



Technische Universität München  
Zentrum Mathematik  
Wissenschaftliches Rechnen

# Dynamic programming with radial basis functions and Shepard's method

Alex Joachim Ernst Schreiber

Vollständiger Abdruck der von der Fakultät für Mathematik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitzender: Univ.-Prof. Dr. F. Kraemer

Prüfer der Dissertation: 1. Univ.-Prof. Dr. O. Junge

2. Univ.-Prof. Dr. M. Gerdtz,

Universität der Bundeswehr München

Die Dissertation wurde am 22.10.2015 bei der Technischen Universität München eingereicht und durch die Fakultät für Mathematik am 22.01.2016 angenommen.



# Abstract

In this thesis, we investigate a discretization of the optimality principle in dynamic programming based on radial basis functions and Shepard's moving least squares approximation method. We prove the convergence of the discrete value function for increasingly dense sets of centres, develop an adaptive version of the algorithm and generalize a Dijkstra-like algorithm of Bertsekas which allows an efficient calculation of the value function. We illustrate the theory with numerous numerical experiments.

# Zusammenfassung

Diese Arbeit beschäftigt sich mit einer Diskretisierung des Optimalitätsprinzips der dynamischen Programmierung, welche auf radialen Basisfunktionen und Shepards Methode der beweglichen Kleinste-Quadrate-Approximation basiert. Wir zeigen die Konvergenz der diskreten Wertefunktion für zunehmend dichtere Zentrenmengen, entwickeln eine adaptive Variante des Algorithmus und verallgemeinern einen Dijkstra-artigen Algorithmus von Bertsekas zur effizienten Berechnung der Wertefunktion. Wir illustrieren die Theorie anhand zahlreicher numerischer Experimente.



# Acknowledgements

First, I would like to thank my supervisor Prof. Dr. Oliver Junge for introducing me to this interesting topic, for his continuous support and for countless helpful and interesting discussions. Thank you for your confidence, help and advice at every stage of my PhD.

I am grateful to Prof. Dr. Roberto Ferretti, Prof. Dr. Lars Grüne, Dr. Péter Koltai and Prof. Dr. Holger Wendland for helpful discussions and to Stephanie Troppmann, Andreas Bittracher, Dr. Daniel Karrasch and Michael Kratzer for proofreading.

Finally, I want to thank all the members of the M3 research unit at the Technische Universität München for the friendly and constructive atmosphere over the years. Working in this group was and continues to be a great experience.



# Contents

<b>0. Introduction</b>	<b>9</b>
<b>1. Preliminaries</b>	<b>13</b>
1.1. Discrete-time control systems	13
1.2. Dynamic programming	16
1.3. Continuous-time control systems	19
1.4. Radial basis functions	20
<b>2. The Shepard discretization of the optimality principle</b>	<b>25</b>
2.1. Discretization of the optimality principle	25
2.2. Stationary vs. non-stationary approximation	28
2.3. Convergence for decreasing fill distance with discounting	29
2.4. Implications of (non-)discounting for dynamic programming	32
2.5. Convergence for decreasing fill distance with the Kružkov transform	33
2.6. Construction of a stabilizing feedback	36
2.7. Summary and outlook	40
<b>3. Implementation and numerical examples</b>	<b>43</b>
3.1. Implementation	43
3.2. Example: A simple 1D example	44
3.3. Example: Shortest path with obstacles	45
3.4. Example: An inverted pendulum on a cart	46
3.5. Example: Magnetic wheel	50
3.6. Dependence on the shape parameter	53
3.7. Dependence on the set of centres	53
3.8. Dependence on the shape function	56
<b>4. Adaptive choice of the centres</b>	<b>61</b>
4.1. Adaptive construction of the centres	61
4.2. Convergence of the algorithm	63
4.3. Implementation	67
4.4. Numerical examples	67
4.4.1. A 1D example with non-smooth solution	67
4.4.2. A basic growth model with explicit solution	69
4.4.3. An inverted pendulum on a cart	71
4.4.4. An office robot	71

*Contents*

<b>5. Shepard value iteration and stochastic shortest path problems</b>	<b>75</b>
5.1. Shortest path problems and Dijkstra's algorithm . . . . .	76
5.2. Stochastic shortest path problems . . . . .	77
5.3. A Dijkstra-like algorithm for stochastic shortest path problems . .	80
5.4. Application on continuous-time control systems . . . . .	85
<b>A. Regularity of the value function without discounting</b>	<b>89</b>
A.1. Lipschitz continuity of the value function . . . . .	89
A.2. Nondifferentiability of the value function . . . . .	96
<b>Bibliography</b>	<b>101</b>

# 0. Introduction

The present thesis presents an approach to efficiently approximating optimal value functions of optimal control problems.

An *optimal control problem* poses the task of finding optimal solutions for *control systems*, which are a generalization of dynamical systems. Dynamical systems are ubiquitous when considering physical or technical systems where they describe the changing state of a system over time. If, in addition, the system can be influenced (by a *control function*) in the course of time, one has a control system. Usually there is also a *cost function* which, like the *dynamics*, depends on the *state* and the *control* at the particular time.

The objective is either steering the system towards a certain state, the *target*, or minimizing the total cost, which accumulates over time.

If one tries to achieve this objective not just for one specific initial state of the system, but for all possible initial conditions at once, an appropriate tool is the (optimal) *value function*. It describes the smallest possible total cost in dependence on the initial condition.

The most concrete examples of optimal control problems are shortest-path problems, which can be imagined as the situation where a virtual driver tries to move a vehicle on a plane (possibly with obstructions) from a given starting position to a certain target on the shortest path or, more generally, in minimal time if the velocity is not constant.

Here, the state space is the plane, the possible controls are the allowed speed vectors and the cost can be constant or e.g. the kinetic energy.

There is an important distinction of control systems whether they are considered in continuous or in discrete time. Our methods apply to the discrete-time setting, but continuous-time control systems can be discretized by standard methods (any numerical integrator for ODEs) from numerical analysis.

Among the approaches towards optimal control problems in continuous time stand out two major methods, Pontryagin's maximum principle ([Pon87]) which gives a necessary condition for an optimal value function, and the Hamilton-Jacobi-Bellman (HJB) equation ([Bel54]) which provides both a necessary and sufficient condition.

This thesis deals exclusively with methods related to the latter approach because it has an analogue in discrete time, the *Bellman equation*. It has the form of a fixed point equation for the optimal value function of the system. In its most

## 0. Introduction

basic and simplified form (without a control input) it splits the value function

$$V(x_0) = \sum_{i=0}^{\infty} c(x_i)$$

along a (discrete) trajectory  $x_0, x_1, x_2, \dots$  with  $x_{k+1} = f(x_k)$  into

$$V(x_0) = c(x_0) + V(x_1) = c(x_0) + V(f(x_0))$$

where  $f$  is the dynamics.

