverage use disjoint partitions of the search area, *e.g.*, triangulation, to convert the continuous sensor positioning problem into a discrete one. However, by a simple example, we show that a highly overlapping set of maximal convex sets is better suited for minimum time coverage. The second subproblem is the combinatorial assignment and ordering of the sets in the cover. Since the Tabu search algorithm is known to perform well on various routing problems, we use it as a part of our proposed solution. Finally, the third subproblem utilizes a particular shortest path sub-routine in order to find the vehicle paths, and calculate the overall objective function used in the Tabu search. The proposed algorithm is illustrated by a number of simulation examples.

Keywords: Surveillance Mission, Minimum-Time Surveillance, Unmanned Ground Vehicles, Combinatorial Optimization

A.1 Introduction

Surveillance is an application area that has received an increasing amount of attention over the last decade. In civilian as well as military applications, automated solutions ranging from security cameras to surveillance UGVs are used in increasing numbers. It is therefore not surprising that the research area of automated positioning and control of surveillance sensors is also active and growing. In this paper we investigate how small scale UGVs, such as the one depicted in Figure A.1, can be used in surveillance and security applications.

For the purpose of this paper, we divide the rich set of work in this field into the following three categories: First moving sensor platforms where the main limitation on field of view is the physical sensor range. Applications where such a formulation is reasonable include

demining, vacuum cleaning and UAV-search. The second category consists of problems dealing with positioning of static sensors where occluding objects and walls present the main limitation on field of view. Such formulations are found in the so called *Art Gallery Problems*, where the number of guards required to monitor a building is to be minimized. The third category consists of problems where moving sensors are to cover an area where again, occluding objects and walls present the main limitation of field of view. This last category includes applications such as pursuit-evasion games or exploration and mapping, in urban or indoor environments. We will now discuss each of these categories in more detail.

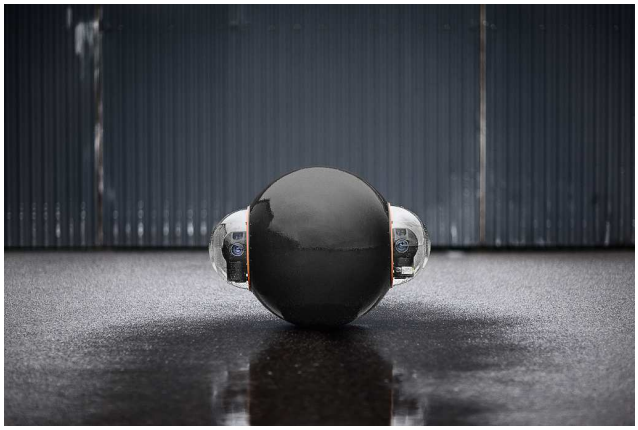


Figure A.1: Within the TAIS project, close cooperation exists with Rotundus AB, the manufacturers of the Surveillance UGV Groundbot, depicted above, which will be used in real-world experiments in mid April, 2009.

In the first category, where sensor range is the main limitation on the extension of the visible area, we find problems such as vacuum cleaning and demining, [1, 2], general coverage [3–5], multi robot coverage [6] and some robotic security applications [7]. Furthermore, a number of UAV surveillance papers, such as [8–10] fall into this category. The last set of papers also consider the combined problems of ordering a set of surveillance areas, and planning the search sweep of each individual area. The pursuit-evasion problem, where a number of pursuers try to find an evader is sometimes also formulated in this way [11].

In the second category, the field of view of stationary sensors is limited by occluding objects instead of physical sensor range. This corresponds to indoor or urban environments, where the distance between, *e.g.*, walls, is in general smaller than the range of the sensor, *e.g.*, a camera or a laser scanner. A large group of results in this category comes from combinatorial geometry, and addresses Art Gallery Problems, see *e.g.*, [12, 13] and the excellent survey in [14]. This work has then been built upon in [15] where a feedback solution to the guard positioning was proposed.

In the third category, where the field of view of moving guards is mainly limited by occluding walls and other objects, we also find results building on the Art Gallery work. In an indoor environment, the pursuit-evasion problem can be solved with a guarantee that the evader will be caught. Such results are found in [16, 17]. Some of this work also deals with the situation where the area is unknown. These problems are sometimes referred to as exploration and mapping, and examples include [18, 19].

Some papers address coverage problems that do not fall into one of the above categories.

Examples include [20], where the mean squared distance from a sensor to a random event is minimized, and [21], where both sensor range and occlusions are incorporated into a combined planning of both UAV and sensor movements.

In the first category, many papers study the problem of covering an area in minimum time. However, when occlusions also limits the field of view, as in the third category, we have found no paper addressing the minimum time coverage problem. The main contribution of this paper is to formulate such a problem, show $\mathcal{NP}$-hardness of it and propose decomposition techniques that allow us to find approximative solutions efficiently in an algorithmic manner. In particular, the proposed decomposition method use the concept of a *maximal convex cover*. A convex cover is a generalization of a convex partitioning, a well known tool for addressing surveillance problems [12]. If the free space to be surveyed is partitioned into convex sets it is enough to visit these sets to survey the whole area. When using a maximal overlapping convex cover instead of a convex partition, we get larger sets, enabling more freedom in the construction of waypoint-paths visiting them, which in turn results in shorter waypoint-paths (see Example A.1 and Remark A.4).

The organization of this paper is as follows. In Section A.2 some concepts and results from combinatorial geometry and multi vehicle routing problems are given. Then, in Section A.3, we state our problem and propose a solution in Section A.4. Simulations illustrating the approach are presented in Section A.5. Finally, the paper is concluded in Section A.6.

A.2 Preliminaries

This paper assumes a previous knowledge of some basic concepts in combinatorial geometry and combinatorial optimization. In particular the so called art gallery problems [12–14] and the multiple traveling salesman problem [22,23] are at principal focus and will be discussed below.

Combinatorial Geometry: Art Gallery Problems

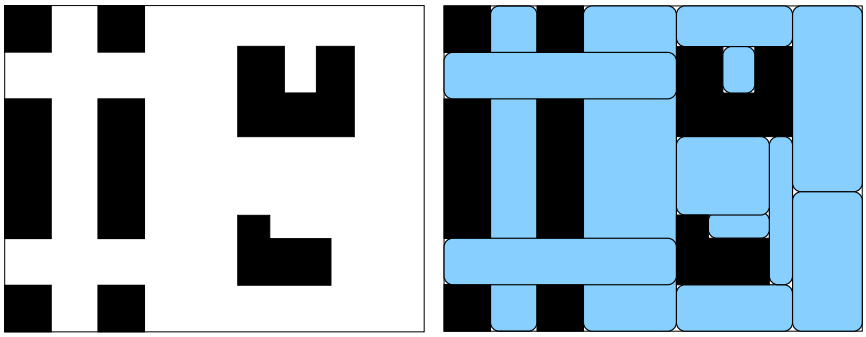
In this section we closely follow Urrutia [14], but add the sensor range R into some of the definitions.

The areas to be searched in this paper are all going to be so called orthogonal polygons with holes (obstacles), thus we denote them A, for area. The orthogonality property is however only important for the maximal convex cover subproblem considered in Section A.4.1. It should be noted that the rest of the proposed solution algorithm is not limited to the orthogonal case, but can handle any general polygon-with-holes type of environment.

Definition A.1 (Orthogonal polygons with holes). *A* polygon Q *in the plane is enclosed by an ordered sequence of points* $q_1, \ldots, q_n \in \mathbb{R}^2,\ n \geq 3$, *called vertices of* Q *together with the line segments* q_i *to* q_{i+1} *and* q_n *to* q_1, *called edges. In the following we assume that none of these edges intersect. A polygon is called* orthogonal *if adjacent edges are orthogonal. Given an orthogonal polygon* Q *and a set of* h *disjoint polygons* $Q_1, \ldots, Q_h$, *that are likewise orthogonal and contained in* Q, *we call the set* $A = Q \backslash \{Q_1 \cup ... \cup Q_h\}$ *an* orthogonal polygon with h holes.

It is now time to state an important result from combinatorial geometry:

Theorem A.1 (Guarding polygons with holes). $\lfloor \frac{v+h}{3} \rfloor$ *point guards are always sufficient and occasionally necessary to guard a polygonal art gallery with* h *holes and a total number of* v *vertices.*



(a) An orthogonal polygon with holes (Definition A.1)

(b) Convex Cover of the polygon. Notice that this cover is not maximal (Definition A.3)

Figure A.2:

The constructive proof of this result can be found in [12,13]. The main idea of the proof is to partition the area A into a set of triangles and note that if there is a guard inside each triangle, then all of that triangle is visible, and hence all of A is guarded. This observation obviously holds for any convex set with diameter less than the maximal sensor range, R. The following definitions and lemma generalizes this argument.

Definition A.2 (Guardiance). *Given two points p and q in A we say that p is* visible *from q if the line segment joining p and q is contained in A*

$$\alpha p + (1-\alpha)q \in A, \quad \forall \alpha \in [0 \;\; 1]$$

and the distance between them is not greater than the sensor range,

$$\|p - q\| \leq R.$$

A set of points $H = \{h_1, \ldots, h_k\} \subset A$ guards *A if for all $p \in A$ there exists $h_i \in H$ such that p is visible from h_i.*

Definition A.3 (Maximal convex cover). *A* convex cover *of A is a set of convex sets $C = \{c_i\}$ such that $|c_i| \leq R$ and $A \subseteq \cup_i c_i$. Here, $|c_i| = \sup_{a,b \in c_i} dist(a,b)$ denotes the diameter of the set c_i. We define a* maximal *convex cover of A to be a convex cover $C = \{c_i\}$ of A, such that for all i, there is no convex set $s \subseteq A$ such that $|s| \leq R$ and $s \supset c_i$.*

Definition A.4 (Visiting waypoint-path). *A* waypoint-path *P is an ordered set of points $P = (p_1, \ldots, p_n) \in \mathbb{R}^{2\times n}$. Any convex cover C is said to be* visited *by the waypoint-path P if $\forall c_i \in C \; \exists p_j \in P : \; p_j \in c_i$.*

Lemma A.1. *If there exists a convex cover C of A such that the waypoint-path P visits C, then P guards A*

Proof. Since P visits C, and every set c_i in C is convex with $|c_i| \leq R$, P guards every set c_i. Furthermore, since $A \subseteq \cup_i c_i$, P guards A. □

Combinatorial Optimization: Multiple Traveling Salesmen Problem

Following [22], the Multiple Traveling Salesmen Problem (MTSP) can be stated as follows. Given a set of cities and the inter-city distances, let there be N salesmen located at given depot nodes. Then the MTSP consists of finding at most N tours, who all start and finish at given depots, such that each city is visited *exactly once* and the total cost of visiting all cities is minimized.

The MTSP can be considered as a relaxation of the closely related Vehicle Routing Problem (VRP), with the capacity restrictions removed [24, 25]. This means that all the formulations and solution methods proposed for the VRP are also valid and applicable to the MTSP by assigning sufficiently large capacities to the salesmen (vehicles). It is a well-known fact that the MTSP and the closely related VRP are $\mathcal{NP}$-hard, and thus represent optimization problems that may be very hard to solve to optimality, see, *e.g.*, [22,24]. Since it has been noted in [24–26] that Tabu search (TS) is an efficient heuristic for a wide range of routing problems, it is reasonable to assume that TS is a good choice for the MTUSP as well and consequently, will be a part of our solution method. Briefly described, TS is a metaheuristic optimization method than can escape local minimas by classifying certain search directions as *tabu*.

A.3 Problem Formulation

In this section we first informally state the Minimum Time UGV Surveillance Problem (MTUSP) and then show that it is $\mathcal{NP}$-hard.

Informally, the problem we are studying is the following: Given a set of surveillance UGVs and a user defined area to be surveyed, find waypoint-paths such that every point of the area can be seen from a point on a waypoint-path and such that the time for cooperatively executing the search in parallel is minimized. As stated earlier, the field of view of the onboard sensors are here assumed to be occluded by the obstacles and limited by a maximal sensor range.

Problem A.1 (Minimum Time UGV Surveillance Problem (MTUSP)). *Given N UGVs and an orthogonal polyhedral area A, find a set of N waypoint-paths $P = (P^1, \ldots, P^N)$ that solve the following optimization problem*

$$\min_{P} \max_{i \in \mathbb{Z}_N^+} \quad \sum_{j=1}^{n_i - 1} \|p_j^i - p_{(j+1)}^i\|$$
$$\textit{such that} \quad \cup_i P^i \ \textit{guards} \ A,$$

where $P^i = (p_1^i, \ldots, p_{n_i}^i), i \in \mathbb{Z}_N^+ = \{1, \ldots, N\}$, and $\| \cdot \|$ denotes the shortest obstacle free distance. The start depots, denoted by p_1^i, are given while the finish depots, $p_{n_i}^i$, may be either free or given.

An example solution to a MTUSP can be found in Figure A.3.

Remark A.1 (Sensor field of view). In the problem statement above we demand that each point in A is visible from some point in P. This is reasonable in the case of omni-directional sensors. It is however also relevant in the case of cameras mounted on pan-tilt units. In these cases the time right before and after passing p_j^i must be used to cover the areas visible from p_j^i. If necessary, the UGVs will have to slow down to facilitate the sensor coverage. A similar argument can be made for the case when the sensor is one or more laser scanners.

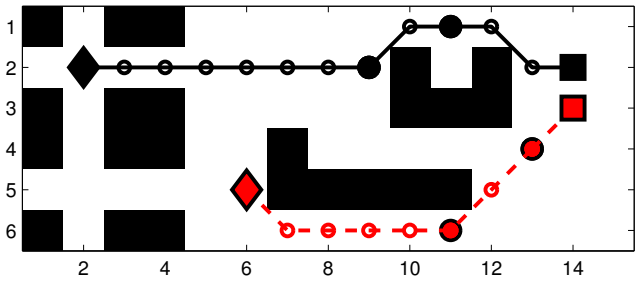


Figure A.3: An approximate solution to the MTUSP involving two UGVs. Note that all the obstacle free area can be seen from some point on the UGV waypoint-paths. Details on this and other simulations can be found in Section A.5.

Remark A.2 (Variations). Throughout this paper we are focusing on the minimum time coverage problem. However, other closely related problems can also be addressed using the same approach. For instance when surveillance of some particular region has higher priority, or when battery power is a scarce resource and the user wishes to minimize the overall distance traveled by the UGVs. A third option is to make the UGVs avoid high threat areas. All these variations can be incorporated into the solution algorithm presented below by simply varying the considered objective function and edge costs.

Remark A.3 (Way-point following). In this problem formulation, it is assumed that there exists a low level controller for each UGV capable of following the generated waypoint-path with bounded error.

We end this section with showing that the problem defined above is $\mathcal{NP}$-hard.

Proposition A.1. *Problem A.1 (MTUSP) is $\mathcal{NP}$-hard.*

Proof. The proof will built upon a polynomial reduction from an *arbitrary* instance of a well-known $\mathcal{NP}$-hard problem, namely the Euclidean-TSP (ETSP)[1] to a *special* instance of MTUSP[2].

Given an ETSP instance, $(n, [d_{ij}])$, where n is the number of cities to be visited and $[d_{ij}]$ denotes the inter-city distances, we are free to choose the following parts of Problem A.1 (MTUSP) such that the achieved optimal solution corresponds to that of the given ETSP.

1. The number of UGVs, N
2. The start and finish depots for all UGVs, $p_1^i, p_{n+1}^i, i \in \mathcal{Z}_N^+$
3. The obstacle configuration
4. The area to be surveyed, A
5. The maximal sensor range, R.

[1] The ETSP is a special case of MTSP discussed in Section A.2 with only one salesmen and the inter-city distances fulfilling the triangle inequality

$$d_{ij} \leq d_{ik} + d_{kj}, \quad \forall i, j, k \in \{1, \ldots, n\}.$$

[2] Consult [23, 27] to read more about showing $\mathcal{NP}$-hardness through reduction.

Regarding the number of UGVs, $N = 1$ is a natural selection. In order to achieve a tour for this single UGV, we may locate the start and finish depot, p_1^1, p_{n+1}^1 at an arbitrary city cite, as long as they are set equal. Further, an obstacle-free environment is chosen and the area A is taken as the union of isolated points located at the city cites. Finally, we set $R = 0$.

Due to these choices, the area A is fully guarded if and only if the UGV visits all the city locations and since the distances are preserved, the optimal solution of this specially designed instance of the MTUSP will coincide with the optimal solution of the given ETSP. This completes the proof. □

Knowing that MTUSP is $\mathcal{NP}$-hard, we can not hope to solve all problem instances to optimality in reasonable time but must adopt heuristic solution methods. The comparative runs presented in Figure A.13 and Table A.1 serve as a testimony of this. In fact, the more comprehensive comparative study [26] shows that for similar class of problems, finding globally optimal solutions (by using an exact Mixed Integer Linear Programming (MILP) formulation) is not a viable approach for larger problem instances.

A.4 Proposed Solution

In this section we will propose a solution to the MTUSP described above. The solution encompasses three subproblems, as illustrated in Figure A.4. In the first subproblem, the

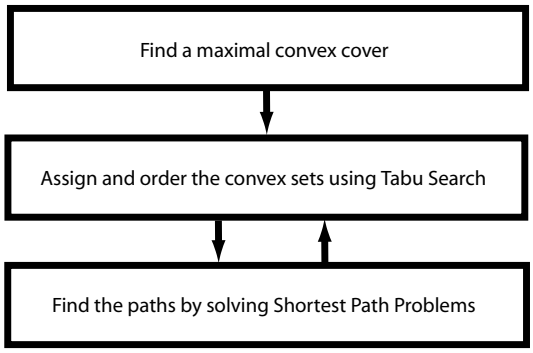


Figure A.4: The proposed solution relies on decomposing the problem into three subproblems.

computationally intractable problem of finding the minimum time waypoint-paths that enable *complete* regional surveillance, is turned into a finite dimensional combinatorial optimization problem. This is achieved by finding a *maximal convex cover* of A, as defined in Section A.2. In the second subproblem, the order in which to visit the sets in the cover is determined using Tabu search (TS). The third subproblem, which is called as a subroutine of the second one to evaluate the objective function in the TS, involves a shortest path problem on a graph, constructed from the given visitation order.

Formally we state the algorithm below.

Algorithm A.1 (Proposed solution for MTUSP)**.** The algorithm consists of the following two steps where the second step involves the iterative solution of two subproblems:

1. Create a maximal convex cover $C = \{c_1, \ldots, c_M\}$ of A in accordance with Algorithm A.2.

2. Solve the following combined assignment and ordering problem using TS:

$$\min_{\pi} F(\pi) = \alpha \max_{i} f_i(\pi) + (1-\alpha)\Sigma_i f_i(\pi), \tag{A.1}$$

where π is a permutation of $\mathcal{Z}^+_{M+N}$, representing the assignment/ordering of the M convex sets to the N vehicles (see Examples A.1 and A.2), $\alpha \in [0\ 1]$, and $f_i(\pi)$ is the optimal path length of UGV i given the visitation constraints dictated by π. The value of $f_i(\pi)$ is found in a sub-routine by using a shortest path formulation to solve the following optimization problem:

$$\begin{aligned} f_i(\pi) = \quad & \min_{P^i} \quad \Sigma_k ||p^i_k - p^i_{(k+1)}|| \\ & \text{s.t.} \quad P^i \text{ guards } \cup_{I^\pi_i} c_j \\ & \qquad P^i \text{ visits } c_{I^\pi_i(j)} \text{ before } c_{I^\pi_i(j+1)}, j \in \mathcal{Z}^+_{|I^\pi_i|-1} \end{aligned} \tag{A.2}$$

In A.2, I^π_i is the index of the sets in C that are assigned to UGV i in the minimization of F. In (A.1), $\alpha = 1$ corresponds to the minimum time problem and $\alpha = 0$ corresponds to the minimum distance problem. Examples of both these options are found in Figures A.17 and A.18 in Section A.5.

Example A.1. A simple example problem with one UGV is depicted in Figure A.5 where the start position, p_1, is given while the end point is a free variable. In step 1 of Algorithm A.1, the maximal convex cover $C = \{c_1, c_2, c_3, c_4\}$ is created. In step 2 the visitation ordering $\pi = (\underline{5}, 2, 3, 4, 1)$ is first tried. In accordance with Example A.2, this permutation corresponds to the UGV (which has id number 5) visiting the convex areas in the following order: $c_2 \rightarrow c_3 \rightarrow c_4 \rightarrow c_1$. The shortest possible waypoint-path visiting the sets in this order is (p_1, p_2, p_3), and is thus returned in step 3, after minimizing $f_i(\pi)$. In the next iteration of step 2 the ordering $\pi = (\underline{5}, 1, 2, 3, 4)$ is chosen, with step 3 returning only (p_1, p_2). After some additional iterations no improvement is found and the algorithm terminates and returns (p_1, p_2).

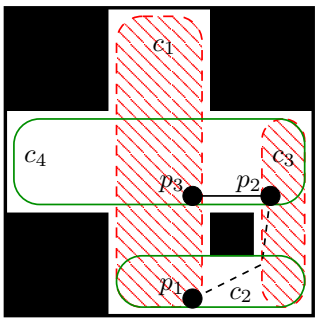


Figure A.5: An example MTUSP problem. One UGV starts at p_1 with free endpoint. The optimal solution corresponds to the dashed line, while a suboptimal one also includes p_3, as explained in Example A.1. The four sets of the maximal convex cover are denoted c_1, c_2, c_3, c_4.

Remark A.4. If a convex partitioning was used instead of a maximal convex cover, the corresponding shortest waypoint-path visiting all sets would not be as short as the solution above (*cf.* Figure A.6).

Remark A.5. Note that decomposing the problem into subproblems, might remove the optimal solution from the new set of feasible solutions. However, since it is shown in Proposition A.1 that Problem A.1 (MTUSP) is $\mathcal{NP}$-hard, our aim is not to solve the problem to optimality, but rather to produce good-enough solutions in reasonable time.

Remark A.6. One straight-forward solution to the MTUSP would be to first solve an art gallery problem as suggested by the constructive proof of Theorem A.1 to find a small set of points guarding A, and then solve an MTSP visiting these points. This approach however is not suitable for the minimum time objective considered in MTUSP since the surveillance points are chosen to be as few as possible, not to permit short vehicle paths.

Having stated Algorithm A.1 we first note that it does indeed result in a complete covering of the surveillance area A, *i.e.*, it produces a feasible solution to Problem A.1.

Proposition A.2. *Algorithm A.1 produces a feasible solution to Problem A.1 (MTUSP).*

Proof. This is clear from Lemma A.1 and the following three observations regarding Algorithm A.1:

1. A convex cover is created.
2. All sets are assigned to different UGVs in (A.1).
3. waypoint-paths visiting all assigned sets are created in (A.2).

□

It is now time to describe and motivate the different subproblems in detail. This will be done in Section A.4.1 through A.4.3.

