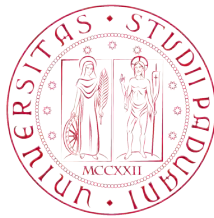


UNIVERSITÀ DEGLI STUDI DI PADOVA

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

DIPARTIMENTO DI INGEGNERIA INDUSTRIALE

Corso di Laurea Magistrale in Ingegneria Aerospaziale



Tesi di Laurea

**OPTIMAL MONITORING
VIA DIFFERENTIAL GAMES**

Dario Paccagnan

Relatore:

Prof. Maria Elena Valcher

Correlatore:

Prof. Alessandro Astolfi - Imperial College London

Prof. Poul Georg Hjorth - Technical University of Denmark

Anno Accademico 2013 - 2014

To my family

*Fool not; for all may have,
If they dare try, a glorious life or grave.*

— George Herbert

Abstract

The problem of continuously monitoring a region using a team of agents (e.g. Unmanned Aerial Vehicles equipped with cameras) is addressed and formulated as a differential game. This allows the use of multiple heterogenous vehicles i.e. agents with different sensor models. Two different approaches are presented to solve the problem. The first one consists in a standard differential game, for which the Hamilton Jacobi Bellman equations provide a sufficient condition. The second formulation results in a non standard differential game that is tackled with two different techniques. The first technique approximates the game as a sequence of infinite horizon optimal control problems. The second one is based on the wavelet decomposition and truncation of the planned trajectory: the differential game is transformed into a non-differentiable optimization problem over \mathbb{R}^p . We present numerical simulations in the case of agents with single integrator dynamics. The results can be exploited to generate a trajectory plan for vehicles with more general dynamics.

Ringraziamenti

Prima di tutto voglio ringraziare la mia famiglia. Senza di loro tutto questo non sarebbe stato possibile. Mi sono stati vicini sia nei momenti felici che in quelli più difficili, mi hanno insegnato a sognare e a non mollare mai. Grazie.

Desidero poi ringraziare i miei relatori: il Professor Poul Hjorth, il Professor Alessandro Astolfi e la Professoressa Maria Elena Valcher. Il Professor Hjorth è stato una guida durante il mio periodo alla DTU fornendo sia consigli tecnici che umani. Il Professor Astolfi mi ha seguito nella preparazione di questa tesi presso l'Imperial College of London. Mi ha fornito sia un interessante argomento che numerosi consigli nella fase di ricerca. Un ringraziamento speciale va alla Professoressa Valcher che ha sempre creduto in me a partire dal momento in cui l'ho conosciuta. La mia carriera universitaria sarebbe stata più difficile senza i suoi consigli, per non parlare del supporto che mi ha dato quando ho presentato domanda per il progetto TIME.

Un ringraziamento speciale a Marco, Alessia e Alessandro, meglio noti come "Little Italy", per i momenti e le esperienze che abbiamo condiviso in giro per il mondo. Ricordatevi che abbiamo un patto da rispettare e non vedo l'ora di ospitarvi a Zurigo.

Un super grazie a Tano per il semestre stupendo che abbiamo passato a Londra e per tutte le cose che abbiamo fatto quest'estate. Non ho trovato solo un fantastico amico, ma anche un ottimo compagno di cordata, con un pizzico di pazzia. Continua così.

Un ulteriore ringraziamento va ad Anna. Sei stata una scoperta inaspettata e una delle persone che mi ha sempre sostenuto negli ultimi due anni.

Per concludere, voglio ringraziare ancora la mia famiglia, questa volta per tutto il supporto che sono sicuro mi offriranno in futuro.

Contents

Abstract	iii
Ringraziamenti	v
1 Introduction	1
2 Optimal Control	5
2.1 Introduction	5
2.2 The Pontryagin Maximum Principle	7
2.3 Dynamic Programming and the Hamilton Jacobi Bellman PDE	12
3 Differential Games	19
3.1 Introduction	19
3.2 Nash and Stackelberg solutions	21
3.2.1 Open loop strategies	21
3.2.2 Closed loop strategies	25
4 Optimal Monitoring via differential game	29
4.1 Introduction	29
4.2 Monitoring using UAVs and related works	32
4.3 Problem formulation	34
4.4 Sensor model and coverage map	35
5 Monitoring as a standard differential game	39
5.1 Standard game formulation	39
5.2 Sufficient conditions	42
5.3 Numerical results	44

6	Monitoring as a non standard differential game	49
6.1	Non standard game formulation	49
6.2	Optimal control approximation	52
6.2.1	Numerical results	55
6.3	Wavelet decomposition	57
6.3.1	Non-smooth Optimization	65
6.3.2	Preliminary results	68
7	Conclusions and future work	71
	Bibliography	75

Introduction

The study and control of multi-agent systems where different subjects mutually interact have been a topic of research in applied mathematics since the beginning of the last century. The development of faster means of communication have recently increased the interest in the field and opened the doors to various applications including automatically controlled vehicles such as Unmanned Aerial Vehicles (UAVs), see an example in figure 1.1. The main advantage of autonomous vehicles comes from the fact that no human being is directly involved in their operation, and thus important but dangerous tasks can be undertaken without jeopardizing human lives [1]. For instance drones equipped with cameras can operate in the event of a natural disaster and retrieve important information from compromised areas, rescuing endangered people.

Due to their versatility and the capability to carry on different payloads, the core application of UAVs is that of aerial surveying and patrolling. Drones have already been used to discover wildfires or for road surveillance [2], not to mention any military activity, where they have been exploited since the Vietnam war [1].

Toward this goal, it is necessary to investigate how UAVs should operate in an arbitrary environment of given shape in order to collect meaningful data. Even without taking into account any real world constraint such as limited amount of fuel or missing data link, the problem is hard to tackle due to the fact that multiple agents need to coordinate their efforts [3]. If furthermore the vehicles are required to perform the task in some *optimal* way such as minimizing the



Figure 1.1: A military Unmanned Aerial Vehicle (“drone”).

mission time, or maximizing the information collected, the problem becomes very tough, and just a few recent literature exists on this [4], [5].

In the present work we analyze the problem of *optimal* monitoring a given region under ideal conditions, i.e. neglecting some of the constraints just mentioned. The main purpose is nevertheless to model and capture the most important features of the problem, including the coordination of multiple agents and avoiding their mutual collision.

The mathematical tools used come from the field of control as well as game theory and the next two chapters help the reader to get acquainted with the main ideas. The concept of coverage map is a main element on which the work is based and provides the foundation to define the criterion we want to optimize. It is first appeared in [5]. Such map describes how well each point of the region has been monitored up to the current time. It depends on the past trajectory of the agents and on how the information is acquired. Data collection is managed introducing a sensor model.

These elements are key to specify what we mean by *optimal monitoring*. The meaning is indeed not clear unless one specifies what ‘optimally’ means in this context. *The search space is monitored by continuously increasing the coverage in the less visited points.* This is the main idea on which the work is based and it first appeared in [4]. A detailed discussion can be found in the third chapter of this thesis.

Two different mathematical formulations are based on the above mentioned idea. The first one consists of a standard differential game as introduced in [4], for which the Hamilton Jacobi Bellman equations (HJB) provide a sufficient condition as presented in chapter four. The problem has been solved numerically,

yielding good results as far as surveying is concerned but not for the case of patrolling. This continues the first new contribution that this thesis provides. The second formulation is the main novelty and results in a non standard differential game that has not been studied previously. Two approximate solutions techniques are presented since no standard methods exists to solve it. The first technique consists in approximating the problem to a series of optimal controls reducing the complexity of the problem, but producing a suboptimal solution. The second one transforms the non standard game into an optimization problem over a finite dimensional coefficient space by selecting a suitable basis for the function space one is interested in. The resulting problem carries on some of the difficulties of the original game, such as the non-differentiability of the objective function, but has the potential to provide solutions that are close to the optimal.

The formulation in terms of non standard game is completely new and provides interesting results when dealing with surveillance, as presented and discussed in the last chapter.

Optimal Control

2.1 Introduction

The modern theory of control dates back to the Fifties when military applications pushed hard the mathematical research. During the Second World War in particular scientists got interested in modifying the evolution of a dynamical system by introducing a controller (a human being, an electronic device, etc.). The laws of nature were not sufficient to take this into account, and a new variable, the *control* u , has been included. Several questions raised at that stage such as stabilizability, optimality and others. For a general introduction see [6]. Optimal control (OC) deals in particular with a dynamical system where the variable u allows to change the dynamics in order to minimize a cost or to maximize a profit. This is the main topic of the chapter, where the most important concepts will be introduced formally and discussed in detail together with the necessary mathematical tools. Optimal control constitutes the foundation for the theory of dynamic games and in particular for differential games. These next pages are intended for the reader to get acquainted with the subject and in particular the techniques that will be used in the following. In case one feels sufficiently familiar with the theory of OC, and in particular with the Pontryagin Maximum principle and the Hamilton Jacobi Bellman equation, he/she can jump directly to the next chapter.

We first present the single agent problem. Subsequently necessary and sufficient conditions for optimality are provided. The section includes discussion of the theoretical results and practical examples.

Let us consider a continuous time dynamical system. The evolution is typically described by a Cauchy problem in the state variable $x \in \mathbb{R}^n$

$$\dot{x} = f(x, t), \quad x(t_0) = x_0,$$

with $f : \mathbb{R}^n \times \mathbb{R} \mapsto \mathbb{R}^n$ and t representing the time. Under mild smoothness conditions on the vector field f , the Picard-Lindelöf theorem [7] guarantees the existence and uniqueness of the trajectory $x(t)$ in a neighborhood of $t = t_0$.

Control theory models and studies the presence of an external agent operating on the system typically to reach a specific goal, e.g. stabilize a certain configuration, steer the system to a specific location, maximize a profit or minimize a cost. The agent modifies the dynamics of the system by means of the so called control function $u(t) : \mathbb{R} \mapsto \mathbb{R}^m$. Thereafter, the mathematical framework is given by a Cauchy problem with the additional presence of the input u

$$\dot{x} = f(x, u, t), \quad x(t_0) = x_0, \quad u(t) \in U. \quad (2.1)$$

Throughout the following we assume that

(H1) The function $u(\cdot)$ is measurable in time and takes values in a compact set U .

The vector field f is differentiable w.r.t. all the variables and has sublinear growth with respect to x , namely

$$\exists C \in \mathbb{R} : \quad |f(x, u, t)| < C(1 + |x|) \quad \text{for all } (x, u, t) \in \mathbb{R}^n \times U \times [t_0, T].$$

The hypothesis (H1) guarantees that for each input u , the Cauchy Problem (2.1) has a unique bounded solution $x(t) \doteq x(t; x_0, u, t_0)$ on the interval of definition $[t_0, T]$, see [8].

OC deals with problems of the form (2.1), where the agent wants to optimize a certain criterion: maximize a profit or minimize a cost. In the continuous case one wants to take into account running costs and terminal cost

$$J(u) = \int_{t_0}^T L(x(t), u(t), t) dt + \psi(x(T)). \quad (2.2)$$

A generic optimal control problem is thus formulated as

$$\begin{cases} \min_{u(t) \in U} \int_{t_0}^T L(x(t), u(t), t) dt + \psi(x(T)), \\ \text{s.t. } \dot{x} = f(x, u, t), \quad x(t_0) = x_0. \end{cases} \quad (2.3)$$

Note that (2.3) is written in terms of minimization of a certain functional. A maximization problem can always be transformed into (2.3) by changing the sign to the original functional. In the following we will always refer to (2.3) with $t_0 = 0$ as the standard formulation.

With no further assumptions (2.3) is difficult to solve, as the optimization is performed on an infinite dimensional space, namely the space of measurable functions $u(t)$. Nevertheless two important tools, the Pontryagin Maximum Principle and the Hamilton Jacobi Bellman PDE, provide necessary and sufficient conditions for $u(t)$ to be optimal.

2.2 The Pontryagin Maximum Principle

The Pontryagin Maximum Principle (PMP) can be thought of as the equivalent of the Lagrange Multiplier Method (LMM) for optimization in an infinite dimensional space. The problem 2.3 is indeed a problem of constraint optimization: one wants to minimize the functional 2.2, subject to the differential constraint 2.1. In analogy to the LMM, we introduce the costate variable $p(t)$, which takes into account the cost for violating the constraint 2.1. Let $u^*(t)$ be optimal for (2.3) and $x^*(t) \doteq x(t; x_0, u^*(t), t_0)$ the corresponding optimal trajectory. The PMP gives a set of necessary conditions for the functions $u^*(\cdot)$ and $x^*(\cdot)$ to satisfy.

THEOREM 2.1 (PMP) *Let $u^*(t)$ and $x^*(t)$ be an optimal control function for (2.3) and the corresponding trajectory. Define $p(t) : \mathbb{R} \mapsto \mathbb{R}^n$ as the solution to the adjoint equation*

$$\dot{p} = -p \cdot \frac{\partial f}{\partial x}(x^*(t), u^*(t), t) - \frac{\partial L}{\partial x}(x^*(t), u^*(t), t) \quad \text{with} \quad p(T) = \frac{\partial \psi}{\partial x}(x^*(T)).$$

The following optimality condition holds at almost every time in $[0, T]$

$$\inf_{u \in U} \{p(t) \cdot f(x^*(t), u, t) + L(x^*(t), u, t)\} = p(t) \cdot f(x^*(t), u^*(t), t) + L(x^*(t), u^*(t), t).$$

□

Different proofs of the theorem are available, for a complete version see [6].

Remarks

- Even if Theorem 2.1 provides just a necessary condition, it suggests a procedure to compute the optimal control:

1. Find the candidate $u^\#$ as a function of t, x, p

$$u^\#(t, x, p) = \arg \inf_{u \in U} \{p \cdot f(x, u, t) + L(x, u, t)\}.$$

Note that the function $u^\# = u^\#(t, x, p)$ may be discontinuous or even multivalued.

2. Solve the boundary value problem

$$\begin{cases} \dot{x} = f(x, u^\#, t), & x(0) = x_0, \\ \dot{p} = -p \cdot \frac{\partial f}{\partial x}(x, u^\#, t) - \frac{\partial L}{\partial x}(x, u^\#, t), & p(T) = \frac{\partial \psi}{\partial x}(x(T)). \end{cases} \quad (2.4)$$

If a solution $(x^\#(t), p^\#(t))$ is found, then $u^\#(t, x^\#(t), p^\#(t))$ is an optimal control candidate as it will naturally satisfy all the conditions of the PMP. Note that existence and uniqueness of the solution for generic BVP is not guaranteed a priori, even if all the functions are smooth.

Following this procedure one obtains an *open loop* control policy $u = u^\#(t, x^\#(t), p^\#(t))$. Thus if the initial condition changes it is necessary to recompute the control.

- If we define the hamiltonian function $H : \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^n \mapsto \mathbb{R}$ to be

$$H(t, x, p) \doteq \inf_{u \in U} \{p \cdot f(x, u, t) + L(x, u, t)\},$$

then the two point boundary value problem (2.4) can be rewritten as an Hamiltonian system in (x, p)

$$\begin{cases} \dot{x} = \frac{\partial H}{\partial p}(t, x, p), & x(0) = x_0, \\ \dot{p} = -\frac{\partial H}{\partial x}(t, x, p), & p(T) = \frac{\partial \psi}{\partial x}(x(T)). \end{cases} \quad (2.5)$$

Other versions of the PMP take into account various constraints on the state of the system $x(t)$. The most common are final constraints of the form $x(T) \in S_T$ or path constraints $x(t) \in S_t$, where S_t and S_T are subsets of the state space. Such formulations are more technical and thus not presented here. For a deeper account see [9].

In general, Theorem 2.1 produces necessary but not sufficient conditions for optimality. Under more specific hypotheses on the Hamiltonian H , the PMP becomes a necessary and sufficient condition as it will be seen in the following.

THEOREM 2.2 (PMP & CONVEXITY \implies OPTIMALITY) Let $u^*(t) \in U$ be measurable, $x^*(t)$ and $p(t)$ be smooth functions satisfying

$$\begin{cases} \dot{x}^* = \frac{\partial H}{\partial p}(t, x^*, p) & x^*(0) = x_0, \\ \dot{p} = -\frac{\partial H}{\partial x}(t, x^*, p) & p(T) = \frac{\partial \psi}{\partial x}(x^*(T)). \end{cases}$$

with $H(t, x, p) \doteq \inf_{u \in U} \{p \cdot f(x, u, t) + L(x, u, t)\} = p \cdot f(x, u^*, t) + L(x, u^*, t)$.

Furthermore assume that the set U is convex and $x \mapsto H(t, x, p(t))$, $x \mapsto \psi(x)$ are convex functions.

Then $u^*(\cdot)$ is an optimal control and $x^*(\cdot)$ the corresponding optimal trajectory.

□

Examples

1. This example is tailored to show that the PMP constitutes *only* a necessary condition. In particular we look at a control problem of the form (2.3), and find a candidate $u^\#(t)$ that satisfies the maximum principle, but is *not* optimal.

Consider the following

$$\begin{cases} \min_{u(t) \in [-1, 1]} \int_0^1 -\frac{x^2}{2} dt, \\ \text{s.t. } \dot{x} = u, \quad x(0) = 0. \end{cases}$$

The objective is to maximize the area under the curve $x^2/2$ with integrator dynamics and control in the compact set $U = [-1, 1]$. The problem satisfies the condition H1. Now the candidate $u^\#(t) = 0$ with the corresponding trajectory $x^\#(t) = 0$ satisfy the PMP. Indeed the adjoint equation is

$$\dot{p} = 0 \quad p(T) = 0 \implies p(t) = 0,$$

and the optimality condition

$$\inf_{u \in U} \{p(t) \cdot f(x^*(t), u, t) + L(x^*(t), u, t)\} = p(t) \cdot f(x^*(t), u^*(t), t) + L(x^*(t), u^*(t), t)$$

is satisfied as

$$\inf_{u \in [-1, 1]} \{0 \cdot u + 0\} = 0.$$

Nevertheless the control $u^\#(t) = 0$ is far from being optimal. In fact it gives a cost of $J^\# = \int_0^1 -\frac{x^2}{2} dt = 0$, while for instance the control $u^{\#\#}(t) = 1/2$ gives $J^{\#\#} = \int_0^1 -\frac{(t/2)^2}{2} dt = -1/24 < J^\#$.

One easily notices that an optimal control policy is the one that takes $x(t)$ as far as possible from the origin at each fixed t , i.e. $u^*(t) = \pm 1$ with a cost of $J^* = -1/6 < J^{\#\#} < J^\#$.

2. This example shows how to use Theorem 2.2 (PMP with convexity hypothesis) to find an optimal control policy.

Consider a system with linear dynamics and quadratics costs

$$\begin{cases} \min_{|u(t)| \leq 10} \int_0^1 \left(\frac{x^2}{2} + \frac{u^2}{2} \right) dt + x^2(1), \\ \text{s.t. } \dot{x} = x + u, \quad x(0) = 1, \end{cases} \quad (2.6)$$

and look for an optimal control policy following the procedure described in the observations to Theorem 2.1. The control candidate is found as

$$u^\#(t, x, p) = \arg \inf_{u \in U} \{p \cdot f(x, u, t) + L(x, u, t)\} = \arg \inf_{|u| \leq 10} \left\{ p \cdot (x+u) + \frac{x^2}{2} + \frac{u^2}{2} \right\}.$$

The minimum is attained at $u^\#(t, x, p) = -p(t)$ if $|p(t)| < 10$ for all $t \in [0, 1]$. Otherwise it lies on the boundary of U .