More generally (if  $c$  also depends on a control input  $u$ ), it is given by ([Bel57])

$$V(x) = \inf_{u \in U} [c(x, u) + V(f(x, u))].$$

The iteration with the corresponding operator

$$\Gamma: V \mapsto \Gamma(V), \quad \Gamma(V)(x) = \inf_{u \in U} [c(x, u) + V(f(x, u))]$$

is called *dynamic programming*.

From the value function it is possible to construct a policy for the choice of optimal controls, a *feedback*, which assigns a control to each point of the state space. If this feedback is plugged into the control system one gets a closed-loop system: a dynamical system with a cost functional. Without *discounting*, it is also possible to approximate the maximal *domain of stability* (points which can be steered towards the target) for this closed-loop system by finding the subset of the state space where the value function is finite. However, most authors consider control systems with discounting, which assigns smaller weights to summands of the cost functional which are added at a later time.

The state space is usually infinite and the Bellman operator nonlinear because of the infimum on the right-hand side. For these reasons control problems do not allow for an exact analytical solution, in general.

Although methods from nonlinear [Cla04] and nonsmooth [CLSW98] analysis might be tried to deal with the non-linearity, usually some kind of space discretization is necessary. The methods which have been used so far include finite difference ([CD83, BCD97]), finite element ([Fal87]) and, more recently, set-oriented methods ([JO04, GJ05]).

Usually their discretization suffers from the *curse of dimensionality*: The numerical effort scales exponentially with the dimension of the problem, because the volume of the state space and hence the size of the discrete state space scale exponentially.

We do not use a discretization of the state space, but instead discretize the space of functions on the state space. To this end we employ linear function

spaces generated by *radial basis functions* (RBF), radially symmetric functions with distinct centres of symmetry.

Usually, one uses translations of a fixed *shape function*, but in some cases one scales the translated functions differently (in spatial direction). Radial basis functions have the form  $\phi_i(x) = \varphi(\sigma\|x - x_i\|)$ , and hence the functions in the RBF space have the form

$$\sum_{i=1}^n \alpha_i \varphi(\sigma\|x - x_i\|), \quad \alpha_i \in \mathbb{R}.$$

Here,  $x_1, \dots, x_n$  are the *centres*,  $\varphi$  is the shape function,  $\sigma$  is the *shape parameter* and the  $\alpha_i$  are the coefficients of the linear combination.

Owing to their self-evident definition, radial basis functions have been used for a long time; recently, interest grew in *compactly supported* radial basis functions, particularly due to the extensive work of Schaback and Wendland ([Sch95, Wen95, SW00b, SW00a, Wen04]).

**What is interesting about radial basis functions?** On the one hand, the discretization of control problems by the use of radial basis functions can be implemented efficiently. This is possible because the radial symmetry of the basis functions allows for the use of *distance matrices* (the matrix of pairwise distances of a set of points) which are sparse for basis functions with compact support.

However, the main advantage of radial basis functions concerns the curse of dimensionality. The discretization is meshfree, at least in principle, in other words, it does not rely on a regular grid, but basically any point set  $\{x_1, \dots, x_n\}$  can be used, though with different quality of the obtained approximation.

Note, however, that per se this does not avoid the curse of dimensionality: For general optimal control problems, it might be impossible to avoid the curse of dimensionality by using radial basis functions with compact support. The volume of their support in relation to the volume of the state space shrinks exponentially so that the number of basis functions which are required for a good approximation grows exponentially. If one uses basis functions with global support, this argument does no longer hold, but there is no advantage to be expected in compensation for the increased numerical effort.

However, radial basis functions can help to alleviate the curse of dimensionality, under the requirement that the centres are distributed according to the relevance of different regions to the control problem. An appropriate distribution of the centres can be chosen a priori for specific control problems; or they can be chosen “automatically” by an adaptive algorithm. The latter approach will also be presented in this thesis.

Radial basis functions have been used with Hamilton-Jacobi-Bellman equations and the related and generally better-known Hamilton-Jacobi equations, e.g. in

## 0. Introduction

[CQO04, HWCL06, AWJR12]. Here we consider radial basis functions in the discrete-time setting, i.e. for the Bellman equation.

For our implementation it is necessary to project functions onto a finite-dimensional approximation space. In most applications of radial basis functions, one uses interpolation at the centres of the basis functions for this projection. However, we deal with a different approach, namely a moving least squares approximation method under the name *Shepard's method* ([She68]) which essentially approximates function values by a convex combination of function values at nearby centres. One of the main advantages of Shepard's method compared to interpolation with RBFs is that the projection with the corresponding Shepard operator is (under additional assumptions) contractive and thus allows for the application of Banach's fixed point theorem.

This thesis is organized as follows.

Chapter 1 gives a brief review on control systems, radial basis functions and Shepard's method.

Chapter 2 is the core of this thesis. We derive the Shepard-RBF discretization of the optimality principle for optimal control problems and prove the convergence of the value iteration scheme. We also prove the convergence of the approximate value function to the exact value function for increasingly dense centre sets. For this proof, one has to distinguish between discounted and undiscounted control systems. Two crucial aspects are that the regularity of the value function (which we have postponed to the appendix) and the dependence of the value function on the target set have to be addressed separately for undiscounted systems. In a subsequent section we develop statements about the constructed approximate optimal feedback and the stability region of the corresponding closed-loop system.

In Chapter 3 we describe the implementation of the algorithm and give several numerical experiments. We also vary some of the parameters and choices from the numerical experiments and compare the convergence behaviour.

Chapter 4 is devoted to the investigation of a variant of the dynamic programming algorithm where an adaptive refinement of the set of centres is used. In numerical experiments we compare the convergence speed with the one from the non-adaptive procedure.

In Chapter 5, we apply a slight generalization of Bertsekas' Dijkstra-like algorithm ([Ber07]) for stochastic shortest path problems on our discretizations of control systems. This is possible because the Bellman equation under the Shepard-RBF discretization corresponds to a fixed point equation obtained for stochastic shortest path problems.

# 1. Preliminaries

## 1.1. Discrete-time control systems

Throughout this thesis we consider time-invariant, or autonomous, discrete-time control systems of the form

$$x_{k+1} = f(x_k, u_k), \quad k = 0, 1, 2, \dots \quad (1.1)$$

Here,  $f: \Omega \times U \rightarrow \Omega$  is a continuous map defined on compact sets  $\Omega \subset \mathbb{R}^s$ ,  $0 \in U \subset \mathbb{R}^d$ , referred to as *state space* and *control space*, respectively. In addition, we assume that a continuous *cost function*  $c: \Omega \times U \rightarrow [0, \infty)$  is given as well as a constant *discount rate*  $0 < \beta = \beta_D \leq 1$ . We use the notation  $\beta_D$  if it is important to distinguish it from  $\beta_K$ , which will appear in (1.10).