A.4.1 Finding a maximal convex cover

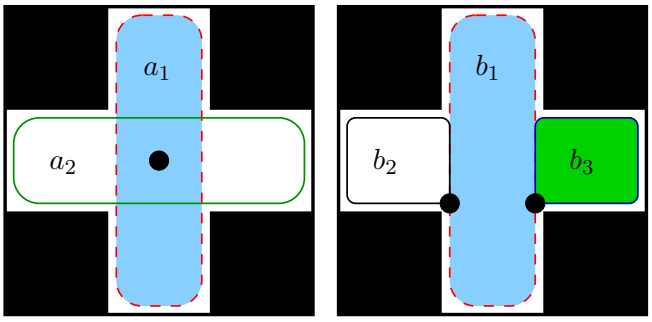
Since the polygons are all orthogonal, one can see that the maximal convex sets must be rectangles aligned with the polygon. With this fact in mind we can apply the following procedure to find a maximal convex cover.

Algorithm A.2 (Maximal convex cover).

1. Make a discretization of the area A and construct the corresponding graph representation, $\mathcal{G}(A)$. Since A is orthogonal, a variable sized grid can be created with grid boundaries intersecting all points in the polygon Q and holes $Q_1, \ldots, Q_m$.
2. Find a yet uncovered cell, p.
3. Start growing a rectangle c_i from p until it is bounded by $|c_i| \leq R$, or the holes on all four sides.
4. While uncovered cells exist, goto 2.

When no more uncovered grid cells can be found the process terminates and A is covered, $A \subseteq \cup_i c_i$. Having described how to find a maximal convex cover in detail, we now discuss a number of related issues.

The benefit of a maximal convex cover is illustrated in Figure A.6 (*cf.* Example A.1). If the area A is cross–shaped, as depicted in the figure, then the entire area can be *instantly* surveyed from any point in $a_1 \cap a_2$. Using disjoint orthogonal sets however, the minimum time waypoint-path for visiting all the orthogonal polygons (b_1, b_2, b_3), and thereby be sure to have surveyed the entire area, is strictly larger than zero.



(a) Maximal convex cover. (b) Rectangular disjoint convex cover.

Figure A.6:

A.4.2 Assignment and ordering of the convex sets

In this section we describe how we propose to solve the optimization problem in (A.1), *i.e.*,

$$\min_{\pi} F(\pi) = \alpha \max_{i} f_i(\pi) + (1-\alpha)\Sigma_i f_i(\pi).$$

Here, π is a permutation representing the assignment/ordering of the M convex sets to the N UGVs, and $f_i(\pi)$ is evaluated by solving another optimization problem, as explained in Section A.4.3 below.

In order to solve the assignment and ordering problems simultaneously we first give the sets and UGVs id-numbers. Assign the id numbers $1, \ldots, M$ to the convex sets $c_i, i \in \mathcal{Z}_M^+$. Let furthermore the N vehicles have id numbers $M+1, \ldots, M+N$. The search space for the TS then consists of permutations of the id numbers, *i.e.*, $\mathcal{Z}_{M+N}^+$. The interpretation of a sequence of id numbers is then best explained by means of the following example.

Example A.2. Let $M = 14$, $N = 3$, and the final sequence be

$$\pi = (\underline{15},\ 1,\ 4,\ \underline{17},\ 10,\ 14,\ 9,\ 3,\ 8,\ \underline{16},\ 2,\ 13,\ 12,\ 7,\ 6,\ 5).$$

This corresponds to the following assignments:

Set with id numbers	assigned to UGV with id number
1 4	15
2 13 12 7 6 5	16
10 14 9 3 8	17

The details of the implementations are, apart from the evaluation of $f_i(\pi)$, identical to those presented in [28]. Hence we refer the interested reader to that paper for a detailed description. We just note that the neighborhood search is performed by pairwise interchanging components in π and the Tabu condition corresponds to requiring a minimum number of iterations before switching a particular pair again.

We now turn to see how the function $f_i(\pi)$ is evaluated in each Tabu step, and how the individual UGV waypoint-paths are found.

A.4.3 Path planning and functional evaluation

In the Tabu step above, each UGV is assigned a number of sets c_i and an order of visitation. Let I_i^π denote this ordered set. The problem is now to decide what part of each convex set to pass through, in order to make the resulting UGV waypoint-path as short as possible while respecting the visitation order dictated by I_i^π. Formally, we need to solve the following optimization problem.

$$\begin{aligned} f_i(\pi) = \quad & \min_{P^i} \quad \Sigma_k ||p_k^i - p_{(k+1)}^i|| \\ & \text{s.t.} \quad P^i \text{ guards } \cup_{I_i^\pi} c_j \\ & \qquad P^i \text{ visits } c_{I_i^\pi(j)} \text{ before } c_{I_i^\pi(j+1)}, j \in \mathcal{Z}^+_{|I_i^\pi|-1} \end{aligned}$$

We will now rewrite this problem as a standard shortest path problem on a so-called Route Graph. Given a pair of starting and finishing positions for each vehicle[3], we construct a particular graph for each vehicle. This graph, which is termed a *Route Graph*, has the starting and finishing positions as its first and last node. As depicted in Figure A.7, the intermediate nodes are extracted from the ordering π and correspond to the nodes of $\mathcal{G}(A)$ inside the convex sets c_j, $j \in I_i^\pi$. To obtain the edge costs for the Route Graph, an *all pairs shortest path problem* [23] is solved in the graph representation of A, $G(A)$.

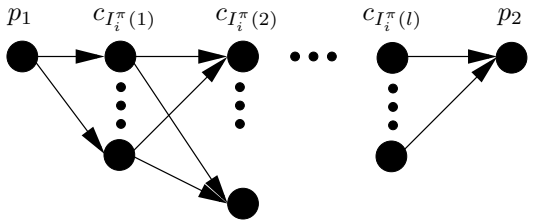


Figure A.7: A graph representation of the route of one UGV.

We illustrate the Route Graph with the following example.

Example A.3. Assume as in the table above that the UGV with id number 15 starts from some point p_1 and is assigned to visit first c_1 and then c_4 on its way to p_2. These sets and positions can be found in Figure A.8.
The shortest waypoint-path starting from p_1, visiting at least one node in c_1 and then visiting at least one node in c_4 and finally ending up at p_2, is plotted in the figure as well as in the Route Graph. As can be seen, the fact that c_1 and c_4 overlap makes $q_{1,4}$ coincide with $q_{4,3}$, enabling a very short waypoint-path.

The evaluation of $f_i(\pi)$ corresponds to solving a shortest path problem in the Route Graph; a task for which polynomial time algorithms such as Dijkstra or A* exist. Note that the solution of this optimization problem yields both the UGV waypoint-path P^i and its length $f_i(\pi)$.

A.5 Simulations

The suggested MTUSP solution method presented in Algorithm A.1 has been implemented both in MATLAB and in C++. The later implementation runs as a part of a demonstration

[3]For applications indifferent to finishing point, it is possible to let the optimization routine choose it freely.

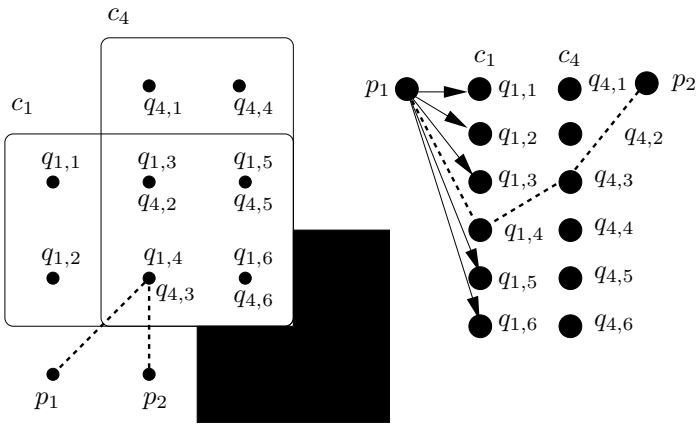


Figure A.8: Example scenario with corresponding Route Graph and optimal waypoint-path (dashed). Note how the fourth node in c_1, $q_{1,4}$, coincides with the third one in c_4, $q_{4,3}$, hence the cost of the edge between them equals zero in the Route Graph.

testbed[4] as depicted in Figure A.9.

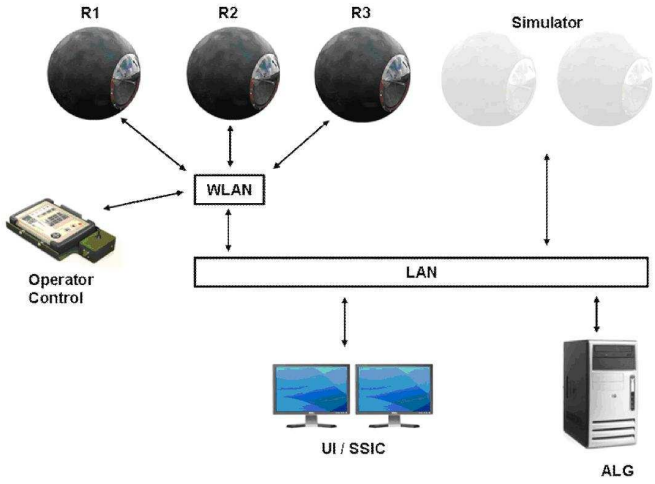


Figure A.9: The high-level design of the demonstration testbed. The algorithm runs on a separate computer and communicates with the UGVs over a network. This setup allows us to work in a mixed HW/SW setting where some of the UGVs only exist in the simulation environment while others are physical robots operating in the real world (R1-R3).

In Figures A.10 and A.11, the areas to be surveyed are bounded within a (white) polygon and the waypoint-paths of the UGVs have been highlighted. The (white) barracks and the (green) tents constitute the obstacles that occlude the onboard cameras. Figure A.12 presents a snapshot view from the onboard cameras.

[4]Developed within the Technologies for Autonomous and Intelligent Systems (TAIS) project, 297316-LB704859.

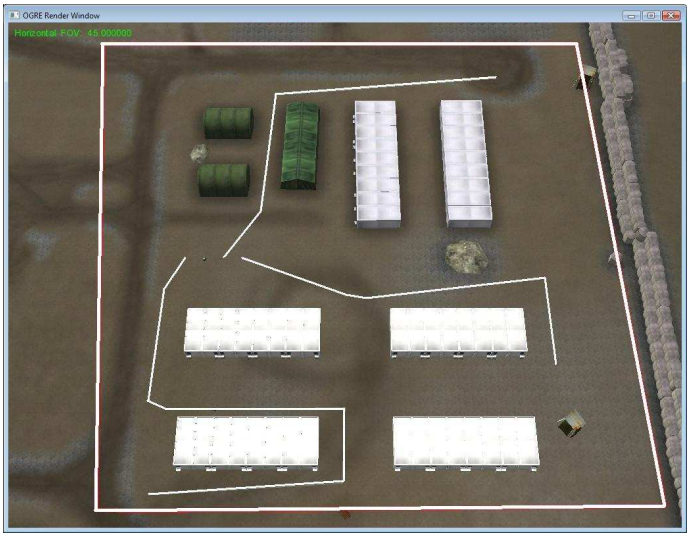


Figure A.10: The convex cover for this particular area contains 20 sets and takes on average 1.0 seconds to generate. The last two subproblems in Algorithm A.1 take on average only 92.5 milliseconds to perform. All computations have been performed on a laptop with Dual Core™, Intel®, 2.0 GHz processors.

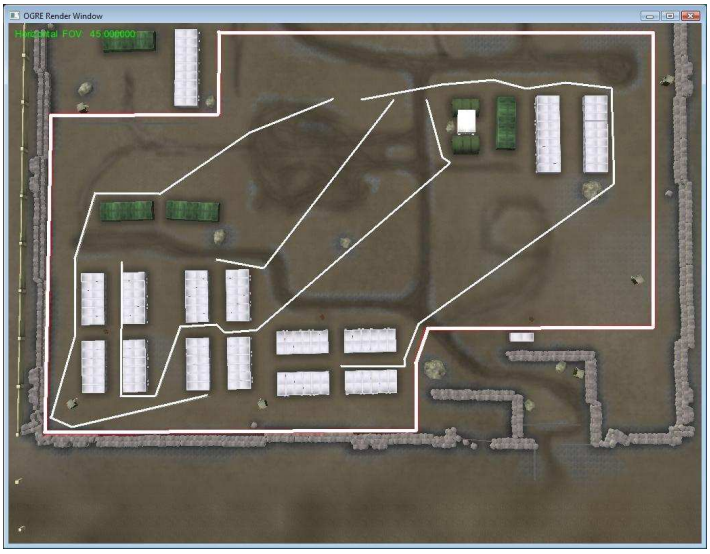


Figure A.11: The convex cover for this larger MTUSP instance contains 47 sets and takes on average 14.2 seconds to generate. The two last subproblems take on average 1.5 seconds to perform.

Figure A.13 and Table A.1 show results from some comparative tests that have been performed in order to validate the claim that finding globally optimal solutions to MTUSP by using an exact Mixed Integer Linear Programming (MILP) formulation, is a viable approach only for smaller problem instances (*cf.* the last paragraph of Section A.3). In this

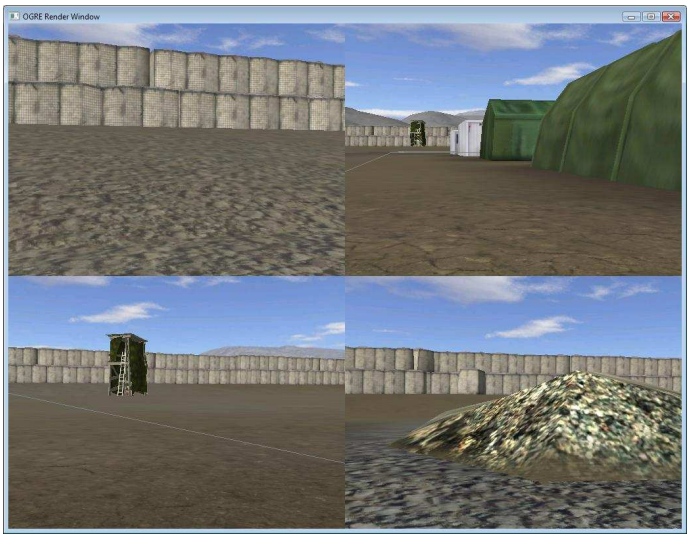


Figure A.12: Snapshot from the onboard cameras of all four UGVs.

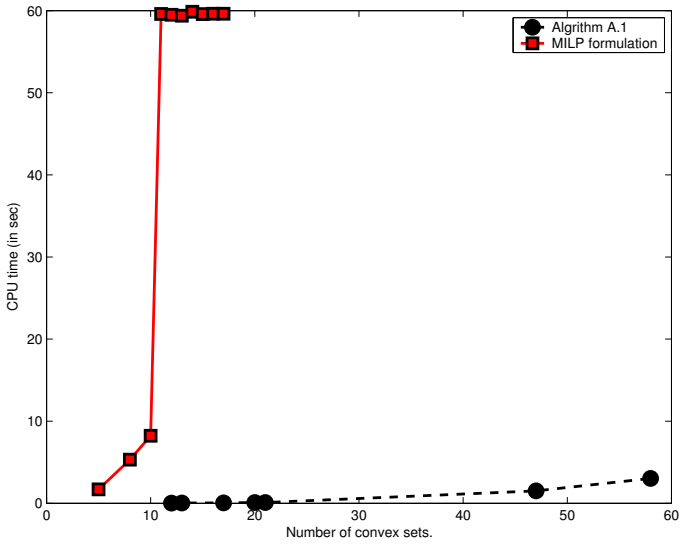


Figure A.13: A comparison between the computational times of the MILP formulation (solid line) and Algorithm A.1 (dashed line). The MILP formulation has been implemented in GAMS and solved by the XPRESS-MP solver via NEOS server. For this, each convex set of the cover is assumed to have 5 nodes and the maximum allowable computation time is set to 60 seconds.

setup, the comparison between the MILP formulation and Algorithm A.1 only considers the assignment/ordering of the convex sets and the path planning subproblem. Hence, the generation of the convex cover is assumed to be done in a similar fashion in both cases. Running times for generation of the convex covers are presented separately in Figure A.14.

# of sets in the cover	5	8	10	11	12	13	15	17
# of binary variables	504	1407	2259	2760	3311	3912	5264	6816
Computation time MILP/XPRESS	1.7 s	5.3 s	8.2 s	60 s	60 s	60 s	60 s	60 s
Relative optimality gap	0	0	0	5.3%	15%	22%	35%	43%

Table A.1: In the exact MILP formulation, the optimal solution was only found for the three smallest problem instances. Notice that the maximum allowable computation time was here set to 60 seconds. For problem instances having more than 12 sets in the convex cover, the relative gap from optimality varied between 15-42%. Hence, the MILP formulation fails to meet our requirements on computational efficiency for this class of problems.

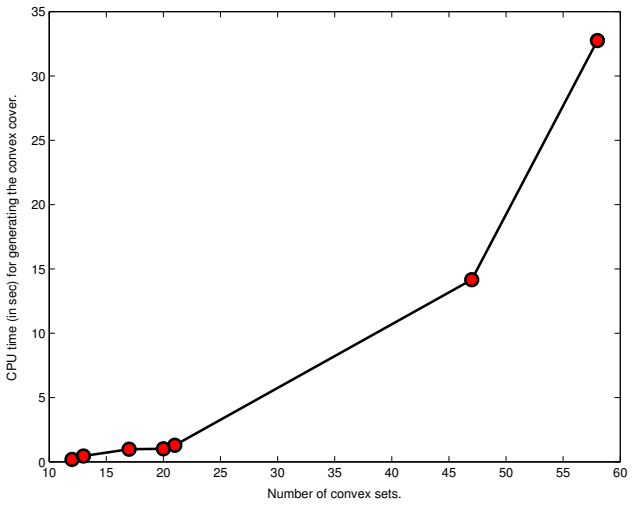


Figure A.14: The running times (in seconds) for generation of the convex covers for areas with different complexities. Notice that in most applications, the generation of the convex cover can be performed off-line. Local modifications to this cover can then be incorporated online as new information about the environment is gathered.

In the MATLAB simulations that follow, the initial position of the UGVs are marked with a square (■), while the final positions are marked with a diamond (♦). These two, together with the filled larger circles represent the surveillance points for guarding A. The search area, A, is chosen to be all of the obstacle free space, *i.e.*, the white area in all figures. It is assumed that the black obstacles have been enlarged with the diameter of the vehicle so that waypoint-paths touching an obstacle do not imply collision.

The first two simulations, found in Figures A.3 and A.15, illustrate the cooperative nature of the MTUSP. The final positions of the vehicles are here *free* variables to be chosen by Algorithm A.1. As can be seen, this extra degree of freedom is used constructively so that the vehicles survey the horizontally and vertically aligned "streets" in a cooperative manner with the common objective of minimizing the search time, *i.e.*, we have chosen $\alpha = 1$ in (A.1). These simulations are also a testimony of the advantage of using a highly overlapping cover.

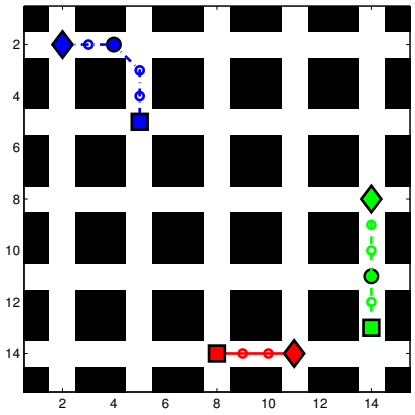


Figure A.15: The Manhattan grid is surveyed cooperatively in minimum time. The starting points of the UGVs are marked with ■.

Figure A.16 illustrates a possible drawback of choosing the objective function as pure minimum time. Here, the route of the vehicle to the left, (dash-dotted), is unnecessarily long since a complete coverage would also have been achieved if the vehicle did not move at all. However, the minimum time objective has no way of distinguishing between these two solutions and regards them as equally good since the time for executing them in parallel is indeed equal.

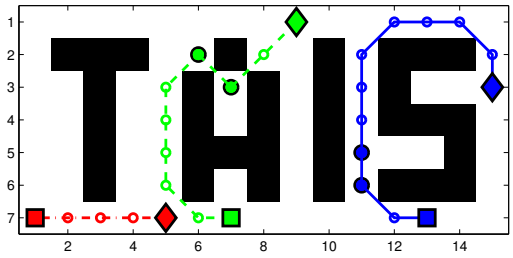
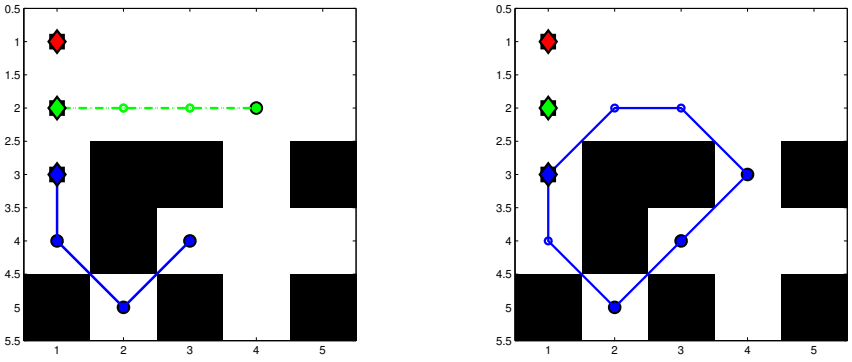


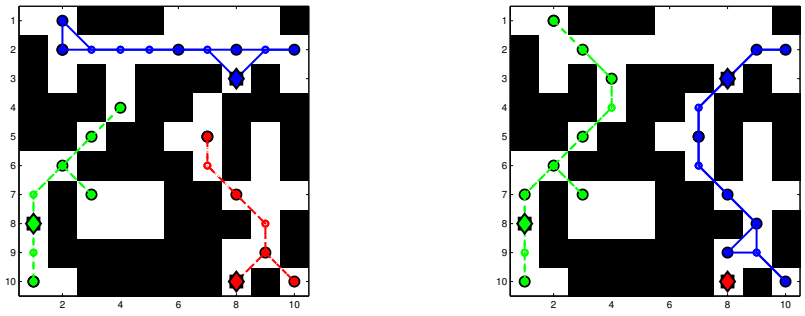
Figure A.16: Minimum time surveillance. Note how the UGVs collectively guard A.

Figures A.17 and A.18 further illuminate the interplay between the choice of the objective function and the obtained solutions. In Figures A.17(a) and A.18(a), the solutions are found by minimizing the total surveillance time. It can be noted that these solutions distribute the work load quite evenly over the vehicle fleet. In Figures A.17(b) and A.18(b) however, the objective has been set to minimize the total distance traveled by the vehicles, *i.e.*, $\alpha = 0$ in (A.1). Since this option does not take into consideration the division of the work load between the different vehicles, the resulting solutions often do not utilize some of the vehicles at all. This may be of tactical interest when, *e.g.*, battery power must be saved, or when unemployed vehicles can be used to perform other missions in parallel.



(a) Minimum time objective. (b) Minimum distance objective.

Figure A.17:



(a) Minimum time objective. (b) Minimum distance objective.