Let us consider the first case and work out the computations starting from the Hamiltonian

$$H(t, x, p) = \{p \cdot f(x, u^\#, t) + L(x, u^\#, t)\} = \frac{x^2}{2} + px - \frac{p^2}{2}.$$

The equations (2.5) become

$$\begin{cases} \dot{x} = x - p, & x(0) = 1, \\ \dot{p} = -x - p, & p(1) = x(1), \end{cases} \quad (2.7)$$

whose solution can be found using the matrix exponential as

$$\begin{pmatrix} x \\ p \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} \sqrt{2} \cosh \sqrt{2}t + \sinh \sqrt{2}t & -\sinh \sqrt{2}t \\ -\sinh \sqrt{2}t & \sqrt{2} \cosh \sqrt{2}t - \sinh \sqrt{2}t \end{pmatrix} \begin{pmatrix} x_0 \\ p_0 \end{pmatrix}.$$

The final constraint on the costate $p(1) = x(1)$ allows to determine the initial condition p_0 , given $x_0 = 1$, namely

$$p_0 = \sqrt{2} \tanh \sqrt{2} + 1.$$

Thus we can determine $p(t)$ and the candidate control $u^\#(t) = -p(t)$

$$u^\#(t) = \frac{1}{\sqrt{2}} \sinh \sqrt{2}t - (\cosh \sqrt{2}t - \frac{1}{\sqrt{2}} \sinh \sqrt{2}t)(1 + \sqrt{2} \tanh \sqrt{2}). \quad (2.8)$$

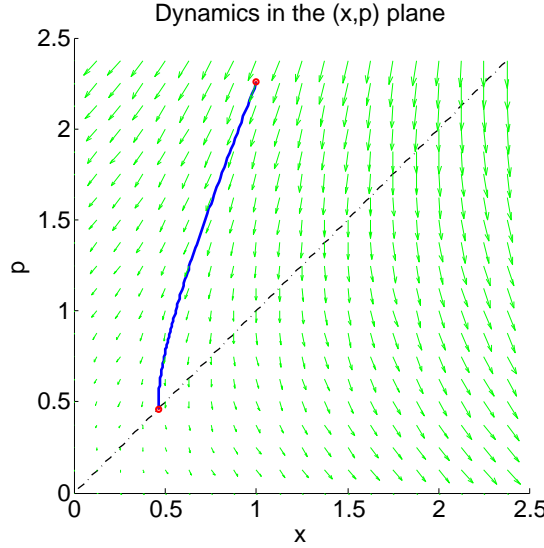


Figure 2.1: The figure presents the solution to the problem (2.6). The optimal trajectory (blue curve) follows the vector field (2.7) (green arrows), satisfies the initial condition $x(0) = 1$ and the terminal constraint $x = p$ (dashed line).

One can easily check that for $t \in [0, 1]$ the constraint on u is satisfied, namely $|u^\#(t)| \leq 10$.

By construction $u^\#(t)$ satisfies the hypothesis of the maximum principle, furthermore the Hamiltonian $H(t, x, p(t))$ and the terminal cost $\psi(x) = x^2/2$ are convex in x . From Theorem 2.2 we conclude that $u^\#(t)$ given in (2.8) is optimal. The corresponding trajectory is given by

$$\begin{aligned} x^\#(t) &= \cosh \sqrt{2}t + \frac{1}{\sqrt{2}} \sinh \sqrt{2}t - \frac{1}{\sqrt{2}} \sinh \sqrt{2}t(1 + \sqrt{2} \tanh \sqrt{2}) \\ &= \frac{\cosh(\sqrt{2}(t-1))}{\cosh \sqrt{2}}. \end{aligned}$$

In Figure 2.1 the evolution of the system is presented in the (x, p) plane, note how the condition $x(0) = 1$ and the terminal condition $x(1) = p(1)$ are satisfied. \square

2.3 Dynamic Programming and the Hamilton Jacobi Bellman PDE

As seen in the previous section, if the Hamiltonian and the final cost are convex in the state x , then the Pontryagin Maximum Principle is a sufficient condition for optimality. Such assumptions are far too restrictive as in several applications they are not satisfied. In this section we introduce the Hamilton Jacobi Bellman PDE (HJB), which constitutes a sufficient condition for optimality, regardless of the structure of H and ψ .

So far we studied an optimal control problem with fixed initial condition $x(t_0) = x_0$. One can consider a family of control problems, where the initial condition is variable.

Let $V(t_0, x_0)$ be the optimal cost corresponding to the initial conditions $(t_0, x_0) \in [0, T] \times \mathbb{R}^n$, i.e.

$$V(t_0, x_0) \doteq \inf_{u: [t_0, T] \rightarrow U} J(u; t_0, x_0), \quad (2.9)$$

where J is the functional in (2.2) and x is governed by (2.1) with initial conditions (t_0, x_0) . The function $V : [0, T] \times \mathbb{R}^n \mapsto \mathbb{R}$ is called *value function* and it represents the cost-to-go starting from $x = x_0$ at $t = t_0$.

When the initial condition is assigned at time $t_0 = T$ (i.e. at the end of the interval) there is no running cost contribution from L , but just final cost ψ , i.e.

$$V(T, x) = \psi(x) \quad \forall x \in \mathbb{R}^n.$$

We now introduce the dynamic programming principle. It describes an important property of the Value function, based on which the Hamilton Jacobi Bellman PDE will be derived in the following.

THEOREM 2.3 (PRINCIPLE OF DYNAMIC PROGRAMMING) *For each $x_0 \in \mathbb{R}^n$ and $0 < t_0 \leq t_1 < T$, the value function satisfies*

$$V(t_0, x_0) = \inf_{u: [t_0, t_1] \rightarrow U} \left\{ V(t_1, x(t_1; t_0, x_0, u)) + \int_{t_0}^{t_1} L(x(t; t_0, x_0, u), u(t), t) dt \right\}.$$

□

For a detailed proof see [10].

Remarks

- The Dynamic Programming Principle can be read as in the following:
Let us solve the problem (2.3) with initial conditions (t_0, x_0) on the whole

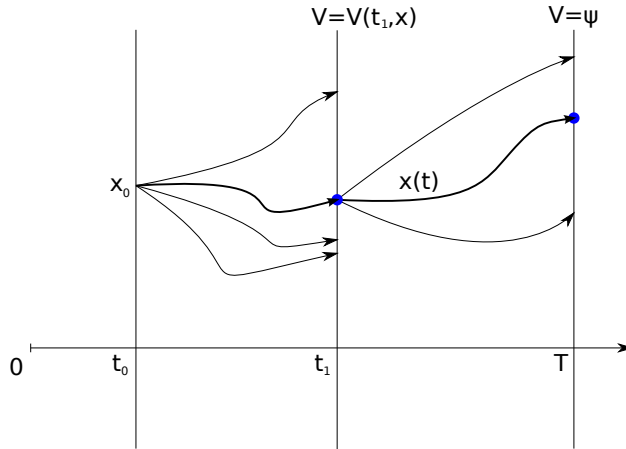


Figure 2.2: Illustration of the dynamic programming principle: the optimal control problem with initial conditions (t_0, x_0) can be split into two subproblems: one on the interval $[t_1, T]$ with final cost $\psi(\cdot)$ and one on the interval $[t_0, t_1]$ with terminal cost $V(t_1, \cdot)$.

interval $[t_0, T]$ and let us call $u^*(t)$ and $x^*(t)$ a pair of optimal control and optimal trajectory.

The minimum cost from (t_0, x_0) is equal to the running cost from t_0 to $t_1 \geq t_0$, following an optimal trajectory $x^*(t)$, plus the optimal cost to go from $(t_1, x^*(t_1))$, namely

$$V(t_0, x_0) = \int_{t_0}^{t_1} L(x^*(t), u^*(t), t) dt + V(t_1, x^*(t_1)).$$

- The Dynamic Programming Principle suggests a procedure to compute backwards the value function by splitting the optimization into smaller problems.
 1. On the interval $[t_1, T]$ let us solve the family of optimal control problems with variable initial condition $x_1 \in \mathbb{R}^n$, running cost L and terminal cost ψ . In this way we obtain $V(t_1, \cdot)$.
 2. On the interval $[t_0, t_1]$ let us solve the family of optimal control problems with initial condition $x_0 \in \mathbb{R}^n$, running cost L and terminal cost $V(t_1, \cdot)$ found in the previous step.

One could apply the same procedure, for instance to find $V(t_1, \cdot)$, splitting again the interval $[t_1, T]$ into $[t_1, t_2] \cup [t_2, T]$.

By taking this process to the limit, i.e. splitting the optimization into problems on an “infinitesimal” time horizon, one gets the Hamilton Jacobi Bellman partial differential equation. In the following V_t and ∇V represents respectively the time derivative and the spatial gradient of the value function V .

THEOREM 2.4 (HAMILTON JACOBI BELLMAN PDE) *Consider the optimal control problem (2.3) and assume the hypothesis H1 is satisfied. Define the Hamiltonian function as*

$$H(t, x, p) = \min_{u \in U} \{p \cdot f(x, u, t) + L(x, u, t)\}.$$

1. *If $V(t, x)$ is the value function of (2.3), then on the region Ω where V is differentiable*

$$V_t + H(t, x, \nabla V) = 0, \quad V(T, x) = \psi(x). \quad (2.10)$$

2. *Vice versa, if a differentiable function $W(t, x)$ satisfies (2.10) on a region Ω , then $W(t, x)$ is the value function for (2.3) in Ω .*

PROOF. (We show only the point 1, for a complete proof see [6].)

By contradiction, let us suppose that there exists a point $(t_0, x_0) \in \Omega$ such that (2.10) does not hold. For instance let us assume that

$$V_t(t_0, x_0) + \min_{u \in U} \{\nabla V(t_0, x_0) \cdot f(x_0, u, t_0) + L(x_0, u, t_0)\} > 0,$$

and take $u^* \in \arg \min_{u \in U} \{\nabla V(t_0, x_0) \cdot f(x_0, u, t_0) + L(x_0, u, t_0)\}$. For continuity there exists a whole neighborhood $B(t_0, x_0) \in \Omega$ such that

$$V_t(t, x) + \nabla V(t, x) \cdot f(x, u^*, t) + L(x, u^*, t) > 0 \quad \forall (t, x) \in B(t_0, x_0).$$

Note that

$$\frac{dV}{dt}(t, x) = V_t(t, x) + \nabla V(t, x) \cdot f(x, u^*, t) > -L(x, u^*, t) \quad \forall (t, x) \in B(t_0, x_0).$$

By applying the constant control $u^*(t) = u^*$ on a sufficiently small time interval $[t_0, t_1]$ such that the trajectory $x^*(t) = x(t; t_0, x_0, u^*)$ stays in $B(t_0, x_0)$, one has

$$\begin{aligned} V(t_1, x^*(t_1)) &= V(t_0, x_0) + \int_{t_0}^{t_1} V_t(t, x^*(t)) + \nabla V(t, x^*(t)) \cdot f(x^*(t), u^*(t), t) dt \\ &> V(t_0, x_0) - \int_{t_0}^{t_1} L(x^*(t), u^*(t), t) dt. \end{aligned}$$

This implies that

$$V(t_0, x_0) < V(t_1, x^*(t_1)) + \int_{t_0}^{t_1} L(x^*(t), u^*(t), t) dt,$$

contradicting the Principle of Dynamic Programming for the value function $V(t, x)$.

In a similar fashion one shows that if

$$V_t(t_0, x_0) + \min_{u \in U} \{ \nabla V(t_0, x_0) \cdot f(x_0, u, t_0) + L(x_0, u, t_0) \} < 0,$$

the Dynamic Programming principle is not satisfied either.

This proves the point 1. □

Remarks

- As a by product of the proof of point 2 not presented here, one obtains a formula to compute an optimal control policy. A map $u^* : (t, x) \mapsto u^*(t, x)$ is an optimal control if

$$u^*(t, x) \in \arg \min_{u \in U} \{ \nabla V(t, x) \cdot f(x, u, t) + L(x, u, t) \}.$$

In particular if for each point (t, x) the argument of minimum is unique, then the optimal control is unique and the map $u^* : (t, x) \mapsto u^*(t, x)$ is the function

$$u^*(t, x) = \arg \min_{u \in U} \{ \nabla V(t, x) \cdot f(x, u, t) + L(x, u, t) \}.$$

Note that in order to compute the optimal policy one needs to compute the gradient of the value function.

- Equation (2.10) is called the Hamilton Jacobi Bellman partial differential equation and has connections with several fields of mathematics and physics where the optimization of a functional plays a role, for instance in calculus of variations, optics, lagrangian mechanics and more. The PDE can be thought as a backward equation in time with final boundary condition

$$V_t = -H(t, x, \nabla V), \quad V(T, x) = \psi(x),$$

carrying on the idea of dynamic programming, where the optimal control problem can split into subproblems going backward in time.

- If the function V is not differentiable, one can still prove that (2.10) has to be satisfied, in the so called *viscosity sense*. For a detailed description of the theory, see [11]. In this seminal paper Crandall, Ishii and Lions prove two key elements: existence and uniqueness of the solution to (2.10) in the viscosity sense.
- Following this procedure one obtains a *feedback* control policy $u = u^*(t, x)$. Thus even if the initial condition changes it is not necessary to recompute the solution, but just to evaluate $u^*(t, x)$ in a different point. On the other hand the PMP has the advantage that the conditions arising are a set of ordinary differential equations, that need to be satisfied only on the specific trajectory. The PMP produces sufficient conditions only under restrictive convexity conditions, while the scope of the HJB PDE is more general.

Examples

1. We revise the example studied in (2.6) and try to find a solution using the HJB equation. This will throw some light on the similarities and differences of the two methods.

Given

$$\begin{cases} \min_{|u(t)| \leq 10} \int_0^1 \frac{x^2}{2} + \frac{u^2}{2} dt + x^2(T), \\ \text{s.t. } \dot{x} = x + u, \quad x(0) = 1, \end{cases}$$

the Hamiltonian function is

$$H(t, x, p) = \inf_{|u| \leq 10} \left\{ p \cdot (x + u) + \left(\frac{x^2}{2} + \frac{u^2}{2} \right) \right\}.$$

The minimum is attained at $u = -p$ as long as $|p(t)| < 10$, otherwise it lies on the boundary of U .

Let us consider the first case and substitute $u = -p$ to get the explicit Hamiltonian

$$H(t, x, p) = \frac{x^2}{2} + px - \frac{p^2}{2},$$

that produces the following HJB

$$V_t + \frac{x^2}{2} + V_x \left(1 - \frac{V_x}{2} \right) = 0, \quad V(1, x) = \frac{x^2}{2}.$$

We solve the previous equation defining a new time $\tau = 1 - t$ so that

$$V_\tau - \frac{x^2}{2} - V_x \left(1 - \frac{V_x}{2} \right) = 0, \quad V(0, x) = \frac{x^2}{2}.$$

It is well known (see for instance [12]) that for linear quadratic problems the value function has quadratic structure in x , so assume that

$$V(\tau, x) = a(\tau) \frac{x^2}{2},$$

and insert this in the HJB equation. One can simplify the term x^2 if $x \neq 0$ and get an ordinary differential equation

$$\dot{a} = -a^2 + 2a + 1, \quad a(0) = 1.$$

After a little bit of calculus the solution is found to be

$$a(\tau) = \sqrt{2} \tanh(\sqrt{2}\tau) + 1 \implies a(t) = \sqrt{2} \tanh[\sqrt{2}(1-t)] + 1.$$

The value function for $x \neq 0$ is thus

$$V(t, x) = (\sqrt{2} \tanh(\sqrt{2}(1-t)) + 1) \frac{x^2}{2},$$

and the optimal control

$$u(t) = -p = -\frac{\partial V}{\partial x} = -(\sqrt{2} \tanh \sqrt{2}(1-t) + 1)x.$$

Note the feedback structure, i.e. the fact that $u(t)$ depends on the current state $x(t)$ of the system.

When using the optimal control, the system dynamics becomes

$$\dot{x} = -\sqrt{2} \tanh(\sqrt{2}(1-t))x, \quad x(0) = 1,$$

whose trajectories are exactly those found with the Pontryagin maximum principle in the previous section

$$x(t) = \frac{\cosh(\sqrt{2}(t-1))}{\cosh \sqrt{2}}.$$

To conclude we verify the condition $|u| \leq 10$, noting that

$$|u| = |\sqrt{2} \tanh(\sqrt{2}(1-t)) + 1||x| < \sqrt{2}|x| < \sqrt{2}|x_0| = \sqrt{2},$$

where the first inequality holds because $|\tanh s| < 1$ and the second because $x(t)$ is monotonically decreasing.

With this method we have found the optimal control not just for $x_0 = 1$, but for all possible initial conditions x_0 , given that $x_0 < \frac{10}{\sqrt{2}}$, i.e. when the constraint $|u| \leq 10$ is satisfied.

This is known as linear quadratic control problem as the dynamics is linear in the state x and the control u , and the cost is quadratic. Such problems can be solved in higher dimension using the HJB equation; in that case one would get the famous Riccati Matrix equations. □

Differential Games

3.1 Introduction

We present here some of the main elements in the theory of differential games. From an historical and scientific point of view, dynamic game theory lies at the intersection of optimal control and game theory. The latter initially developed by the pioneering work of Von Neumann and Morgensten [13] and was mainly concerned with the study of multi-agents decision making within the framework of finite games. On the other hand control theory developed quickly during the Second World War thanks to the contribution of several important scientists such as Bellman [14] and Pontryagin [15]. At that time control theory was mainly focused on the study of dynamical systems and their optimization.

The main feature of dynamic game theory is the modelization of conflicting situations amongst multiple agents. While in optimal control a *single* actor is trying to minimize a certain cost (see problem (2.3)), here multiple *players* are involved. Each of them is trying to optimize his personal payoff, which in turn depends also on the other players choice. The evolution of the system is governed by an ordinary differential equation.

It is important to get acquainted with the language and the tools, as in the next chapter they will be widely used. First the differential game setup is presented together with the two main definitions of solution. The difference between open loop and closed loop strategies is made clear, and necessary or sufficient

conditions are presented. Examples are also discussed.

Let us start with a two players game and identify with $u_1(t)$ and $u_2(t)$ their control functions. A two agent differential game can be written as

$$P_1 : \begin{cases} \min_{u_1(t) \in U_1} J_1(u_1, u_2), \\ P_2 : \begin{cases} \min_{u_2(t) \in U_2} J_2(u_1, u_2), \\ \text{s.t. } \dot{x} = f(x, u_1, u_2, t), \quad x(0) = x_0, \end{cases} \end{cases} \quad (3.1)$$

where

$$J_i = \int_0^T L_i(x(t), u_1(t), u_2(t), t) dt + \psi_i(x(T)), \quad i = 1, 2.$$

Player number one is trying to minimize his own functional that depends on his decision $u_1(t)$ and also on the other player choice $u_2(t)$. The control of the second player influences the payoff of the first player both directly via $L_1(\cdot, \cdot, u_2(t), \cdot)$ and indirectly via the evolution of the system state x . The same holds with complete symmetry for the other player.