Consider the cost functional

$$J(x_0, (u_k)) := \sum_{k=0}^{\infty} \beta^k c(x_k, u_k)$$

which assigns the total cost along the trajectory to the initial point  $x_0$  and the control sequence  $(u_k)_{k \in \mathbb{N}_0}$ . The *optimal value function*, or simply *value function*, is defined by

$$V(x_0) := \inf_{(u_k)} J(x_0, (u_k)), \quad (1.2)$$

i.e. the greatest lower bound on the total cost for trajectories starting from  $x_0$ . Note that both  $J$  and  $V$  have values in  $[0, \infty]$ , so they may attain the value  $\infty$ . It is a priori not clear whether the infimum on the right-hand side is attained for some control sequence.

### Finite horizon approximation and discounting

A control system is *discounted* if the discount rate  $\beta$  is less than one, and *undiscounted* if  $\beta = 1$ . In general, we are more interested in undiscounted systems, although they are more difficult to treat both in theory and numerically.

As an approximation to undiscounted systems, one often considers the finite horizon approximation

$$J_K(x_0, (u_k)) := \sum_{k=0}^K c(x_k, u_k)$$

## 1. Preliminaries

for some large fixed integer  $K$ . This is a rough simplification which makes computation easy as long as  $K$  is not too large. Moreover, it also directly implies the continuity of the corresponding optimal value function in a straight-forward way.  $J_K$  is continuous as a composition of continuous maps and the correspondingly defined  $V_K$  is continuous because the space  $U^{K+1}$  of finite control sequences is compact. It is also easy to give an a priori bound, namely  $V_K \leq K \max_{\Omega \times U} c(x, u)$ . The finite horizon approximation has, besides its approximation error, the disadvantage that the resulting optimal value function does not fulfil the optimality principle (1.4). For this reason, we do not investigate it further.

However, the use of a discount rate  $0 < \beta < 1$  shares some similarity with the use of the finite horizon approximation, as both approximations give lower weight to those summands of the cost functional which arise at later times of the trajectory. With a discount rate the weights are  $1, \beta, \beta^2, \dots$  while for the finite horizon approximation it is  $1, \dots, 1, 0, \dots$ .

Discounting also leads to a bounded value function with an a priori bound. In fact, it is immediate that one has  $V \leq \max_{\Omega \times U} c(x, u)(1 + \beta + \beta^2 + \dots) = \max_{\Omega \times U} c(x, u) \frac{1}{1-\beta}$ .

### Equilibrium points and target sets

In the undiscounted case, finiteness of  $V$  can not be expected generally. A necessary condition for finiteness is that  $\lim_{k \rightarrow \infty} c(x_k, u_k) = 0$  for some control sequence  $(u_k)$ .

To avoid infiniteness of  $V$  everywhere in  $\Omega$  one requires the existence of an *equilibrium point*, which by convention is the point  $0 \in \Omega$ , and fulfils  $f(0, 0) = 0$  and  $c(0, 0) = 0$ . Here, obviously  $V(0) = 0$  for the optimal control sequence  $u_k \equiv 0$ .

For many problems, one also requires a compact *target set*  $T \subset \Omega$  and that  $c(x, u)$  is bounded from below by a constant  $\delta > 0$  for all  $x \in \Omega \setminus T$  and  $u \in U$ . If one assumes  $c(x, u) > 0$  for all  $0 \neq x \in \Omega, u \in U$  and  $T$  has  $0$  as an interior point, then the existence of such a  $\delta > 0$  is immediate because then  $0 \in \overset{\circ}{T}$ , so  $0$  is not in the compact set  $\Omega \setminus \overset{\circ}{T}$ , which implies  $\min_{(\Omega \setminus \overset{\circ}{T}) \times U} c(x, u) > 0$ .

One then redefines  $J$  and  $V$  as if the cost function were zero on all of  $T$ , specifically  $c(x, u) = 0$  for all  $x \in T, u \in U$ , and as if the trajectory can be kept in  $T$  after this set has been reached so that  $V_T \equiv 0$ .

So the accumulated cost is

$$J(x_0, (u_k)) := \sum_{k=0}^{K^*-1} c(x_k, u_k), \quad K^* = \inf\{k \in \mathbb{N} \mid x_k \in T\},$$

and again  $V(x_0) := \inf_{(u_k)} J(x_0, (u_k))$ . Note the difference from the finite horizon approximation where  $K$  is a constant while here,  $K^* = K^*(x_0)$  depends on the trajectory.

In order to unify our notation, we set  $T = \{\}$  whenever no target set is given. Alternatively, one could choose  $T = \{0\}$  with the same effect.

For some problems, one is actually not interested in the accumulated cost but only in steering the system towards the target. Then the goal is to design control sequences for each initial point  $x_0$  that stabilize the system in the sense that discrete trajectories defined by  $x_{k+1} = f(x_k, u_k), k = 0, 1, 2, \dots$ , reach  $T$  in a finite number of steps for all initial conditions  $x_0$  in a maximal subset  $\mathcal{S} \subset \Omega$ , the region of stabilization. Points  $x_0$  which allow such control sequences are said to be *controllable* to the target.

In those cases a cost function  $c$  need not be given in the problem formulation. In order to use the methods presented in this thesis to tackle control problems, it can be adequate to construct “artificial” cost functions. A generic choice is  $c(x, u) = C_x \|x\|^2 + C_u \|u\|^2$  for constants  $C_x, C_u > 0$ , the easiest way to define a function with  $c(0, 0) = 0$  and otherwise  $c(x, u) > 0$  for  $(x, u) \in \Omega \times U$ .

If one considers discounted systems, there is no need for stabilization or for an equilibrium point, let alone a target set. Still, in practice there will usually be an equilibrium point because discounted systems are in most cases approximations to undiscounted systems.

## Feedbacks

So far, we considered the situation where control sequences  $(u_k) \in U^{\mathbb{N}}$  had to be chosen in dependence on a given initial point  $x_0$  with the aim to obtain a low accumulated cost.

The definition of the trajectory and the accumulated cost is such that the trajectory “has no memory”: The optimal control  $u_k$  in the  $k$ th step only depends on  $x_k$ , but neither on the way on which the trajectory reached  $x_k$  nor on the number  $k$  of steps from  $x_0$  to  $x_k$ . This is true even in the discounted case in accordance with the fact that the discounting sequence  $(1, \beta, \beta^2, \dots)$  has been chosen as a geometric sequence: If the point  $x_k$  is given, the task of minimizing

$$\sum_{i=k}^{\infty} \beta^i c(x_i, u_i)$$

is essentially the same as minimizing

$$\sum_{i=0}^{\infty} \beta^i c(\tilde{x}_i, \tilde{u}_i)$$

for the initial point  $\tilde{x}_0 := x_k$ ; they just differ by the constant rate  $\beta^k$ .