Figure A.18:

A.6 Concluding Remarks

The Minimum Time UGV Surveillance Problem (MTUSP), where it is both occlusion and maximum sensor range, that are the main limitation to the sensors' field of view, is at the focal point of this paper. We initially show that this problem is in fact $\mathcal{NP}$-hard, hence we cannot hope to solve all instances to optimality in reasonable time. We then proceed by proposing a decomposed solution method that encompasses finding a maximal convex cover, performing Tabu search on the assignment and ordering of the convex cover and finally, solving shortest path problems in the so called Route Graphs. The simulations demonstrate the advantage of using a highly overlapping convex cover, the cooperative nature of the MTUSP, but also the interplay between minimum time- and minimum distance solutions.

A.7 References

[1] Acar, E., Choset, H., Zhang, Y., and Schervish, M., "Path Planning for Robotic Demining: Robust Sensor-based Coverage of Unstructured Environments and Probabilistic Methods." *International Journal of Robotics Research*, Vol. 22, No. 7, 2003, pp. 441–466.

[2] Guo, Y., Parker, L., and Madhavan, R., "Towards Collaborative Robots for Infrastructure Security Applications," *Proceedings of The 2004 International Symposium on Collaborative Technologies and Systems*, 2004, pp. 235–240.

[3] Choset, H., "Coverage for Robotics–A Survey of Recent Results," *Annals of Mathematics and Artificial Intelligence*, Vol. 31, No. 1, 2001, pp. 113–126.

[4] Hazon, N. and Kaminka, G., "Redundancy, Efficiency and Robustness in Multi-Robot Coverage," *Robotics and Automation, 2005. Proceedings of the 2005 IEEE International Conference on*, 2005, pp. 747–753.

[5] Huang, W., "Optimal Line-Sweep-Based Decompositions for Coverage Algorithms," *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, Vol. 1, 2001.

[6] Ge, S. and Fua, C., "Complete Multi-Robot Coverage of Unknown Environments with Minimum Repeated Coverage," *Proceedings of the IEEE International Conference on Robotics and Automation*, 2005.

[7] Carroll, D., Everett, H., Gilbreath, G., and Mullens, K., "Extending Mobile Security Robots to Force Protection Missions," *AUVSI Unmanned Systems*, 2002, pp. 9–11.

[8] Ablavsky, V., Stouch, D., and Snorrason, M., "Search Path Optimization for UAVs Using Stochastic Sampling With Abstract Pattern Descriptors," *Proceedings of the AIAA Guidance Navigation and Control Conference, Austin, TX, August*, 2003.

[9] Panton, D. M. and Elbers, A. W., "Mission Planning for Synthetic Aperture Radar Surveillance," *Interfaces*, Vol. 29, No. 2, 1999, pp. 73–88.

[10] John, M., Panton, D., and White, K., "Mission Planning for Regional Surveillance," *Annals of Operations Research*, Vol. 108, 2001, pp. 157–173.

[11] Hespanha, J., Kim, H., and Sastry, S., "Multiple-Agent Probabilistic Pursuit-Evasion Games," *Decision and Control, 1999. Proceedings of the 38th IEEE Conference on, Phoenix, AZ*, Vol. 3, 1999, pp. 2432–2437.

[12] Bjorling-Sachs, I. and Souvaine, D., "A Tight Bound for Guarding Polygons With Holes," Tech. rep., Report LCSR-TR-165, Lab. Comput. Sci. Res., Rutgers Univ., New Brunswick, NJ, 1991.

[13] Hoffmann, F., Kaufmann, M., and Kriegel, K., "The art gallery theorem for polygons with holes," *32nd Annual Symposium on Foundations of Computer Science (San Juan, PR, 1991)*, IEEE Comput. Soc. Press, Los Alamitos, CA, 1991, pp. 39–48.

[14] Urrutia, J., "Art Gallery and Illumination Problems," *Handbook of computational geometry*, edited by J.-R. Sack and J. Urrutia, North-Holland Publishing Co., 2000, pp. 973–1027.

[15] Ganguli, A., Cortes, J., and Bullo, F., "Distributed Deployment of Asynchronous Guards in Art Galleries," *Proceedings of the 2006 American Control Conference*, 2006.

[16] Gerkey, B., Thrun, S., and Gordon, G., "Visibility-Based Pursuit-Evasion with Limited Field of View," *The International Journal of Robotics Research*, Vol. 25, No. 4, 2006, pp. 299–315.

[17] Efrat, A., Guibas, L., Har-Peled, S., Lin, D., Mitchell, J., and Murali, T., "Sweeping Simple Polygons With a Chain of Guards," *Proceedings of the 11th ACM-SIAM Symposium on Discrete Algorithms*, 2000, pp. 927–936.

[18] Yamauchi, B., "Frontier-Based Exploration Using Multiple Robots," *Proceedings of the second international conference on Autonomous agents*, 1998, pp. 47–53.

[19] Burgard, W., Moors, M., Fox, D., Simmons, R., and Thrun, S., "Collaborative Multi-Robot Exploration," *IEEE International Conference on Robotics and Automation (ICRA)*, Vol. 1, 2000.

[20] Cortes, J., Martinez, S., Karatas, T., and Bullo, F., "Coverage Control for Mobile Sensing Networks," *IEEE Transactions on Robotics and Automation*, Vol. 20, No. 2, 2004, pp. 243–255.

[21] Skoglar, P., Nygards, J., and Ulvklo, M., "Concurrent Path and Sensor Planning for a UAV - Towards an Information Based Approach Incorporating Models of Environment and Sensor," *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2006*, Vol. 1, 2006.

[22] Bektas, T., "The multiple traveling salesman problem: an overview of formulations and solution procedures," *Omega*, Vol. 34, No. 3, Jun. 2006, pp. 209–219.

[23] Papadimitriou, C. H. and Steiglitz, K., *Combinatorial optimization: algorithms and complexity*, Dover Publications Inc., Mineola, NY, 1998.

[24] Laporte, G., "The vehicle routing problem: An overview of exact and approximate algorithms," *European Journal of Operational Research*, Vol. 59, No. 3, Jun. 1992, pp. 345–358.

[25] Cordeau, J.-F., Gendeau, M., Laporte, G., Potvin, J.-Y., and Semet, F., "A guide to vehicle routing heuristics," *Journal of Operational Research Society*, Vol. 53, No. 5, May 2002, pp. 512–522.

[26] Thunberg, J., Anisi, D. A., and Ögren, P., "A Comparative study of task assignment and path planning methods for multi-UGV missions," *Optimization and Cooperative Control Strategies*, edited by M. Hirsch, C. Commander, P. Pardalos, and R. Murphey, Lecture Notes in Control and Information Sciences, Springer Verlag, 2008.

[27] Garey, M. and Johnson, D., *Computers and Intractability: A Guide to the Theory of NP-Completeness*, WH Freeman & Co. New York, NY, USA, 1979.

[28] Ögren, P., Wirkander, S.-L., Stefansson, A., and Pelo, J., "Formulation and Solution of the UAV Paparazzi Problem," *AIAA Guidance, Navigation, and Control Conference and Exhibit; Keystone, CO, USA*, 21–24 August 2006.

Paper B

Connectivity Constrained Surveillance Using Multiple UGVs

Connectivity Constrained Surveillance using Multiple Unmanned Ground Vehicles

David A. Anisi, Petter Ögren and Xioaming Hu

Abstract

This paper addresses the problem of connectivity constrained surveillance of a given area with obstacles using a group of Unmanned Ground Vehicles (UGVs). The considered communication restrictions may involve both line-of-sight constraints and limited sensor range constraints. In this paper, the focus is on dynamic information graphs, $\mathcal{G}$, which are required to be kept *recurrently* connected at the time instants when the UGVs perform the surveillance mission, *i.e.*, when they gather and transmit sensor data. The main motivation for introducing this weaker notion of connectivity is security and surveillance applications where the sentry vehicles may have to split temporary in order to complete the given mission efficiently but are required to establish contact recurrently in order to exchange information or to make sure that all units are intact and well-functioning. From a theoretical standpoint, recurrent connectivity is shown to be sufficient for exponential convergence of consensus filters for the collected sensor data.

Keywords: Surveillance Mission, Unmanned Ground Vehicles, Connectivity Constraints, Combinatorial Optimization.

B.1 Introduction

THE main focus of this work lies on the so called Connectivity Constrained UGV Surveillance Problem (CUSP), which informally can be stated as follows: Given a set of surveillance UGVs and a user defined area to be covered, find waypoint-paths such that:

1. the area is completely surveyed,
2. the time for performing the search is minimized,
3. the induced information graph is kept *recurrently* connected at the time instants when the UGVs perform the surveillance mission.

In this formulation, the field of view of the onboard sensors are assumed to be occluded by the obstacles and limited by a maximal sensor range. Also, connectivity constraints of both line-of-sight and limited sensor range types are considered. A more formal statement of

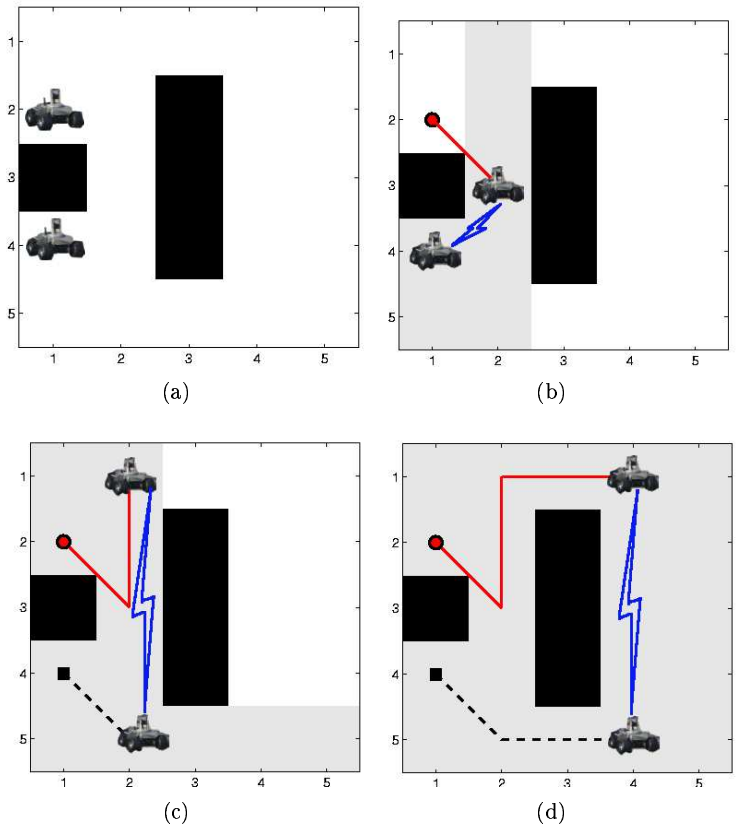


Figure B.1: Four snapshots during an illustrative CUSP mission. The two UGVs start off to the left, depicted in Figure B.1(a), and maintain connectivity *recurrently* at the three surveillance instances, Figures B.1(b)- B.1(d). The area already visited is shadowed and the mission is completed once the entire area is surveyed. Notice how the two UGVs are allowed to split temporarily in order to pass on different sides of the rightmost obstacle.

the CUSP is provided in Section B.3 and a typical CUSP mission can be seen in Figure B.1.

The main contribution of this paper is to formulate the CUSP, show that this optimization problem is $\mathcal{NP}$-hard and subsequently, present decomposition techniques that allow efficient algorithmic solutions. One of the most distinguishing features of the CUSP formulation is that it considers connectivity constraints of both line-of-sight and limited sensor range types in the presence of obstacles. In this context, we also introduce and utilize the notion of recurrent connectivity of a graph, which is a significantly more flexible connectivity constraint than, *e.g.*, the 1-hop connectivity constraints frequently considered in the literature (*cf.* Figure B.1). Finally, from a theoretical standpoint, we show that recurrent connectivity of the information graph is sufficient for convergence of consensus filters for the collected sensor data.

The remainder of this paper is organized as follows. Section B.1.1 provides a concise exposition of related work. Section B.2 presents some basic concepts that underlie the rest of the paper. The considered problem is formally defined in Section B.3 and the proposed algorithm for solving it efficiently can be found in Section B.4. Then, a set of theoretical

results are found in Section B.5. Simulations illustrating the approach are presented in Section B.6. Finally, the paper is concluded in Section B.7.

B.1.1 Related Work

A distinguishable number of publications have been devoted to consider different aspects of the communication maintenance problem. However, a great amount of such research consider *static* sensors and hence lies outside the sphere of interest of this work. In addition, the great majority of the papers that deal with controlled/mobile platforms have solely focused on the sensor range constraint and oftenly deal with obstacle-free environments, see, *e.g.*, [1–7]. To the best of our knowledge [8] and [9] are the only works that consider communication restrictions involving both limited sensor range and line-of-sight constraints in the presence of obstacles. In the following, both these papers will be discussed in detail.

The problem formulation in [8] is reminiscent of the one considered in [10], namely optimal path planning for a number of relay vehicles that have the mission of maintaining a chain of line-of-sight communication links that connect a given leader vehicle to the ground station. In accordance with their previous work in multi-vehicle path planning, Schouwenaars *et al.* use binary variables to capture connectivity between subsequent relay vehicles and end up solving a mixed integer linear program (MILP).

The problem considered in [9] is the most closely related one to our work. However, Esposito and Dunbar:

- aim at reaching a final configuration while our high-level objective is to complete the surveillance mission.
- utilize a potential function to synthesize a feasible movement direction for the vehicles.
- consider the case when a *fixed* information graph is given *a priori* and maintain these given links intact throughout the entire duration of the motion[1].

Dynamic or time-varying information graphs have also been studied in frameworks other than the one considered in this paper. In particular, results in various applications of information consensus, such as flocking ([11]), rendezvous ([12]) and formation stabilization ([13]), heavily rely on some notion of (joint) connectivity of the underlying information graph. Consequently, a fundamental research issue over the last few years has been the search for a less restrictive notion of connectivity which still renders the consensus control convergent (see [14–16]). Proposition B.3 relates these results to the notion of recurrent connectivity introduced in Section B.5.1.

B.2 Preliminaries

This paper assumes a previous knowledge of some basic concepts in combinatorial geometry, combinatorial optimization and graph theory. In particular the so called art gallery problems [17–19], multiple traveling salesman problems [20,21] and the information graph [13,21] are at principal focus and will be discussed below.

Combinatorial Geometry: Art Gallery Problem

In this section we closely follow Urrutia [19], but add the sensor range R into some of the definitions.

[1] This is equivalent with maintaining 1-hop connectivity of the information graph.

The areas to be searched in this paper are all going to be so called orthogonal polygons with holes (obstacles), thus we denote them A, for area. The orthogonality property is however only important for the maximal convex cover subproblem considered in Algorithm B.1. It should be noted that the rest of the proposed solution algorithm is not limited to the orthogonal case, but can handle any general polygon-with-holes type of environment.

Definition B.1 (Orthogonal polygons with holes). *A* polygon Q *in the plane is enclosed by an ordered sequence of points* $q_1, \ldots, q_n \in \mathbb{R}^2,\ n \geq 3$, *called vertices of* Q *together with the line segments* q_i *to* q_{i+1} *and* q_n *to* q_1, *called edges. In the following we assume that none of these edges intersect. A polygon is called* orthogonal *if adjacent edges are orthogonal. Given an orthogonal polygon* Q *and a set of* h *disjoint polygons* $Q_1, \ldots, Q_h$, *that are likewise orthogonal and contained in* Q, *we call the set* $A = Q \backslash \{Q_1 \cup ... \cup Q_h\}$ *an* orthogonal polygon with h holes.

Definition B.2 (Guardiance). *Given two points* p *and* q *in* A *we say that* p *is* visible *from* q *if the line segment joining* p *and* q *is contained in* A

$$\alpha p + (1 - \alpha) q \in A, \quad \forall \alpha \in [0\ \ 1]$$

and the distance between them is not greater than the sensor range,

$$\|p - q\| \leq R.$$

A set of points $H = \{h_1, \ldots, h_k\} \subset A$ guards A *if for all* $p \in A$ *there exists* $h_i \in H$ *such that* p *is visible from* h_i.

Definition B.3 (Maximal convex cover). *A* convex cover C *of* A *is a set of convex sets* $C = \{c_i\}$ *such that* $|c_i| \leq R$ *and* $A \subseteq \cup_i c_i$. *Here,* $|c_i| = \sup_{a,b \in c_i} dist(a, b)$ *denotes the diameter of the set* c_i. *We define a* maximal *convex cover of* A *to be a convex cover* $C = \{c_i\}$ *of* A, *such that for all* i, *there is no convex set* $s \subseteq A$ *such that* $|s| \leq R$ *and* $s \supset c_i$.

Definition B.4 (Visiting waypoint-path). *A* waypoint-path P *is an ordered set of points* $P = (p_1, \ldots, p_n)$. *Any convex cover* C *is said to be* visited *by the waypoint-path* P *if* $\forall c_i \in C\ \exists p_j \in P :\ p_j \in c_i$.

Combinatorial Optimization: Multiple Traveling Salesman Problem

Following [20], the Multiple Traveling Salesman Problem (MTSP) can be stated as follows: Given a set of cities and inter-city distances, let there be N salesmen located at given depot nodes. Then the MTSP consists of finding at most N tours, who all start and finish at given depots, such that each city is visited *exactly once* and the total cost of visiting all cities is minimized.

The MTSP can be considered as a relaxation of the closely related Vehicle Routing Problem (VRP), with the capacity restrictions removed. This means that all the formulations and solution methods proposed for the VRP are also valid and applicable to the MTSP by assigning sufficiently large capacities to the salesmen (vehicles). It is a well-known fact that the MTSP and the closely related VRP are $\mathcal{NP}$-hard, and thus represent optimization problems that are very hard to solve to optimality, see, *e.g.*, [20, 22].

Graph Theory: Information Graphs

A concept needed for making a formal statement of the CUSP is concerned with the information graph.

Definition B.5 (Information graph). *Let $\mathcal{V} = \{v_1, \ldots, v_N\}$ denote the vertex set representing the N UGVs. The* communication graph, *$\mathcal{G}_c(t) = (\mathcal{V}, \mathcal{E}_c(t))$, is induced by*

$$e_{ij} \in \mathcal{E}_c(t) \Leftrightarrow \|p_i(t) - p_j(t)\| \leq R,$$

where R denotes the limited sensor range[2] and $p_i(t) \in \mathbb{R}^2$ is the position of UGV i at time t. The sensing graph, *$\mathcal{G}_s(t) = (\mathcal{V}, \mathcal{E}_s(t))$, is induced by free line-of-sight, more precisely,*

$$e_{ij} \in \mathcal{E}_s(t) \Leftrightarrow \alpha p_i(t) + (1-\alpha)p_j(t) \in A, \quad \forall \alpha \in [0\ 1].$$

Finally, the information graph, *$\mathcal{G}(t) = (\mathcal{V}, \mathcal{E}(t))$, is defined as the union[3] of the sensing- and communication graph,* i.e., *$\mathcal{E} = \mathcal{E}_c \cup \mathcal{E}_s$.*

In this setting, the sensing graph captures the *passive* information flow among the UGVs gathered by the on-board sensors, while the communication graph represents *active* transmission of inter-vehicle information (*cf.* [13]). This distinction is important to make in various applications, *e.g.*, military missions where passive sensing is encouraged while active transmission, which might imply enemy exposure and thereby jeopardize the mission, are to be avoided.

B.3 Problem Formulation

In this section we first informally state the Connectivity Constrained UGV Surveillance Problem (CUSP). We then make a formal statement using concepts from Section B.2.

Informally, the problem we are studying is the following: Given a set of surveillance UGVs and a user defined area to be covered, find waypoint-paths such that every point of the area can be seen from a waypoint on a path, the induced information graph is kept *recurrently* connected at the time instants when the UGVs perform the surveillance mission, and the time for cooperatively executing the search in parallel is minimized.

This problem formulation is an extension of the one considered in Paper A where connectivity constraints where not taken into account.

Problem B.1 (Connectivity Constrained UGV Surveillance Problem (CUSP)). *Given N UGVs and an orthogonal polyhedral area A, find a set of N waypoint-paths $P = (P^1, \ldots, P^N)$ that solve the following optimization problem*

$$\begin{aligned} \min_{P} \quad & \sum_{j=1}^{n-1} \max_{i \in \mathbb{Z}_N^+} ||p_j^i - p_{(j+1)}^i|| \\ \textit{such that} \quad & \cup_i P^i \textit{ guards } A \\ & \mathcal{G}_{P(j)} \in \mathcal{C}, \ \ \forall j \end{aligned}$$

Here $P(j) = (p_j^1, \ldots, p_j^N)$ denotes the UGV positions at time instance j and $\mathcal{G}_{P(j)}$ is the induced *information graph when the UGVs are at $P(j)$. Further, $\mathcal{C}$ is the set of connected graphs on N vertices. In accordance with Problem A.1 (MTUSP), $P^i = (p_1^i, \ldots, p_n^i), i \in \mathbb{Z}_N^+ = \{1, \ldots, N\}$, and $\| \cdot \|$ denotes the shortest obstacle free distance. Here, the start depots, denoted by p_1^i, are given while the finish depots, p_n^i, may be either free or given.*

[2] *Assuming a uniform bound on the range of all sensors, R, is merely a matter of notational convenience. An extension to allow different sensor ranges is straightforward.*

[3] *Intersection can also be used without any conceptual implications.*

Remark B.1 (Re-connection instances)**.** It is important to note that in CUSP, the re-connection of the information graph does not occur at arbitrary time instances. In fact, $\mathcal{G}$ is required to re-connect at the *surveillance critical time instances*, *i.e.*, exactly when the overall mission is being collectively solved.

Remark B.2 (Interdependence)**.** In CUSP, the inter-vehicle dependence stems both from the imposed guarding and connectivity constraints, as well as the non-separable objective function.

Remark B.3 (Way-point following)**.** In this problem formulation, it is assumed that there exists a low level controller for each UGV capable of following the generated waypoint-path with bounded error.

Proposition B.1. *Problem B.1 (CUSP) is $\mathcal{NP}$-hard.*

Proof. Follows directly from the proof of Proposition A.1 since in the single UGV case, the connectivity constraint is trivially fulfilled. □

B.4 Proposed Solution

This section will propose a solution to the CUSP described in Section B.3. As a consequence of Proposition B.1, we can not hope to solve all CUSP instances to optimality in reasonable time but must adopt heuristic solution methods. A natural way to address the CUSP is to add the connectivity constraint to the third subproblem of the MTUSP solution (*i.e.*, Algorithm A.1). Due to Proposition B.4 however, this would turn the third, polynomial time, subproblem of finding the paths into an $\mathcal{NP}$-hard one, and having $\mathcal{NP}$-hard subproblems is in general not a good idea. In [23], we proposed and compared four different algorithms for solving Problem B.1 (CUSP). In this paper, we present an algorithm which is tailored for handling the hard connectivity constraints explicitly, and turned out to give a good trade-off between computational complexity and solution performance. The proposed algorithm builds upon the notion of a *connectivity-primitive*, see Figure B.2.