Example (duopolistic competition)

Two companies compete for market share as they both sell the same product. Let us call $x_1(t) = x(t) \in [0, 1]$ the market share of the first company and $x_2(t) = 1 - x(t)$ the market share of the second one. Each company advertises its product with effort $u_i(t)$ so that the Lanchester model [16] describes the dynamics as

$$\dot{x} = (1 - x)u_1 - xu_2, \quad x(0) = x_0 \in [0, 1].$$

This means that the share of the first company increases due to the advertisement effort u_1 , with a coefficient proportional to the share of the second company. On the other hand x decreases due to u_2 with a coefficient proportional to its own share.

Each of the players wants to maximize the revenue and thus the market share, but the more advertisement the effort, the higher the cost they incur. Consequently each of the companies wants to maximize (minimize the opposite)

$$J_i(u_1, u_2) = \int_0^T \left(a_i x_i(t) - c_i \frac{u_i^2(t)}{2} \right) dt + S_i x_i(T),$$

with constants $a_i, c_i, S_i > 0$.

What is the best advertisement policy for each company to follow?

The problem will be revised and tackled in the following, after discussing some of the solution concepts. \square

Note that the differential game presented so far included just two players, but the formulation can be easily extended to $M \geq 2$ players

$$\begin{array}{l}
 P_1 : \\
 \vdots \\
 P_M :
 \end{array}
 \left\{ \begin{array}{l}
 \min_{u_1(t) \in U_1} \int_0^T L_1(x(t), u_1(t), \dots, u_M(t), t) dt + \psi_1(x(T)), \\
 \vdots \\
 \min_{u_M(t) \in U_M} \int_0^T L_M(x(t), u_1(t), \dots, u_M(t), t) dt + \psi_M(x(T)), \\
 \text{s.t. } \dot{x} = f(x, u_1, \dots, u_M, t), \quad x(0) = x_0.
 \end{array} \right.$$

For the sake of simplicity in the following we will present the main concepts for two players game, but they can be easily extended to the multiplayer case.

3.2 Nash and Stackelberg solutions

The first difficulty arises when trying to define what is an ‘optimal solution’ for the problem (3.1). In general it is impossible to find controls $u_1(t)$ and $u_2(t)$ that minimize both cost functionals at the same time. Indeed an outcome that is optimal for one player, can be worse off for the other. Furthermore, if we keep in mind the main feature of differential games, i.e. the modelization of conflicts, this will hardly be the case. Several notions of solution have been introduced (see [17]), here we are going to deal with the most used ones: *Nash Solution* and *Stackelberg solution*.

While for optimal control problems we did not dwell on the differences between open loop and closed loop solutions, for differential games it is necessary to do so as the concepts are subtly different, in particular for the Stackelberg case. In the following we will introduce first open loop strategies and subsequently closed loop solutions. Note that the terms *solution* and *equilibrium* will be used with the same meaning.

3.2.1 Open loop strategies

Open loop solutions to differential games are strategies that depend on the time and implicitly on the initial condition $x(0) = x_0$, even if this is not made clear when writing as it is commonly done $u_i(t)$. We first give the definition of open loop Nash equilibrium and apply the PMP theorem to obtain necessary conditions. Finally we introduce the concept of open loop Stackelberg solution.

DEFINITION 3.1 (OPEN-LOOP NASH SOLUTION) *A pair of control functions $(u_1^*(t), u_2^*(t))$ is a Nash equilibrium for (3.1) if the following holds:*

- i) *The control $u_1^*(t)$ is a solution to the optimal control problem for the first player with fixed $u_2(t) = u_2^*(t)$, i.e.*

$$\begin{cases} \min_{u_1(t) \in U_1} \int_0^T L_1(x(t), u_1(t), u_2^*(t), t) dt + \psi_1(x(T)), \\ \text{s.t. } \dot{x} = f(x, u_1, u_2^*, t), \quad x(0) = x_0. \end{cases}$$

- ii) *The control $u_2^*(t)$ is a solution to the optimal control problem for the second player with fixed $u_1(t) = u_1^*(t)$, i.e.*

$$\begin{cases} \min_{u_2(t) \in U_2} \int_0^T L_2(x(t), u_1^*(t), u_2(t), t) dt + \psi_2(x(T)), \\ \text{s.t. } \dot{x} = f(x, u_1^*, u_2, t), \quad x(0) = x_0. \end{cases}$$

This means that neither of the players can improve his situation by changing unilaterally his own strategy, as long as the other player sticks to the equilibrium solution. The Nash equilibrium concept was first introduced by A. Cournot in [18] while discussing the theory of oligopoly, but is named after J. Nash who proved the existence of at least one such equilibrium in a more specific case [19]. The main feature of this solution concept is that it describes a *competitive* game in a scenario where the players are not allowed to exchange information to get a better payoff.

For the first time it should be clear why we have spent some effort in the previous section giving an introduction to optimal control problems: the definition of Nash Solution itself poses a two player differential game as a coupled optimal control problem. The techniques we have seen for ‘single’ player problems (i.e. Pontryagin maximum principle and HJB) will be widely used to formulate necessary and sufficient condition for Nash optimality.

Finding Nash Equilibrium candidates using the PMP

Based on the PMP we present a procedure to compute a candidate Nash Solution to (3.1). To do so let us assume that there exist unique controls $u_1^\#$ and $u_2^\#$ that minimize the corresponding pre-Hamiltonian

$$\begin{aligned} u_1^\#(t, x, p_1, p_2) &= \arg \inf_{u_1 \in U_1} \{p_1 \cdot f(x, u_1, u_2^\#, t) + L_1(x, u_1, u_2^\#, t)\}, \\ u_2^\#(t, x, p_1, p_2) &= \arg \inf_{u_2 \in U_2} \{p_2 \cdot f(x, u_1^\#, u_2, t) + L_2(x, u_1^\#, u_2, t)\}. \end{aligned} \quad (3.2)$$

Consequently the Pontryagin Maximum Principle (see Theorem 2.1) gives the following set of necessary conditions

$$\begin{cases} \dot{x} = f(x, u_1^\#, u_2^\#, t), & x(0) = x_0, \\ \dot{p}_1 = -p_1 \cdot \frac{\partial f}{\partial x}(x, u_1^\#, u_2^\#, t) - \frac{\partial L_1}{\partial x}(x, u_1^\#, u_2^\#, t), & p_1(T) = \frac{\partial \psi_1}{\partial x}(x(T)), \\ \dot{p}_2 = -p_2 \cdot \frac{\partial f}{\partial x}(x, u_1^\#, u_2^\#, t) - \frac{\partial L_2}{\partial x}(x, u_1^\#, u_2^\#, t), & p_2(T) = \frac{\partial \psi_2}{\partial x}(x(T)), \end{cases}$$

with initial condition on the state and final condition on the adjoint variables. Note that the previous system does not constitute a Cauchy Problem, but is a mixed boundary value problem. The initial condition is given on the state, but final conditions are specified for p_1 and p_2 . If the problem has a solution, then the candidate controls are found as

$$u_1^\#(t) = u_1^\#(t, x(t), p_1(t), p_2(t)), \quad u_2^\#(t) = u_2^\#(t, x(t), p_1(t), p_2(t)).$$

Recall that the PMP is a necessary condition for optimality, thus even if one finds solutions to the previous BVP, we cannot conclude that they are optimal. Sufficient conditions for optimality via HJB will be presented in the Closed-Loop section.

Example (Open-Loop Nash Equilibrium)

We are now in the position to write a necessary condition in order to find the best advertisement policy for the duopolistic competition model previously presented.

1. Following the procedure introduced above, let us start by computing the optimal control as

$$u_1^\#(t, x, p_1, p_2) = \arg \inf_{u_1 \in U_1} \{p_1 \cdot f(x, u_1, u_2^\#, t) + L_1(x, u_1, u_2^\#, t)\} = (1-x) \frac{p_1}{c_1},$$

$$u_2^\#(t, x, p_1, p_2) = \arg \inf_{u_2 \in U_2} \{p_2 \cdot f(x, u_1^\#, u_2, t) + L_2(x, u_1^\#, u_2, t)\} = \frac{p_2}{c_2}.$$

2. The BVP given by the Pontryagin Maximum principle is

$$\begin{cases} \dot{x} = (1-x)u_1^\# + xu_2^\# = (1-x)^2 \frac{p_1}{c_1} + x^2 \frac{p_2}{c_2}, & x(0) = x_0, \\ \dot{p}_1 = -p_1(u_1^\# + u_2^\#) - a_1 = -p_1 \left((1-x) \frac{p_1}{c_1} + x \frac{p_2}{c_2} \right) - a_1, & p_1(T) = S_1, \\ \dot{p}_2 = -p_2(u_1^\# + u_2^\#) - a_2 = -p_2 \left((1-x) \frac{p_1}{c_1} + x \frac{p_2}{c_2} \right) - a_2, & p_2(T) = S_2. \end{cases}$$

□

Open Loop Stackelberg Equilibrium

The *Stackelberg equilibrium* describes a situation in which the symmetry is lost. The first player, called *leader*, chooses his strategy $u_1(t)$ and communicates it to the second player, called *follower*, before the beginning of the game. Player number one enforces his decision, i.e. is in the position to stick to what he has previously announced. Player number two receives such a piece of information, but can only choose $u_2(t)$ in order to minimize his own cost, given the strategy $u_1(t)$.

The leader though can predict what the follower will choose when he announces the strategy $u_1(t)$. Thus he will choose the control $u_1(t)$ that together with the best reply of the follower $u_2(t)$ will give him the smallest possible cost.

The game is asymmetric because the leader optimizes ‘twice’ his cost: once because he can choose the strategy $u_1(t)$ and the second time because he knows what the second agent will play in response to his strategy. The solution takes the name from the german economist H. Von Stackelberg who first described the scenario in [20]. The model is widely used in economics and firms competition study.

Before giving the formal definition of open loop Stackelberg equilibrium, we need to introduce the concept of best reply set.

Best reply set: Given an admissible control $u_1^\# : [0, T] \mapsto U_1$, the best reply set $\mathcal{R}_2(u_1^\#)$ is the set of all admissible functions $u_2 : [0, T] \mapsto U_2$ such that they minimize the cost for the second player corresponding to $u_1^\#(t)$. This means that $u_2 \in \mathcal{R}_2(u_1^\#)$ if it solves the problem

$$\begin{cases} \min_{u_2(t) \in U_2} \int_0^T L_2(x(t), u_1^\#(t), u_2(t), t) dt + \psi_2(x(T)), \\ \text{s.t. } \dot{x} = f(x, u_1^\#, u_2, t), \quad x(0) = x_0. \end{cases}$$

DEFINITION 3.2 (OPEN-LOOP STACKELBERG SOLUTION) *A pair of control functions $u_1^*(t)$, $u_2^*(t)$ is a Stackelberg equilibrium for (3.1) if the following holds:*

i) $u_2^* \in \mathcal{R}_2(u_1^*)$,

ii) *Among all the possible controls u_1 and best replies $u_2 \in \mathcal{R}_2(u_1)$, one has*

$$J_1(u_1^*, u_2^*) \leq J_1(u_1, u_2).$$

In order to find a Stackelberg equilibrium, the leader has to calculate the best reply of the follower for each of his admissible controls u_1 , i.e. to construct

$\mathcal{R}_2(u_1)$. He will then choose the control function u_1^* that minimizes his own cost.

Note that with the definition we have given, we are taking the optimistic view that, if the second player has several best replies to a strategy u_1^* , he will choose the most favorable for player one. The problem of nonuniqueness of the best reply is quite difficult and will not be discussed here. In the following we will always assume uniqueness of the best reply if not differently mentioned.

3.2.2 Closed loop strategies

Closed loop solutions to differential games are strategies that depend on the time t and on the state of the system x . They are typically expressed as $u_i(t, x)$ and can be implemented only if all the players can observe the current state of the system $x(t)$. While open loop strategies strongly depend on the initial condition x_0 , in the closed loop case, it is natural to ask for the strategy to be optimal regardless of the initial condition $x(\tau) = y$ with $(\tau, y) \in [0, T] \times \mathbb{R}^n$. This is the essential difference with respect to open loop controls. We first give the definition of closed loop Nash equilibrium and apply the HJB theorem to obtain sufficient conditions for optimality. Subsequently we do the same with the more delicate concept of closed loop Stackelberg solution.

DEFINITION 3.3 (CLOSED-LOOP NASH SOLUTION) *A pair of control functions $u_1^*(t, x)$, $u_2^*(t, x)$ is a Closed-Loop Nash equilibrium for (3.1) if the following hold:*

- i) *The control $u_1^*(t, x)$ is a closed loop solution to the optimal control problem for the first player with fixed $u_2(t, x) = u_2^*(t, x)$ and generic initial condition $x(\tau) = y$, with $(\tau, y) \in [0, T] \times \mathbb{R}^n$:*

$$\left\{ \begin{array}{l} \min_{u_1(t) \in U_1} \int_0^T L_1(x(t), u_1(t, x), u_2^*(t, x), t) dt + \psi_1(x(T)), \\ \text{s.t. } \dot{x} = f(x, u_1, u_2^*, t), \quad x(\tau) = y. \end{array} \right.$$

- ii) *The control $u_2^*(t, x)$ is a closed loop solution to the optimal control problem for the second player with fixed $u_1(t, x) = u_1^*(t, x)$ and generic initial condition $x(\tau) = y$, with $(\tau, y) \in [0, T] \times \mathbb{R}^n$:*

$$\left\{ \begin{array}{l} \min_{u_2(t) \in U_2} \int_0^T L_2(x(t), u_1^*(t, x), u_2(t, x), t) dt + \psi_2(x(T)), \\ \text{s.t. } \dot{x} = f(x, u_1^*, u_2, t), \quad x(\tau) = y. \end{array} \right.$$

Sufficient condition for Nash Equilibrium using the HJB equations

As the definition of Closed-Loop Nash equilibrium is described in terms of a coupled optimal control problems, the solution will naturally satisfy the Dynamic Programming Principle. In particular let x^* be the trajectory corresponding to the Nash Equilibrium and u_1^* , u_2^* be the optimal controls. We can define one value function per each player as the minimum cost-to-go they incur

$$V_i(\tau, y) \doteq J_i(u_1^*, u_2^*; \tau, y) = \int_{\tau}^T L_i(x^*(t), u_1^*(t), u_2^*(t), t) dt + \psi_i(x(T)),$$

where the trajectory $x^*(t)$ propagates from (τ, y) following the ODE

$$\dot{x}^* = f(x^*, u_1^*, u_2^*, t), \quad x^*(\tau) = y.$$

As consequence of Theorem 2.4, each value function would satisfy a system of coupled HJB equations

$$\begin{cases} V_{1t} = -\nabla V_1 \cdot f(t, x, u_1^*, u_2^*) - L_1(x^*(t), u_1^*(t), u_2^*(t), t), \\ V_{2t} = -\nabla V_2 \cdot f(t, x, u_1^*, u_2^*) - L_2(x^*(t), u_1^*(t), u_2^*(t), t), \end{cases} \quad (3.3)$$

where the controls are computed as in (3.2) and evaluated in $p_i = \nabla V_i$, i.e.

$$u_i^* = u_i^\#(t, x, \nabla V_1, \nabla V_2).$$

The system constitutes a PDE initial value problem as it is completed with the terminal condition on the value functions

$$V_1(T, x) = \psi_1(x), \quad V_2(T, x) = \psi_2(x).$$

As shown in [21], the system is in general *not* hyperbolic, and the linearized Cauchy problem is ill posed both backward and forward in time¹. This means that small perturbations on the initial condition can propagate and become ‘large’ in finite time.

Closed Loop Stackelberg Equilibrium

Giving a well posed and solid definition of Closed-Loop Stackelberg Equilibrium is not a trivial task. If one indeed defines the Closed-Loop Stackelberg Equilibrium in a similar fashion to what done for the Closed-Loop Nash case (i.e. asking $u_1^*(t, x)$ and $u_2^*(t, x)$ to be Stackelberg optimal for all possible initial conditions $x(\tau) = y$), he will notice that such a solution would not verify the

¹In the case of zero sum games one can prove that the system is at least hyperbolic.

Principle of Dynamic Programming. This is due to the lack of symmetry that does not allow to split the problem into smaller subproblems, as on the other hand is done in the statement of the Dynamic Programming. Consequently one would no longer be able to use standard tools of optimal control theory (HJB in this specific case). Several workarounds have been conceived but the simplest way to define the Closed-Loop Stackelberg equilibrium is to ask for weaker requirements.

In particular in [22] the continuous problem is discretized and the leader announces and enforces his strategy only stage-wise. Taking the limit as the discrete problem becomes continuous, one obtains a (well posed) definition of Closed-Loop Stackelberg solution. Loosely speaking the Closed-Loop Stackelberg solution is the solution of a leader-follower game where the leader has only infinitesimal advantage over the follower. For a more detailed discussion on the definition see [23]. For such weaker solutions one can show that the Closed-Loop Stackelberg equilibrium satisfies the Dynamic Programming Principle, and thus can be characterized using the HJB partial differential equation, as we shortly present here.

Sufficient condition for Stackelberg Equilibrium via HJB equations

Let us assume that the best reply of the second player to what the leader announces is unique. It can therefore be computed as

$$u_2^\#(t, x, p_1, p_2, u_1) = \arg \inf_{u_2 \in U_2} \{p_2 \cdot f(x, u_1, u_2, t) + L_2(x, u_1, u_2, t)\}, \quad (3.4)$$

and does depend on the choice of u_1 .

The leader will then optimize given the knowledge of the best reply, i.e.

$$u_1^\#(t, x, p_1, p_2) = \arg \inf_{u_1 \in U_1} \{p_1 \cdot f(x, u_1, u_2^\#(t, x, p_1, p_2, u_1), t) + L_1(x, u_1, u_2^\#(t, x, p_1, p_2, u_1), t)\}, \quad (3.5)$$

with the corresponding one for the follower being

$$u_2^\#(t, x, p_1, p_2) = u_2^\#(t, x, p_1, p_2, u_1^\#(t, x, p_1, p_2)).$$

As a consequence of Theorem 2.4 and as shown in [21], a sufficient condition to find a closed loop Stackelberg equilibrium is given by

$$\begin{cases} V_{1t} = -\nabla V_1 \cdot f(t, x, u_1^*, u_2^*) - L_1(x^*(t), u_1^*(t), u_2^*(t), t), \\ V_{2t} = -\nabla V_2 \cdot f(t, x, u_1^*, u_2^*) - L_2(x^*(t), u_1^*(t), u_2^*(t), t), \end{cases}$$

where the controls are computed as in (3.4), (3.5) and evaluated in $p_i = \nabla V_i$, i.e.

$$u_i^* = u_i^\#(t, x, \nabla V_1, \nabla V_2).$$

The system constitutes a PDE initial value problem as it is completed with the terminal condition on the value functions

$$V_1(T, x) = \psi_1(x), \quad V_2(T, x) = \psi_2(x).$$

Note that the only difference with the system of PDE obtained in the Nash case is in the choice of the optimal policies $u_i^*(t)$. In the Nash case they were obtained from equation (3.2), while here the first player chooses for u_1^* being aware of the best reply of the second player, see equations (3.4) and (3.5).