For that reason, one can restrict to the search for a map  $\mathbf{u}: \Omega \rightarrow U$  called *feedback*, and choose  $u_k := \mathbf{u}(x_k)$  to define a control sequence implicitly. So the iteration (1.1) changes to

$$x_{k+1} = f(x_k, \mathbf{u}(x_k)), \quad k = 0, 1, 2, \dots, \quad (1.3)$$

## 1. Preliminaries

A feedback turns a control system into a *closed-loop system*, i.e. a dynamical system  $x \mapsto f(x, \mathbf{u}(x))$  without control.

In the next section, we provide a rigorous argument why it is possible to restrict to feedbacks in order to find optimal control sequences.

## 1.2. Dynamic programming

In order to construct such a feedback, we utilize the *optimality principle*, which is given by the *Bellman equation*, cf. [Bel57, Ber05],

$$V(x) = \inf_{u \in U} [c(x, u) + \beta V(f(x, u))], \quad x \in \Omega \setminus T, \quad (1.4)$$

where  $V : \Omega \rightarrow [0, \infty]$  is the optimal value function. This equation is valid because the accumulated cost for  $x$  consists of the cost in the first step  $c(x, u)$  and the accumulated cost from  $f(x, u)$ , discounted by the factor  $\beta$ . It can be derived from the definition of  $V$ :

$$\begin{aligned} V(x_0) &= \inf_{(u_k)} J(x_0, (u_k)) \\ &= \inf_{(u_k)} \sum_{k=0}^{\infty} \beta^k c(x_k, u_k) \\ &= \inf_{(u_k)} [c(x_0, u_0) + \sum_{k=1}^{\infty} \beta^k c(x_k, u_k)] \\ &= \inf_{u_0} [c(x_0, u_0) + \beta \inf_{(u_k)_{k \geq 1}} \sum_{k=0}^{\infty} \beta^k c(x_{k+1}, u_{k+1})] \\ &= \inf_{u_0} [c(x_0, u_0) + \beta V(x_1)] \\ &= \inf_{u_0} [c(x_0, u_0) + \beta V(f(x_0, u_0))]. \end{aligned}$$

From the right-hand side of the Bellman equation, one defines the *Bellman operator*

$$\Gamma(V)(x) := \inf_{u \in U} [c(x, u) + \beta V(f(x, u))]. \quad (1.5)$$

By means of the optimal value function  $V$ , we define a feedback by

$$\mathbf{u}(x) = \operatorname{argmin}_{u \in U} [c(x, u) + \beta V(f(x, u))],$$

whenever the minimum exists, e.g. if  $V$  is continuous, as in this case the right-hand side is continuous in  $u \in U$  and  $U$  is a compact set.

If  $\beta < 1$ , this is a  $\beta$ -contraction. Consequently, by Banach's fixed point theorem, there is a unique fixed point.

**Lemma 1.1.** *If  $\beta < 1$ , the Bellman operator  $\Gamma: L^\infty(\mathbb{R}^s, \mathbb{R}) \rightarrow L^\infty(\mathbb{R}^s, \mathbb{R})$  is a contraction and thus possesses a unique fixed point.*

If  $\beta = 1$ ,  $\Gamma$  is just non-expanding. In this case, the existence of a fixed point of  $\Gamma$  is not clear. On the other hand, if there is a fixed point  $V$ , it is not unique, because then  $V + C$ , for any constant  $C \in \mathbb{R}$ , is a fixed point as well. For the undiscounted case, however, one needs a point of equilibrium or even a target set anyway, as explained in the last section. This yields the boundary conditions  $V|_T = 0$  and  $V|_{\mathbb{R}^s \setminus \Omega} = \infty$ . If no target set is specified, the boundary condition reads as  $V(0) = 0$ , with the additional requirement that  $V$  be continuous in 0, because in this case trajectories might never reach 0 but only converge to 0.

In the remainder of this chapter, we consider only undiscounted control systems with target sets which contain zero as interior point.

### The Kruřkov transform.

In general, some part of the state space  $\Omega$  will not be controllable to the target set  $T$ . By definition, if  $\beta = 1$ ,  $V(x) = \infty$  for points  $x$  in this part of  $\Omega$ . From a theoretical viewpoint, this does not pose any problem: On the contrary, the optimal value function allows thus to characterize the controllable subset of  $\Omega$ .

However, the case *beta* = 1 might not only have the problem that the Bellman operator might no longer be a contraction. Also for the numerical implementation parts of  $\Omega$  with  $V = \infty$  could lead to the problem that infinite values at certain points might propagate through the approximation procedure to other points. This could lead to infinite approximate values also for points which have actually finite values.

An elegant way to deal with the fact that  $V$  might attain the value  $\infty$  (and also with the missing contraction, as we will see), is by the *Kruřkov transform* (cf. [Kru75]) on  $V$  to get the function

$$v(x) := \exp(-V(x)), \tag{1.6}$$

where we use the continuous extension  $\exp(-\infty) := 0$ . Under this transformation, the optimality principle (1.4), after applying  $\exp(-\cdot)$  on both sides, takes the form

$$v(x) = \sup_{u \in U} \left[ e^{-c(x,u)} v(f(x,u)) \right], \quad x \in \Omega \setminus T, \tag{1.7}$$

the boundary conditions transform to  $v|_T = 1$  and  $v|_{\mathbb{R}^s \setminus \Omega} = 0$ . The right-hand side of this fixed point equation yields the transformed Bellman operator

$$\Gamma(v)(x) := \begin{cases} \sup_{u \in U} \left[ e^{-c(x,u)} \bar{v}(f(x,u)) \right] & x \in \Omega \setminus T, \\ 1 & x \in T, \\ 0 & x \in \mathbb{R}^s \setminus \Omega \end{cases} \tag{1.8}$$

## 1. Preliminaries

on the Banach space  $L^\infty(\mathbb{R}^s, \mathbb{R})$ , where

$$\bar{v}(x) := \begin{cases} v(x) & x \in \mathbb{R}^s \setminus T, \\ 1 & x \in T. \end{cases} \quad (1.9)$$

Since we assumed  $c$  to be bounded from below by  $\delta > 0$  outside of the target set, we obtain  $\|\Gamma(v) - \Gamma(w)\|_\infty \leq \beta_K \|v - w\|_\infty$  with

$$\beta_K := e^{-\delta} = \sup_{x \in \Omega \setminus T} \sup_{u \in U} e^{-c(x,u)} < 1. \quad (1.10)$$

The contraction rate  $\beta_K$  is specific to the Kruřkov transform, like the contraction rate  $\beta_D$  for discounted system.