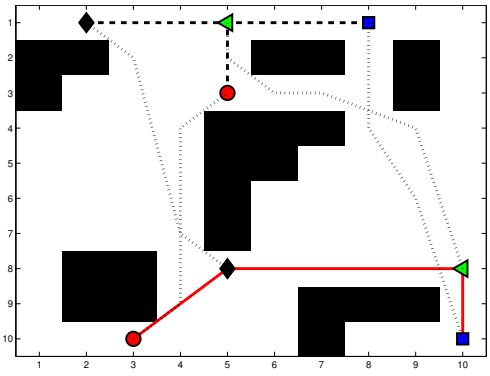


Figure B.2: Two connectivity-primitives (in solid and dashed) composed of four UGV positions each. The connectivity-primitives are created in step two of the algorithm. The dotted waypoint-paths indicate the result of step three, *i.e.*, how the UGVs should move from one primitive to another. The maximum of these path-lengths is then used as distance d_{ij} between the two primitives when solving the TSP in step four to find the order in which to visit the primitives.

Definition B.6 (Connectivity-primitive). *A connectivity-primitive* $s = \{p_1, \ldots, p_N\}$ *in* $\mathbb{R}^{2\times N}$*, is a collection of* N *UGV positions* $p_i \in \mathbb{R}^2$*, that induce an information graph that is* connected.

As depicted in Figure B.3, the suggested algorithm for solving CUSP consists of four subproblems. We first state the algorithm and then discuss the four steps in detail.

Algorithm B.1 (Proposed solution for CUSP). The algorithm consists of the following four steps:

1. Create a maximal convex cover $C = \{c_1, \ldots, c_M\}$ of area A in accordance with Algorithm A.2.
2. Create a set of connectivity-primitives, $S = \{s_1, s_2, ..., s_p\}$, that completely covers the area, *i.e.*, the convex cover C is visited by the points in $\cup_i s_i$.
3. Solve an assignment problem for each pair (s_i, s_j) of connectivity-primitives in S, where the cost of an assignment of $p_k \in s_i$ to $p_l \in s_j$ is equal to the shortest obstacle free path from p_k to p_l. Let d_{ij} be the maxima of these path-lengths in the optimal assignment.
4. Solve a Traveling Salesmen Problem (TSP) having the p connectivity-primitives in S as cities and the total path lengths d_{ij} as inter city distances. While the maximal number of iterations have not been reached, goto 2.

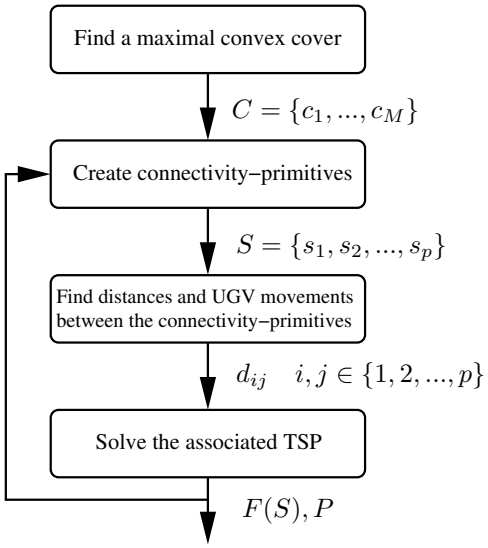


Figure B.3: Schematic overview of the suggested algorithm for solving CUSP.

The first subproblem is the same as in Algorithm A.1, finding a maximal convex cover, and is solved by Algorithm A.2.

In the second subproblem, a set of connectivity-primitives, $S = \{s_1, s_2, ..., s_p\}$, is created in such a way that the entire area is covered, *i.e.*, there is at least one connectivity-primitive node in every set of the cover, C. As will be explained in Proposition B.4, it is hard to find a connectivity-primitive visiting a set of gives sets. Thus we propose to iteratively pick connected positions in the yet uncovered part of A and then see what additional sets they cover. Let $V \subseteq C$ be the set of convex sets that have been visited by a connectivity-primitive already. In order to keep the number of connectivity-primitives low, it is preferable that the

next connectivity-primitive has some nodes in the set $C \setminus V$. Pick one that can be connected and add it to the primitive. This procedure is repeated until $C \setminus V = \varnothing$ which means that by visiting the connectivity-primitives, $S = \{s_1, s_2, ..., s_p\}$, the entire area is surveyed.

In order to decide in which order to visit the primitives (subproblem four) we need some notion of how far apart they are. In the third subproblem we find such distances d_{ij} by solving an assignment problem for each pair of connectivity-primitives. One such assignment is depicted in Figure B.2. We let the distance be the maximal length any UGVs need to travel to get from primitive i to primitive j. The $\frac{p(p-1)}{2}$ assignment problems as well as the underlying shortest path problem can be solved by polynomial time algorithms [21].

The distances between the connectivity-primitives, denoted d_{ij} in Figure B.3, are then passed down to the fourth and last subproblem which is to determine the order in which to visit the connectivity-primitives as well as the objective function, $F(S)$.This is in fact a TSP where connectivity-primitives play the role of cities and d_{ij} the corresponding distances. This also explains our earlier concern to keep the number of connectivity-primitives low in subproblem two. Various heuristic solution methods exist for solving TSPs [20]. In our particular case, a Simulated Annealing algorithm has been adopted [24]. This algorithm generates a feasible solution to the CUSP and the algorithm may stop. However, in order to get closer to an optimal solution, it may be beneficial to re-do the process for a particular number of iterations by repeating the last three subproblems.

Proposition B.2. *Algorithm B.1 produces a feasible solution to Problem B.1 (CUSP).*

Proof. This is true since all connectivity-primitives are visited and they are designed to both respect the connectivity constraints and provide a complete coverage of area, A. □

B.5 Theoretical Analysis

The theoretical results presented in this section includes a theorem relating the re-occurring connectivity of the CUSP to consensus filters, but also, $\mathcal{NP}$-completeness of the problem of determining existence of a connectivity-primitive that visits a set of given sets.

B.5.1 Recurrent Connectivity and Consensus Filters

In this section, the precise definition of the notion of recurrent connectivity will be given. Subsequently, a result concerning the relationship between this weaker form of connectivity and convergence of so called "Laplacian consensus filters" [14, 25] is presented. But before embarking, let us mention the motivation for introducing the notion of recurrent connectivity, why it is important and how it can be used in practical surveillance missions.

Many applications do not require connectivity of the information graph at all time instances. As a matter of fact, typical surveillance missions are solved more effectively if the UGVs are allowed to split up temporary. Seeking through the area in Figure B.1 serves as an illustrative example of such a scenario. This motivates the introduction of the notion of recurrent connectivity.

The main focus of this paper is on planning waypoint-paths for UGVs. Let us now assume that along these waypoint-paths, each UGV measures some quantity $y_i \in \mathbb{R}$ using the on-board sensor. In accordance with Definition B.5, the position of the UGVs, induce an information graph, $\mathcal{G}(t)$. The sensor measurements can therefore be propagated through the links of $\mathcal{G}(t)$ in order to share information among the UGVs. In other words, $\mathcal{G}(t)$ provides the *infrastructure* for information sharing. For practical purposes, we would like the robots to reach consensus regarding the measurements y_i. It is therefore important to

study the issue of convergence of information filters for the collected sensor data when the underlying graph is *recurrently* connected. This is the essence of Proposition B.3.

Definition B.7 (Recurrent connectivity)**.** *A graph, $\mathcal{G}$, is said to be* recurrently connected *with a dwell-time, $\Delta > 0$, if for all $t \in \mathbb{R}^+$ there exists $T(t) \geq t + \Delta$ such that*

$$\mathcal{G}(\tau) \in \mathcal{C}, \quad \forall \tau \in [T - \Delta \quad T].$$

Here, $\mathcal{C}$ denotes the set of connected graphs.

This definition implies that one is always able to find a time-interval of length Δ, during which the graph is connected.

Remark B.4. In a discrete time setting, the sampling time provides a natural selection of dwell-time (*cf.* [14, 15]).

As mentioned previously, along the waypoint-paths, each UGV measures some quantity $y_i \in \mathbb{R}$ at time t_0 that is then communicated to the others through the links of the induced information graph, $\mathcal{G}(t)$, and used as input to a consensus filter

$$\dot{x}_i = \sum_{j \in N_i} a_{ij}[x_j(t) - x_i(t)], \quad x_i(t_0) = y_i, \tag{B.1}$$

where $N_i = \{j \in \mathbb{Z}_N^+ : \varepsilon_{ij} \in \mathcal{E}(t)\}$ are the neighbors of UGV i in $\mathcal{G}(t)$. Let $x = (x_1, \cdots, x_N)^T$, then using standard notations in graph theory, (B.1) can be rewritten as

$$\dot{x} = -Lx, \quad x(t_0) = y \tag{B.2}$$

where L denotes the weighted Laplacian matrix.

It is well known (see, *e.g.*, [14, 25]) that if the graph is (jointly) connected, then a consensus can be reached in (B.2). The next proposition shows that a consensus can also be reached for (B.2) if the graph is recurrently connected.

Proposition B.3 (Consensus Reaching)**.** *Suppose the information graph is recurrently connected with a dwell time Δ and that*

$$\hat{T} \triangleq \sup_{t \in \mathbb{R}^+} T(t) - t < \infty.$$

Then the frequently adopted "Laplacian consensus filter" (B.2) will exponentially *converge to a consensus in the agreement subspace.*

Proof. Starting at the initial time instance, t_0, a sequence of half-open time intervals, $\{T_i\}$, with $T_i = [t_i^L \quad t_i^R)$ is constructed as follows:

1. Initially, set $t_1^L = t_0$.
2. The information graph $\mathcal{G}$ being recurrently connected, ensures the existence of $T(t_1^L)$ such that $\mathcal{G}(\tau) \in \mathcal{C}, \forall \tau \in [T - \Delta \quad T]$. Set then $t_1^R = T(t_1^L)$. Thus, the first time interval, $T_1 = [t_1^L \quad t_1^R)$, is constructed.
3. Having the first k terms of the sequence of time intervals at hand, $T_1, \ldots, T_k$, the $(k+1)^{\text{th}}$ time interval is set to

   $$T_{k+1} = [t_{k+1}^L \quad t_{k+1}^R) = [t_k^R \quad T(t_k^R)),$$

   where again, $T(t_k^R)$ exists due to the recurrent connectivity assumption.

The terms of the sequence of time intervals $\{T_i\}$ constructed this way, fulfill the following properties[4]:

- $|T_i| > 0, \forall i$ (non-empty)
- $T_i \cup T_{i+1} = [t_i^L \quad t_{i+1}^R)$ (contiguous)
- $T_i \cap T_j = \emptyset$ (non-overlapping)
- $|T_i| \leq \hat{T}\ \forall i$ (bounded)

The first property follows since $\Delta > 0$. The following two are consequences of having $t_{k+1}^L = t_k^R$. In addition to this,

$$\mathcal{G}_i = \cup_{t \in T_i}\ \mathcal{G}(t)$$

is *jointly connected* for all i. Thus, the sequence $\{T_i\}$ fulfills the hypothesis of Theorem 2 in [14] which implies exponential convergence of any consensus state to the agreement subspace. □

Remark B.5. The assumption on boundedness of the time-intervals implied by,

$$\hat{T} \triangleq \sup_{t \in \mathbb{R}^+} T(t) - t < \infty,$$

is not required for *asymptotic* convergence to the agreement subspace, see [15].

B.5.2 Finding Connectivity-Primitives

Our next result gives more insight into the choice of solution for the CUSP. As will be shown, given N sets for the UGVs to visit, determining whether these sets contain a connectivity-primitive or not is an $\mathcal{NP}$-complete problem. This explains why a straight forward extension of the MTUSP algorithm does not perform well on the CUSP problem but also the proposed heuristic algorithm for creating the connectivity-primitives covering A.

Assume that the N UGVs are to visit sets $c_1, \cdots, c_N$. Let further V_k denote the nodes in A associated with the set c_k, $k \in \mathbb{Z}_N^+$. For the generation of UGV positions that visits all the N sets, we would like to solve the following problem:

Problem B.2. *Find a connectivity-primitive, $s = \{p_1, \ldots, p_N\}$, such that $p_i \in V_i$, $i \in \mathbb{Z}_N^+$,* i.e.*, each UGV is at a node inside the appropriate convex set.*

Problem B.2 is hence concerned with selecting one of the nodes in each convex set (hence-forth referred to as the *representative* of that set) that induce a connected graph.

Proposition B.4. *Problem B.2 is $\mathcal{NP}$-complete.*

Proof. The proof will built upon polynomial reduction of the satisfiability problem (SAT) [21]. From an *arbitrary* SAT instance with N clauses, $C_1, \ldots, C_N$, involving boolean literals, $l_1, \ldots, l_q$, we construct a special instance of Problem B.2 as follows (*cf.*, Figure B.4).

1. Let there be N sets containing only one node. These will represent the clauses, $C_1, \ldots, C_N$ and will be referred to as *clausal sets.*
2. Let there be q additional *literal sets* having two nodes each. The first node in the i^{th} literal set, denoted l_i^t, corresponds to the literal l_i having the *true* value, while the second node, denoted l_i^f, corresponds to the literal l_i having the *false* value.

[4]Here, $|T_i|$ denotes the interval length, *i.e.*,

$$|T_i| = \sup_{a,b \in T_i} \text{dist}(a,b) = t_i^R - t_i^L.$$

3. Connect every clausal set to the node in the literal set that is dictated by its expression (*cf.* Example B.1).
4. Finally, introduce one last additional set that connects to all nodes in the literal sets.

Now, choosing a representative for the clausal sets and the additional set is trivial since they only contain one node. Choosing representative for the literal sets however, corresponds to determining the *truth/false* assignments of the q literals. If we would have been able to choose representatives among these $N+q+1$ sets that induce a connected graph in polynomial time, we would have solved this *arbitrary* instance of the SAT problem. Since SAT is a well-known $\mathcal{NP}$-complete problem, we conclude that Problem B.2 is $\mathcal{NP}$-complete as well. □

Example B.1. Solving the following SAT instance

$$\begin{aligned} \text{SAT} &= C_1 \wedge C_2 \wedge \cdots \wedge C_N \\ C_1 &= (l_1 \vee l_2) \\ C_2 &= (\bar{l}_2 \vee l_q) \\ \vdots &\quad \vdots \\ C_N &= (l_1 \vee \bar{l}_2 \vee \bar{l}_q) \end{aligned}$$

is equivalent with finding representatives among the $N+q+1$ sets in Figure B.4 that induce a connected graph. Here, $\bar{l}_i$ denotes the negation of l_i.

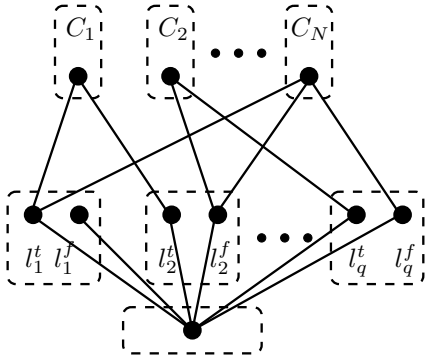


Figure B.4: Determining which of the two nodes to visit in each *literal set* corresponds to determining the *true/false* value of the literals $l_1, \ldots, l_q$.

Having discussed the algorithm and its properties in detail, it is now time to run some simulation examples.

B.6 Simulations

In this section, a small selection of the simulations made is presented. The objective is to highlight some of the key characteristics of the proposed solution methods. Throughout this section, the search area, A, is chosen to be all of the obstacle free space, *i.e.*, the white area in all figures.

In Figure B.5, the starting positions of the UGVs are chosen randomly while the final positions are optimized by Algorithm B.1. The most important aspect to notice is that

the three UGVs are not restricted to pass on the same "side" of the obstacles but are nevertheless *recurrently* connected at the five surveillance instances, Figure B.5(b)- B.5(f). Also, notice that the randomly selected initial positions in Figure B.5(a) do not necessarily induce a connected information graph and that the area is completely surveyed.

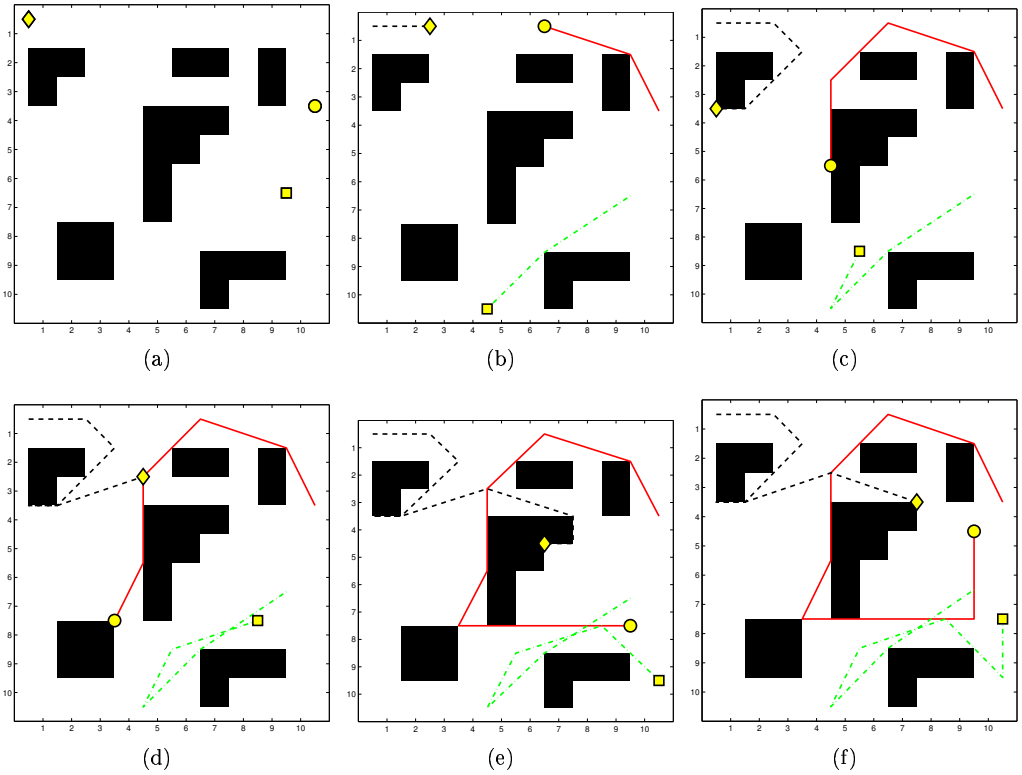


Figure B.5: An example CUSP simulation. The most important aspect to notice is that the UGVs are not restricted to pass on the same "side" of the obstacles but are nevertheless *recurrently* connected at the five surveillance instances in Figure B.5(b)– B.5(f).

Another CUSP simulation can be seen in Figure B.6, where the area representation is taken as a random matrix with obstacle density $\rho = 0.3$.

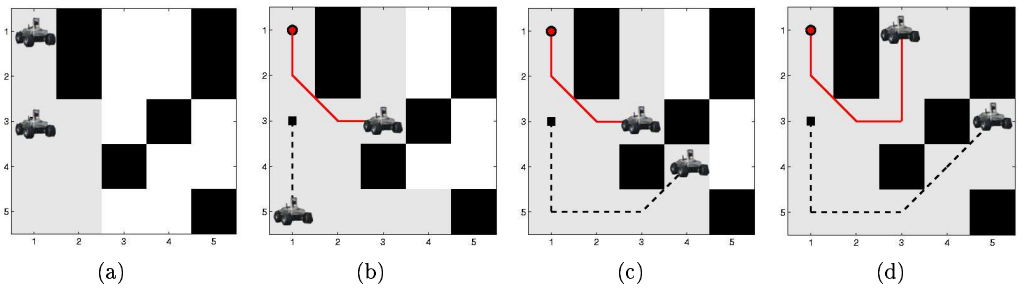


Figure B.6: Another CUSP simulation.

Figure B.7 illustrates connectivity constrained surveillance of the so called "Manhattan grid". In this example, the cooperative nature of the solution becomes even more apparent. The two UGVs are dropped off at the upper left corner in A and move downwards in order to fulfill their common goal of complete coverage. In essence, the UGV whose waypoint-path has been depicted in dashed/black surveys the vertically aligned streets while the other one (solid/red) covers the others. Notice how the inter-vehicle connectivity is maintained cooperatively as the UGVs timely pass the horizontally aligned streets. Also in this example, the fact that the final solution is merely locally optimal is apparent from the dashed/red waypoint-path, which could rather be a straight line segment.

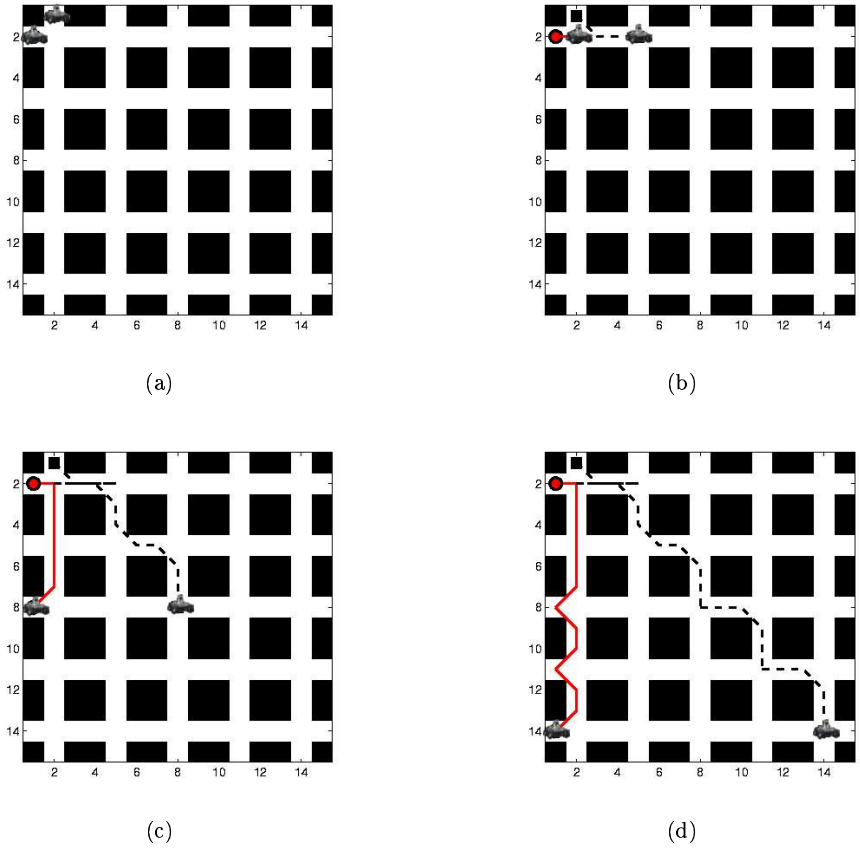


Figure B.7: Complete surveillance of the so called "Manhattan grid". Notice how the inter-vehicle connectivity is maintained cooperatively as they timely pass the horizontally aligned streets.

B.7 Concluding Remarks

An important problem in cooperative UGV surveillance is to make sure that the sensor data can reach all team members but also be transmitted back to the operator. In this paper, we presented a cooperative path and task planning algorithm that made sure that the whole

surveillance area was covered, and at the same time the entire UGV group was *recurrently connected* in order to exchange information and upload it to the operator.