A similar system of HJB will be obtained in the next chapter when dealing with the differential game arising from the problem of optimal monitoring.

Optimal Monitoring via differential game

4.1 Introduction

The ancestor of the unmanned flight vehicles dates back at least to the 1800 when the Austrian artilleryist Franz von Uchatius conceived the idea of flying unmanned balloons carrying explosives to attack Venice. The balloons were uncontrolled and their flight path was determined mainly by the weather conditions. The first mission indeed failed because of unfavorable wind, but the second attempt got “luckier” and caused minimal damages to the city.

If we restrict to vehicles capable of generating lift and on which there was a minimum control, the kite flown by Douglas Archibald in 1883 can be considered the ancestor of the modern Unmanned Aerial Vehicles (UAVs). It was capable of flying up to 350 *m*, measuring the wind speed with an anemometer and taking one of the first aerial pictures. William Eddy exploited the ideas of Douglas and used a similar kite to take various pictures during the Spanish-American war in 1898. That was the first use of an UAV in combat, and since then, the development of UAVs has been mainly driven by military applications.

During the First World War Charles Kettering developed the Kettering Aerial Torpedo, an unmanned biplane commonly known as the ‘Bug’ that was capable of flying for roughly 70 km carrying 80 kg of explosive material. The vehicle

was guided by a preset controller. The main breakthrough though came a few years later, when Prof. Montgomery Low solved the aerial data link problem and made the first radio controlled flight in 1924. After that time UAVs got perfected and had been used constantly in the Second World War as bomb carrier. Well known for this are the German's V1 and V2.

From the Vietnam War, UAVs started being used also for reconnaissance as deep penetrators. They are indeed capable of infiltrating in the hostile territory without jeopardizing human lives, and can provide priceless pieces of information. In the War in Afghanistan they have been widely used to locate and keep track of important targets and have proved to be a decisive weapon [1].

On the civil side, NASA started in the early 1970s to investigate automatically controlled drones. The first project was called PA-30 and featured a self guided airplane even though a pilot was in the cockpit for safety reasons. NASA actively engaged in other programs such as the F-15 Spin Research Vehicle in order to bring the potential of UAV to the civilian [24]. In the 1990s NASA started a pilot program to attract industrial partners and opened the door to the modern era of UAVs.

The market for unmanned vehicles is in rapid expansion as the data from [25] show in figure 4.1. Here we give a quick overview on the civil areas of interest.

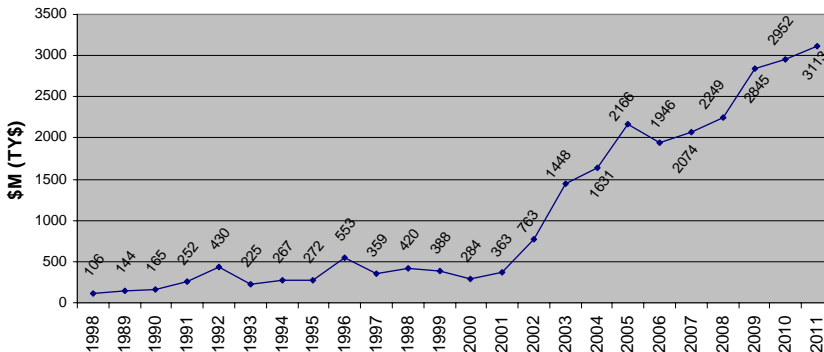


Figure 4.1: Annual funding profile from the Department of Defense for UAVs.

- Aerial surveillance and patrolling: this is one of the main applications of UAVs. Due to their versatility and the capability to carry on different payloads (such as cameras), they constitute an essential tool in surveillance. They have been used for instance to discover wildfires or for road patrolling [2].
- Disaster relief - Search and rescue: UAVs can be extremely useful in dangerous location thanks to the fact that no human being is directly involved

in their operations. For instance drones can retrieve important information from compromised areas and help rescuing endangered people.

- **Filmmaking:** though not the main application, drones are being used by movie makers to shot special scenes due to their higher flexibility when compared with standard shooting devices.
- **Oil and gas exploration:** when fitted with special payloads such as magnetometers, UAVs help to understand the nature of the underground rock structure. Magnetic field data are indeed already used to locate the position of mineral deposits.
- **Postal delivery:** this is a novel application that is under investigation. The german company DHL has started a pilot project and in 2013 was it capable of delivering a parcel of around 1kg with the small quadcopter shown in picture 4.2.



Figure 4.2: The UAV used by DHL to deliver a small parcel across the Rhine.

- **Scientific research:** this is a wide area and contains several applications that differ significantly one from the other. The National Oceanic and Atmospheric Administration, for instance, used UAVs to study and track the formation of hurricanes.

This chapter has begun with a brief historical background on UAVs and their evolution both in the military and civil sector. In the next section the problem of monitoring is introduced.

4.2 Monitoring using UAVs and related works

As previously mentioned, one of the core applications for UAVs is surveillance. Toward such goal, it is necessary to investigate how drones should coordinate in an arbitrary environment of given shape in order to collect meaningful data. In a real scenario several constraints must be taken into account, such as limited amount of fuel, poor or missing data link, time restrictions and others.

In the remaining of this work we will analyze the problem of monitoring a given region by means of UAVs under more ideal conditions, i.e. by neglecting some of the constraints just mentioned. The main purpose is nevertheless to model and capture the most important features of the problem, including the coordination of multiple agents and avoiding their mutual collision.

In the case when the domain to be controlled has relatively small size compared to the sensors capabilities, the problem of surveillance can be turned into a static optimization problem where the position of the sensors is fixed. Given a region of known shape, the optimization of static sensors location has been deeply studied in the past, for instance in [26], [27]. The solution is well known to be a Voronoi partition, where each sensor is responsible for a single Voronoi cell and its location coincides with the centroid of the cell it belongs to [28]. Loosely speaking, given a region to cover and a number of points (the agents), the region is subdivided into cells such that each point of the region is allocated to the the closest cell. A good introduction to the main difficulties is provided in [3], where the authors discuss the challenges that come along with multi agents networks for sensing and estimation.

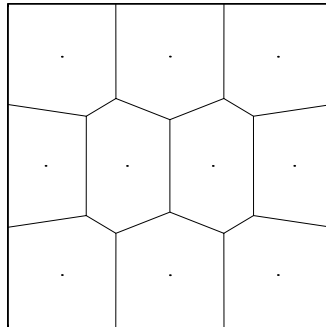


Figure 4.3: Example of a 10 point centroidal Voronoi tessellation of a square. Every region includes all the points that need to be controlled from each camera located in the center (black points).

When the area to be monitored increases, a static solution no longer provides a sufficient coverage, but mobile sensors need to be considered. The literature in this field is quite new and the survey [29] by Li and Cassandras presents a valid starting point. They discuss both static and dynamic scenarios and analyze the problem from different perspectives such as coverage, communication costs and others.

The same authors consider the case of a random event occurring in a given region and study the problem of capturing it [30]. They model the probability distribution of the random event to be captured and successively try to maximize the rate of capture, minimizing at the same time the communications. Given an initial position of the agents they develop a maximization algorithm that converge to a local optimum. The long term solution coincides with a stationary configuration.

Similar issues have been studied in [31] and [32]. The first deals with redeployment of the agents in order to improve the coverage, while in the second one the authors focus on surveillance tasks.

Modifications of the Voronoi solution for a dynamic case have been introduced in [33] and [34], the main issue being the computational effort required when computing the Voronoi tessellation continuously as in the above mentioned works. In the previously mentioned papers the authors focused on redeploying the agents, i.e. given a known initial condition, they determine a final configuration to which the agents should converge and a way to reach this goal.

A completely new perspective on the problem has been introduced by Hussein and Stipanović in [5]. Given an initial configuration they devise a way to control the agents in order to guarantee that a certain minimum coverage value is achieved across all the region. One of the main contributions of the above mentioned work ‘*Effective Coverage Control for Mobile Sensor Networks With Guaranteed Collision Avoidance*’ is the formulation of the problem itself, regardless of the solution they present. They indeed introduce the instantaneous coverage and the coverage maps. Mylvaganam and Astolfi in [4] worked in the same direction as Hussein and Stipanović but introduced concepts from game theory and formulated the problem for the first time in terms of differential game. The papers [4] and [5] constitute the foundation on which the work presented here is based.

This thesis introduces some novelties in the field of monitoring. They are briefly presented in this paragraph, where the structure of the chapter is presented. In section 4.3 the main goal is formally discussed, the instantaneous coverage and the coverage map are introduced in 4.4. The mathematical problem is formulated in section 5.1 as a standard differential game, following what has been done in [4], where Mylvaganam and Astolfi look for approximate solutions. In contrast to this, we write down the HJB equations and try to solve them numerically.

The problem is then reformulated as a novel non standard game in section 6.1.

Two different approximate solution methods are discussed. In 6.2 we adopt an optimal control approach, while in 6.3 a decomposition based on wavelets is presented. The reformulation and the wavelet approach constitute the main novelty presented in this thesis.

4.3 Problem formulation

Let us introduce in more detail the problem of optimal monitoring via UAVs.

Given a region $\Omega \subset \mathbb{R}^n$, $n \in \{1, 2, 3\}$, of known shape, consider a number $m \geq 1$ of UAVs. The dynamics of each agent is governed by a differential equation

$$\dot{x}_i = f_i(x_i, u_i), \quad i = 1, \dots, m,$$

where $x_i(t) \in \Omega$ represents the agent position and the function $u_i(t)$ can be used to control its movement. Every agent is equipped with a sensor capable of collecting data on the region Ω . Note that both the agents and the sensors are heterogeneous, i.e. the dynamics f_i and the sensors model can be different for different UAVs. The main question we want to address is the following.

How should the agents move in order to monitor the region Ω ‘optimally’, in the given mission time T ?

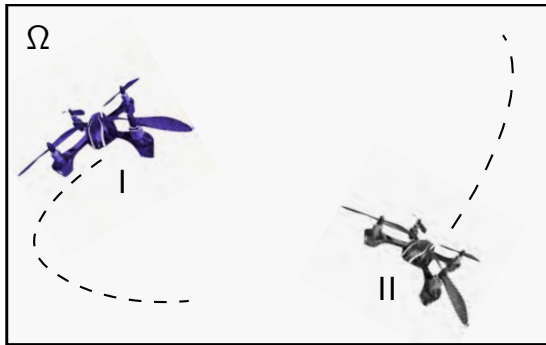


Figure 4.4: Sketch of a region Ω to be monitored and two UAVs flying around.

The problem is not well defined unless one specifies what ‘optimally’ means in this context. As a matter of fact depending on the application, one may want to optimize different criteria. It is not possible at this stage to formally specify what we mean by ‘optimal’, as more tools are needed in order to do so.

Nevertheless we want to present the main idea that will guide us:

The agents should move in order to continuously maximize the minimum coverage of Ω .

The tools we need to introduce include the sensor model and the coverage map. They are presented in the following subsection, after which a precise mathematical formulation of the problem is given.

4.4 Sensor model and coverage map

Similarly to what is done in [5], let us introduce the *sensor model* and the *coverage map*.

- The **sensor model** S_i represents what the i -th agent can perceive of the surrounding domain Ω and how good is in capturing each single point $q \in \Omega$. The sensor model is completely identified by the positive real valued map

$$S_i(x_i, q) : \Omega \times \Omega \mapsto \mathbb{R}^+,$$

that describes how effectively the i -th agent located in x_i senses the point $q \in \Omega$.

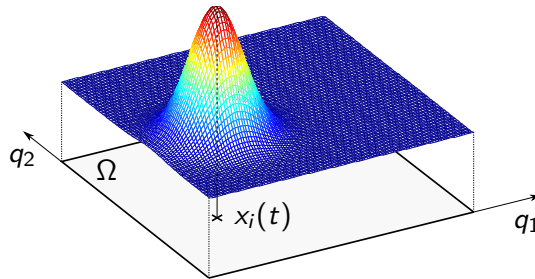


Figure 4.5: Example of sensor model $S_i(x_i, q) = e^{-\alpha_i \|x_i - q\|^2}$ in \mathbb{R}^2 . The higher the value of $S_i(x_i, q)$, the better the sensor located in x_i can perceive what is happening at the point q .

We consider sensors that satisfy the following properties:

1. The maximum sensing capacity is attained in the exact point x_i where

the agent is located, i.e.

$$S_i(x_i, x_i) > S_i(x_i, q), \quad \forall q \neq x_i, \quad q, x_i \in \Omega.$$

This is true for all the agents.

2. Each sensor capacity decays with the euclidean distance from the position of the corresponding agent x_i . This means that the sensor model has a circular symmetry.

$$S_i(x_i, \tilde{q}) < S_i(x_i, q) \quad \forall \tilde{q}, q, x_i \in \Omega \quad \text{s.t.} \quad \|\tilde{q} - x_i\| > \|q - x_i\|,$$

$$S_i(x_i, \tilde{q}) = c \quad \forall \tilde{q}, x_i \in \Omega \quad \text{s.t.} \quad \|\tilde{q} - x_i\| = d_c \text{ const.}$$

This holds true for all the agents.

To understand why we assume such properties let us suppose the payload of the considered drone is an ideal camera. The most focused point of the image is clearly the center (prop. **1**), while the quality of the image decreases with the distance from it and has circular symmetry (prop. **2**). For a real camera the field of view represents the area that can be sensed by the instrument and is *finite*. This implies that the sensor model should have compact support and the value of S_i should be zero when the distance is large from the center x_i . One could enforce this, but it would introduce further complications in the following. Thus from now on we consider a sensor of the form

$$S_i(x_i, q) = e^{-\beta_i \|x_i - q\|^2}, \quad \beta_i > 0, \quad (4.1)$$

that respects the properties **1** and **2** and has *quick* decay, i.e. in some weaker sense resemble the compact support property. The considered sensor model is depicted in figure 4.5 for a fixed value of x_i .

- The **coverage map** takes into account the past history and represents at time t how well each point $q \in \Omega$ has been surveyed following the trajectories of the agents $x_i(s)$, $s \in [0, t]$. The coverage map is given by the integral of the sensor model following the trajectory of the agents. The contribution of the i -th agent to the coverage map is given by

$$J_t^i(x_i(s), q) = \int_0^t S_i(x_i(s), q) ds,$$

while when multiple UAVs are involved, the contributions sum up

$$J_t(x(s), q) = \int_0^t \sum_{i=1}^m S_i(x_i(s), q) ds, \quad (4.2)$$

where $x(s) = [x_1(s), \dots, x_m(s)]$ represents the positions of all the agents. Note that the coverage map J_t depends on the sensor model and on how the agents have moved from the initial time $s = 0$ up to $s = t$, i.e. J_t is a functional that maps the trajectories $x_i(s)$, $s \in [0, t]$ into the coverage value for each point $q \in \Omega$. An example of coverage map is shown in figure 4.6.

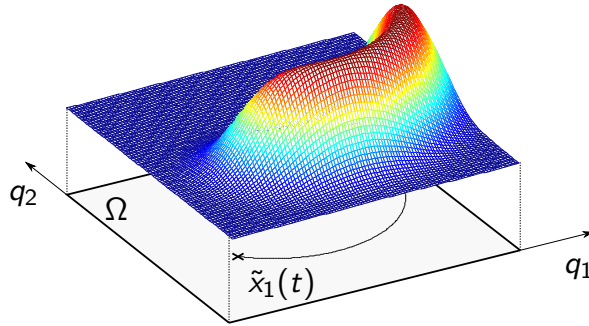


Figure 4.6: Example of coverage map for a single agent moving with constant speed on an arc of circle. The sensor model used is (4.1).

Monitoring as a standard differential game

5.1 Standard game formulation

We are now in the position to formulate the optimal monitoring problem in terms of differential games. The idea has been presented for the first time in [4], where the authors introduced the concept of virtual players.

In section 4.3 we have stated the problem explicitly; let us recall it here and use it as a starting point for the discussion:

How should the agents move in order to monitor the region Ω ‘optimally’, in the given mission time T ?

As already mentioned we need to give a clear meaning to the concept of ‘optimal’ monitoring. Loosely speaking we have said that:

The agents should move in order to continuously maximize the minimum coverage in Ω .

With this in mind, *virtual players* $\tilde{x}_i(t)$, $i = 1, \dots, m$ with *fast dynamics* are introduced. The purpose is to use these agents to look for the less covered points and then to ask the real agents $x_i(t)$ to track their trajectory i.e. to maximize the coverage following the less covered points. Since the players $\tilde{x}_i(t)$

are fictitious, they can be equipped with arbitrary dynamics. In particular let us introduce simple integrator dynamics for them, i.e.

$$\dot{\tilde{x}}_i = w_i, \quad i = 1, \dots, m,$$

where w_i represents the control of the i -th virtual player.

It is assumed that the real players also follow single integrator dynamics

$$\dot{x}_i = u_i, \quad i = 1, \dots, m.$$

Even if this may seem unrealistic, often the problem of path planning for aerial vehicles is studied with a very simple dynamics or even neglected [1]. At a preliminary stage one wants to look for the optimal trajectory for the problem without taking too much into account the dynamics of the specific vehicle used. At a further stage, when the trajectory is found, it is possible to pose the problem of tracking such a motion by assuming a more realistic vehicle model. Furthermore, as it will be clearer in the following, the problem is extremely difficult to tackle and even in this simple scenario preserves the main features. In real applications the control effort u_i is limited by the hardware capabilities, this needs to be taken into account asking for the control functions $u_i(t)$ to satisfy at each time

$$\|u_i(t)\| \leq u_M, \quad \forall t \in [0, T], \quad (5.1)$$

and for all the agents $i = 1, \dots, m$.

As already discussed the virtual players can be attributed arbitrary dynamics, but also in this case it is necessary to specify a bound on the control value, that could otherwise go to infinity. We ask the fictitious players to satisfy a similar condition

$$\|\tilde{u}_i(t)\| \leq \tilde{u}_M, \quad \forall t \in [0, T], \quad (5.2)$$

but we want them to be faster than the real players in order to quickly reach the less covered points. Thus we want the following condition to be satisfied

$$\tilde{u}_M \gg u_M. \quad (5.3)$$

The problem can consequently be formulated as a differential game, where the fictitious agents (first players) look for the less covered points, while the real players (second players) want to increase the coverage in those specific points i.e.

$$\begin{cases} \min_{w(t)} \int_0^T \sum_{i=1}^m S_i(x_i(s), \tilde{x}_i(s)) ds, \\ \max_{u(t)} \int_0^T \sum_{i=1}^m S_i(x_i(s), \tilde{x}_i(s)) ds, \\ \text{s.t.} \quad \dot{x}_i = u_i, \quad \dot{\tilde{x}}_i = w_i, \\ \text{with} \quad \|\tilde{u}_i(t)\| \leq \tilde{u}_M, \quad \|u_i(t)\| \leq u_M, \quad \tilde{u}_M \gg u_M. \end{cases} \quad (5.4)$$

Note that a further constraint on the trajectories $x_i(t)$ and $\tilde{x}_i(t)$ must be included, namely

$$x_i(t) \in \Omega, \quad \tilde{x}_i(t) \in \Omega,$$

since this is not guaranteed by the fact that the controls are bounded. In particular this constraint is necessary as the player looking for the less covered points would otherwise leave the set Ω . As a matter of fact in Ω the coverage is greater than outside.