Now for each  $x \in \Omega$ ,  $u \in U$  and  $v_1, v_2 \in L^\infty(\mathbb{R}^s, \mathbb{R})$  we have

$$\begin{aligned} |e^{-c(x,u)} \bar{v}_1(f(x,u)) - e^{-c(x,u)} \bar{v}_2(f(x,u))| &= e^{-c(x,u)} |\bar{v}_1(f(x,u)) - \bar{v}_2(f(x,u))| \\ &\leq e^{-c(x,u)} |v_1(f(x,u)) - v_2(f(x,u))| \\ &\leq \beta_K |v_1(f(x,u)) - v_2(f(x,u))|, \end{aligned}$$

and hence

$$\begin{aligned} |\Gamma(v_1)(x) - \Gamma(v_2)(x)| &= \left| \sup_{u \in U} \left[ e^{-c(x,u)} \bar{v}_1(f(x,u)) \right] - \sup_{u \in U} \left[ e^{-c(x,u)} \bar{v}_2(f(x,u)) \right] \right| \\ &\leq \sup_{u \in U} \left| e^{-c(x,u)} \bar{v}_1(f(x,u)) - e^{-c(x,u)} \bar{v}_2(f(x,u)) \right| \\ &\leq \beta_K |v_1(f(x,u)) - v_2(f(x,u))|. \end{aligned}$$

We then have by the Banach fixed point theorem

**Lemma 1.2.** *If  $\beta = 1$ , the Kruřkov transformed Bellman operator  $\Gamma: L^\infty(\mathbb{R}^s, \mathbb{R}) \rightarrow L^\infty(\mathbb{R}^s, \mathbb{R})$  is a contraction and thus possesses a unique fixed point.*

It might seem redundant to define  $\bar{v}$  and plugging it into (1.8) instead of plugging in  $v$  instead. This is indeed the case if one only considers iterations with  $\Gamma$  because  $\overline{\Gamma(v)} = \Gamma(v)$  with the operator from (1.9).

However, in (2.2) we will consider iterations of a composed operator  $S \circ \Gamma$  for which it is important to enforce the condition  $v_T \equiv 1$  on the function resulting from this operator.

Note that for discounted systems, the transformed Bellman equation would read

$$v(x) = \sup_{u \in U} \left[ e^{-c(x,u)} v(f(x,u))^\beta \right], \quad x \in \Omega \setminus T, \quad (1.11)$$

and the Bellman operator

$$\Gamma(v)(x) := \begin{cases} \sup_{u \in U} \left[ e^{-c(x,u)} v(f(x,u))^\beta \right] & x \in \Omega \setminus T, \\ 1 & x \in T, \\ 0 & x \in \mathbb{R}^s \setminus \Omega. \end{cases} \quad (1.12)$$

The latter is no contraction, because  $v \mapsto v^\beta$  is expanding close to  $v = 0$ .

So one can consider the Kruřkov transform as a substitute for the discounting in the sense that both lead to Bellman operators which are contractions, but the combination of discounting and the Kruřkov transform does not work in general. In the whole thesis we will implicitly assume the Kruřkov transform for Shepard discretized undiscounted control systems.

## 1.3. Continuous-time control systems

The methods which we develop in this thesis apply to discrete-time control systems. However, in many cases one considers discrete-time control systems which are time-discretizations of continuous-time control systems. For that reason we collect some facts about continuous-time control systems in this section. We refer to [BCD97] for a standard work on this topic.

Assume a continuous map  $f: \Omega \times U \rightarrow \mathbb{R}^s$  and a continuous cost function  $c: \Omega \times U \rightarrow [0, \infty)$  on compact sets  $\Omega \subset \mathbb{R}^s$ ,  $U \subset \mathbb{R}^d$ ,  $0 \in U$ , as state and control space, respectively. For an initial point  $x_0$  and the choice of a measurable *control function*  $u: [0, \infty) \rightarrow U$  the trajectory is given by the solution of the initial value problem

$$x(0) = x_0, \quad \dot{x}(t) = f(x(t), u(t)).$$

The cost functional is

$$J(x_0, u) := \int_0^\infty c(x(t), u(t)) e^{-\lambda t} dt$$

with discount rate  $\lambda \geq 0$ , and the optimal value function is

$$V(x_0) := \inf_{u \in \mathbf{U}} J(x_0, u), \tag{1.13}$$

where  $\mathbf{U} = \{u: [0, \infty) \rightarrow U \text{ measurable}\}$ .

For time- $\tau$ -sampling the correspondence between the discrete and continuous discount rates  $\beta$  and  $\lambda$  is given by  $\beta = e^{-\lambda\tau}$ . The continuous-time analogue of the Bellman equation is the *Hamilton-Jacobi-Bellman* equation ([Bel54, Bel57, CD83]). It reads

$$\lambda V(x) = \sup_{u \in U} [-f(x, u) \cdot \nabla V(x) - c(x, u)],$$

where  $\nabla V$  is the gradient of  $V$ , hence  $f(x, u) \cdot \nabla V(x)$  is the directional derivative of  $V$  in the direction of  $f(x, u)$ . If there is no discounting,  $\lambda = 0$ , a possible interpretation for the equation is that  $-c(x, u)$  describes the decrease of  $V$  along the trajectory in the direction  $f(x, u)$ .

Clearly, a discrete-time dynamical system can be obtained from a continuous-time one via the time- $\tau$  map,  $\tau > 0$ ,

## 1. Preliminaries

$$f^\tau(x_0) = x(\tau)$$

if the flow of the vector field is used.

For control systems this is possible as well, but a difference is that for a complete equivalence it is necessary to use the infinite-dimensional control space  $\mathbf{U}^\tau := \{u: [0, \tau) \rightarrow U \text{ measurable}\}$ . For  $x_0 \in \Omega$  and  $u \in \mathbf{U}^\tau$  the discrete-time system map is given by

$$f^\tau(x_0, u) = x(\tau),$$

where  $x(\cdot)$  is the solution of the ODE

$$x(0) = x_0, \quad x'(t) = f(x(t), u(t)),$$

and the discrete-time cost function is

$$c^\tau(x_0, u) = \int_0^\tau c(x(t), u(t)) dt.$$

## 1.4. Radial basis functions

In this section we recall some facts about radial basis functions and Shepard's method. We refer to [Fas07] for a more detailed discussion.

### Approximation with radial basis functions

The value function  $V$  from (1.2) or its Kruřkov transform  $v$  from (1.6) can be expressed in closed form only for simple optimal control problems, such as linear-quadratic ones. In general, we need to approximate it numerically within a finite-dimensional *approximation space*  $\mathcal{W}$ .

In this thesis we are going to use *radial basis functions* for this purpose, expressly radially symmetric functions  $\varphi_i: \mathbb{R}^s \rightarrow \mathbb{R}$  of the form  $\varphi_i(x) = \varphi^\sigma(\|x - x_i\|_2) = \varphi(\sigma\|x - x_i\|_2)$  on some set  $X = \{x_1, \dots, x_n\} \subset \mathbb{R}^s$  of *centres*. The *shape parameter*  $\sigma$  controls the “width” of the radial basis functions (see Figure 1.1, left) and has a significant impact on the numerical results.