The main motivation for introducing this weaker notion of connectivity is security and surveillance applications where the vehicles may have to split temporary in order to complete the given mission efficiently but are required to establish contact recurrently in order to exchange information and/or to make sure that all units are intact and well-functioning. From a theoretical standpoint, recurrent connectivity of the information graph is shown to be sufficient for convergence of Laplacian consensus filters for the collected sensor data. It should be noted that this work considers connectivity constraints of both line-of-sight and limited sensor range types in the presence of obstacles.

B.8 References

[1] Spanos, D. P. and Murray, R. M., "Robust connectivity of networked vehicles," *Proc. of the* 43^{rd} *IEEE Conference on Decision and Control (CDC)*, Vol. 3, Paradise Island, Bahamas, Dec. 2004.

[2] Kim, Y. and Mesbahi, M., "On maximizing the second smallest eigenvalue of a state-dependent graph laplacian," *IEEE transactions on automatic control*, Vol. 51, No. 1, 2006, pp. 116–120.

[3] Zavlanos, M. M. and Pappas, G. J., "Controlling Connectivity of Dynamic Graphs," *Proc. of the* 44^{th} *IEEE Conference on Decision and Control (CDC)*, Seville, Spain, Dec. 2005, pp. 6388–6393.

[4] Zavlanos, M. M. and Pappas, G. J., "Potential Fields for Maintaining Connectivity of Mobile Networks," *IEEE Transactions on Robotics*, Vol. 23, No. 4, Aug. 2007, pp. 812–816.

[5] De Gennaro, M. C. and Jadbabaie, A., "Decentralized Control of Connectivity for Multi-Agent Systems," *Proc. of the* 45^{th} *IEEE Conference on Decision and Control (CDC)*, San Diego, CA, Dec. 2006, pp. 3628–3633.

[6] Ji, M. and Egerstedt, M., "Distributed Coordination Control of Multiagent Systems While Preserving Connectedness," *IEEE Transactions on Robotics*, Vol. 23, No. 4, Aug. 2007, pp. 693–703.

[7] Srivastava, K. and Spong, M. W., "Multi-Agent Coordination under Connectivity Constraints," *Proc. of the American Control Conference (ACC)*, Seattle, WA, USA, Jun. 2008, pp. 2648–2653.

[8] Schouwenaars, T., Feron, E., and How, J., "Multi-vehicle path planning for non-line of sight communication," *Proc. of the American Control Conference (ACC)*, Minneapolis, Minnesota, USA, Jun. 2006, pp. 5757–5762.

[9] Esposito, J. M. and Dunbar, T. W., "Maintaining wireless connectivity constraints for swarms in the presence of obstacles," *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, Orlando, Florida, May 2006, pp. 946–951.

[10] Nguyen, H., Pezeshkian, N., Raymond, M., Gupta, A., and Spector, J., "Autonomous communication relays for tactical robots," *Proc. of the* 11^{th} *International Conference on Advanced Robotics (ICAR)*, Coimbra, Portugal, 2003, pp. 35–40.

[11] Olfati-Saber, R., "Flocking for multi-agent dynamic systems: algorithms and theory," *IEEE Transactions on Automatic Control*, Vol. 51, No. 3, Mar. 2006, pp. 401–420.

[12] Lin, J., Morse, A., and Anderson, B., "The multi-agent rendezvous problem," *Proc. of the* 42^{nd} *IEEE Conference on Decision and Control (CDC)*, Dec. 2003, pp. 1508–1513.

[13] Fax, J. and Murray, R., "Graph Laplacians and stabilization of vehicle formations," *Proc. of the* 15^{th} *IFAC World Congress*, 2002, pp. 283–288.

[14] 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.

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

[16] 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.

[17] Bjorling-Sachs, I. and Souvaine, D., "A Tight Bound for Guarding Polygons With Holes," Tech. rep., Report LCSR-TR-165, Lab. Comput. Sci. Res., Rutgers Univ., New Brunswick, NJ, 1991.

[18] Hoffmann, F., Kaufmann, M., and Kriegel, K., "The art gallery theorem for polygons with holes," *32nd Annual Symposium on Foundations of Computer Science (San Juan, PR, 1991)*, IEEE Comput. Soc. Press, Los Alamitos, CA, 1991, pp. 39–48.

[19] Urrutia, J., "Art Gallery and Illumination Problems," *Handbook of computational geometry*, edited by J.-R. Sack and J. Urrutia, North-Holland Publishing Co., 2000, pp. 973–1027.

[20] Bektas, T., "The multiple traveling salesman problem: an overview of formulations and solution procedures," *Omega*, Vol. 34, No. 3, Jun. 2006, pp. 209–219.

[21] Papadimitriou, C. H. and Steiglitz, K., *Combinatorial optimization: algorithms and complexity*, Dover Publications Inc., Mineola, NY, 1998.

[22] Laporte, G., "The vehicle routing problem: An overview of exact and approximate algorithms," *European Journal of Operational Research*, Vol. 59, No. 3, Jun. 1992, pp. 345–358.

[23] Anisi, D. A., Lindskog, T., and Ögren, P., "Algorithms for the Connectivity Constrained Unmanned Ground Vehicle Surveillance Problem," *Proc. of the European Control Conference (ECC)*, Budapest, Hungary, Aug. 2009.

[24] Kirkpatrick, S., Gelatt, C., and Vecchi, M., "Optimization by Simulated Annealing," *Science*, Vol. 220, No. 4598, 1983, pp. 671–680.

[25] Tsitsiklis, J., Bertsekas, D., and Athans, M., "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *IEEE Trans. Automat. Control*, Vol. 31, No. 9, 1986, pp. 803–812.

Paper C

Online Trajectory Planning for Aerial Vehicles: Safety and Task Completion

Online Trajectory Planning for Aerial Vehicles: Safety and Task Completion

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

Abstract

This work addresses the problem of designing a real time high performance trajectory planner for aerial vehicles. The objective is to use information about terrain and enemy threats to fly low and avoid radar exposure on the way to a given target. The proposed algorithm builds on the well known approach of Receding Horizon Control (RHC) combined with a sporadically updated terminal cost that captures the global characteristics of the environment and mission objectives. However, since the terminal cost is most often calculated from a graph representation of the environment, it might lead to trajectories that turn out to be dynamically infeasible in the future. Thus, neither safety nor task completion can be guaranteed *a priori*. Using a novel safety maneuver combined with a task completing trajectory, and under an assumption on the maximal terrain inclination, the main contribution of this paper is to prove safety as well as task completion for the proposed algorithm. The safety maneuver is incorporated in the short term optimization, which is performed using Nonlinear Programming (NLP). Some key characteristics of the trajectory planner are highlighted through simulations.

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

C.1 Introduction

ON-LINE trajectory planning for an aerial vehicle subject to simultaneous kinematic and dynamic constraints, is the main topic of this paper. Given current information about the target, terrain and positions of enemy threats, the generated trajectory should use the terrain to reduce exposure to enemy radar by flying low, while at the same time keeping control efforts, as well as total time of flight, small. Furthermore, we would also like to be able to formally verify that the vehicle will in fact reach the target without crashing into the terrain. Often, the trajectory planning problem is formulated as an Optimal Control Problem (OCP). In our case however, an underlying assumption is that due to imperfect information, the terrain elevation, as well as the location of the target 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 "good enough" solutions provided by fast computational algorithms for iteratively solving the trajectory optimization problem.

In the field of trajectory planning and control, Receding Horizon Control (RHC) or Model Predictive Control (MPC), is a well known tool to achieve computationally efficient, "good enough" solutions to many unmanned vehicle control problems [4–11]. In RHC, the doubtful viability of long term optimization under uncertain conditions is adhered, so that instead of solving the OCP on the full time 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 t_c forward in time, the optimization step is iteratively performed. This closes the loop and obtains a certain robustness against modeling errors or disturbances. Unfortunately, it is known that in the absence of particular precautions, closed-loop stability[1] cannot be assured. Hence, an important issue with RHC is to make sure that the greedy, short term optimization does not lead to long term problems. In the vehicle control domain, this often boils down to two things: not getting into situations where a collision is unavoidable, and making sure that the destination is actually reached. To this end, it must be noted that although formulated as an OCP, finding a provably collision free trajectory 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, which in our particular case will be to generate collision free trajectories that lead us to the target. Hence-forth, a collision free trajectory is called *safe* and a trajectory reaching the target set will be referred to as a *task completing* trajectory. As will be shown in Section C.4, the proposed algorithm has provable safety properties, as well as a guaranteed finite time task completion. This is the main contribution of this paper which merges our earlier work on online trajectory optimization for aerial vehicles [11, 12].

Regarding safety concerns, collision avoidance for ground vehicles or helicopters can be achieved by making sure that every planned trajectory ends in a standstill [4]. Similarly, for fixed wing aerial vehicles, a guaranteed obstacle free circular loitering pattern ensures safety [5,8]. This is achieved 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. Kuwata and How [9,10] have considered safe RHC of autonomous vehicles in a 2D setting which rely on visibility graphs for environmental representation. They 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. However, task completion is not guaranteed in any of these papers.

Task completion has been previously considered by Richards and How [6, 7]. 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. Task completion is then guaranteed by imposing a hard terminal equality constraint on the target state which restricts the solution candidates to those that end up in the target set at the end of the planning horizon. This is a computationally demanding constraint that beside the introducement of binary variables, requires needlessly long planning horizon. In addition, early termination of the optimization routine may cause violation of the equality constraint and consequently

[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, aiming at controlling the vehicle into a target set which not necessarily contains any equilibrium points or is control invariant.

jeopardize the task completion objective. The alternative solution proposed in this paper utilizes monotonic decay of the terminal cost to establish task completion.This is due to a conditional plan changing strategy, where, starting from a feasible solution, a new plan for the remaining part of the mission is only accepted if it gives an incremental decrease to the terminal cost. This decouples the length of the planning horizon from task completion and thus allows us to choose the planning horizon only taking optimality requirements and computational resources into account.

Most of the papers above consider planar problems, while our formulation is in 3D. This setting makes the problem both more realistic and easier to address, since there is always an obstacle free sky above the terrain. As a direct consequence of this, our work differs from the mentioned papers by the fact that in our case, safety and task completion are intimately connected. This is due to an elaborate choice of safety maneuver which is augmented by a closed form task completing trajectory, that is collision free by design. By iteratively replacing old safe plans with new safe plans, there is always a safe plan available for execution if some step of the update procedure should fail.

The high-level framework utilized in this work for trajectory planning in the three dimensional space, is reminiscent of the one presented in [13]. 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 [13], the term "off-line" is rather to be interpreted as, at a much slower sampling rate than the trajectory planning 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 online planner through an updated terminal cost, as discussed in [14].

In what follows, the considered trajectory optimization problem is presented in Section C.2. Section C.3 then describes the proposed solution in detail. The properties of safety and task completion are proved in Section C.4, followed by simulation examples in Section C.5. Finally, this paper is concluded in Section C.6.

C.2 Problem Formulation

In this section we will state the problem in terms of vehicle model and control objectives.

C.2.1 Vehicle Model

Let the aerial vehicle model be given in discrete time by

$$\begin{aligned} p_{i+1} &= p_i + h\, v_i \\ v_{i+1} &= v_i + h\, a_i \\ \|v_i\|_\infty &\leq v_{\max} \\ \|a_i\|_\infty &\leq a_{\max} \\ d(p_i) &\leq 0 \\ (0,0,1)\, v_i^T &\leq ||v_i|| \sin\bar{\gamma}, \end{aligned} \tag{C.1}$$

that is, a discrete time, double integrator with time step h and upper bounds on magnitude of speed, $v_i \in \mathbb{R}^3$, and acceleration $a_i \in \mathbb{R}^3$. There is also a terrain collision constraint on

position, $p_i \in \mathbb{R}^3$, as well as an upper bound $\bar{\gamma} \in (0, \pi/2]$ on the flight path angle. The collision constraint is

$$d(p_i) \triangleq H(x_i, y_i) + \Delta - z_i \leq 0,$$

where $H(x_i, y_i)$ denotes the altitude of the terrain at the point $p_i = (x_i, y_i, z_i)$ and $\Delta > 0$ denotes a minimum clearance distance, set by the operator. It should be noted, that the theory below admits considerably more general dynamics and control constraints; as long as the NLP below terminates fast enough, any model can be used.

In order to state safety properties of the proposed trajectory planner, we make the following assumption about the terrain.

Assumption C.1. The maximal terrain inclination, α, is smaller than the maximum flight path angle $\bar{\gamma}$.

In instances where this assumption does not hold at isolated spots, a virtual "inclination-smoothed" terrain above the real one can be calculated and used instead of the original, thus fulfilling the assumption.

C.2.2 Control Objectives

Designing the trajectory planner, we would like to meet the following objectives.

- Operate the vehicle safely (*i.e.*, within the state and control limits)
- Arrive at target position
- Compute trajectories in real time
- Allow for information updates
- Use small control effort
- Achieve low threat and radar exposure
- Achieve a short time of flight

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. The objective function might therefore be of the following form

$$\sum_i h\Big(\|a_i\|_2^2 + bg(p_i) + c\Big), \tag{C.2}$$

where p_i, a_i, h are defined above, $b, c > 0$ are scalar weights, g represents a measure of threat and radar exposure and the summation stretches over the whole mission.

C.2.3 Problem Statement

We formulate, somewhat loosely, the overall trajectory planning problem in the following way.

Problem C.1. *Iteratively and efficiently choose controls a_i such that the model (C.1) starting from p_0 safely reaches the target, p_f, while approximately minimizing the objective function (C.2).*

The term approximately is added, since solving the problem to optimality turns out to be computationally intractable for problems with longer horizons. However, if we are willing to settle for a good, but not necessarily optimal solution in terms of the objective function (C.2), the problem becomes tractable.

C.3 Proposed Solution

In this section we will propose a solution to Problem C.1 described above. The solution will be a Receding Horizon Control (RHC) scheme using Nonlinear Programming (NLP) for the short term planning. The central idea behind the proposed solution is to use a conditional plan changing strategy, where, starting from a feasible solution, a new plan for the remaining part of the mission is only accepted if it gives an incremental decrease to the terminal cost. The key problem is then to retain safety and finite time task completion when the plan is changed. This problem is solved by including a safety maneuver, which guarantees the existence of a safe task completing trajectory. The safety maneuver is incorporated into the short term planning NLP problem. We will furthermore discuss the choice of terminal cost function for the long term planning, as well as how we can make sure that the vehicle actually arrives at the target without crashing into the terrain.

C.3.1 Solution Outline and Receding Horizon Control

As discussed earlier, RHC has proved to be a powerful tool to achieve good performance in a computationally tractable way. The main idea in RHC is to divide a planning problem into a short horizon and a long horizon part. Over the short horizon a detailed plan is calculated and over the long horizon only a coarse plan, or no plan at all is needed. The short term plan is then iteratively updated as time evolves, making the horizon of the plan *recede*, always extending a fixed amount of time into the future.

The way we apply RHC to this problem is illustrated in Figure C.1. The figure depicts the aerial vehicle and corresponding plan at two different time instants. At the first instant a short term plan (solid and dotted lines) is calculated using NLP. This trajectory includes a safety maneuver (dotted) and is augmented with a task completing trajectory (dashed). The existence of the safety maneuver at the end of the short term plan is a *constraint* of the NLP. At the end of the safety maneuver the trajectory is directed towards the target and has the climb angle α, which by definition is equal to the maximal terrain inclination.

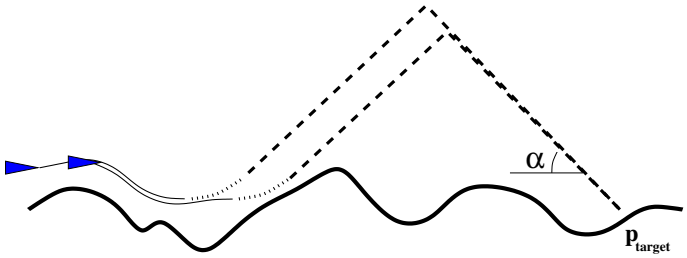


Figure C.1: Two consecutive plans. Note the short term plan (solid and dotted), safety maneuver (dotted), and task completing trajectory (dashed), reaching the target, p_f, on the ground (thick solid).

This fact enables us to take the final part of the trajectory as a steady climb and a steady descent towards the target at α degrees inclination, *collision free* by design. That a collision free trajectory is obtained also after re-planning, *i.e.*, that safety is retained, follows from the observation that if the new planning for some reason fails, the old plan is still valid and can be executed all the way to the target [4].

Having a plan, the vehicle now proceeds to execute the first part of the plan over an execution horizon T_e, which must be smaller than the short term planning horizon, $T_p = Nh$.

Figure C.1 also depicts the situation after T_e, when the NLP planning is performed once more and a new plan is constructed.

Remark C.1. It is useful, but not necessary, that T_p, T_e are chosen so that the safety maneuver is shorter in duration than $T_p - T_e$ since then the re-planning commences before the vehicle has started the safety maneuver. In fact, from a performance perspective, it is in general desirable that the vehicle actually *never* has to execute the safety maneuver.

In order to apply the above solution strategy one has to find a way to negotiate the instant 2α turn at the top of the task completing trajectory in some way, so that the constraints in model (C.1) are met. There are several ways of doing this, as discussed below, but we shall make the standing assumption that the issue has been resolved for the first planning instance.

Having outlined the principle behind our formulation of the trajectory planning problem as an RHC problem, we now proceed to a somewhat more detailed look at its various components.

C.3.2 Short term planning and Nonlinear Programming

The NLP alluded to above, for the solution of the short term planning subproblem, can be described as follows. At the time of planning, let the vehicle be in state p_c, v_c. Let all planning variables have two subscripts where the first represent the index of the plan, and the second represents the time scale on which the vehicle dynamics in (C.1) are defined, *i.e.*, $p_{k,i}$ is the planned position of plan k at $h \cdot i$ time units after the plan was initiated.

Definition C.1. *By a short term plan, we mean the best known solution to the following NLP.*

$$
\begin{aligned}
\underset{a_k}{\text{minimize}} \quad & \sum_{i=0}^{N} h\Big(\|a_i\|_2^2 + bg(p_i) + c\Big) + \Psi(p_{k,N}) && && \text{(C.3)}\\
\textit{s.t.} \quad & p_{k,i+1} = p_{k,i} + h\, v_{k,i} && i = 0, \cdots, N-1\\
& v_{k,i+1} = v_{k,i} + h\, a_{k,i} && i = 0, \cdots, N-1\\
& d(p_{k,i}) \le 0 && i = 0, \cdots, N\\
& \|v_{k,i}\|_\infty \le v_{\max} && i = 0, \cdots, N\\
& \|a_{k,i}\|_\infty \le a_{\max} && i = 0, \cdots, N\\
& (0,0,1)\, v_{k,i}^T \le \|v_{k,i}\| \sin\bar{\gamma} && i = 0, \cdots, N\\
& p_{k,0} = p_c, \quad v_{k,0} = v_c\\
& \Psi(p_{k,N}) \le \Psi(p_{k-1,N}) - \varepsilon\\
& v_{k,N} = v_{\max}(\cos\phi\cos\alpha,\ \sin\phi\cos\alpha,\ \sin\alpha)\\
& \phi = \textit{arctan2}\big((0,1,0)\,(p_f - p_{k,N})^T, (1,0,0)\,(p_f - p_{k,N})^T\big)
\end{aligned}
$$

where k is plan index, $N = T_p/h$, and the terminal cost, $\Psi : \mathbb{R}^3 \to \mathbb{R}^+$, and cost decrement margin, $\varepsilon > 0$, will be defined below. The final two constraints make the last part of the trajectory climb in the direction of the target. All other variables are defined as their counterparts, without index k, in (C.1). If there is no feasible solution to (C.3) *then the short term plan is undefined.*

Remark C.2. As part of the RHC scheme, we will always be on our way to execute an old plan when a new plan is constructed. Thus the old plan can often serve as a feasible starting solution in the optimization.

In the case when no new feasible plan is found, *i.e.*, the NLP solver fails to provide a solution to (C.3), the execution continues along the old plan. The infeasibility may have different sources, including;

- The set of controls that fulfill the terminal cost decrement constraint being empty.
- Since the terminal cost is most often calculated from a graph representation of the environment, it might lead to trajectories that turn out to be dynamically infeasible in the future.
- Various optimization routine failures including non–convergence and abnormal termination.

Remark C.3. Optimization routine failures may always occur regardless the choice of terminal cost, and may therefore not be neglected in any case.

C.3.3 Task completing trajectory

In this section we will see how the short term plan produced by the NLP can be canonically extended all the way to the target, in a way that is collision free by design, as suggested in Figure C.1. This is explicitly done for every plan instance, k.

Definition C.2. *A task completing trajectory is defined as a feasible terminal trajectory which reaches the target, p_f. By an extended plan, we mean a short term plan P_k over time horizon T_p, augmented with a task completing trajectory. If no such can be found, the extended plan is undefined. Likewise, if no feasible short term plan is found in the NLP* (C.3)*, the extended plan is undefined.*

A canonical way of defining an extended plan is to augment the safety maneuver at the end of the short term plan with a straight line steady climb at α degrees towards the target, followed by a maximum rate turn and a straight line descent ending at the target. The dive is initiated at a point p_{top} (see Figure C.2), defined as the top vertex of a triangle, $\{p_{k,N}, p_{top}, p_f\}$, in the vertical plane with edge inclinations α, for the two upper edges. In instances when p_{top} and p_f are so close that the dive towards p_f violates the vehicles dive constraints, or when p_f is inside the maximum turn circle from p_{top} and cannot be reached, the extended plan is undefined.

When planning a trajectory on-line and either the NLP (C.3), or the formation of an extended plan according to Definition C.2 fails, no new plan is selected and the aerial vehicle keeps executing the old plan.

Note also that any choice of task completing trajectory above the triangle $\{p_{k,N}, p_{top}, p_f\}$ is also collision free by design. By using information about the highest terrain altitude, one can form more elaborate, and less conservative, extended plans. The important thing is that the construction is of low computational complexity.

C.3.4 Long term planning and terminal cost

The purpose of the terminal cost, $\Psi : \mathbb{R}^3 \to \mathbb{R}^+$, in the NLP (C.3) is to guide the short term plan in directions that make the flown overall trajectory good, in terms of the objective

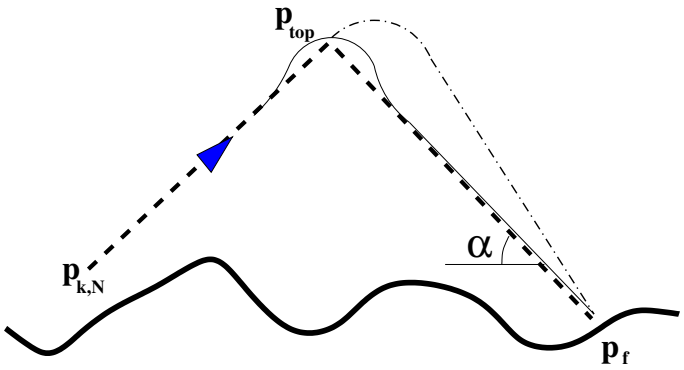


Figure C.2: Two alternatives for forming a task completing strategy. The dash-dotted curve illustrates the alternative described in Definition C.2. Note that this is only one of many ways to reach the target while flying above the safe dashed line and satisfying the vehicle dynamics (C.1). If there is a gap between $\bar{\gamma}$ and α, the thin solid curve is a less conservative such option. The important thing is that the construction is of low computational complexity.

function (C.2). It will also be used below to make sure there is not an infinite number of plan changes.