A simple way to satisfy the previous constraints is to include a barrier function that penalizes the players when they leave the set Ω , see [35]. We want to penalize virtual and real players in the same way, thus let us introduce the barrier $b(x, \tilde{x})$ in (5.4) as in the following

$$\begin{cases} \min_{w(t)} \int_0^T \sum_{i=1}^m S_i(x_i(s), \tilde{x}_i(s)) + b(x(s), \tilde{x}(s)) ds, \\ \min_{u(t)} \int_0^T - \sum_{i=1}^m S_i(x_i(s), \tilde{x}_i(s)) + b(x(s), \tilde{x}(s)) ds, \end{cases}$$

where $b(x, \tilde{x}) > 0$ and in particular is zero when inside Ω , but grows rapidly when reaching the boundary $\partial\Omega$. An example of barrier function is the one depicted in figure 5.1.

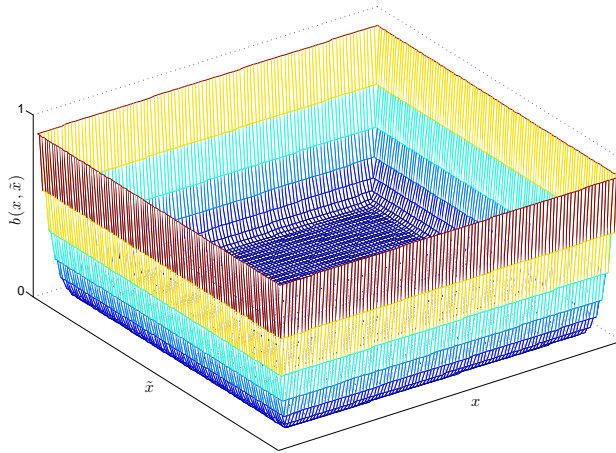


Figure 5.1: Example of barrier function on a squared domain $\Omega \times \Omega$. Barrier functions are used to penalize the agents when they leave the domain Ω . They are chosen so that $b(x, \tilde{x}) > 0$ and in particular to be zero when inside Ω , but growing rapidly when reaching the boundary $\partial\Omega$.

Finally the hard constraints (5.1) and (5.2) can be transformed into soft conditions introducing another term that penalizes high control activity. The final differential game assumes the form

$$\left\{ \begin{array}{l} \min_{w(t)} \int_0^T \sum_{i=1}^m S_i(x_i(s), \tilde{x}_i(s)) + b(x(s), \tilde{x}(s)) + \alpha_1 \frac{\|w\|^2}{2} ds, \\ \min_{u(t)} \int_0^T -\sum_{i=1}^m S_i(x_i(s), \tilde{x}_i(s)) + b(x(s), \tilde{x}(s)) + \alpha_2 \frac{\|u\|^2}{2} ds, \\ \text{s.t.} \quad \dot{x}_i = u_i, \quad \dot{\tilde{x}}_i = w_i, \\ \text{with} \quad \alpha_1 \ll \alpha_2, \end{array} \right. \quad (5.5)$$

where $x(s) = [x_1(s), \dots, x_m(s)]$ and $\tilde{x}(s) = [\tilde{x}_1(s), \dots, \tilde{x}_m(s)]$ represent the positions of real and fictitious players respectively, while $w(s) = [w_1(s), \dots, w_m(s)]$ and $u(s) = [u_1(s), \dots, u_m(s)]$ represent the control for real and fictitious vehicles respectively.

5.2 Sufficient conditions

As the differential game (5.5) arising from optimal monitoring is in *standard form*, conventional techniques presented in the previous chapter can be applied. Closed loop solutions are in particular always preferred when dealing with real world applications. They are more robust than open loop strategies that need to be recomputed in the case a perturbation steers the system away from the optimal trajectory.

Let us consider the case of a single agent monitoring the segment $\Omega = [a, b] \subset \mathbb{R}$ in the interval $[0, T]$, and devise the HJB sufficient condition for Nash optimality as presented in the previous chapter in (3.3).

The game is given by

$$\left\{ \begin{array}{l} \min_{w(t)} \int_0^T S(x(s), \tilde{x}(s)) + b(x(s), \tilde{x}(s)) + \alpha_1 \frac{w^2}{2} ds, \\ \min_{u(t)} \int_0^T -S(x(s), \tilde{x}(s)) + b(x(s), \tilde{x}(s)) + \alpha_2 \frac{u^2}{2} ds, \\ \text{s.t.} \quad \dot{\tilde{x}} = w, \quad \dot{x} = u, \\ \text{with} \quad \alpha_1 \ll \alpha_2. \end{array} \right. \quad (5.6)$$

- **Optimal policy:** The optimal policies $w^*(\tilde{x}, x, \tilde{p}, p)$ and $u^*(\tilde{x}, x, \tilde{p}, p)$ are obtained as functions of the state (\tilde{x}, x) and of the momenta (\tilde{p}, p) , as

shown in equation (3.2)

$$\begin{aligned} w^* &= \arg \inf_w \left\{ \tilde{p}_1 w + \tilde{p}_2 u + S(x, \tilde{x}) + b(x, \tilde{x}) + \alpha_1 \frac{w^2}{2} \right\} = -\frac{\tilde{p}_1}{\alpha_1}, \\ u^* &= \arg \inf_u \left\{ p_1 w + p_2 u - S(x, \tilde{x}) + b(x, \tilde{x}) + \alpha_2 \frac{u^2}{2} \right\} = -\frac{p_2}{\alpha_2}. \end{aligned}$$

- **Hamiltonian:** Let us denote by H and \tilde{H} the hamiltonian functions for the real and fictitious player. One gets

$$\begin{aligned} \tilde{H}(x, \tilde{x}, p, \tilde{p}) &= \left\{ \tilde{p}_1 w^* + \tilde{p}_2 u^* + S(x, \tilde{x}) + b(x, \tilde{x}) + \alpha_1 \frac{w^{*2}}{2} \right\} \\ &= -\frac{\tilde{p}_1^2}{2\alpha_1} - \frac{\tilde{p}_2 p_2}{\alpha_2} + S(x, \tilde{x}) + b(x, \tilde{x}), \\ H(x, \tilde{x}, p, \tilde{p}) &= \left\{ p_1 w^* + p_2 u^* - S(x, \tilde{x}) + b(x, \tilde{x}) + \alpha_1 \frac{u^{*2}}{2} \right\} \\ &= -\frac{p_2^2}{2\alpha_2} - \frac{p_1 \tilde{p}_1}{\alpha_1} - S(x, \tilde{x}) + b(x, \tilde{x}). \end{aligned}$$

- **HJB:** The partial differential equations that the value functions $V(t, x, \tilde{x})$ and $\tilde{V}(t, x, \tilde{x})$ need to satisfy are given by (3.3) as

$$\begin{cases} \tilde{V}_t + \tilde{H}(x, \tilde{x}, \nabla V, \nabla \tilde{V}) = 0, \\ V_t + H(x, \tilde{x}, \nabla V, \nabla \tilde{V}) = 0, \end{cases} \quad (5.7)$$

together with the final conditions

$$\begin{cases} \tilde{V}(T, x, \tilde{x}) = b(x, \tilde{x}), \\ V(T, x, \tilde{x}) = b(x, \tilde{x}), \end{cases} \quad (5.8)$$

since we want to penalize the final position when it is outside of the domain Ω . The equations (5.7) take the explicit form

$$\begin{cases} \tilde{V}_t = \frac{\tilde{V}_{\tilde{x}}^2}{2\alpha_1} + \frac{\tilde{V}_x V_x}{\alpha_2} - S(x, \tilde{x}) - b(x, \tilde{x}), \\ V_t = \frac{V_x^2}{2\alpha_2} + \frac{\tilde{V}_{\tilde{x}} V_{\tilde{x}}}{\alpha_1} + S(x, \tilde{x}) - b(x, \tilde{x}), \end{cases} \quad (5.9)$$

where V_x and V_t represent respectively the space and time partial derivatives of the value functions.

Note that in this case it is not crucial to specify which equilibrium we are interested in. As a matter of fact it is easy to see that with the given formulation

the Nash solution coincides with the Stackelberg one.

The system (5.9) together with the terminal conditions (5.8) constitutes a initial value problem for partial differential equations.

5.3 Numerical results

A commonly used numerical method to solve such problem is the finite difference scheme with the use of *upwind*, see [36]. We have adapted the optimal control toolbox provided by Mitchell in [37] and computed the numerical solutions together with the optimal policies $u^*(t, \tilde{x}, x, \nabla \tilde{V}, \nabla V)$, $w^*(t, \tilde{x}, x, \nabla \tilde{V}, \nabla V)$ for the monodimensional problem discussed here. The value functions are presented in figure 5.2 and 5.3 only for $t = 0.25$ since the shape does not change consistently when the time passes by.

The results are obtained with the following values of the parameters

$$\begin{cases} \alpha_1 = 1, & \alpha_2 = 5, \\ a = 0, & b = 5, \\ \beta = 1, & T = 1. \end{cases} \quad (5.10)$$

It is of great interest to consider the closed loop system, i.e. the system when the feedback control is applied, in this case simply

$$\begin{cases} \dot{\tilde{x}} = w^*(t, \tilde{x}, x, \nabla V, \nabla \tilde{V}), \\ \dot{x} = u^*(t, \tilde{x}, x, \nabla V, \nabla \tilde{V}). \end{cases} \quad (5.11)$$

The study of this dynamical system will provide with qualitative information on the behavior of the agents when they follow the optimal strategy. The system is bidimensional, thus it is possible to represent the vector field for different values of $t \in [0, T]$. Since only the magnitude of the vector field changes significantly with time, but not its direction, in figure 5.4 the phase portrait of the closed loop system is shown for $t = 0.25$. The direction of the arrows is representative of the behavior for all times $0 \leq t \leq T$.

Remarks

- As discussed in the survey paper [38], in general the numerical solution for a system of HJB equations is difficult to obtain, and the computational burden grows exponentially with the dimension of the state. In dimension

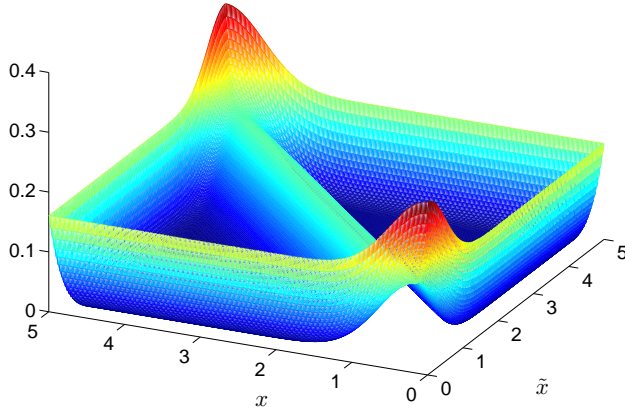


Figure 5.2: Value function at time $t = 0.25$ for the first player of the optimal monitoring game (5.6). Parameters as from (5.10).

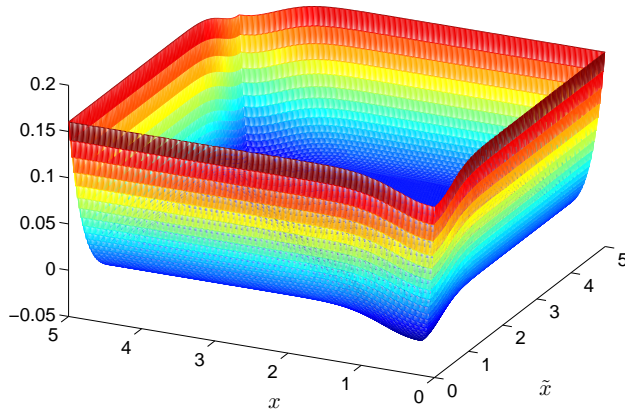


Figure 5.3: Value function at time $t = 0.25$ for the second player of the optimal monitoring game (5.6). Parameters as from (5.10).

$n \leq 3$ the solution is quite viable, while higher dimension problems have been solved numerically just in a few cases. Consequently such technique does not allow the introduction of more players or the study of planar problem i.e. with $x_i \in \mathbb{R}^2$. Furthermore from the numerical side, the HJB system is very sensitive to the errors and shows instability issues. Artificial terms need to be introduced in the numerical solution, for example using the Lax-Friedrichs scheme [39].

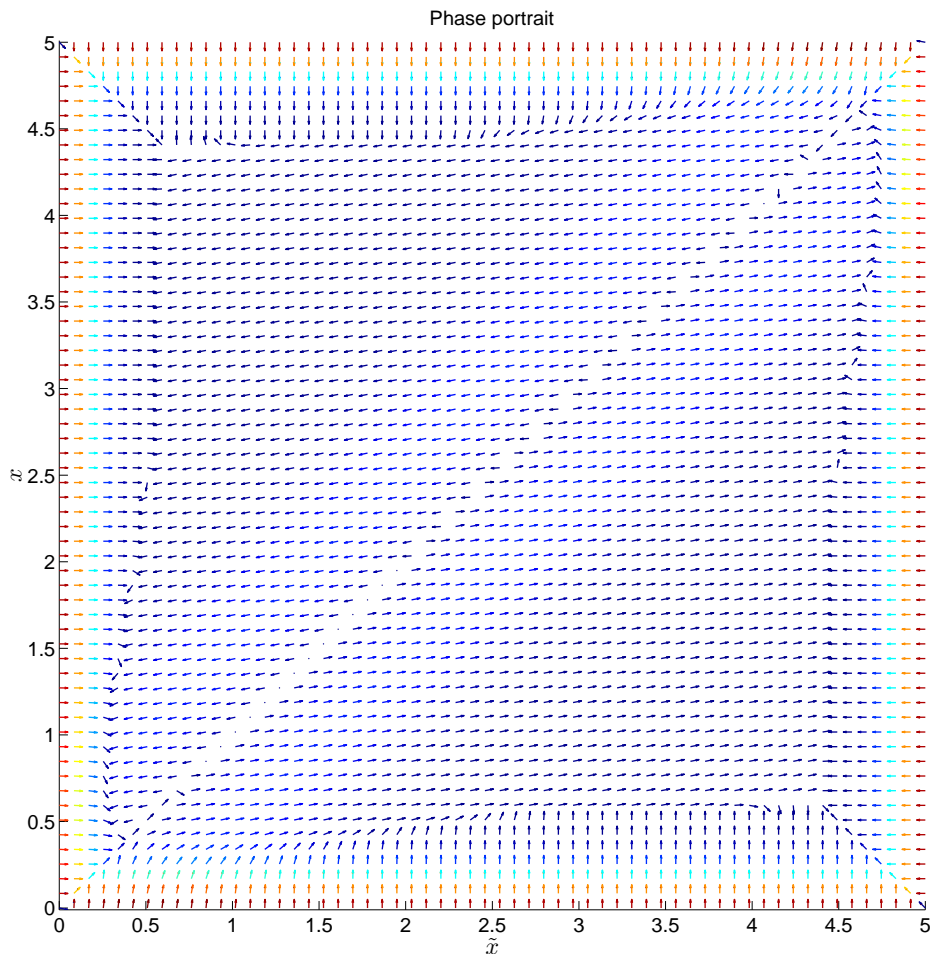


Figure 5.4: Phase portrait of the closed loop dynamical systems (\tilde{x}, x) . The color represent the magnitude of the field. Red stands for high velocity, while blue for low. Note that when the magnitude is close to zero i.e. dark blue color, the direction of the arrows may be affected by numerical errors. This is the case for instance along the diagonal $x = \tilde{x}$. If the initial condition lies below the diagonal, the system moves toward the equilibrium in the top right corner. On the contrary when the initial condition is located above the diagonal, the dynamics moves in the direction of the equilibrium in the bottom left corner.

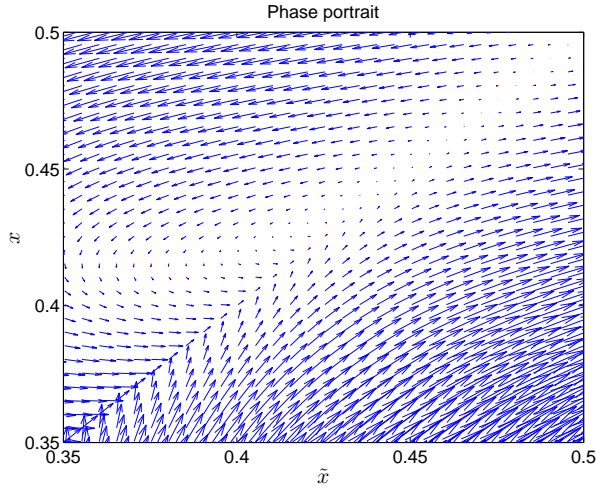


Figure 5.5: Zoom of the bottom left corner for the closed loop phase portrait of (5.11) at $t = 0.25$. Here the length of the arrows is representative of the magnitude of the vectors. Note how trajectories coming from the upper side cannot cross the diagonal as they will anyway stop at an equilibrium point.

- Let us comment the result shown in figure 5.4.

In the considered case a single agent is monitoring a segment $\Omega = [a, b]$ starting from one of its extremes. One would expect the solution (trajectory that continuously maximize the minimum coverage) to show some sort of oscillatory nature: the agent starts from $x = a$, goes toward $x = b$ and comes back at multiple times. On the contrary the numerical solution does *not* incorporate this behavior. At a first glance there seems to be some sort of periodic solution at least for t far from the end of the interval $[0, T]$, but if one looks closer to the top right and bottom left corners of the phase portrait (see the zoom in figure 5.5) this is not the case. The dynamical system indeed has two equilibrium points in such corners, where the dynamics is attracted. As a matter of fact no oscillatory behavior occurs and an agent starting from $x = a$ will move towards $x = b$ by the end of the interval $[0, T]$, no matter where \tilde{x} starts from.

This is due to the fact that the mission has finite horizon (and not infinite) and is also a consequence of the specific formulation in terms of the standard game presented in equation (5.5). Since this is a minimal feature that we want to include in the model, in the next section we explain with more details why (5.5) cannot produce such result, and how to modify the model in order to reach the goal.

Monitoring as a non standard differential game

6.1 Non standard game formulation

With the formulation of section 5.1, virtual players P_1 have been introduced with the aim to let them look for the less covered points. Subsequently the real agents P_2 where asked to track such trajectory by trying to maximize the coverage along the paths of P_1 i.e. $\tilde{x}_i(s)$. The differential game assumed the form (5.4) that we recall here

$$\begin{array}{l}
 P_1 \left\{ \begin{array}{l} \min_{w(t)} \int_0^T \sum_{i=1}^m S_i(x_i(s), \tilde{x}_i(s)) ds, \\ \\ \\ \end{array} \right. \\
 P_2 \left\{ \begin{array}{l} \max_{u(t)} \int_0^T \sum_{i=1}^m S_i(x_i(s), \tilde{x}_i(s)) ds, \\ \\ \text{s.t. } \dot{x}_i = u_i, \quad \dot{\tilde{x}}_i = w_i, \\ \text{with } \|\tilde{u}_i(t)\| \leq \tilde{u}_M, \quad \|u_i(t)\| \leq u_M, \quad \tilde{u}_M \gg u_M. \end{array} \right.
 \end{array}$$

With such structure the virtual players *do not* look for the less covered point(s) at each fixed instant $t \in [0, T]$, but they look for the points $x_i(s)$ that produce the minimum coverage at the *end* of the interval. This explains the non oscillatory

behavior seen in figure 5.4 and produces interesting results when the goal is to *survey* the region Ω . On the contrary as far as patrolling is concerned, we would require the agents to travel more than once through the same location $q \in \Omega$. As discussed, this does not happen with the actual formulation (5.5), thus we need to adjust the game in order for the fictitious player to look for the less covered points at *each* instant of time and for the real player to track them. This will be clarified in the next lines as we discuss the modifications that need to be introduced and we reformulate the game. It is crucial to read carefully and understand the coming paragraph in order to follow the rest of the chapter.