We assume the *shape function*  $\varphi: \mathbb{R} \rightarrow [0, \infty)$  to be nonnegative, typical examples include the Gaussian  $\varphi(r) = \exp(-r^2)$  and the *Wendland functions*  $\varphi(r) = \max\{0, P(r)\}$ , normalized in the sense that  $\varphi(0) = 1$  and at least once continuously differentiable. Here,  $P$  is an appropriate polynomial which leads to  $\varphi$  being strictly positive definite (see next subsection), having a certain smoothness and the minimal possible degree. Moreover they have the structure  $P(r) = (1-r)^l Q(r)$  where  $Q$  is a polynomial with positive coefficients; as a consequence  $P(r) > 0$  for  $0 < r < 1$  and  $P(r) < 0$  for  $r > 1$  and so

$$\varphi^\sigma(r) = \begin{cases} P(\sigma r) & \text{for } 0 < r < 1/\sigma, \\ 0 & \text{for } r > 1/\sigma. \end{cases}$$

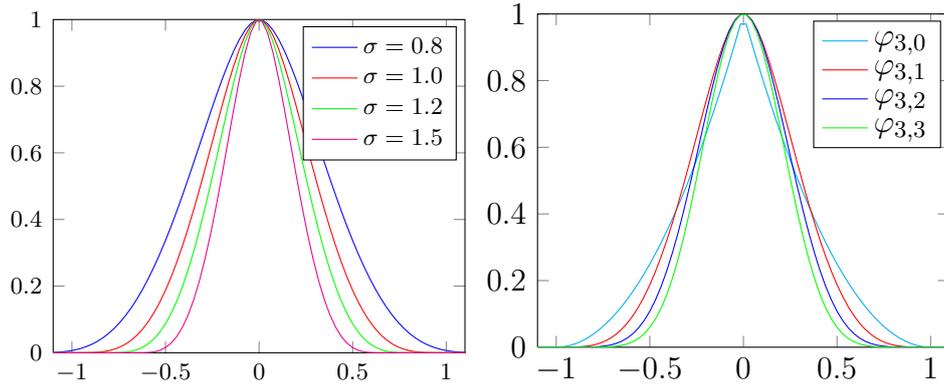


Figure 1.1.: Left: Wendland's function  $\varphi_{3,1}$  for some shape parameters. Right: The shape of some Wendland functions.

See Figure 1.1, right, for some examples for Wendland function.

In case that the shape function  $\varphi$  has compact support, we will need to require that the supports of the  $\varphi_i$ 's cover  $\Omega$ .

We only use Wendland functions in this thesis.

## Interpolation with radial basis functions

One way to use radial basis functions for approximating some function  $g: \Omega \rightarrow \mathbb{R}$  is by (scattered data) interpolation: One uses the ansatz

$$\tilde{g}(x) = \sum_{i=1}^n \alpha_i \varphi_i(x), \quad \alpha_i \in \mathbb{R},$$

and requires  $\tilde{g}$  to fulfil the interpolation conditions  $\tilde{g}(\xi_i) = g(\xi_i)$  at some points  $\xi_i \in \Omega, i = 1, \dots, n$ . The coefficient vector  $\alpha = (\alpha_1, \dots, \alpha_n)$  is then given by the solution of the linear system  $A\alpha = g(\xi)$  with  $A = (\varphi_j(\xi_i))_{ij}$  and  $g(\xi) = (g(\xi_1), \dots, g(\xi_n))$ . In order to guarantee the unique solvability of this system, one usually aims at making  $A$  positive definite by choosing suitable shape functions (cf. [Fas07], Chapter 3).

Formally, for some function  $g: \Omega \rightarrow \mathbb{R}$ , we can define its *interpolation approximation*  $Ig: \Omega \rightarrow \mathbb{R}$  by

$$Ig = \sum_{i=1}^n g(x_i) u_i^*,$$

where the  $u_i^*: \Omega \rightarrow \mathbb{R}$  are *cardinal basis functions* associated with the centres  $X$ , i.e. a Kronecker basis of the approximation space  $\text{span}(\phi_1, \dots, \phi_n)$  with  $u_i^*(x_i) = 1$  and  $u_i^*(x_j) = 0$  for  $i \neq j$ . Note that  $Iv$  depends linearly on  $v$ .

However, using interpolation for approximation has some shortcomings in our context. As an improvement, we use a least-squares type approach for function

## 1. Preliminaries

approximation known as *Shepard's method*, cf. [She68], which we will develop in the next sections. Shepard's method has several advantages over interpolation:

- (a) Computing the approximation on a finite set of points only requires a matrix-vector product (in contrast to solving a system of linear equations for interpolation),
- (b) the discretized Bellman operator remains a contraction, since the operator associated to the Shepard approximation is non-expanding (cf. Lemma 1.3), and
- (c) the approximation behaviour for an increasing number of centres is more favourable. For Shepard's method, in contrast to interpolation, one gets convergence if one considers shape functions with compact support and the supports of the basis functions are scaled proportionally to the fill distance (to be defined in (2.8)) of the centres, giving rise to sparse matrices in the implementation.

### Weighted least squares

Given some approximation space  $\mathcal{A} = \text{span}(a_1, \dots, a_m)$ ,  $a_i: \Omega \rightarrow \mathbb{R}$ , the set of centres  $X = \{x_1, \dots, x_n\}$ ,  $m \leq n$ , and a weight function  $w: \Omega \rightarrow (0, \infty)$ , we define the discrete inner product

$$\langle g_1, g_2 \rangle_w := \sum_{i=1}^n g_1(x_i) g_2(x_i) w(x_i),$$

for  $g_1, g_2: \Omega \rightarrow \mathbb{R}$  and the induced norm  $\|\cdot\|_w$ . The *weighted least squares approximant*  $\tilde{g} \in \mathcal{A}$  of some function  $g: \Omega \rightarrow \mathbb{R}$  is then defined by minimizing  $\|g - \tilde{g}\|_w$ . The solution is given by  $\tilde{g} = \sum_{i=1}^m \alpha_i a_i$ , where the optimal coefficient vector  $\alpha = (\alpha_1, \dots, \alpha_m)$  solves the linear system  $G\alpha = g_{\mathcal{A}}$  with Gram matrix  $G = (\langle a_i, a_j \rangle_w)_{ij}$  and  $g_{\mathcal{A}} = (\langle g, a_j \rangle_w)_j$ .