As in, *e.g.*, [4], [15], the terminal cost function, Ψ, is derived from a shortest path problem in a graph. This is done in a standard fashion and is therefore only described briefly here.

The graph is created by taking a horizontally equidistant mesh and using the terrain altitude $z = H(x, y)$ as values for the nodes. Above this set of nodes, four additional node layers are added at $z = H(x, y) + j\Delta_H$ meters, $j = 1, \cdots, 4$, where Δ_H is the inter layer distance. Each node is then connected by edges to its 8 neighbors in the same layer and the 9 neighbors above, and below.

The edge cost is a weighted sum of Euclidean distance and threat exposure, in accordance with the choices of $g(\cdot), b, c$ in the objective function (C.2). In order to use Ψ to decide when to change plans, we demand that Ψ is always positive, hence $\Psi : \mathbb{R}^3 \to \mathbb{R}^+$.

Again, as in [4], [15] we calculate the optimal cost to go from each node using a Dijkstra type of algorithm and then interpolate the node values to find Ψ. The interpolation routine used, can be shown to be consistent and free from local minimums inside each cube in the grid.

It is also worth noting here that the possibility of updating the "off-line" computed terminal should not be overlooked. The term "off-line" is rather to be interpreted as, at a much slower sampling rate than the trajectory planning loop, *i.e.*, in the order of tens of seconds (*cf.* [13]). As new information about the environment or mission objectives is gathered, it can be processed and fed back regularly to the online planner through an updated terminal cost.

Finally, if the planning and re-planning can be executed on a time scale which is faster than the vehicle velocity one can, by joining segments of short term plans where the safety maneuver has been omitted, plan an increasingly longer trajectory ahead with better optimality properties. In the extreme case where all this planning is successful, this will produce a chain of short term plans all the way to the target and neither the safety maneuver nor the terminal part of the extended plan will ever have to be executed.

C.3.5 The Algorithm

In this section we state the proposed algorithm, after making an initial assumption.

Assumption C.2. The first step in Algorithm C.1 returns a feasible solution.

In view of the discussion above, this assumption is very reasonable but excludes cases when the target position is extremely close to the launch position, and can not be reached even with a maximal turn.

Algorithm C.1.

1. The vehicle is launched at inclination α and velocity $v_{\max}$ in the direction of the target. At launch, the vehicle follows a default extended plan constructed by augmenting the initial state according to Definition C.2.
2. The vehicle executes the given plan for time T_e.
3. While executing, a new plan is sought according to Definitions C.1 and C.2.
4. If new informations arrive and Ψ needs to be recalculated, this is performed in the background. Once a new Ψ is calculated it is applied in Definition C.1 as well as in 5 below.
5. Let $\varepsilon > 0$ be fixed throughout the mission.
   If a new plan P_{k+1} according to Definitions C.1 and C.2 is found, then the new plan is activated instead of the old one, P_k.
6. If the target is not reached, go to 2.

Remark C.4. The choice of proceeding along the old plan or switching to a new one, thereby incrementally improving the terminal cost, can be viewed as a conditional version of the satisficing [16,17] control strategies used in some forms of RHC. In these RHC applications, the cost acts as a control Lyapunov function with which one can construct a control with local optimality properties and guaranteed global stability. In the trajectory optimization problem considered here, the counterpart of stability is safety and task completion, which are both already guaranteed by any single plan. The terminal cost function here acts by providing a means of improving the cost for completing the mission by changing to a new plan (while retaining safety and task completion) which is executed on the condition that the terminal cost is incrementally decreased.

C.4 Theoretical Properties

In this section we will prove that the proposed solution will indeed solve Problem C.1.

Proposition C.1. *If the terrain altitude does not increase within the short term planning horizon, and beyond that, does not change so that the maximal terrain inclination, α, is exceeded, then Algorithm C.1 solves Problem C.1.*

Proof. First note that each accepted plan is safe and reaches the goal by construction. So if there is only a finite number of changes of plan the aerial vehicle will reach the target. Then note that the Ψ-value of each accepted plan must be at least ε better than the previous one. Since $\Psi \geq 0$, there can at most be $\lceil \Psi(p_0)/\varepsilon \rceil$ changes.

Furthermore, the algorithm is iterative by design, and the RHC scheme makes it computationally efficient. Finally, the objective function (C.2) is approximately minimized. This proves the proposition. □

Remark C.5. If the vehicle starts out with a user defined upper bound on time to reach target, and only new plans that satisfy the bound are accepted, task completion will be guaranteed. The argument works equally well with a given fuel limit. In these instances one can thus drop the requirement on decay on the terminal cost Ψ from NLP (C.3).

Having discussed the proposed solution method and some of its properties, it is now time to run some simulation examples.

C.5 Simulations

For environmental representation, real terrain elevation data over the Cascade mountains, WA, have been used (see Figure C.3). The dataset used is a subset extracted from the one appearing in [18][2]. 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 [13], 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 16^{th} and 256^{th} 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, as explained above. The non-uniform grid built this way, can be seen as stretching out the layers of a uniform grid on the terrain surface.

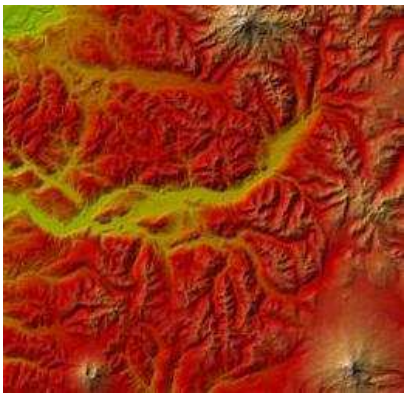


Figure C.3: The terrain elevation map used in the simulations represents an area of more than 82km $\times$82km taken from the Cascade range, WA. It contains the summit of Mt. Rainier, Mt. Adams and Mt. St. Helen's.

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.

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

Running the algorithm we get the trajectory shown in Figure C.4. The plot also shows a subset of the safety maneuvers. Note that none of these were actually executed. New improved plans were iteratively replacing the old ones, before the vehicle reached the climbing part of the trajectories.

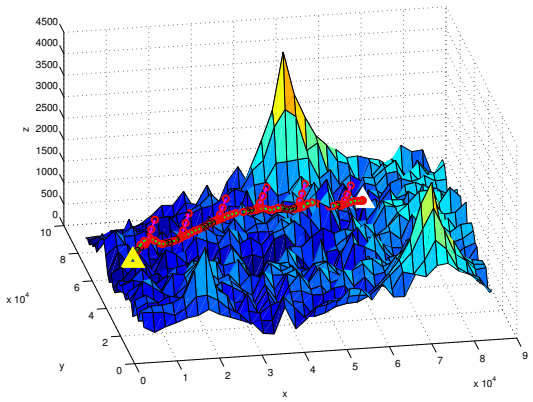


Figure C.4: At the end of the safety maneuver the trajectory is directed towards the target and has the climb angle α, which by definition is equal to the maximal terrain inclination. Here, to simplify exposition, the safety maneuver is shown at every tenth time step.

As discussed earlier, 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 in the graph representation of the environment. Figure C.5 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 marked 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

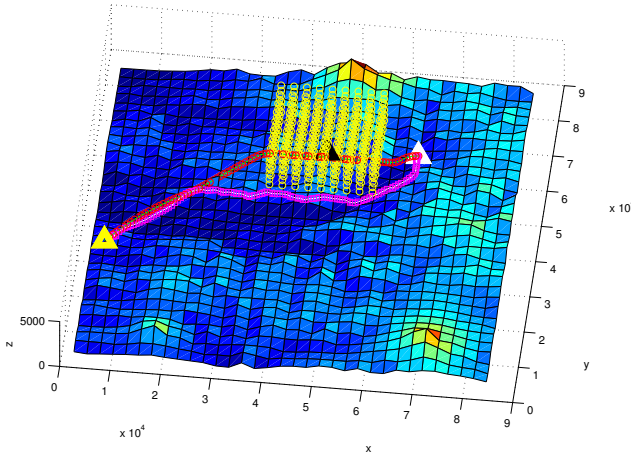


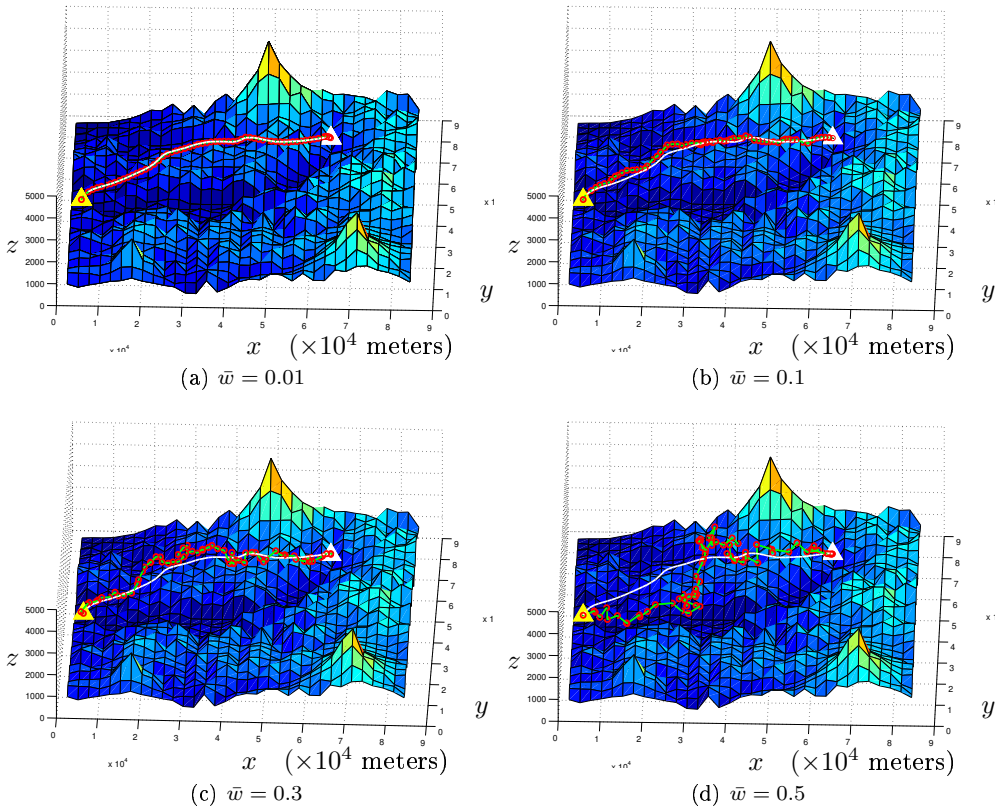
Figure C.5: The effect of threat exposure on the generated path.

exposure is now minimized by circumventing and flying at a much lower altitude, utilizing the protection provided by the terrain and thereby avoiding detection.

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 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, $p_{k,0} = p_{k-1,1}$, is modified to

$$p_{k,0} = p_{k-1,1} + w,$$

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 C.6 shows the generated paths corresponding to four increasing values on the noise parameter.



(a) $\bar{w} = 0.01$ (b) $\bar{w} = 0.1$

(c) $\bar{w} = 0.3$ (d) $\bar{w} = 0.5$

Figure C.6: 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.

As can be seen in the third column of Table C.1, the greatest disturbance corresponds to a displacement of more than 40%.Table C.1 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 can be read from the second column. The relative disturbance, *i.e.*, the ratio between the maximum offset and the maximum inter-node spacing, 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 $\bar{w}$.

$\bar{w}$	Max. offset	Relative disturbance	$\|a\|_2$
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 C.1: Impact of the added disturbance.

C.6 Concluding Remarks

In this paper, results regarding trajectory optimization for aerial vehicles in the three dimensional space has been presented. In particular, properties such as *safety* and *task completion* were in focus.

The alternative outlined in this work, 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 due to a combination of ideas which include a novel safety maneuver combined with a task completing trajectory and a conditional plan-changing strategy, where, starting from a feasible solution, a new plan for the remaining part of the mission is only accepted if it gives an incremental decrease to the terminal cost. The safety maneuver also has the beneficial side effect that it makes it possible to cope with hard real-time constraints.

Because of the computational burden it introduces, task completion is here not achieved by merely prolonging 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.

C.7 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] 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.

[5] Schouwenaars, T., How, J., and Feron, E., "Decentralized Cooperative Trajectory Planning of Multiple Aircraft with Hard Safety Guarantees," *AIAA Guidance, Navigation, and Control Conference and Exhibit, Providence, Rhode Island, August*, 2004.

[6] 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 (ACC)*, Vol. 5, Jun. 2003, pp. 4034–4040.

[7] 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.

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

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

[10] 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.

[11] Anisi, D. A., Robinson, J. W., and Ögren, P., "On-line trajectory planning for aerial vehicles: a safe approach with guaranteed task completion," *Proc. of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, Keystone, Colorado, Aug. 2006.

[12] Anisi, D. A., Ögren, P., and Robinson, J. W., "Safe receding horizon control of an aerial vehicle," *Proc. of the* 45^{th} *IEEE Conference on Decision and Control (CDC)*, San Diego, CA, Dec. 2006.

[13] 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.

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

[15] Kuwata, Y. and How, J. P., "Three dimensional receding horizon control for UAVs," *Proc. of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, Providence, Rhode Island, Aug. 2004, pp. 16–19.

[16] 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.

[17] 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.

[18] 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.

Paper D

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 temporal 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.

D.1 Introduction

THE paradigm of qualitative control design, that is associating a measure of the "utility" with 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 [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 D.2 some background material regarding computational methods for solving optimal control problems is presented. Subsequently in Section D.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 D.4, the benefits of utilizing the suggested method are confirmed by a missile guidance example. Finally, this paper is concluded in Section D.5 with some expository remarks.

D.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, S_f, 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 D.1). For a given OCP, indirect methods, which are based on the cal-

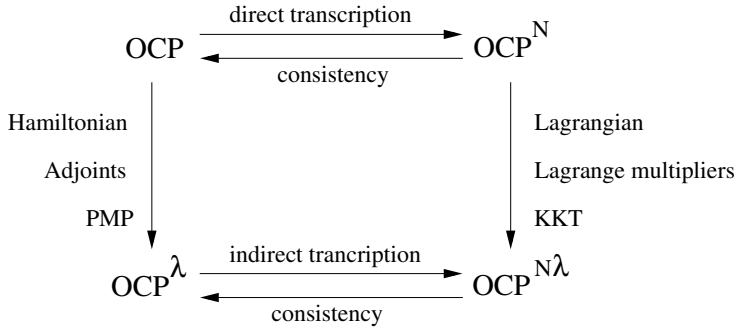


Figure D.1: Direct and indirect transcription methods.

culus 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^{λ} in Figure D.1). In contrast, direct methods transcribe the OCP directly, hence turning it into a large NLP (denoted OCP^{N} in Figure D.1). The dual to this NLP and the Lagrange multipliers may be achieved by way of the Lagrangian and the Karush-Kuhn-Tucker (KKT) conditions. The direct- and indirect methods have a particular simple relation for the so called *complete* methods [13], for which transcription and dualization indeed commutes, so that the Lagrange multipliers of the NLP are a multiple of the discretized values of the adjoint variables associated with the PMP.

Although indirect methods are considered to produce more accurate results, they are not typically used to solve problems having complex dynamics or constraint set. Neither are they suitable for problems where the underlying OCP is considered to be changeable in terms of the final manifold, S_f and/or the constraint set, $g(x, u)$. This is mainly due to the 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 states and/or controls 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]).

D.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 [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, 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 D.2 illustrates the average, and maximum run-times of NPOPT which is the solver used for all simulations in this paper. 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 D.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].

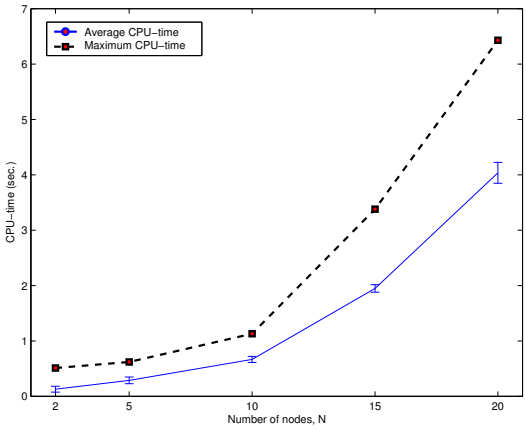


Figure D.2: The increasing average and maximum run-times of NPOPT as a function of N. Computations are performed on a shared Linux cluster, using one of its four 2.80 GHz Intel Xeon processors.

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

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

As an example, $w \equiv 1$ gives rise to the 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 additional 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:

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

which is to be augmented with the underlying NLP. From (D.1) it is 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 D.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 D.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.

D.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 D.3).

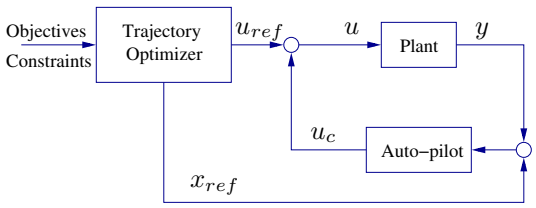


Figure D.3: Two level separation of the missile guidance problem.

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.

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), as seen in Figure D.4.

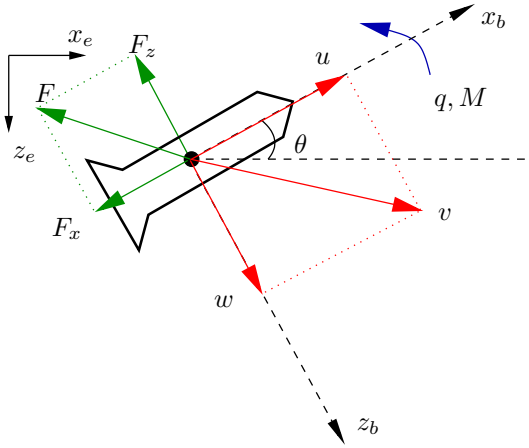


Figure D.4: Missile system variables.

The governing dynamic equations are

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

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 D.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 D.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. 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.

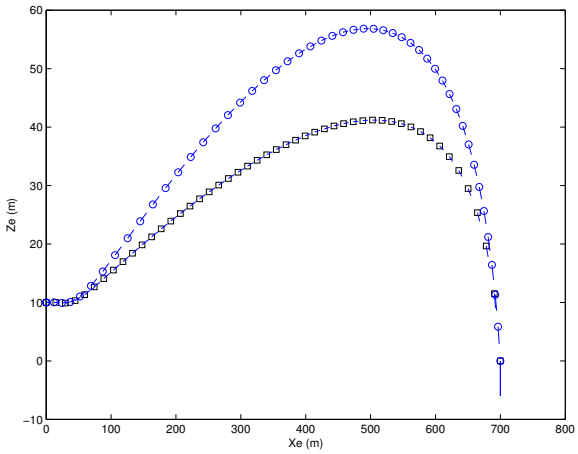


Figure D.5: Reference trajectories: static (○) and adaptive (□).

Figure D.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 D.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 D.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.

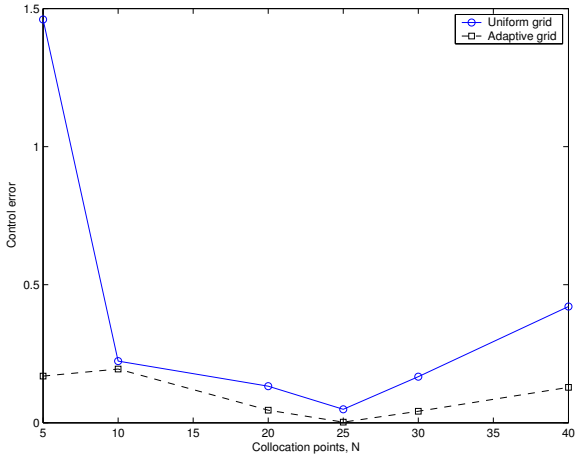


Figure D.6: The accuracy of the approximating control.

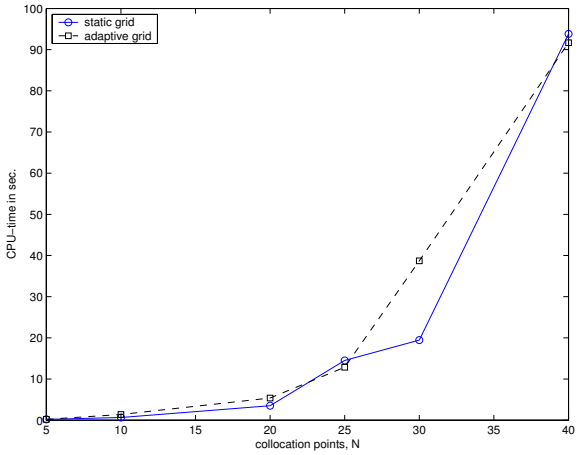


Figure D.7: The average CPU-time for the uniform- and adaptive grid generation scheme as a function of N.

D.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 augmented with the underlying NLP.

The missile guidance problem considered, showed that the extra degree of freedom provided by distributing the nodes may be 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.

D.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," *AIAA 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," *AIAA 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," *AIAA Journal of Guidance, Control, and Dynamics*, Vol. 23, No. 6, 2000, pp. 988–1000.

Paper E

Nonlinear Observability and Active Observers for Mobile Robotic Systems

Nonlinear Observability and 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.

E.1 Introduction

SINCE 1970's there has been an extensive study on the design of observers for nonlinear control systems, [1–7]. It is well 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. In [8] 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. However, study from the systems and control point of view in terms of observer design still lacks, [9].

This paper considers the problem of active observer design for a class of mobile robotic systems and an alternative design method is presented. It extends previous work in [10] by providing more formal and generalized definitions of the two sets utilized in the proposed design method, namely the set of feasible configurations and the set of output flow equivalent states. Also in this paper, the notion of small-time observability for mobile robotic systems is introduced. It is shown that the inter-relation between these two sets implies small-time observability and hence may serve as the basis for design of active observers. Furthermore, for the specific design study on the unicycle robot model, we consider the case of having more than 2 range-sensors but also demonstrate the robustness properties of the suggested observer in the face of noisy measurements.

The disposition of the paper is as follows; In Section E.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 E.3. To illustrate the concepts introduced in Section E.3.1, a case study is given in Section E.4. The simulation results thereof are presented in Section E.5 and finally, some concluding remarks are made in Section E.6.

E.2 Preliminaries

Consider the nonlinear control system

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

with state $x \in \mathcal{X}$, control $u(\cdot) \in \mathcal{U}$ 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.*,

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

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

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

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

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

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

Property E.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 E.2. Some of these modifications are considered below[1].

Definition E.3 (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 $x \in \mathcal{X}$.*

Remark E.1. The notion of weak observability (Definition E.3) is a relaxation of Property E.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 $\Omega \times \mathcal{U}_\Omega^{x_0}$ and Ω respectively. Here $\mathcal{U}_\Omega^{x_0}$ denotes the subset of all admissible inputs that generates trajectories that start in x_0 and lie in Ω.