Game reformulation

1. To make things easier, let us neglect the dynamics of the agents and suppose we can select directly the set of trajectories of the players $x(s) \in \mathcal{X}$, where \mathcal{X} is a suitable function space.¹ As mentioned the objective is to control the agents in order to *continuously* maximize the minimum coverage in Ω .
2. Suppose trajectories $x(s) \in \mathcal{X}$ are given, then at each instant of time $0 \leq t \leq T$ the coverage map introduced in (4.2) can be computed as

$$J_t(x(s), q) = \int_0^t \sum_{i=1}^m S_i(x_i(s), q) ds.$$

One is interested in finding the minima i.e. in solving

$$\arg \min_{q \in \Omega} \int_0^t \sum_{i=1}^m S_i(x_i(s), q) ds, \quad (6.1)$$

The set Ω is assumed to be compact and the functions S_i as well as the trajectories $x(s)$ to be continuous. It follows that the minimum is reached in Ω , but it could be attained in more than one point. If we assume that the argument of minimum $q^* \in \Omega$ is unique, then for each instant of time one can construct the map $q^*(t)$ that returns the less covered point at time t , given the trajectories $x(s)$.²

3. As the objective is to maximize the coverage of the less known areas, we are interested in computing the coverage map at the end of the mission

¹Recall that $x(s)$ contains the trajectories of all the UAVs, i.e. $x(s) = [x_1(s), \dots, x_m(s)]$ where m is the total number of vehicles.

²In case the argument of minimum is not unique, it is sufficient to select one of those points. In particular it makes sense to allocate to each agent the closest q^* .

following the less covered points i.e.

$$J_{min} = \int_0^T \sum_{i=1}^m S_i(x_i(s), q^*(s)) ds .$$

With this process we have associated to each set of trajectories $x(s)$, a real number J_m that describes how good is the coverage following the minima. The task is to select the trajectories $x(s)$ that maximize J_m

$$\max_{x(s) \in \mathcal{X}} \int_0^T \sum_{i=1}^m S_i(x_i(s), q^*(s)) ds , \quad (6.2)$$

where $q^*(s)$ depends on $x(s)$ via the minimization in equation (6.1).

One can take a further step and introduce the dynamics

$$\dot{x}_i = f_i(x_i, u_i), \quad i = 1, \dots, m . \quad (6.3)$$

The maximization of equation (6.2) has no longer to be performed with respect to $x(s) \in \mathcal{X}$, but on $u \in U$.

The novel differential games we have introduced assumes the self consistent form

$$P_1 \left\{ \begin{array}{l} \min_{q \in \Omega} \int_0^t \sum_{i=1}^m S_i(x_i(s), q) ds , \\ P_2 \left\{ \begin{array}{l} \max_{u(t)} \int_0^T \sum_{i=1}^m S_i(x_i(s), q) ds , \\ \text{s.t.} \quad \dot{x}_i = f_i(x_i, u_i) \quad i = 1, \dots, m \\ u(t) \in U , \end{array} \right. \end{array} \right. \quad (6.4)$$

where we are interested in the *Stackelberg* solution with P_2 as a leader and P_1 as a follower. This is due to the way we have constructed the differential game: the real players P_2 announce their policy u^* and enforce it. The virtual player selects the best reply $q^* = \mathcal{R}_1(u^*)$ as in definition 3.2. The leader knows what the follower will play and thus optimizes taking that into consideration. This is exactly what was presented in the steps 1 and 2 from the previous page.

Remarks

- The game (6.4) is a *non* standard differential game, due to the presence of t as upper extremum on the first integral. This denies the opportunity to use the classical tools introduced in the previous chapter for optimal control and differential games. New techniques and numerical schemes need to be developed.

- The minimization (6.1) is highly non linear and the function q^* may be discontinuous, both in terms of the time and of the dependency on $x(s)$. This makes the problem more complicated.
- From the modellistic point of view the game (6.4) does not penalize the use of control and does not guarantee collision avoidance amongst the agents. This is not an issue as additional terms can be included later.
- No barrier function is required if the region Ω is convex. As a matter of fact the virtual player selects q^* directly belonging to Ω and consequently the real agents would not leave the set Ω throughout the mission.

6.2 Optimal control approximation

In this section we approximate the differential game (6.4) and transform it into a sequence of optimal control problems.

The mission horizon T is split into smaller intervals of length τ and a generic timestep is labelled t_k . In each interval $[t_k, t_{k+1}]$ suppose that the minimizer of the coverage map is fixed and call it q_k^* . This is not true in general as in $[t_k, t_{k+1}]$ the agents $x(s)$ will move and thus the coverage map will change. Clearly, the smaller τ , the better the approximation.

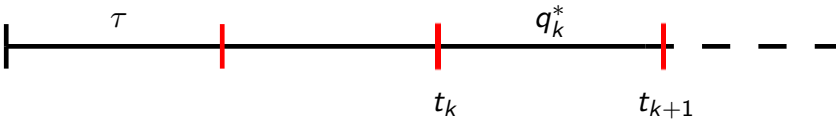


Figure 6.1: Discretization of the mission horizon T into intervals of length τ .

The algorithm is based on a discretization of the game where the players policy is chosen in two different stages as follows.

Stage 1: Suppose the solution has been computed up to time t_k so that the state of the system and the previous history is known as $x(s)$, $s \in [0, t_k]$. The coverage map at time t_k is then given by

$$J_{t_k}(x_i(s), q) = \int_0^{t_k} \sum_{i=1}^m S_i(x_i(s), q) ds,$$

while the minimizer can be easily found solving the following optimization over $q \in \Omega \subset \mathbb{R}^n$

$$q_k^* = \arg \min_{q \in \Omega} \int_0^{t_k} \sum_{i=1}^m S_i(x_i(s), q) ds.$$

In case of nonuniqueness the same observations made previously hold.

Stage 2: Since q^* is fixed to be q_k^* , it is sufficient to solve the optimal control problem on the interval $[t_k, t_{k+1}]$ for P_2

$$\max_{u(\cdot)_{[t_k, t_{k+1}]}} \int_{t_k}^{t_{k+1}} \sum_{i=1}^m S_i(x_i(s), q_k^*) ds,$$

i.e. to maximize the coverage in q_k^* . To this purpose the usual techniques can be implemented, in particular the HJB equation or the PMP.

Given the solution $u^*(\cdot)_{[t_k, t_{k+1}]}$ the dynamics is used to determine the evolution of the system solving the Cauchy problem

$$\dot{x}_i = f_i(x_i, u_i^*), \quad x(t_k) = x_k, \quad i = 1, \dots, m.$$

This will produce the trajectory up to time $t = t_{k+1}$. One can return to Stage 1 and repeat the procedure with $k \mapsto k + 1$ until the final time is reached i.e. $t = T$.

Initial.: At the first step one has $t = 0$ and $x(0) = x_0$ given. The coverage map is simply

$$J_0(x_i(s), q) = \int_0^0 \sum_{i=1}^m S_i(x_i(s), q) ds = 0 \quad \forall q \in \Omega,$$

thus there is an infinite number of minimizers, namely all the points $q \in \Omega$. In order not to consume additional fuel u , the agents $x(s)$ are kept stationary in the initial configuration while the coverage map builds up. At the next step

$$J_{t_1}(x_0, q) = \int_0^{t_1} \sum_{i=1}^m S_i(x_0, q) ds,$$

that is *not* constant along the points $q \in \Omega$.

It is worth noting that this approximation allows to consider the control cost and to avoid collision. Usually when dealing with applications one wants to

penalize the use of control as this turns out to be connected with the amount of fuel used. This can be done introducing in Stage 2 a penalizing term, namely

$$\max_{u(\cdot)_{[t_k, t_{k+1}]}} \int_{t_k}^{t_{k+1}} \sum_{i=1}^m S_i(x_i(s), q_k^*) - u^t C u ds,$$

where C is a positive definite matrix. The players $x(s)$ want to increase the coverage but need to take into account the fuel u they will use.

With similar reasoning a collision avoidance term $d_c(x(s))$ can be introduced. The function d_c depends on the distance amongst different agents

$$d_{ij} = \|x_i(s) - x_j(s)\| \quad \forall i \neq j, \quad i, j = 1, \dots, m,$$

and penalizes the performance index when at least two of them are too close. A valid implementation [4] is for instance

$$d_c = \sum_{i=1}^m \left(\max \left\{ 0, \sum_{j=1, j \neq i}^m R_i^2 - d_{ij}^2 \right\} \right)^2,$$

where R_i represents the minimum safe distance between $x_i(s)$ and $x_j(s)$ with $i \neq j$.

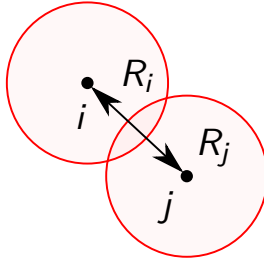


Figure 6.2: Collision avoidance can be implemented penalizing two agents when they are too close. The term for the i -th agent switches on only if the distance from the j -th agent is smaller than R_i , i.e. if the j -th player enters the red circle centered in $x_i(s)$.

In case both control costs and collision avoidance are included, the solution is found as

$$\max_{u(\cdot)_{[t_k, t_{k+1}]}} \int_{t_k}^{t_{k+1}} \sum_{i=1}^m S_i(x_i(s), q_k^*) - u^t C u - d_c(x(s)) ds.$$

6.2.1 Numerical results

The optimal control approximating algorithm has been implemented and tested in two scenarios: a monodimensional problem where a single agent is asked to monitor a segment $\Omega = [a, b]$ and a two-dimensional case where two agents monitor a squared region $\Omega = [a, b] \times [a, b]$.

In the first case $\Omega = [-10, 10]$, the mission horizon is $T = 600$, while the sensor is taken to follow the exponential model (4.1) with $\beta = 0.1$. It is assumed a simple integrator dynamics, and the control cost has been included by selecting $C = 1$. The optimal trajectory as a function of time is shown in figure 6.3, while in 6.4 the evolution of the coverage map is depicted. Figure 6.3 in particular describes what we expected: the agent starts from $x = -5$, and after the first stationary step, looks for the less covered point i.e. $q_1^* = 10$ and moves in that direction. After x reaches $q_1^* = 10$, the agent again seeks the less controlled location that is $q_2^* = -10$ and goes in that direction. The procedure repeats until the end of the mission.

In figure 6.4 one can follow the evolution of J_t . At the beginning of the mission the area is completely unknown. As the time proceeds forward, the agent moves and gathers meaningful data that increase the coverage. It is worth noting that the horizontal stripes are due to the fact that the agent spends quite some time close to each the minimum point q^* and thus improves significantly J_t close by.

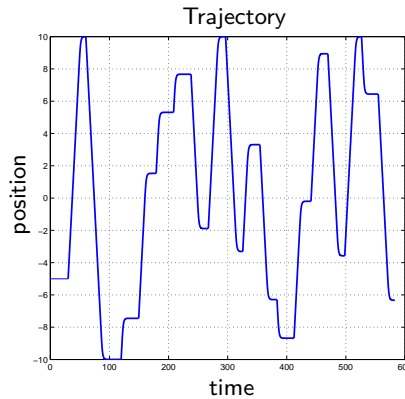


Figure 6.3: Numerical solution for the monodimensional monitoring problem representing the trajectory of the agent on the segment $[-10, 10]$ as a function of time. The agent starts from $x = -5$ and after the first stationary step, looks for the less covered point i.e. $q_1^* = 10$ and moves in that direction. The procedure repeats until the end of the mission.

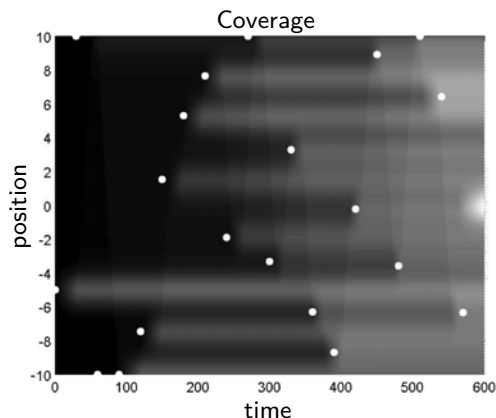


Figure 6.4: Evolution of the coverage map for the monodimensional problem. The darker, the less covered. Each vertical slice represent the coverage map at a fixed time, while the white points identify the minimizers of the coverage map at different instants of time.

In the two-dimensional case $\Omega = [-10, 10] \times [-10, 10]$, the mission horizon is $T = 600$, while the sensors are assumed to follow an identical exponential model (4.1) with $\beta = 0.1$. Simple integrator dynamics is considered. Both control cost and collision avoidance have been included, by selecting $C = 1$ and $R_i = 1$ for $i = 1, 2$. The trajectory is depicted in figure 6.5, while in 6.6 the final coverage is shown.

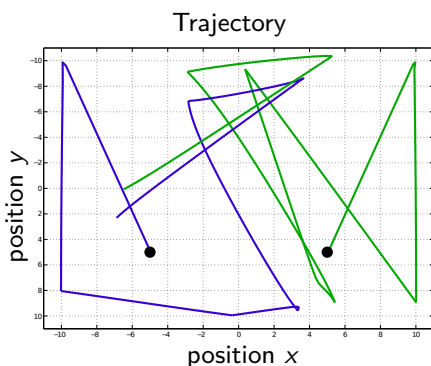


Figure 6.5: Numerical solution for the two dimensional monitoring problem representing the trajectory of the agents on the square $[-10, 10] \times [-10, 10]$.

As in this case two UAVs are operating, if two minimizers of J_t exist, they are allocated one to each of the agents. Note that for a short interval of time both of them have been allocated the same minimizer q^* and are moving in that direction. The collision avoidance term comes into play and repels the green agent outside the region of interest.

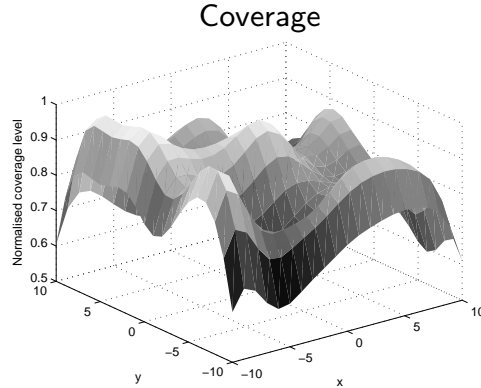


Figure 6.6: The coverage map J_T has been divided by its maximum value to produce the normalized coverage at the end of the mission, here plotted as a function of $q \in \Omega = [-10, 10] \times [-10, 10]$.

6.3 Wavelet decomposition

Optimization problems over infinite dimensional spaces have been firstly formulated and studied within the Calculus of Variations, of which the Brachistochrone problem is certainly the most notorious one [40]. Various approximate methods have appeared in literature; one of them consists in looking for the solution in a finite dimensional space.

The idea of reducing the problem to finite dimension has been used since the introduction of the series expansions and it consists of two steps. First, let us call \mathcal{X} the function space the solution belongs to, and select a suitable set of functions $x_i \in \mathcal{X}$. We want $\{x_i\}_{i \in \mathbb{N}}$ to be a basis for \mathcal{X} . Secondly, the finite dimensional space is taken as $\tilde{\mathcal{X}} = \text{span}\{x_i\}_{i \in A}$ with $A \subset \mathbb{N}$, i.e. A is a proper subset of \mathbb{N} . Therefore the approximate solution $\tilde{x} \in \tilde{\mathcal{X}}$ takes the form $\tilde{x} = \sum_{i \in A} \tilde{c}_i x_i$, for which only the coefficients c_i have to be determined. Note that the number of coefficients is finite since A is so.

In the same spirit, a fast method used for calculus of variation is presented in [41]. The algorithm uses wavelets as basis functions, due to their interesting

properties. Loosely speaking, when suitable wavelets are selected, the solution can be approximated using just a few coefficients, thus producing a fast code.

In the following we first introduce the notion of wavelets and their main properties. Successively we use them as basis functions in order to project the differential game (6.4) into a finite dimensional space.

Wavelet Expansion

Let us consider the *Hilbert* space $\mathcal{L}^2(\mathbb{R})$. The wavelet is formally defined as in the following.

DEFINITION 6.1 (WAVELET) *Let $\psi \in \mathcal{L}^2(\mathbb{R})$ and define the functions $\psi_{j,k}$ as*

$$\psi_{j,k}(x) \doteq 2^{j/2} \psi(2^j x - k), \quad j, k \in \mathbb{Z}, \quad x \in \mathbb{R}.$$

The function ψ is called a wavelet if $\{\psi_{j,k}\}_{j,k \in \mathbb{Z}}$ represents an orthonormal basis for $\mathcal{L}^2(\mathbb{R})$.

The functions $\psi_{j,k}$ form a *special* orthogonal basis for $\mathcal{L}^2(\mathbb{R})$ as they all need to be produced via dilation $2^{j/2}$ and translation $2^j x - k$ from ψ .

Recall that by definition of basis and due to orthogonality, every $f \in \mathcal{L}^2(\mathbb{R})$ can be written as

$$f = \sum_{j,k \in \mathbb{Z}} c_{j,k} \psi_{j,k} = \sum_{j,k \in \mathbb{Z}} \langle f, \psi_{j,k} \rangle \psi_{j,k},$$

where $\langle f, \psi_{j,k} \rangle$ indicates the inner product, i.e.

$$\langle f, \psi_{j,k} \rangle = \int_{-\infty}^{\infty} f(s) \psi_{j,k}(s) ds.$$

As far as applications are involved, we are interested in constructing wavelets such that the coefficients $c_{j,k}$ decay *fastly*. If this is the case, then just a few terms $c_{j,k}$ can be used to approximate $f \in \mathcal{L}^2(\mathbb{R})$. The notion of vanishing moment is crucial in this sense, as the next theorem shows.

DEFINITION 6.2 (VANISHING MOMENTS) *Let $n \in \mathbb{N}$. A wavelet ψ is said to have n vanishing moments if*

$$\int_{-\infty}^{\infty} s^l \psi(s) ds = 0 \quad l = 0, 1, \dots, n-1.$$

THEOREM 6.3 (DECAY OF WAVELET COEFFICIENTS) Consider a compactly supported wavelet with n vanishing moments, $\psi \in \mathcal{L}^2(\mathbb{R})$. Then for any $f \in \mathcal{L}^2(\mathbb{R})$ that is n times differentiable and with $f^{(n)}$ bounded, there exists a constant C such that

$$|\langle f, \psi_{j,k} \rangle| \leq C \cdot 2^{-jn} 2^{-j/2} \quad \forall j \geq 1, \quad k \in \mathbb{Z}.$$

The theorem gives an upper bound on the magnitude of the coefficients $|\langle f, \psi_{j,k} \rangle|$. In particular it shows that $c_{j,k}$ decay exponentially with j and n . Thus the higher the number of vanishing moments, the faster the coefficients will decay and the better the approximation to f will be when chopping the series.