### Moving least squares

When constructing a least squares approximation to some function  $g: \Omega \rightarrow \mathbb{R}$  at  $x \in \Omega$ , it is often natural to require that only the values  $g(x_j)$  at some centres  $x_j$  close to  $x$  should play a significant role. This can be modelled by introducing a *moving* weight function  $w: \Omega \times \Omega \rightarrow \mathbb{R}_0^+$ , where  $w(\xi, x)$  is small (or even zero) if  $\|\xi - x\|_2$  is large. In the following, we will use a radial weight

$$w(\xi, x) = \varphi^\sigma(\|\xi - x\|_2),$$

where  $\varphi^\sigma = \varphi(\sigma \cdot)$  is the shape function introduced before. The corresponding discrete inner product is

$$\langle g_1, g_2 \rangle_{w(\cdot, x)} := \sum_{i=1}^n g_1(x_i) g_2(x_i) w(x_i, x).$$

The *moving least squares approximation*  $\tilde{g}$  of some function  $g: \Omega \rightarrow \mathbb{R}$  is then given by  $\tilde{g}(x) = \tilde{g}^x(x)$ , where  $\tilde{g}^x \in \mathcal{A}$  is minimizing  $\|g - \tilde{g}^x\|_{w(\cdot, x)}$ . The optimal coefficient vector  $\alpha^x$  is given by the solution of the Gram system  $G^x \alpha^x = g_{\mathcal{A}}^x$  with  $G^x = (\langle a_i, a_j \rangle_{w(\cdot, x)})_{ij}$  and  $g_{\mathcal{A}}^x = (\langle g, a_j \rangle_{w(\cdot, x)})_j$ .

## Shepard's method

We now simply choose  $\mathcal{A} = \text{span}(1)$  as approximation space. Then both the Gram matrix  $G^x = \langle 1, 1 \rangle_{w(\cdot, x)} = \sum_{j=1}^n w(x_j, x)$  and the right-hand side  $g_{\mathcal{A}}^x = \langle g, 1 \rangle_{w(\cdot, x)} = \sum_{i=1}^n g(x_i) w(x_i, x)$  are scalar. Thus we get  $\alpha^x = g_{\mathcal{A}}^x / G^x = \sum_{i=1}^n g(x_i) \psi_i(x)$ , where

$$\psi_i(x) := \frac{w(x_i, x)}{\sum_{j=1}^n w(x_j, x)}. \quad (1.14)$$

We define the *Shepard approximation*  $Sg: \Omega \rightarrow \mathbb{R}$  of  $g: \Omega \rightarrow \mathbb{R}$  as

$$Sg(x) = \alpha^x \cdot 1 = \sum_{i=1}^n g(x_i) \psi_i(x), \quad x \in \Omega. \quad (1.15)$$

Note again that  $Sg$  depends linearly on  $g$ . What is more, for each  $x$ ,  $Sg(x)$  is a convex combination of the values  $g(x_1), \dots, g(x_n)$ , since the  $\psi_i$  form a partition of unity,  $\sum_{i=1}^n \psi_i(x) = 1$  for all  $x \in \Omega$ .

Another important property of the Shepard operator is that it is non-expansive. This yields a contractive Bellman-Shepard operator, to be defined in (2.1).

For the next lemma, let  $\mathcal{W} := \text{span}(\psi_1, \dots, \psi_n)$ .

**Lemma 1.3.** *The Shepard operator  $S: C^0(\mathbb{R}^s, \mathbb{R}) \rightarrow \mathcal{W} \subset C^0(\mathbb{R}^s, \mathbb{R})$  has norm 1.*

*Proof.* Since, by assumption, the  $\psi_i$  are nonnegative and, as mentioned above, for each  $x \in \Omega$ ,  $Sg(x)$  is a convex combination of the values  $g(x_1), \dots, g(x_n)$ , we have for each  $x \in \Omega$

$$|Sg(x)| \leq \sum_{i=1}^n |g(x_i) \psi_i(x)| \leq \max_{i=1, \dots, n} |g(x_i)| \sum_{i=1}^n \psi_i(x) = \max_{i=1, \dots, n} |g(x_i)| \leq \|g\|_\infty,$$

so that  $\|Sg\|_\infty \leq \|g\|_\infty$ . Moreover, for constant  $g$  one has  $Sg = g$  and consequently  $\|Sg\|_\infty = \|g\|_\infty$ .  $\square$



## 2. The Shepard discretization of the optimality principle

In this chapter we define the Bellman-Shepard operator, which gives an iterative method to approximate the value function of a control problem. This operator makes use of the Shepard approximation, which depends on a set of centres. We prove that if the *fill distance* of a sequence of sets of centres shrinks to zero, the approximate value function converges to the exact value function. The distinction of control problems on whether they are discounted or not, leads to the split of the convergence proof for decreasing fill distance into Sections 2.3 and 2.5.

For undiscounted problems, we also show in Section 2.6 results about the quality of the approximate feedback (which one gets from the approximate value function) with regard to its ability to find trajectories along which the value function decreases.

The content of this chapter has mainly been published in [JS15].

### 2.1. Discretization of the optimality principle

We want to compute an approximation to the fixed point  $V$  resp.  $v$  of the Bellman operator (1.5) resp. (1.8) by *value iteration*, which means by iterating  $\Gamma$  on some initial function  $\tilde{V}^0$  resp.  $\tilde{v}^0$ .<sup>1</sup> We would like to perform this iteration inside the finite-dimensional approximation space  $\mathcal{W} \subset L^\infty(\mathbb{R}^s, \mathbb{R})$ , i.e. after each application of  $\Gamma$ , we need to map back into  $\mathcal{W}$ .

**Discretization using radial basis functions** Abstractly, choosing some (finite dimensional) *approximation space*  $\mathcal{W} \subset L^\infty(\mathbb{R}^s, \mathbb{R})$  as well as some projection operator  $\Pi : C^0(\mathbb{R}^s, \mathbb{R}) \rightarrow \mathcal{W}$ , one defines the *discrete Bellman operator*

$$\Pi \circ \Gamma : \mathcal{W} \rightarrow \mathcal{W}.$$

If the operator  $\Pi$  is constructed such that  $\|\Pi\| \leq L$  and  $\Gamma$  such that  $\|\Gamma\| \leq \beta$ , then  $\Pi \circ \Gamma$  has the norm  $\|\Pi \circ \Gamma\| \leq L\beta$ . If these maps can be arranged such that  $L\beta < 1$ ,  $\Pi \circ \Gamma$  is a contraction, possessing a unique fixed point  $\tilde{V} = (\Pi \circ \Gamma)[\tilde{V}]$  by Banach's fixed point theorem.

---

<sup>1</sup>In this section we always write  $V, V^0$ , etc. rather than  $v, v^0$ , etc., but the considerations concern both the discounted and the Kruřkov case.

## 2. The Shepard discretization of the optimality principle

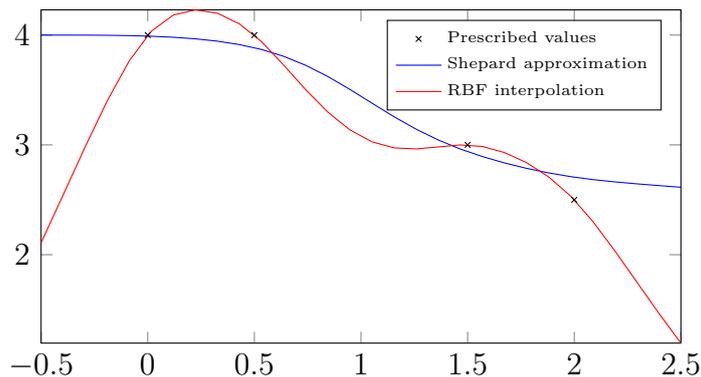


Figure 2.1.: Example of a Shepard approximation and an interpolation with radial basis functions.