Definition E.4 (Ω-indistinguishability). *Two initial states, $x_1, x_2 \in \Omega$ are said to be* Ω-indistinguishable *if*

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

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

Definition E.5 (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 $x \in \mathcal{X}$.*

Remark E.2. The notion of strong observability (Definition E.5) restricts Property E.2. Hence, 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 E.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 $x_0 \in \mathcal{X}$.*

Remark E.3. The notion of instant observability (Definition E.6) relaxes Property E.1, while restricting Property E.2.

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

Definition E.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 E.1, commutes and the error trajectory $z(t) - \Psi(x(t))$ converges to zero as $t \to \infty$.*

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

[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.

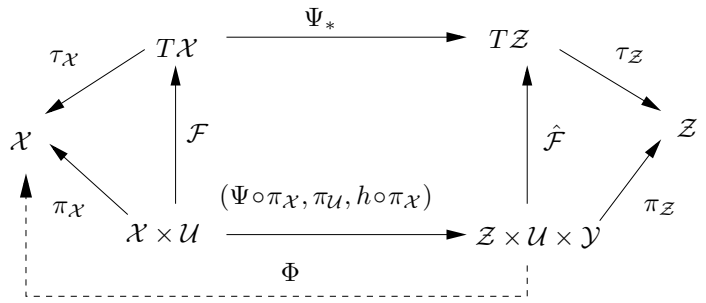


Figure E.1: Commutative diagram defining an observer.

means that all the maps indicated by arrows should be *consistent* regardless of point of arrival

According to Definition E.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

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

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

E.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:

$$\Sigma_{rob}: \begin{cases} \dot{x}_w &= f_w(x) + g_w(x)u \\ \dot{x}_c &= f_c(x) + g_c(x)u \\ y &= h_e(x) \\ 0 &\neq e(x_w), \end{cases}$$

where $x_w \in \mathcal{W}$, $x_c \in \mathcal{C}$, $u(\cdot) \in \mathcal{U}$ and $y \in \mathcal{Y}$. The map $e(q) = 0$ defines obstacles (walls) in the environment. We use the notation h_e to indicate the interaction of the sensors with the environment but also to emphasize the dependence of the output on the *environmental map*. 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 the hyper-surface defined by $e(q) = 0$ can be parameterized as

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

where θ is assumed to be 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 that the map is 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 E.1. Consider the unicycle vehicle model 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 $\mathcal{W}$ (*cf.* Figure E.2).

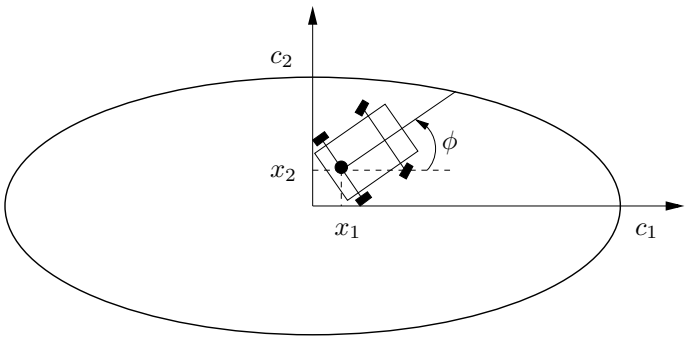


Figure E.2: The unicycle robot equipped with one range-measuring sensors moving inside an elliptic field.

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 $s_\theta = \tau$. 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 E.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

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

E.3.1 Observabiity and Active Observers

As pointed out in Section E.2, observability is an *infinite-horizon* concept (Property E.3). Notice that the observability notions listed in Section E.2 relax/restrict Properties E.1 and E.2 but there is no natural setting for solely modifying Property E.3. To adapt this for the area in mind, the following is suggested.

Definition E.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 any $T > 0$, there exists a control, $u(\cdot) \in \mathcal{U}$ and $\tilde{t} \in [0 \quad T]$, such that*

$$h_e(x(\tilde{t}, x_1, u(\cdot))) \neq h_e(x(\tilde{t}, x_2, u(\cdot))).$$

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

Remark E.4. The notion of small-time observability (Definition E.8) restricts Property E.3.

Remark E.5. Although not made explicit, modified versions of Definition E.8 (*i.e.*, weakly/strongly small time observability) can be obtained in apparent manners.

To stress the distinction between the newly introduced concept of small-time observability (Definition E.8) and those of Section E.2, recall that Ω-distinguishability, the underlying concept of Definition E.5, only involves separation of points in the restricting Ω. In extension, the term "instantaneously" in Definition E.6 implies that a point must be distinguishable from its *instant neighbors* through trajectories that stay in the same *instant neighborhood*. Therefore, there is no natural setting for solely modifying Property E.3, without necessarily modifying Property E.1 and/or E.2 (cf. Remarks E.1, E.2 and E.3). In contrast, the notion of small-time observability does not put any constraints on the generated trajectories nor the part of state-space that has to be distinguished, but *solely* restricts the time-interval that has to be considered in order to find deviating output.

Given the environmental map, $\theta(s_\theta)$, the sensor measurements are considered as a mapping, $h_e : \mathcal{X} \to \mathcal{Y}$. For a given measurement, $y \in \mathcal{Y}$, the inverse image of y under h_e is the set of all $x \in \mathcal{X}$ such that $h_e(x) = y$. 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 $\mathcal{X}$. To cope with this, the following definitions are made.

Definition E.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 h_e in the state-space,* i.e.*,*

$$\mathcal{FS}_y = \{x \in \mathcal{X} : h_e(x) = y\}.$$

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 E.10 (Value-function). *A continuously differentiable, 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 E.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 E.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 E.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 $t \in [0 \quad T]$,*

$$h_e(x(t, x_1, u(\cdot))) - h_e(x_1) \equiv h_e(x(t, x_0, u(\cdot))) - h_e(x_0).$$

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

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

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

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.*,

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

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

$$h_e(x_1) = h_e(x_0) = y, \tag{E.3}$$

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

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

□

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

Theorem E.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 $\mathcal{OF}^u_x$) one can uniquely determine the state, $x(t)$, then a dynamical system is said to be small-time observability.

E.4 Observer Design Study

In this section, we revisit the robot model from Example E.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 E.3, 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

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

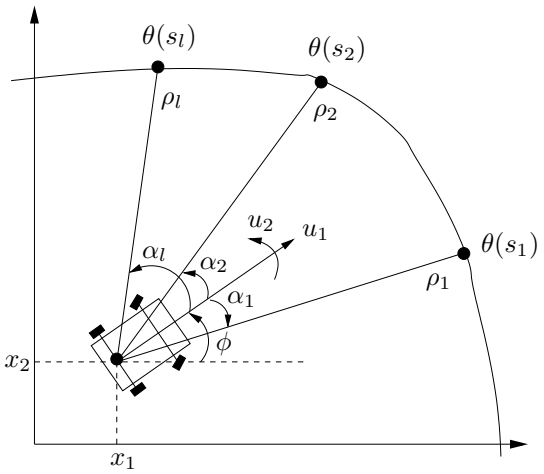


Figure E.3: The unicycle robot equipped with l range-measuring sensors.

The goal is to reconstruct the full state, $x = [x_1,\ x_2,\ \phi]$, based on sensor readings $\rho = (\rho_1, \cdots, \rho_l)$. As remarked in conjunction with Definition E.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:

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

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

$$V_\rho(z) = \frac{1}{2} \sum_{i=1}^{l-1} \sum_{j>i} v_{ij}(z)^T v_{ij}(z).$$

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 (E.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 (E.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. Definition E.11 implies that on the set of output flow equivalent states, the geometric relation (E.4) must be preserved on a non-empty time interval, which implies that $Q_i = 0$ for all i. Hence, the mapping defined by

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

contains 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 [10] for details), it can be shown that this set and the set of feasible states together fulfill the condition of Theorem E.1, which implies that we are bound to have weakly small-time observability. The way the two sets fulfill the condition of Theorem E.1 is that the differential of Q, denoted $\partial_z Q$, should not be perpendicular/orthogonal to the kernel of the map $\partial_z V_\rho$. We can then conclude that the two sets are not locally parallel and hence will cross each other in one single point. This is the requirement of Theorem E.1.

From the results of [10], it is concluded that the exciting control must be chosen to satisfy

$$\begin{bmatrix} \sin\alpha_1 & -\rho_1 \\ \vdots & \vdots \\ \sin\alpha_l & -\rho_l \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \neq 0. \tag{E.5}$$

Next, we show two special families of exciting control that fulfill (E.5), are bounded, have bounded derivative, and generate periodic trajectories that will remain inside the elliptic environment, denoted Ω. The first one uses trigonometric functions to generate motion that moves back and forth while the other one follows the curvature of an elliptic reference path back to the initial position.

Exciting Control 1 (Trigonometric Functions)**.** We first elaborate with exciting controls of the following form

$$u_{x_0}(t) = \begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix} = \begin{bmatrix} a_1 \sin(k_1 + p_1 t) \\ a_2 \sin(k_2 + p_2 t) \end{bmatrix}.$$

Below, it is shown that for particular choices of the parameters, k_1, p_1, k_2, p_2, the exciting control, $u_{x_0}(t)$, generates periodic trajectories. Hence for a given initial position, x_0, in the interior of Ω, by adjusting the remaining free gain parameters, a_1, a_2, we are able to keep the generated trajectory inside Ω for all time.

Starting with the vehicle orientation, ϕ, by integrating its dynamics, we obtain

$$\phi(t) = \phi_0 - \frac{a_2}{p_2}\cos(k_2 + p_2 t).$$

Inserting this into the position dynamic yields

$$\begin{aligned}\dot{x}_1 &= u_1(t)\cos\left(\phi_0 - \frac{a_2}{p_2}\cos(k_2 + p_2 t)\right) \triangleq u_1(t)c_1(t),\\ \dot{x}_2 &= u_1(t)\sin\left(\phi_0 - \frac{a_2}{p_2}\cos(k_2 + p_2 t)\right) \triangleq u_1(t)c_2(t).\end{aligned}$$

Hence,

$$\begin{aligned}x_1(t) &= x_1(t_0) + \int_{t_0}^{t_0+t} u_1(\tau)c_1(\tau)\mathrm{d}\tau,\\ x_2(t) &= x_2(t_0) + \int_{t_0}^{t_0+t} u_1(\tau)c_2(\tau)\mathrm{d}\tau.\end{aligned}$$

To obtain periodicity of motion, we want to put constraints on the function u_1 such that

$$\int_{t_0}^{t_0+\bar{T}} u_1(t)c_i(t)\mathrm{d}t = 0, \quad i = 1, 2,$$

for some $\bar{T} \in \mathbb{R}^+$, *i.e.*, we return to the initial position at time $t_0 + \bar{T}$.

To this end, notice that $c_i(t)$, $i = 1, 2$, are periodic functions with period $\frac{2\pi}{p_2}$ and that the choice $k_2 = 0$ gives

$$c_i(t) = c_i(\frac{2\pi}{p_2} - t), \quad i = 1, 2.$$

Then if we choose u_1 as a periodic function with period $\frac{4\pi}{p_2}$ which further satisfies

$$u_1(t) = -u_1(\frac{4\pi}{p_2} - t),$$

we get

$$\int_{t_0}^{t_0+\frac{4\pi}{p_2}} u_1(t)c_i(t)\mathrm{d}t = 0, \quad i = 1, 2,$$

which implies that the vehicle will return to it's starting position at $t = t_0 + \frac{4\pi}{p_2}$.

The family of functions, $u_1(t)$, fulfill the above mentioned requirements if we set

$$\begin{aligned}p_1 &= \frac{p_2}{2},\\ k_1 + \frac{2\pi}{p_2}p_1 &= \pi n,\end{aligned}$$

for some $n \in \mathbb{Z}$. In the simulations presented in Section E.5, we have set $a_1 = 0.5, k_1 = -\pi, p_1 = \pi, a_2 = 1.5, k_2 = 0$ and $p_2 = 2\pi$, which fulfill the requirements with $n = 0$.

Exciting Control 2 (Curvature Following)**.** Another family of controls that meet our objectives can be extracted by applying standard curvature following techniques.

From the system dynamics we obtain that

$$\begin{aligned} \phi &= \arctan \frac{\dot{x}_2}{\dot{x}_1}, \\ u_1 &= \sqrt{\dot{x}_1^2 + \dot{x}_2^2}, \\ u_2 &= \dot{\phi} = \frac{\dot{x}_1 \ddot{x}_2 - \ddot{x}_1 \dot{x}_2}{\dot{x}_1^2 + \dot{x}_2^2}. \end{aligned}$$

Let

$$\begin{aligned} x_1^r &= a_1 \cos(wt) \\ x_2^r &= a_2 \sin(wt) \end{aligned}$$

denote the periodic reference trajectory, in this case, an ellipse. We have

$$\begin{aligned} \dot{x}_1^r &= -a_1 w \sin(wt) \\ \ddot{x}_1^r &= -a_1 w^2 \cos(wt) \\ \dot{x}_2^r &= a_2 w \cos(wt) \\ \dot{x}_1^r &= -a_2 w^2 \sin(wt), \end{aligned}$$

and hence we take

$$\begin{aligned} u_1(t) &= w\sqrt{a_1^2 \sin^2(wt) + a_2^2 \cos^2(wt)} \\ u_2(t) &= \frac{a_1 a_2 w^3}{u_1^2} = \frac{a_1 a_2 w}{a_1^2 \sin^2(wt) + a_2^2 \cos^2(wt)}. \end{aligned}$$

Remark E.6. In principle, the choice of the gain parameters, a_1 and a_2, should depend on the initial state, x_0, or more precisely, the distance to the closest obstacle which can be estimated by the range sensors.

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

$$\dot{z} = -k_V \Big[\frac{\partial V_\rho}{\partial z}\Big]^T - k_Q \Big[\frac{\partial Q}{\partial z}\Big]^T Q, \tag{E.6}$$

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

The first term in (E.6) serves to drive the state to the set of feasible states, $\mathcal{FS}_y$, while the objective of including the second term is to strive towards the set of output flow equivalent states, $\mathcal{OF}_x^u$. Since these two sets fulfill the condition of Theorem E.1, which implies that the system is weakly small-time observable, any two states can be locally distinguished.

To obtain a full state observer, the full observer mapping, Φ, is to be decided (*cf.* Figure E.1). By relation (E.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 E.1. *Assume the control and it's first derivative are both bounded and generate periodic motions. Then the estimation error of the proposed observer* (E.6) *is locally bounded and can be made arbitrarily small by tuning the observer parameters,* k_V *and* k_Q*.*

Proof. Let $z(t)$ and $\hat{z}(t)$ denote the true and estimated states respectively and set $\delta z = \hat{z} - z$. Defining a candidate Lyapunov function

$$V(\delta z) = \frac{1}{2}\delta z^T \delta z = \frac{1}{2}\|\delta z\|^2,$$

the total derivate of V along the observer dynamics (E.6) becomes

$$\dot{V} = -\delta z^T \Big[k_V \frac{\partial V_\rho^T(\hat{z})}{\partial \hat{z}} + k_Q \frac{\partial Q(\hat{z})^T}{\partial \hat{z}} Q(\hat{z}) \Big] - \delta z^T \dot{z}.$$

If δz is sufficiently small, then we have

$$v_{ij}(\hat{z}) \approx \frac{\partial v_{ij}(z)}{\partial z}\delta z, Q(\hat{z}) \approx \frac{\partial Q(z)}{\partial z}\delta z.$$

Now,

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

thus,

$$\dot{V} = -\delta z^T A(t) \delta z - \delta z^T \dot{z} + \mathcal{O}(\|\delta z\|^2)$$

where,

$$A(t) = k_V \sum_{i=1}^{l-1}\sum_{j>i} \frac{\partial v_{ij}(z)^T}{\partial z}\frac{\partial v_{ij}(z)}{\partial z} + k_Q \frac{\partial Q(z)^T}{\partial z}\frac{\partial Q(z)}{\partial z}.$$

Since the condition of Theorem E.1 is fulfilled,

$$a(t) = \big[\sqrt{k_V}\frac{\partial v_{12}(z)^T}{\partial z}, \cdots, \sqrt{k_V}\frac{\partial v_{l-1,l}(z)^T}{\partial z}, \sqrt{k_Q}\frac{\partial Q(z)^T}{\partial z}\big]$$

is of full rank, therefore

$$A(t) = a(t)^T a(t)$$

is always of full rank, and thus, positive definite for any fixed t. Due to the periodicity and differentiability assumptions (on u and the smoothness of the other mappings), $A(t)$ is periodic and $\dot{z}(t)$ is bounded. Thus if we let $k_V > k, k_Q > k$, we have

$$\dot{V} \leq -k\gamma\|\delta z\|^2 + P\|\delta z\|,$$

where $\gamma, P > 0$. This shows that the size of the design parameter k, bounds and tunes the estimation error. □

E.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 E.1 and depicted in Figure E.2 (with $c_1 = 3$ and $c_2 = 2$). 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 $k_V = 5, k_Q = 1$.

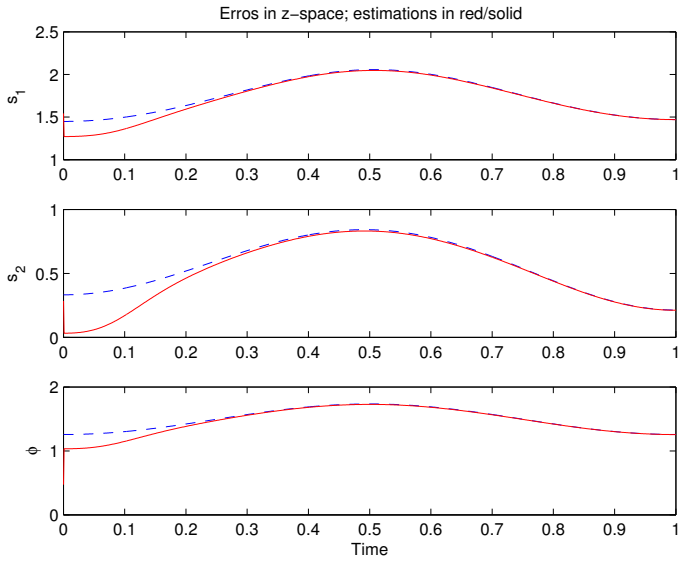


Figure E.4: Observation error in $\mathcal{Z}$-space.

Figure E.4 shows the trajectory of the components of $z(t)$ (in dashed/blue) and $\hat{z}(t)$ (in solid/red). This figure 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, $\mathcal{X}$. Figure E.5 shows the observation errors as measured after mapping $\hat{z}$ into $\hat{x}$ by means of the full observer mapping, Φ.

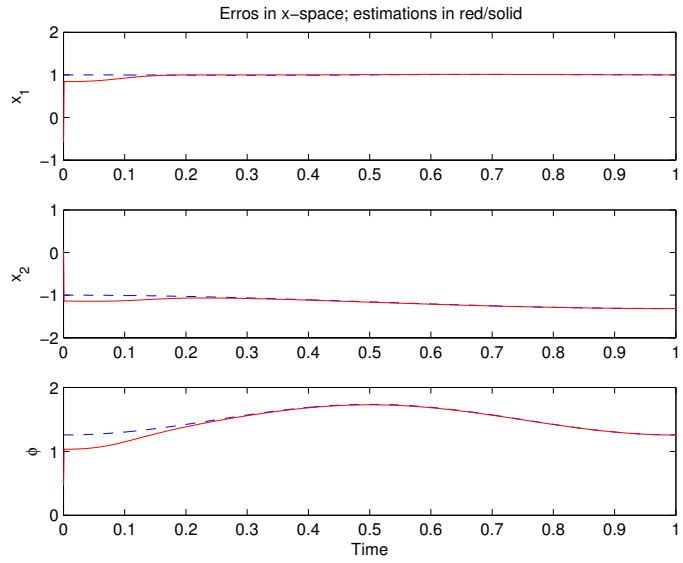


Figure E.5: Observation error in $\mathcal{X}$-space.

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 E.6, 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 E.7). In cases when (E.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 E.7 that, in the presence of measurement noise, the position estimation is much better than the estimation of the orientation angle, ϕ.

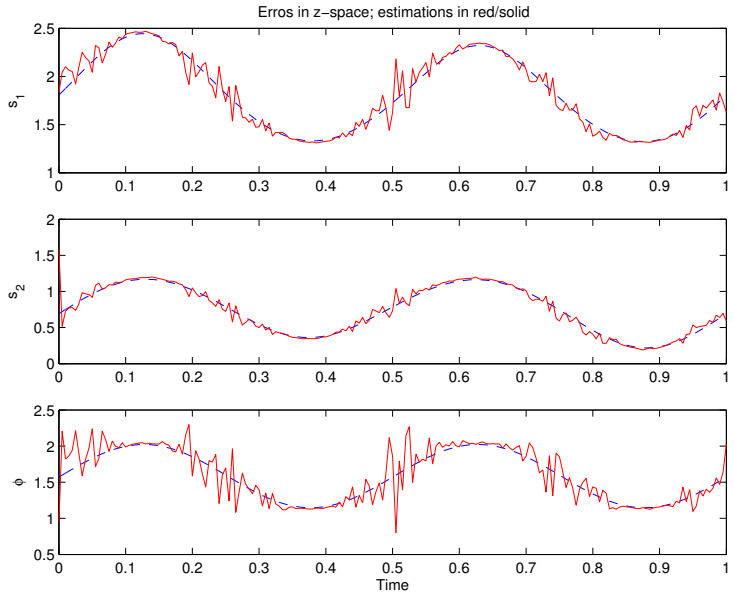


Figure E.6: Observation error in $\mathcal{Z}$-space with noise.

E.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 and 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 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

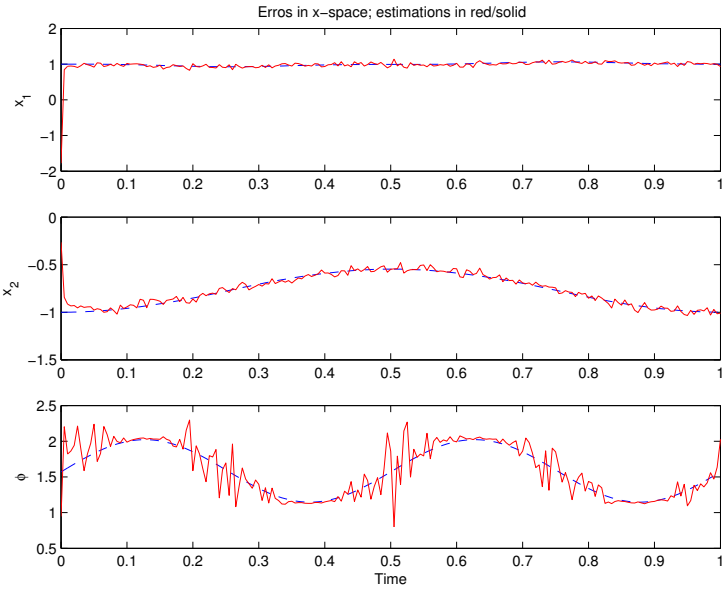


Figure E.7: Observation error in $\mathcal{X}$-space with noise.

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.

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

[10] Cederwall, S. and Hu, X., "Nonlinear Observers for Unicycle Robots With Range Sensors," *IEEE Transactions on Automatic Control*, Vol. 52, No. 7, Jul. 2007, pp. 1325–1329.