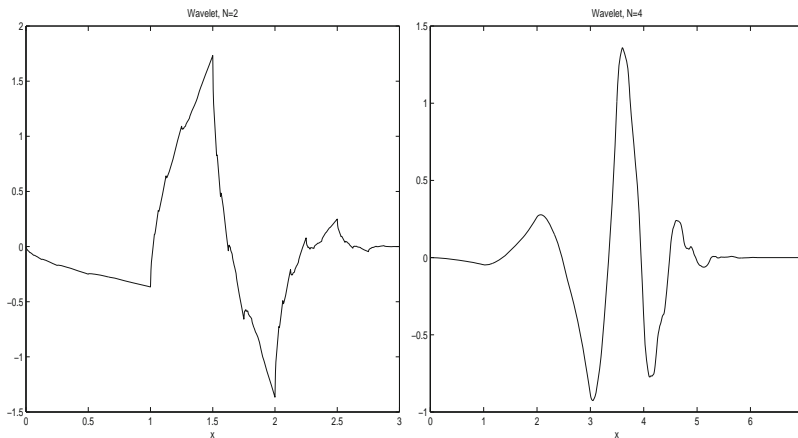


Figure 6.7: Daubechies' wavelets with $n = 2$ and $n = 4$. Image from [42].

This moves the attention on how to create a wavelet ψ that has compact support and n vanishing moments. The systematic study of wavelet bases started around 1985 and the main result is due to the introduction of the so called *multi-resolution* analysis. The key point is the capability of constructing wavelets with a generic number n of vanishing moments. Daubechies' wavelets are an example and the case with $n = 2$ and $n = 4$ is shown in figure 6.7. For a deeper account, see [42]. Similar results hold in the case when one considers the functions space $\mathcal{X} = \mathcal{L}^2([a, b])$ that will be used in the next section.

Reduction over a finite dimensional space

Using the wavelets as basis functions, we look for solutions to (6.4) on a finite dimensional space. In particular to make things easier, we consider the case of a single agent monitoring the segment $\Omega = [a, b]$. The procedure can be extended to higher dimension.

1. Let us suppose we can control directly the trajectory $x(s)$ and neglect the dynamics. The latter can be included in a second stage. Assume the trajectory $x(s) \in \mathcal{L}^2([a, b]) = \mathcal{X}$. Given ψ a wavelet, $\{\psi_{j,k}\}_{j,k \in \mathbb{Z}}$ forms a basis for \mathcal{X} .

It follows that each function can be written as $x(s) = \sum_{j,k \in \mathbb{Z}} c_{j,k} \psi_{j,k}(s)$. The finite subspace we are interested in is the one obtained by stopping the series $\tilde{\mathcal{X}} = \text{span}\{\psi_{j,k}\}_{j,k \in A}$ with $A \subset \mathbb{Z}$.

We look for approximate solutions $\tilde{x} \in \tilde{\mathcal{X}}$, i.e. of the form

$$\tilde{x}(s, \tilde{c}_{j,k}) = \sum_{j,k \in A} \tilde{c}_{j,k} \psi_{j,k}(s), \quad \tilde{c}_{j,k} \in \mathbb{R}, \quad (6.5)$$

where the dependence of \tilde{x} on the coefficients $\tilde{c}_{j,k}$ has been made clear.

2. We follow the steps 2 and 3 as in section 6.1 exploiting the particular structure (6.5) of $\tilde{x}(s, \tilde{c}_{j,k})$.

The coverage map introduced in (4.2) depends now on the trajectory only via $\tilde{c}_{j,k}$. The less covered points q^* can be found from

$$\arg \min_{q \in \Omega} \int_0^t S(\tilde{x}(s, \tilde{c}_{j,k}), q) ds, \quad (6.6)$$

that allows to construct the minimizer's map $q^*(t)$. To be more specific let us write explicitly the fact that q^* depends not only on time, but also on the coefficients $\tilde{c}_{j,k}$ via \tilde{x} , namely

$$q^* = q^*(t, \tilde{c}_{j,k}), \quad j, k \in A.$$

The performance index to maximize is finally obtained as in (6.2)

$$J_{min} = \int_0^T S(\tilde{x}(s, \tilde{c}_{j,k}), q^*(s, \tilde{c}_{j,k})) ds. \quad (6.7)$$

Note that this time we can interpret J_{min} as a function

$$J_{min} : \mathbb{R}^p \mapsto \mathbb{R},$$

where p represents the number of coefficients $\tilde{c}_{j,k}$ considered.

3. The objective is to maximize J_{min} , i.e. to maximize a real function over a finite dimensional domain \mathbb{R}^p , namely

$$\max_{\tilde{c} \in C} J_{min}(\tilde{c}), \quad (6.8)$$

where $\tilde{c} \in C \subset \mathbb{R}^p$ represents all the coefficients $\tilde{c}_{j,k}$. The problem has been transformed into a simpler one.

In case one wants to take into account the dynamics, it is sufficient to select $\tilde{u}(s, \tilde{g}_{j,k}) \in \text{span}\{\psi_{j,k}\}_{j,k \in A}$, i.e.

$$\tilde{u}(s, \tilde{g}_{j,k}) = \sum_{j,k \in A} \tilde{g}_{j,k} \psi_{j,k}(s), \quad \tilde{g}_{j,k} \in \mathbb{R}.$$

The trajectory is consequently obtained solving the Cauchy problem (6.3) with $x(0) = x_0$. In the same way we would be able to construct a map from the space of coefficients $\tilde{g}_{j,k}$ to the performance index J_{min} , thus reducing to an optimization problem.

Continuous but non differentiable optimization

One issue becomes evident when transforming the differential game into (6.8). The map $q^*(t, \tilde{c}_{j,k})$ obtained from (6.6) is in general a multivalued function since more than one minimizer can exist for each fixed time $t \in [0, T]$. As discussed before one option is to select the closest point q^* to the agent when multiple ones are available. Even in this case though the function $q^*(t, \tilde{c}_{j,k})$ may be discontinuous with respect to the coefficients $\tilde{c}_{j,k}$ (and this would in general be the case). If we want to apply standard optimization techniques to $J_{min}(\tilde{c})$, we need to guarantee a certain degree of smoothness of (6.8), and the fact that $q^*(t, \tilde{c}_{j,k})$ may be discontinuous puts us in a difficult situation. The performance index J_{min} is indeed obtained by composing S with q^* as in (6.7).

In the following we show that due to the discontinuous nature of $q^*(t, \tilde{c}_{j,k})$, the objective function J_{min} turns out to be continuous and Lipschitz, but *not differentiable*.

THEOREM 6.4 *Given a function $f(t, c) : [0, T] \times \mathbb{R}^p \mapsto \mathbb{R}$. Suppose f has a jump discontinuity along the locus described by $t^*(c)$ with $c \in \mathcal{C} \subset \mathbb{R}^p$ and $t^*(c)$ continuous. Suppose f is continuous elsewhere. Then*

$$J(c) = \int_0^T f(t, c) dt \quad \text{is continuous for all } c \in \mathbb{R}^p.$$

PROOF. Let us consider the case where $p = 1$ and $\mathcal{C} = [a, b]$. The points where

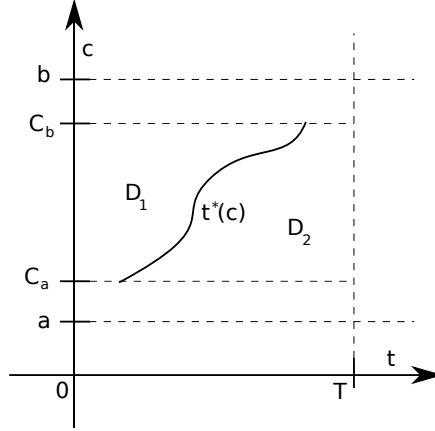


Figure 6.8: The curve $t^*(c)$ splits the domain into two subsets D_1 and D_2 . The first one includes all the points in between the image of $t^*(c)$ and the y axis. The second one consists of the points in between the image of $t^*(c)$ and the line $t = T$.

the function $f(t, c)$ is not continuous are shown in figure 6.8.

We want to prove that $J(c)$ is continuous $\forall c \in [a, b]$. When $c < c_a$ for each value of the time $t \in [0, T]$ the function $f(t, c)$ is continuous, and so is $J(c)$. The same holds in the case when $c > c_b$.

We need to prove it also for $c_a \leq c \leq c_b$, in this case we can split the integral as

$$J(c) = \int_0^T f(t, c) dt = \int_0^{t^*(c)} f(t, c) dt + \int_{t^*(c)}^T f(t, c) dt.$$

In the region D_1 and D_2 (see figure 6.8) separately, the function $f(t, c)$ is continuous. Thus one can write

$$\int_0^{t^*(c)} f(t, c) dt + \int_{t^*(c)}^T f(t, c) dt = F_1(t, c)|_0^{t^*(c)} + F_2(t, c)|_{t^*(c)}^T,$$

where $F_1(t, c)$ and $F_2(t, c)$ are the smooth primitives of f in the two regions D_1 and D_2 respectively. Thus $J(c)$ can be computed as

$$\begin{aligned} J(c) &= \int_0^T f(t, c) dt = F_1(t, c)|_0^{t^*(c)} + F_2(t, c)|_{t^*(c)}^T \\ &= F_1(t^*(c), c) - F_1(0, c) + F_2(T, c) - F_2(t^*(c), c), \end{aligned} \quad (6.9)$$

and due to the smoothness of both $t^*(c)$ and $F_i(t, c)$, we conclude that $J(c)$ is continuous also for $c_a \leq c \leq c_b$. Consequently $J(c)$ is continuous in all the points

of the domain. □

This setup extends also in higher dimension when it is possible to parametrize the locus of discontinuity as a function $t^*(c)$ and thus to construct the regions D_1 and D_2 . The sketch of figure 6.9 provides insight in the case $p = 2$.

It should be clear that the previous theorem applies also to more complex

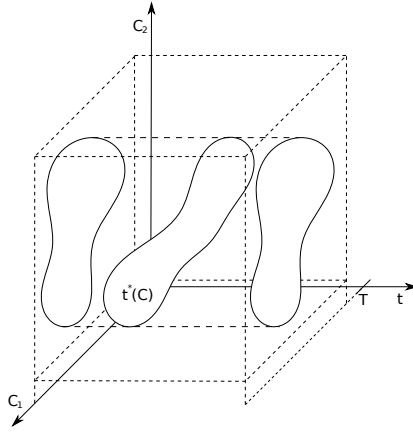


Figure 6.9: The surface $t^*(c)$ splits the domain into two subsets D_1 and D_2 . The first one includes the points in between the image of $t^*(c)$ and $t = 0$, while the second the points between $t^*(c)$ and $t = T$.

scenarios as the case for which the locus of discontinuity can be parametrized by multiple functions $t_i^*(c)$ (see figure 6.10). It is sufficient to split the integration at each discontinuity point encountered as done in (6.9).

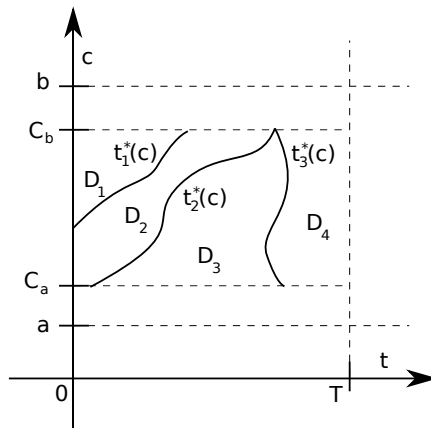


Figure 6.10: The curves $t_i^*(c)$ split the domain into multiple subsets D_i .

If we apply the theorem to the problem of optimal monitoring and select $f(t, \tilde{c}) = S(\tilde{x}(s, \tilde{c}_{j,k}), q^*(s, \tilde{c}_{j,k}))$ as in (6.7), we can conclude that $J(\tilde{c})$ is continuous as long as it is possible to parametrize the locus of discontinuity $\Gamma \subset \mathbb{R} \times \mathbb{R}^p$ as

$$\Gamma = \{(t, c) \text{ such that } t = t_i^*(c) \text{ for some continuous functions } t_i^* : \mathcal{C}_i \subset \mathbb{R}^p \mapsto \mathbb{R}\}.$$

The only setting where the theorem does not apply is when the discontinuity locus is parallel to the time axis for some nonzero interval, i.e. $f(t, c)$ jumps at least on one curve $c = \gamma(t)$ of the form

$$\gamma(t) = c^* \text{ fixed} \quad \forall t \in [t_a, t_b], \quad t_b > t_a.$$

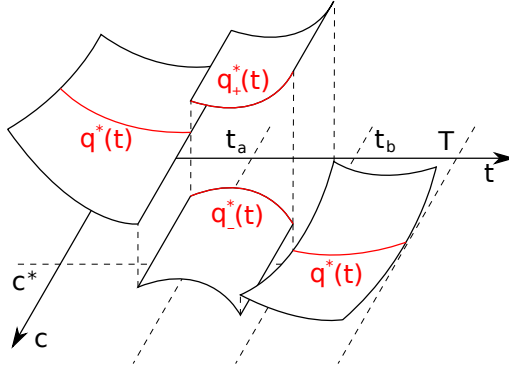


Figure 6.11: Discontinuity lying on a segment parallel to the time axis. For $t_a \leq t \leq t_b$ there are two minimizers, namely $q_-^*(t)$ and $q_+^*(t)$.

This case is however covered by the following observation.

Recall that $q^*(s, \tilde{c}_{j,k})$ is found as solution to the minimization problem (6.6). In particular let us consider a neighborhood of $c = c^*$. When $t < t_a$ or $t > t_b$ the minimizer is unique and we call it $q^*(t)$. For $t_a \leq t \leq t_b$ we call $q_-^*(t)$ and $q_+^*(t)$ the two minimizers corresponding to the right and left neighborhood of c^* . Let us call $Q_+^*(t)$ and $Q_-^*(t)$ the full path obtained following $q^*(t)$ for $t < t_a$ or $t > t_b$ and respectively $q_+^*(t)$ or $q_-^*(t)$ for $t_a \leq t \leq t_b$.

The situation is portrayed in figure 6.11.

Since $q_-^*(t)$ and $q_+^*(t)$ are found solving (6.6), the following must hold

$$\begin{aligned} \min_{q \in \Omega} \int_0^t S(\tilde{x}(s, \tilde{c}_{j,k}^*), q) ds &= \\ &= \int_0^t S(\tilde{x}(s, \tilde{c}_{j,k}^*), Q_-^*(t)) ds = \int_0^t S(\tilde{x}(s, \tilde{c}_{j,k}^*), Q_+^*(t)) ds \quad \text{for all } t \in [t_a, t_b]. \end{aligned} \quad (6.10)$$

Consequently $S(\tilde{x}(s, \tilde{c}_{j,k}^*), q_-^*(t)) = S(\tilde{x}(s, \tilde{c}_{j,k}^*), q_+^*(t))$, indeed equation (6.10) can be rewritten as

$$\int_0^t \left(S(\tilde{x}(s, \tilde{c}_{j,k}^*), Q_-^*(t)) - S(\tilde{x}(s, \tilde{c}_{j,k}^*), Q_+^*(t)) \right) ds = 0, \quad \forall t \in [t_a, t_b].$$

It follows from an important theorem of the calculus of variation [43] that the integrand function has to be zero almost everywhere in $[t_a, t_b]$, i.e.

$$S(\tilde{x}(s, \tilde{c}_{j,k}^*), q_-^*(t)) = S(\tilde{x}(s, \tilde{c}_{j,k}^*), q_+^*(t)), \quad a.e. s \in [t_a, t_b], \quad \forall t \in [t_a, t_b].$$

Consequently when computing J_{min} close to c^* , the integrand function S assumes the same values in a left or right neighborhood of c^* , and thus continuity is guaranteed also in this case.

Using theorem 6.4 and the observation just presented, we can conclude that the function J_{min} is continuous with respect to the coefficients. With a similar procedure one can show that J_{min} is also Lipschitz. Unfortunately $J_{min}(\tilde{c}_{j,k})$ is not differentiable. This is due to the fact that the derivatives of $q^*(s, \tilde{c}_{j,k})$ are involved when computing J'_{min} , and in the case just discussed the right and left differentials do not match at the point.

To conclude, we have transformed the differential game into a non-smooth optimization problem over a finite dimensional space. Thus appropriate tools from non-smooth analysis need to be introduced as it is done in the next section.

6.3.1 Non-smooth Optimization

Since standard optimization techniques cannot be applied to maximize J_{min} , we turn our attention to non-smooth optimization. The notion of generalized derivative is introduced, as well as the equivalent of the stationary condition $\nabla f(x) = 0$ for non smooth functions.

Traditional optimization techniques are based on the idea of descent direction, i.e. a direction on the domain space where the function decreases. In the case of smooth functions, the gradient provides such information and the hessian can also be used to improve the model. Basic algorithms exploiting these concepts are for instance the steepest descent or Newton's method [35]. In the non smooth case, the gradient does not exist at every point of the domain, making it difficult to find descent directions. Furthermore the stationary condition $\nabla f(x) = 0$, used to stop the above mentioned methods, does no longer apply as the gradient may not be defined at the optimal point. Non-smooth analysis extends the usual concept of derivative to these cases as in the following.

DEFINITION 6.5 (CLARKE GENERALIZED DERIVATIVE) *Let $f : \mathbb{R}^p \mapsto \mathbb{R}$ be a locally Lipschitz continuous function of $x \in \mathbb{R}^p$. The generalized directional*

derivative of f at x in the direction $d \in \mathbb{R}^p$ is given by

$$f^\circ(x; d) \doteq \limsup_{\substack{y \rightarrow x \\ h \downarrow 0}} \frac{f(y + hd) - f(y)}{h}. \quad (6.11)$$

More explicitly the previous definition can be written as

$$f^\circ(x; d) = \lim_{\epsilon \rightarrow 0} \left\{ \sup \left\{ r(y, h, d) : (y, h) \in \mathbb{R}^{p+1} \cap B((x, 0), \epsilon), h > 0 \right\} \right\},$$

where $r(y, h, d) = \frac{f(y+hd) - f(y)}{h}$ and $B((x, 0), \epsilon)$ is an ϵ neighborhood of the point $(x, 0) \in \mathbb{R}^{p+1}$. Loosely speaking, this means that $f^\circ(x; d)$ is found as the supremum amongst all the difference quotients $r(y, h, d)$ for y close to x , with $h \rightarrow 0^+$.

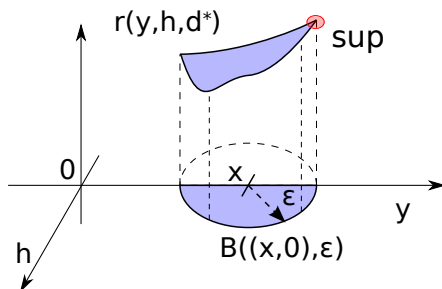


Figure 6.12: The construction used to define the Clarke generalized derivative (6.11), represented for a fixed value of the direction $d = d^*$.

In the case of a differentiable function this definition coincides with the usual one of directional derivative.

DEFINITION 6.6 (CLARKE SUBDIFFERENTIAL) Given $f : \mathbb{R}^p \mapsto \mathbb{R}$ a locally Lipschitz continuous function of $x \in \mathbb{R}^p$, the subdifferential of f at x is defined as the set $\partial f(x)$ of vectors $\xi \in \mathbb{R}^p$ such that

$$\partial f(x) = \{ \xi \in \mathbb{R}^p \mid f^\circ(x; d) \geq \xi^T d \text{ for all } d \in \mathbb{R}^p \}.$$

In particular the subdifferential represents the convex hull of all the possible gradients close to the point x , as the next theorem claims.

THEOREM 6.7 (SUBDIFFERENTIAL AS A CONVEX HULL) *Given $f : \mathbb{R}^p \mapsto \mathbb{R}$ a locally Lipschitz continuous function. Then the subdifferential $\partial f(x)$ is given by*

$$\partial f(x) = \text{conv}\{\xi \in \mathbb{R}^p \mid \nabla f(x_i) \rightarrow \xi, x_i \rightarrow x \text{ and } f \text{ is differentiable at } x_i\},$$

where $\text{conv}(S)$ indicates the convex hull of S . □

The subdifferential is an important tool in nonsmooth analysis due to the following theorem that provides a necessary condition for optimality.

THEOREM 6.8 (FIRST ORDER CONDITIONS) *Given $f : \mathbb{R}^p \mapsto \mathbb{R}$ a locally Lipschitz continuous function in $x \in \mathbb{R}^p$. If f attains a local maximum in x , then*

1. $0 \in \partial f(x)$,

2. $f^\circ(x; d) \leq 0 \forall d \in \mathbb{R}^p$. □

Theorem 6.8 and in particular point 1. can be regarded as the extension of the stationary condition $\nabla f(x) = 0$ to the case when $f(x)$ is non smooth. Note that this is a necessary but not sufficient condition, thus there could be points satisfying 1. and 2. without being maximizers. See [44] for the Clarke's seminal paper on generalized derivatives, or [45] for a more comprehensive picture.

Example (Absolute value)

Consider the function $f(x) = -|x|$. It is well known that f has a maximum in the point $x = 0$. The subdifferential is given by

$$\partial f(x) \begin{cases} = 1 & \text{for } x < 0, \\ \in [-1, 1] & \text{for } x = 0, \\ = -1 & \text{for } x > 0. \end{cases}$$

Clearly $0 \in \partial f(0)$ and the generalized derivative assumes the value $f^\circ(0; d) = -1$ for $x = 0$.

6.3.2 Preliminary results

Several methods have been presented in literature to solve non-smooth optimization problems (NSO). Methods for solving NSO can be divided into three categories: subgradient methods [46], bundle methods [45], and gradient sampling methods [47]. Recall that only the objective function value and one arbitrary generalized gradient can be computed at each point.

Subgradient methods are based on standard gradient methods (such as steepest descent), where the idea is to replace the gradient with an arbitrary subgradient. They are extremely simple and thus widely used even though they may suffer from some serious drawbacks. An extensive overview of various subgradient methods can be found in [46].

Bundle methods are considered more effective and reliable than subgradient's one. They are based on the subdifferential theory we presented here, where first order necessary conditions can be described in terms of $\partial f(x)$. The main idea of bundle methods is to approximate the subdifferential set (the set of gradients) of the objective function by storing subgradients from previous iterations into a bundle. In this way, more information is obtained when compared to the use of a single subgradient [45].

The newest approach is to use gradient sampling algorithms developed by Burke, Lewis and Overton [47]. The gradient sampling method applies to nonsmooth and/or nonconvex problems. Gradient sampling methods are based on the steepest descent algorithm, with the main point being the approximation of the subdifferential through sampling of gradients near the point of interest.

Let us turn the attention to the non smooth optimization problem arising from the combination of optimal monitoring and wavelet. In the following plot is represented the function $J_{min}(\tilde{c}_{j,k})$, where the trajectory $x(s)$ has been decomposed using only two Haar Wavelet's functions with coefficients \tilde{c}_1 and \tilde{c}_2 . This is the maximum dimension that can be visualized. As previously proved, one can verify here that the objective function is continuous, Lipschitz but non differentiable.

Hamiltonian steepest descent

Due to time constraints, it has not been possible to implement any of the advanced methods just mentioned. On the contrary a modified version of the Steepest descent algorithm has been tested in low dimension. The method is presented in the following. Consider the problem of maximizing a function $J_{min}(\tilde{c}) : \tilde{C} \subset \mathbb{R}^p \mapsto \mathbb{R}$. This is equivalent to minimizing the function $f(\tilde{c}) = -J_{min}(\tilde{c})$.

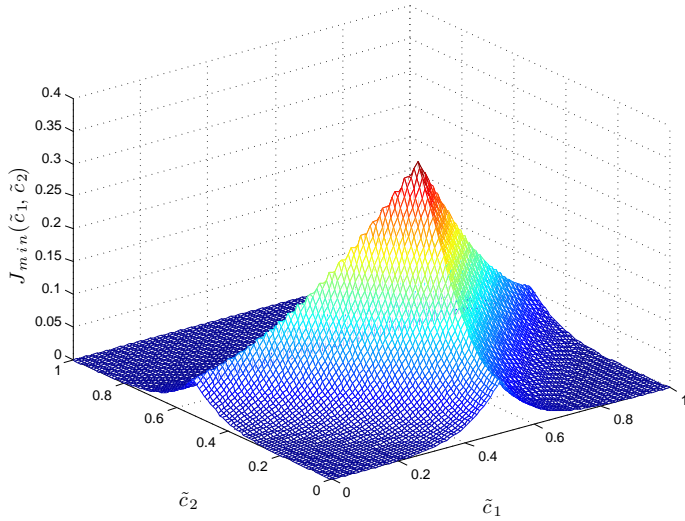


Figure 6.13: Objective function J_{min} obtained using the first two elements of the Haar Wavelet's basis. Note that the maximizer is exactly a point where the gradient is not defined.

In analogy to mechanics we consider the following dynamical system

$$\begin{cases} \dot{\tilde{c}} = v, \\ \dot{v} = -\frac{\partial f}{\partial \tilde{c}} - \epsilon v, \end{cases} \quad (6.12)$$

where a particle with position \tilde{c} is moving under the effect of the potential $f(\tilde{c})$ and friction $-\epsilon v$.

The total energy of the system is

$$H(c, v) = \frac{v^2}{2} + f(\tilde{c}),$$

which can easily be proven to decrease along the trajectories of (6.12).

Indeed one has

$$\dot{H}(\tilde{c}, v) = v\dot{v} + \frac{\partial f}{\partial \tilde{c}}v = v\left(-\frac{\partial f}{\partial \tilde{c}} - \epsilon v\right) + v\frac{\partial f}{\partial \tilde{c}} = -\epsilon v^2,$$

this means that the sum of $f(\tilde{c})$ and $v^2/2$ decreases with time. Since $H(\tilde{c}, v)$ has a minimum in $(\tilde{c}^*, 0)$, where \tilde{c}^* is the minimizer of $f(\tilde{c})$, we conclude that H is a Lyapunov function and the dynamics (6.12) evolves towards this point. In such a way we have constructed a method that is guaranteed to converge to

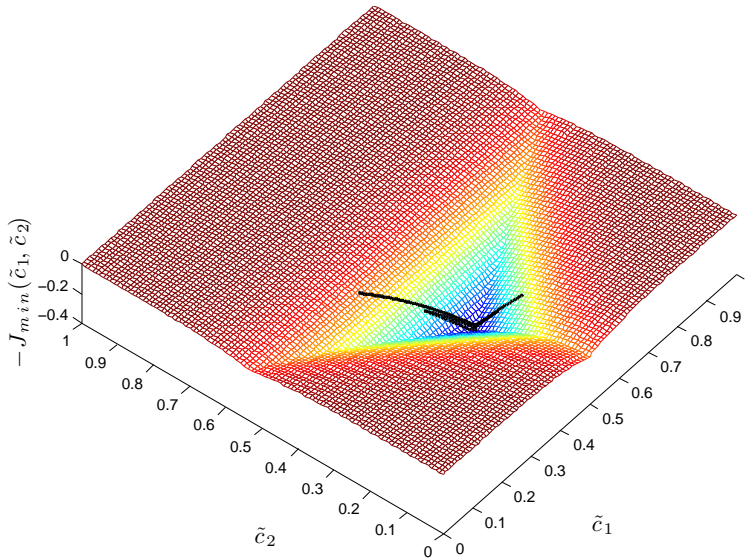


Figure 6.14: Objective function $-J_{min}$ and a trajectory obtained with the method just presented. Note that the solution oscillates around the solution before being attracted thanks to the term $-\epsilon v$.

the minimum of $f(\tilde{c})$, i.e. to the maximum of $J_{min}(\tilde{c})$. Recall that the gradient of $f(\tilde{c})$ is not defined for all the points of the domain, but it exists almost everywhere [45]. This is sufficient to give sense to the differential equation (6.12) in the Caratheodory sense [48]. When such a method is applied to the case of figure 6.13, the solution is easily found as picture 6.14 shows. An even simpler version of the method presented here can be obtained from the continuous steepest descent, where equations (6.12) are substituted with $\dot{\tilde{c}} = -\frac{\partial f}{\partial \tilde{c}}$. This has the drawback that when a nondifferentiable curve for $f(\tilde{c})$ is found, the method very often gets stuck or becomes slow as it jumps from one side to the other of the “valley”. On the contrary (6.12) has the advantage that some energy is stored as velocity in the term $v^2/2$, and it is possible to avoid following non differentiable paths.

Conclusions and future work

In this work the problem of optimal monitoring for UAVs has been introduced using the notion of coverage map and elements of game theory. Two different formulations have been presented.

In the first one, the monitoring problem has been rewritten in terms of a standard differential game, for which the Hamilton Jacobi Bellman equations provide a sufficient condition. The first new contribution is given by the numerical results of the aforementioned system. They have proven this approach to be useful when performing surveying but not when patrolling or surveillance are involved.

For this reason a novel formulation based on non standard differential game has been introduced. Agents are continuously asked to look for the less visited points and to increase the coverage in there. The problem is more difficult to attack since game theory provides neither a sufficient nor a necessary condition for this case. Two different solutions techniques have been developed.

The first one is based on the reduction of the differential game to a series of optimal control problems and has produced interesting results for one and two dimensional surveillance applications.

The second method is rather new and leverages on the opportunity to approximate an optimization problem over an infinite dimensional function space with a similar problem in a finite dimensional coefficient space. Wavelets have been selected as basis functions, due to their approximating properties and the resulting objective function has been proven to be Lipschitz but not differentiable. Consequently non-smooth optimization has been considered and a basic method

has been implemented in low dimension.

Future work in this direction consists in implementing an advanced numerical solver such as the gradient sampling method, and to use it to find the solution of the finite dimensional differential game.

Another development consists in further exploiting the structure of the game. The formulation is indeed similar to the one of max-min problems even though it is not exactly the same. The study could provide valuable information and restrict the search for maximizers to some specific points (conjecture: the points of non differentiability of J_{min}).

Finally it would be interesting to take into account and model the difficulties in the communications amongst agents. In this work indeed we have assumed perfect communication and designed a centralized control.

Bibliography

- [1] P. Fahlstrom and T. Gleason, *Introduction to UAV systems*. John Wiley & Sons, 2012.
- [2] K. Ro, J.-S. Oh, and L. Dong, “Lessons learned: Application of small uav for urban highway traffic monitoring,” in *45th AIAA Aerospace Sciences Meeting and Exhibit*, pp. 2007–596, 2007.
- [3] A. Ganguli, S. Susca, S. Martinez, F. Bullo, and J. Cortes, “On collective motion in sensor networks: sample problems and distributed algorithms,” in *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC '05. 44th IEEE Conference on*, pp. 4239–4244, Dec 2005.
- [4] T. Mylvaganam and A. Astolfi, “Approximate optimal monitoring: preliminary results,” in *American Control Conference (ACC), 2012*, pp. 4745–4750, IEEE, 2012.
- [5] I. I. Hussein and D. M. Stipanovic, “Effective coverage control for mobile sensor networks with guaranteed collision avoidance,” *Control Systems Technology, IEEE Transactions on*, vol. 15, no. 4, pp. 642–657, 2007.
- [6] A. Bressan and B. Piccoli, *Introduction to the mathematical theory of control*, vol. 2. American institute of mathematical sciences Springfield, 2007.
- [7] V. Arnold, *Ordinary Differential Equations*. Springer-Textbook, New York, 1992.
- [8] J. D. Meiss, *Differential dynamical systems*, vol. 14. Siam, 2007.
- [9] R. Vinter, *Optimal Control*. Modern Birkhäuser Classics, Springer, 2010.

-
- [10] D. Bertsekas, *Dynamic Programming and Optimal Control*. No. v. 2 in Athena Scientific optimization and computation series, Athena Scientific, 2007.
- [11] M. Crandall, H. Ishii, and P. Lions, *User's Guide to Viscosity Solutions of Second Order Partial Differential Equations*. American Mathematical Society, 1992.
- [12] D. Liberzon, *Calculus of variations and optimal control theory: a concise introduction*. Princeton University Press, 2012.
- [13] J. Neumann and O. Morgenstern, *Theory of games and economic behavior*. Princeton, NJ: Princeton Univ. Press, 3. ed. ed., 1953.
- [14] R. Bellman, "Dynamic programming and lagrange multipliers," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 42, no. 10, p. 767, 1956.
- [15] L. S. Pontryagin, *Mathematical theory of optimal processes*. CRC Press, 1987.
- [16] F. Lanchester, "Mathematics in warfare," *The World of Mathematics*, vol. 4, 1956.
- [17] T. Basar and G. J. Olsder, *Dynamic noncooperative game theory*, vol. 200. SIAM, 1995.
- [18] A. Cournot, N. Bacon, and I. Fisher, *Researches Into the Mathematical Principles of the Theory of Wealth*. Economic classics, Macmillan, 1897.
- [19] J. F. Nash *et al.*, "Equilibrium points in n-person games," *Proceedings of the national academy of sciences*, vol. 36, no. 1, 1950.
- [20] H. Von Stackelberg, D. Bazin, R. Hill, and L. Urch, *Market structure and equilibrium*. Springer, 2010.
- [21] A. Bressan, "Noncooperative differential games. a tutorial," *Department of Mathematics, Penn State University*, 2010.
- [22] A. Haurie, "Feedback equilibria in differential games with structural and modal uncertainties," in *Advances in Large Scale Systems*, Citeseer, 1984.
- [23] M. Simaan and J. B. Cruz Jr, "Additional aspects of the stackelberg strategy in nonzero-sum games," *Journal of Optimization Theory and Applications*, vol. 11, no. 6, pp. 613–626, 1973.
- [24] T. H. Cox, C. J. Nagy, M. A. Skoog, I. A. Somers, and R. Warner, "Civil uav capability assessment," *NASA, Tech. Rep., draft Version*, 2004.

- [25] S. A. Cambone, *Unmanned aircraft systems roadmap 2005-2030*. Defense Technical Information Center, 2005.
- [26] Z. Drezner, *Facility location: a survey of applications and methods*. Springer Verlag, 1995.
- [27] A. Okabe and A. Suzuki, "Locational optimization problems solved through voronoi diagrams," *European Journal of Operational Research*, vol. 98, no. 3, pp. 445–456, 1997.
- [28] Q. Du, V. Faber, and M. Gunzburger, "Centroidal voronoi tessellations: applications and algorithms," *SIAM review*, vol. 41, no. 4, pp. 637–676, 1999.
- [29] C. G. Cassandras and W. Li, "Sensor networks and cooperative control," *European Journal of Control*, vol. 11, no. 4, pp. 436–463, 2005.
- [30] W. Li and C. G. Cassandras, "Distributed cooperative coverage control of sensor networks," in *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC'05. 44th IEEE Conference on*, pp. 2542–2547, IEEE, 2005.
- [31] Y. Zou and K. Chakrabarty, "Sensor deployment and target localization based on virtual forces," in *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, vol. 2, pp. 1293–1303, IEEE, 2003.
- [32] S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M. B. Srivastava, "Coverage problems in wireless ad-hoc sensor networks," in *in IEEE INFOCOM*, pp. 1380–1387, 2001.
- [33] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," in *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, vol. 2, pp. 1327–1332, IEEE, 2002.
- [34] P. R. Chandler, M. Pachter, and S. Rasmussen, "Uav cooperative control," in *American Control Conference, 2001. Proceedings of the 2001*, vol. 1, pp. 50–55, IEEE, 2001.
- [35] S. Wright and J. Nocedal, *Numerical optimization*, vol. 2. Springer New York, 1999.
- [36] R. Courant, E. Isaacson, and M. Rees, "On the solution of nonlinear hyperbolic differential equations by finite differences," *Communications on Pure and Applied Mathematics*, vol. 5, no. 3, pp. 243–255, 1952.

-
- [37] I. M. Mitchell, “A toolbox of level set methods,” *Dept. Comput. Sci., Univ. British Columbia, Vancouver, BC, Canada*, <http://www.cs.ubc.ca/~mitchell/ToolboxLS/toolboxLS.pdf>, *Tech. Rep. TR-2004-09*, 2004.
- [38] R. Buckdahn, P. Cardaliaguet, and M. Quincampoix, “Some recent aspects of differential game theory,” *Dynamic Games and Applications*, vol. 1, no. 1, pp. 74–114, 2011.
- [39] R. J. LeVeque and R. J. Le Veque, *Numerical methods for conservation laws*, vol. 132. Springer, 1992.
- [40] I. M. Gelfand, I. M. Gelfand, and S. Fomin, *Calculus of variations*. Courier Dover Publications, 2000.
- [41] C.-H. Hsiao, “Haar wavelet direct method for solving variational problems,” *Mathematics and computers in simulation*, vol. 64, no. 5, pp. 569–585, 2004.
- [42] O. Christensen, *Functions, spaces, and expansions: mathematical tools in physics and engineering*. Springer, 2010.
- [43] P. R. Halmos, P. R. Halmos, P. R. Halmos, P. R. Halmos, and H. Mathematician, *Measure theory*, vol. 2. van Nostrand New York, 1950.
- [44] F. H. Clarke, *Optimization and nonsmooth analysis*, vol. 5. Siam, 1990.
- [45] M. M. Makela and P. Neittaanmaki, *Nonsmooth optimization*. World Scientific, 1992.
- [46] N. Z. Shor, K. C. Kiwiel, and A. Ruszcay ski, *Minimization methods for non-differentiable functions*. Springer-Verlag New York, Inc., 1985.
- [47] J. V. Burke, A. S. Lewis, and M. L. Overton, “Approximating subdifferentials by random sampling of gradients,” *Mathematics of Operations Research*, vol. 27, no. 3, pp. 567–584, 2002.
- [48] D. Biles and P. Binding, “On carathéodory’s conditions for the initial value problem,” *Proceedings of the American Mathematical Society*, vol. 125, no. 5, pp. 1371–1376, 1997.