[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.

Paper F

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.

F.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

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

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 F.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 F.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 F.1 is commutative (*cf. *[1] and [2]).*

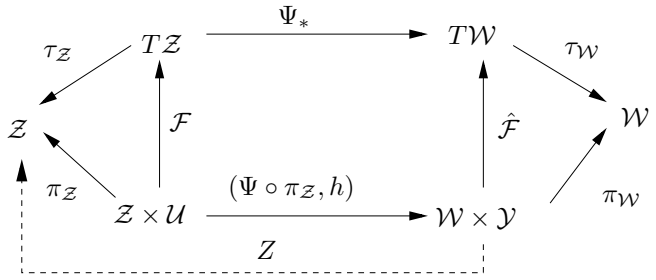


Figure F.1: Commutative diagram defining an observer.

In the diagram of Figure F.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 F.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 F.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 F.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 F.1, this implies Property F.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

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

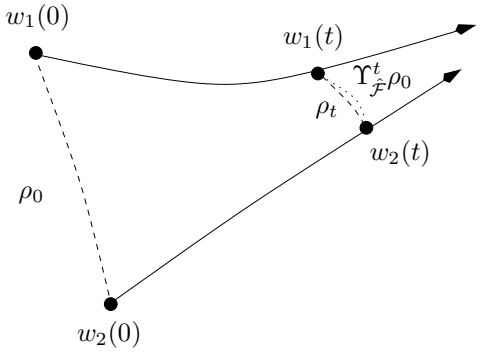


Figure F.2: The length of the geodesic curve ρ_t, between two trajectories decreases if $\mathcal{L}_{\hat{\mathcal{F}}} G < 0$.

However, the assumption that the observer dynamics is contractive, is very restrictive and in most cases, Property F.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 velocity 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 F.2 is devoted to introducing some preliminary concepts of tangent bundle geometry (Section F.2.1) and Euler-Lagrange systems (Section F.2.2). The design of the observer is the subject of Section F.3, while Section F.4 is devoted to the convergence properties of it. Finally, these properties are illustrated in Section F.5, where we present some simulation results.

F.2 Preliminaries

F.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

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

so that

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

and the the Ricci identity,

$$Y^i{}_{|j|k} - Y^i{}_{|k|j} = R_m{}^i{}_{jk} Y^m, \tag{F.1}$$

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 F.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 F.1. These expressions will be used for the component–wise contraction analysis in Section F.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 F.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

$$\begin{aligned}
\mathcal{L}_{Y^{\mathbb{V}}}\begin{bmatrix}\mathrm{d}x^i\\ \mathrm{D}v^i\end{bmatrix} &= \left[\begin{array}{c|c}0 & 0\\ \hline Y^i_{|j} & 0\end{array}\right]\begin{bmatrix}\mathrm{d}x^j\\ \mathrm{D}v^j\end{bmatrix} \qquad \text{(F.2)}\\
\mathcal{L}_{Y^{\mathbb{H}}}\begin{bmatrix}\mathrm{d}x^i\\ \mathrm{D}v^i\end{bmatrix} &= \left[\begin{array}{c|c}Y^i_{|j} & 0\\ \hline R_k{}^i{}_{jl}\,v^k\,Y^l & 0\end{array}\right]\begin{bmatrix}\mathrm{d}x^j\\ \mathrm{D}v^j\end{bmatrix}\\
\mathcal{L}_{\overset{\triangledown}{Z}}\begin{bmatrix}\mathrm{d}x^i\\ \mathrm{D}v^i\end{bmatrix} &= \left[\begin{array}{c|c}0 & \delta^i_j\\ \hline R_k{}^i{}_{jl}\,v^k\,v^l & 0\end{array}\right]\begin{bmatrix}\mathrm{d}x^j\\ \mathrm{D}v^j\end{bmatrix}\\
\mathcal{L}_{\mathbb{I}(R(\cdot,Y))^{\mathbb{V}}}\begin{bmatrix}\mathrm{d}x^i\\ \mathrm{D}v^i\end{bmatrix} &= \left[\begin{array}{c|c}0 & 0\\ \hline (R_m{}^i{}_{kl}Y^l)_{|j}\,v^k\,v^m & (R_j{}^i{}_{kl}+R_k{}^i{}_{jl})Y^lv^k\end{array}\right]\begin{bmatrix}\mathrm{d}x^j\\ \mathrm{D}v^j\end{bmatrix}.
\end{aligned}$$

F.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

$$\boxed{\mathcal{F} = \overset{\nabla}{Z} - (\mathrm{grad}\, U)^{\mathbb{V}} + F^{\mathbb{V}}}$$

F.3 Observer Design

For the class of systems, Σ, described in Section F.2.2, we now introduce a full state identity observer, $\hat{\Sigma}$.

Referring to Figure F.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 $R^\alpha = \mathbb{I}(R(\cdot, \mathrm{grad}\, S))^\alpha = R_\beta{}^\alpha{}_{\gamma\iota}\eta^\beta S^\gamma \eta^\iota$, 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.

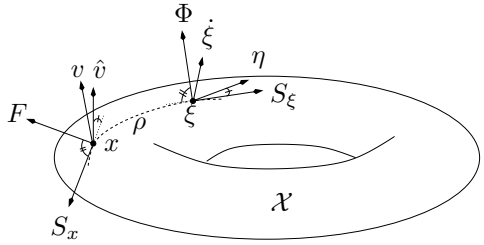


Figure F.3: The system- and observer variables, are denoted by Latin and Greek letters respectively.

The parallel transport operator, K^i_α, has the following properties, which are easily verified in Fermi coordinates:

$$K^i_{\alpha|\beta}S^\beta = 0. \tag{F.3}$$

$$K^i_\alpha S^\alpha = -S^i \tag{F.4}$$

$$K^i_\alpha K^j_\beta g^{\alpha\beta} = g^{ij} \tag{F.5}$$

Upon introducing this notation, the following observer dynamics is suggested for Σ:

$$\begin{aligned} \dot{\xi}^\alpha &= \eta^\alpha - Ag^{\alpha\beta}S_\beta, \qquad \alpha = 1,\dots,n \\ \mathrm{D}_t\eta^\alpha &= -Bg^{\alpha\beta}S_\beta - g^{\alpha\beta}U_{,\beta} + \Phi^\alpha + CR^\alpha, \end{aligned} \tag{F.6}$$

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 (F.6) satisfies the diagram property of Definition F.1. As observer output mapping, Z, we may for instance use $Z_1 = (\xi^\alpha, \eta^\alpha)$, or $Z_2 = (x^i, K^i_\alpha\eta^\alpha)$. Choosing the latter approach[3], the velocities, v^i, are estimated as

$$\hat{v}^i = K^i_\alpha\eta^\alpha. \tag{F.7}$$

Thus, putting (F.6) and (F.7) together, the following observer, $\hat{\Sigma}$, is suggested for Σ

$$\boxed{\hat{\Sigma}: \left\{ \begin{aligned} \dot{\xi}^\alpha &= \eta^\alpha - Ag^{\alpha\beta}S_\beta \\ \mathrm{D}_t\eta^\alpha &= -Bg^{\alpha\beta}S_\beta - g^{\alpha\beta}U_{,\beta} + \Phi^\alpha + CR^\alpha \\ \hat{v}^i &= K^i_\alpha\eta^\alpha \end{aligned} \right.}$$

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 F.4.3, we follow [5], by choosing $C = 1$ and the output mapping Z_1, while in Section F.4.4 we use a general C and Z_2.

F.4 Convergence Analysis

In this section, convergence issues are treated by means of contraction analysis (Section F.4.3) and, in the case of constant curvature, by means of a conventional Lyapunov method (Section F.4.4). To this end however, we devote Sections F.4.1 and F.4.2 to deriving expressions for the variation of some quantities along a geodesic.

F.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

$$S_\alpha S^\alpha - 2S = 0, \tag{F.8}$$

Taking the covariant derivative of (F.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

$$S^\beta{}_{|\alpha}S^\alpha - S^\beta = 0. \tag{F.9}$$

[3] In the face of noisy measurements, it might be advantageous to consider Z_1 instead.

Then, combining (F.3), (F.4), (F.8) and (F.9), it follows that

$$S^i{}_{|\beta}S^\beta - S^i = 0. \tag{F.10}$$

The covariant derivative of (F.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

$$S^\beta{}_{|\gamma|\alpha}S^\alpha = S^\beta{}_{|\gamma} - R_\iota{}^\beta{}_{\alpha\gamma}S^\iota S^\alpha - S^\beta{}_{|\alpha}S^\alpha{}_{|\gamma}. \tag{F.11}$$

In a similar fashion, we obtain

$$K^i_{\beta|\gamma|\alpha}S^\alpha = R_\beta{}^\epsilon{}_{\alpha\gamma}K^i_\epsilon S^\alpha + K^i_{\beta|\alpha}S^\alpha{}_{|\gamma}. \tag{F.12}$$

It should be possible to derive Grönwall-like estimates of $S^\beta{}_{|\gamma}$ and $K^i_{\beta|\gamma}$ from (F.11) and (F.12). In the present paper, however, we focus on spaces of constant curvature.

F.4.2 Constant Curvature

In the case when $\mathcal{X}$ has constant curvature, *i.e.*, when

$$R_\iota{}^\beta{}_{\alpha\gamma} = \kappa(\delta^\beta_\gamma g_{\iota\alpha} - \delta^\beta_\alpha g_{\iota\gamma}), \tag{F.13}$$

equation (F.11) may be explicitly solved for $S^\beta{}_{|\gamma}$ by means of the Ansatz

$$S^\beta{}_{|\gamma} = \Upsilon_1(S)\delta^\beta_\gamma + \Upsilon_2(S)S^\beta S_\gamma. \tag{F.14}$$

Multiplying with S^γ from the right and using (F.9) immediately yields that $\Upsilon_1 + 2S\Upsilon_2 = 1$. Substituting this back into (F.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 F.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

$$K^i_\alpha = \Upsilon_3(S)S^i{}_{|\alpha} + \Upsilon_4(S)S^i S_\alpha. \tag{F.15}$$

Multiplying (F.15) with S^α from the right and using (F.4), and (F.10), we obtain $\Upsilon_3 + 2S\Upsilon_4 = -1$. Substituting this back in the Ansatz (F.15) and utilizing Property (F.3), we get

$$K^i_{\alpha|\beta} S^\beta \big(2S\Upsilon_3' + \Upsilon_3 - \Upsilon_3\Upsilon_1\big)(S^i{}_{|\alpha} - \frac{1}{2S} S^i S_\alpha) = 0.$$

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

$$K^i_\alpha = -\frac{\sin\sqrt{2\kappa S}}{\sqrt{2\kappa S}} S^i{}_{|\alpha} - \kappa \frac{\sqrt{2\kappa S} - \sin\sqrt{2\kappa S}}{(\sqrt{2\kappa S})^3} S^i S_\alpha.$$

Noticing Remark F.1, we differentiate (F.15) with respect to ξ^β and obtain

$$K^i{}_{\alpha|\beta} = \Upsilon_3'(S) S_\beta S^i{}_{|\alpha} + \Upsilon_3(S) S^i{}_{|\alpha|\beta} + \Upsilon_4'(S) S_\beta S^i S_\alpha + \Upsilon_4(S) S^i{}_{|\beta} S_\alpha + \Upsilon_4(S) S^i S_{\alpha|\beta}.$$

Regarding the first and fourth term, from (F.15), we have

$$S^i{}_{|\alpha} = \frac{1}{\Upsilon_3(S)} \big[K^i_\alpha + \Upsilon_4(S) s^i S_\alpha\big].$$

The third term is satisfactory, while $S_{\alpha|\beta}$ in the fifth term can be calculated via (F.14) and equals

$$S_{\alpha|\beta} = \Upsilon_1(S) g_{\alpha\beta} + \Upsilon_2(S) S_\alpha S_\beta. \tag{F.16}$$

Paying attention to the second term, since we have

$$S^i{}_{|\alpha|\beta} = g^{ij} S_{\alpha|\beta|j}, \tag{F.17}$$

we differentiate (F.16) with respect to x^j, yielding

$$S_{\alpha|\beta|j} = \Upsilon_1'(S) S_j g_{\alpha\beta} + \Upsilon_2'(S) S_j S_\alpha S_\beta + \Upsilon_2(S) S_{\alpha|j} S_\beta + \Upsilon_2(S) S_\alpha S_{\beta|j}.$$

Substituting this back into (F.17), we get

$$S^i{}_{|\alpha|\beta} = \Upsilon_1'(S) S^i g_{\alpha\beta} + \Upsilon_2'(S) S^i S_\alpha S_\beta + \Upsilon_2(S) S^i{}_{|\alpha} S_\beta + \Upsilon_2(S) S_\alpha S^i{}_{|\beta}.$$

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

$$\begin{cases} K^i_{\alpha|\beta} &= \Upsilon(S)(S^i g_{\alpha\beta} + K^i_\beta S_\alpha), \\ \Upsilon(S) &= \kappa \frac{\tan\frac{1}{2}\sqrt{2\kappa S}}{\sqrt{2\kappa S}}. \end{cases} \tag{F.18}$$

By manipulating (F.18), with the roles of x and ξ reversed, we also obtain

$$K^j_{\alpha|\beta} K^\alpha_m K^\beta_k = K^j_{\alpha|k} K^\alpha_m = \Upsilon(S)(S^j g_{km} - \delta^j_m S_k). \tag{F.19}$$

F.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

$$\mathcal{L}_{\hat{\mathcal{F}}}G = \mathcal{L}_{\underset{Z}{\nabla}}G - A\mathcal{L}_{(\text{grad } S)^{\mathbb{H}}}G - B\mathcal{L}_{(\text{grad } S)^{\mathbb{V}}}G - \mathcal{L}_{(\text{grad } U)^{\mathbb{V}}}G + \mathcal{L}_{\Phi^{\mathbb{V}}}G + C\mathcal{L}_{\mathcal{R}}G.$$

Using the formulas (F.2), we arrive at

$$\mathcal{L}_{\hat{\mathcal{F}}}G = \begin{bmatrix} \mathrm{d}\xi^\alpha \\ \mathrm{D}\eta^\alpha \end{bmatrix}^T \otimes \mathcal{M} \begin{bmatrix} \mathrm{d}\xi^\beta \\ \mathrm{D}\eta^\beta \end{bmatrix}, \tag{F.20}$$

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

$$\begin{aligned} M_{\alpha\beta} &= -AS_{\alpha|\beta} \\ N_{\alpha\beta} &= g_{\alpha\beta} \\ Q_{\alpha\beta} &= C(R_{\beta\alpha\gamma\iota} + R_{\gamma\alpha\beta\iota})S^\iota\eta^\gamma \\ P_{\alpha\beta} &= Y_{\alpha\beta\gamma\iota}\eta^\gamma\eta^\iota + AR_{\gamma\alpha\beta\iota}\eta^\gamma S^\iota - BS_{\alpha|\beta} - U_{\alpha|\beta} + g_{mn}F^m K^n_{\alpha|\beta}, \end{aligned}$$

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

$$\mathcal{M} = \begin{bmatrix} 2c & b \\ b & 0 \end{bmatrix} Y_{\alpha\beta\gamma\iota}\eta^\gamma\eta^\iota + \mathcal{O}(\eta). \tag{F.21}$$

Based on these calculations, the following result can be formulated.

Theorem F.1. *The contracting neighborhood of the set $S = 0$ shown in [5], is* infinitely thin *as $|\eta|_g \to \infty$.*

Proof. We outset from (F.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. □

F.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

$$\dot{V} = g_{ij} \Delta v^i (\mathrm{D}_t v^j - \mathrm{D}_t \hat{v}^j) + BS_i \dot{x}^i + BS_\alpha \dot{\xi}^\alpha, \tag{F.22}$$

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 (F.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

$$\mathrm{D}_t \hat{v}^j = K^j_{\alpha|k} v^k \eta^\alpha + K^j_{\alpha|\beta} \dot{\xi}^\beta \eta^\alpha + BS^j + F^j + CK^j_\alpha R^\alpha. \tag{F.23}$$

Concentrating on the last term in (F.23), from (F.13) and (F.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 (F.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 $\hat{\Sigma}$, (F.3), $\eta^\alpha = K^\alpha_m \hat{v}^m$ and (F.19) respectively. Then by substituting $\eta^\alpha = K^\alpha_m \hat{v}^m$ in the first term as well, (F.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

$$\mathrm{D}_t \hat{v}^j = K^j_{\alpha|k} K^\alpha_m \hat{v}^m \Delta v^k + BS^j + F^j. \tag{F.24}$$

Consider next the last terms in the total derivative (F.22). From (F.4), in conjunction with dynamics of $\hat{\Sigma}$ and the Hamilton-Jacobi equation (F.8), it follows that

$$S_\alpha \dot{\xi}^\alpha = -K^i_\alpha S_i (\eta^\alpha - A g^{\alpha\beta} S_\beta) = -S_i \hat{v}^i - A g^{\alpha\beta} S_\alpha S_\beta = -S_i \hat{v}^i - AS^\beta S_\beta = -S_i \hat{v}^i - 2AS.$$

Substituting these modified terms back into (F.22) yields

$$\begin{aligned}\dot{V} &= g_{ij}\Delta v^i\big[\cancel{P^j} - K^j_{\alpha|k}K^\alpha_m\hat{v}^m\Delta v^k - BS^j - \cancel{P^j}\big] + BS_iv^i + B\big[-S_i\hat{v}^i - 2AS\big]\\ &= -g_{ij}\Delta v^iK^j_{\alpha|k}K^\alpha_m\hat{v}^m\Delta v^k - Bg_{ij}\Delta v^iS^j + BS_i\Delta v^i - 2ABS\\ &= -g_{ij}\Delta v^iK^j_{\alpha|k}K^\alpha_m\hat{v}^m\Delta v^k - \cancel{BS_i\Delta v^i} + \cancel{BS_i\Delta v^i} - 2ABS\\ &= -g_{ij}\Delta v^iK^j_{\alpha|k}K^\alpha_m\hat{v}^m\Delta v^k - 2ABS.\end{aligned}$$

By using (F.19), we arrive at

$$\begin{aligned}\dot{V} &= -g_{ij}\Delta v^i\Upsilon(S)\big[S^jg_{km} - \delta^j_mS_k\big]\hat{v}^m\Delta v^k - 2ABS\\ &= -\Upsilon(S)\big[g_{km}S_i - g_{im}S_k\big]\hat{v}^m\Delta v^i\Delta v^k - 2ABS. \qquad \text{(F.25)}\end{aligned}$$

Theorem F.2. *If it is known that* $\sup_t|v(t)|_g \le v_{\max}$ *, 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

$$-\Upsilon(S)\big[g_{km}S_i - g_{im}S_k\big]\hat{v}^m\Delta v^i\Delta v^k \le |\Upsilon(S)|\Big[|g_{km}S_i\hat{v}^m\Delta v^i\Delta v^k| + |g_{im}S_k\hat{v}^m\Delta v^i\Delta v^k|\Big].$$

From (F.8) and the assumption $\sup_t|v(t)|_g \le v_{\max}$, it follows that

$$|g_{km}S_i\hat{v}^m\Delta v^i\Delta v^k| \le \sqrt{2S}|\eta|_g(v_{\max}+|\eta|_g)^2,$$

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

$$\dot{V} \le 2|\Upsilon(S)|\sqrt{2S}|\eta|_g(v_{\max}+|\eta|_g)^2 - 2ABS.$$

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 $V(0) = \frac{1}{2}g_{ij}v^i(0)v^j(0) \le \frac{1}{2}v^2_{\max}$. At any arbitrary time instance t, it then follows that

$$V(t) = \frac{1}{2}g_{ij}\Delta v^i\Delta v^j + BS \le \frac{1}{2}v^2_{\max},$$

which implies $BS \le \frac{1}{2}v^2_{\max}$. Since $S = \frac{1}{2}d(x,\xi)^2$, by requiring $B > (\frac{v_{\max}}{\rho})^2$ we obtain

$$d(x,\xi) \le \frac{v_{\max}}{\sqrt{B}} < \rho,$$

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.

F.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

$$g_{.,.} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \sin^2 x_1 & 0 \\ 0 & 0 & \sin^2 x_1 \sin^2 x_2 \end{bmatrix}$$

which implicitly gives the distance function, S, as

$$\cos\sqrt{2S} = \cos x_1 \cos\xi_1 + \\ \sin x_1 \sin\xi_1 [\cos x_2 \cos\xi_2 + \cos(x_3 - \xi_3) \sin x_2 \sin\xi_2].$$

The exterior forces, F, are given by $-\mathrm{grad}\, W$, 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 F.4 show the convergence of the observer when the initial data are

$$\begin{cases} x_1(0) = \xi_1(0) = 1 \\ x_2(0) = \xi_2(0) = 0.7 \\ x_3(0) = \xi_3(0) = 2 \end{cases} \quad \begin{cases} v_1(0) = 2.25 \\ v_2(0) = 1.25 \\ v_3(0) = 4 \end{cases} \quad \begin{cases} \hat{v}_1(0) = 0 \\ \hat{v}_2(0) = 0 \\ \hat{v}_3(0) = 0. \end{cases}$$

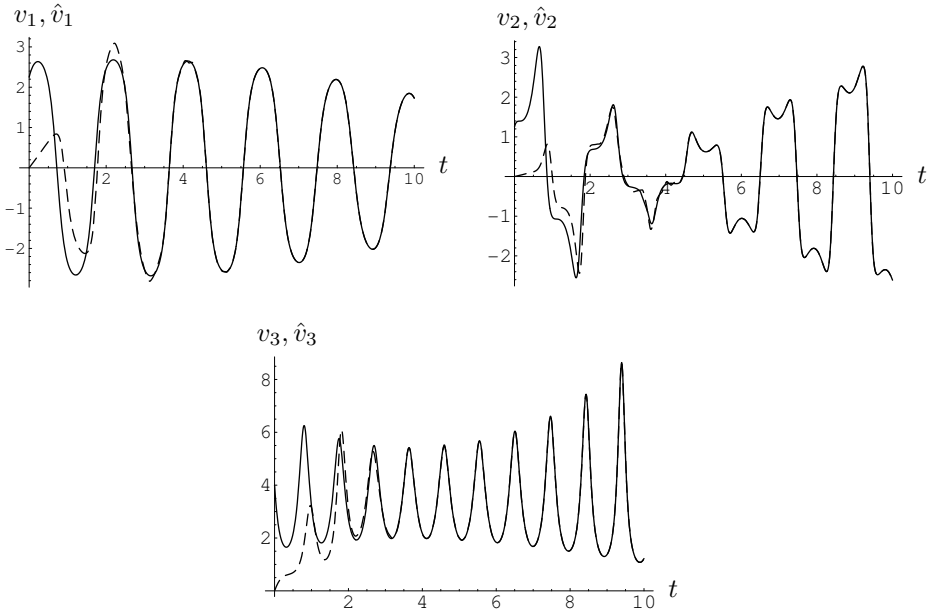


Figure F.4: The solid line refers to the original system, while the dashed line represents the observer.

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).

F.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]).

F.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*, 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 (ACC)*, 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.