In the following we will consider RBF interpolation and Shepard's method for radial basis functions as choices for the approximation operator  $\Pi$ .

As a simple motivation, we plot in Figure 2.1 an example of both approximation methods in a one-dimensional case with four prescribed values (black x's in the figure) and the two approximating functions. One sees that the Shepard approximation does not interpolate the given values exactly. However, it looks smoother than the RBF interpolation and fulfils monotonicity: Its maximum and minimum are bounded by the maximum and minimum of the prescribed values. This is a consequence of Lemma 1.3.

Next, we consider RBF interpolation and Shepard's method for the discretization of the Bellman operator.

### Interpolation

Choosing  $\mathcal{A} = \text{span}(\varphi_1, \dots, \varphi_n)$ , we define the *Bellman interpolation operator* to be

$$I \circ \Gamma : \mathcal{A} \rightarrow \mathcal{A}.$$

In general, the operator  $I$  is expansive and not necessarily monotone. If the maximal expansion rate is denoted by  $L$ , in the worst case the contraction rate  $\beta$  of  $\Gamma$  has to compensate for  $L$ , specifically  $\beta L < 1$ . However,  $L$  is not known, in general. For that reason, one cannot rely on the iteration with  $I \circ \Gamma$  to be convergent (although in most of our numerical experiments it turned out to converge) and move our focus in this thesis towards the value iteration with Shepard's method.

### Shepard's method

Correspondingly, we define the *Bellman-Shepard operator* as

$$\tilde{\Gamma} := S \circ \Gamma : \mathcal{W} \rightarrow \mathcal{W} \tag{2.1}$$

for the Shepard space  $\mathcal{W} = \text{span}(\psi_1, \dots, \psi_n)$ .<sup>2</sup> Explicitly, the value iteration then reads

$$\tilde{V}^{k+1} := S(\Gamma[\tilde{V}^k]), \quad k = 0, 1, 2, \dots, \quad (2.2)$$

where, as mentioned, some initial function  $\tilde{V}^0 \in \mathcal{W}$  has to be provided.

As a composition of the contraction  $\Gamma$  and the non-expansive Shepard operator  $S$ , cf. Lemmas 1.1 resp. 1.2, and 1.3, we get the following theorem about the Shepard value iteration.

**Theorem 2.1.** *The Bellman-Shepard operator  $\tilde{\Gamma} : (\mathcal{W}, \|\cdot\|_\infty) \rightarrow (\mathcal{W}, \|\cdot\|_\infty)$  is a contraction,*

$$\|\tilde{\Gamma}(V_1) - \tilde{\Gamma}(V_2)\| \leq \beta \|V_1 - V_2\|,$$

thus the iteration (2.2) converges to the unique fixed point  $\tilde{V} \in \mathcal{W}$  of  $\tilde{\Gamma}$ .

## Explicit formulas for the Bellman-Shepard discretization

In the case of a Kruřkov transformed value iteration, (1.8) yields the iteration

$$v^{k+1}(x) = \Gamma[v^k](x) = \begin{cases} \sup_{u \in U} \{e^{-c(x,u)} v^k(f(x,u))\} & x \in \Omega \setminus T, \\ 1 & x \in T, \\ 0 & x \in \mathbb{R}^s \setminus \Omega, \end{cases} \quad (2.3)$$

and if we expand this in a Shepard basis as in (1.15), we get

$$S \circ \Gamma[v^k](x) = \sum_{i=1}^n \psi_i(x) \begin{cases} \sup_{u \in U} \{e^{-c(x_i,u)} v^k(f(x_i,u))\} & x_i \in \Omega \setminus T, \\ 1 & x_i \in T, \\ 0 & x_i \in \mathbb{R}^s \setminus \Omega. \end{cases} \quad (2.4)$$

For later use in Section 5.2, we will now consider the related iteration with  $\Gamma \circ S$ , instead of with  $S \circ \Gamma$ , which converges to the fixed point  $\Gamma(v)$  instead of  $v$ .

If we expand  $v^k$  in (2.3) a Shepard basis as in (1.15), we get

$$\Gamma \circ S[v^k](x) = \begin{cases} \sup_{u \in U} \{e^{-c(x,u)} \sum_{i=1}^n v^k(x_i) \psi_i(f(x,u))\} & x \in \Omega \setminus T, \\ 1 & x \in T, \\ 0 & x \in \mathbb{R}^s \setminus \Omega. \end{cases} \quad (2.5)$$

As in Chapter 1 was specified,  $v, v^k$ , etc. refer to Kruřkov transformed value functions, while  $V, V^k$ , etc. refer to discounted systems. For discounted control systems instead of Kruřkov transformed ones, the previous equation becomes

$$\Gamma \circ S[V^k](x) = \inf_{u \in U} \left\{ c(x,u) + \beta \sum_{i=1}^n V^k(x_i) \psi_i(f(x,u)) \right\}, \quad (2.6)$$

and so the fixed point equation has the form

$$V(x) = \inf_{u \in U} \left\{ c(x,u) + \beta \sum_{i=1}^n V(x_i) \psi_i(f(x,u)) \right\}. \quad (2.7)$$

---

<sup>2</sup>To simplify the notation, we use the letter  $\mathcal{W}$  to denote both an abstract approximation space and the Shepard space.

## 2. The Shepard discretization of the optimality principle

### 2.2. Stationary vs. non-stationary approximation

Our aim is to prove that for decreasing fill distance, the approximate value function converges to the exact one. The approximate value function is the fixed point of the Bellman-Shepard operator. As an auxiliary result we show in this section that the sequence of Bellman-Shepard operators corresponding to an increasingly dense set of centres converges to the identity.

For the set of centres  $X$  we define the number

$$h := h_{X,\Omega} := \max_{x \in \Omega} \min_{\xi \in X} \|x - \xi\|, \quad (2.8)$$

which is called the *fill distance* of  $X$  in  $\Omega$ . This is the radius of the largest open ball inside  $\Omega$  that is disjoint from  $X$ .

In *non-stationary* approximation with radial basis functions, the shape parameter  $\sigma$  from Section 1.4 is kept constant while the fill distance  $h$  goes to 0. Non-stationary interpolation is convergent (cf. [Fas07, Theorem 15.3]). On the other hand, *stationary* interpolation, i.e. letting  $1/\sigma$  shrink to 0 proportionally with  $h$  ( $\sigma = C_1/h$  for a constant  $C_1 > 0$ ), does not converge (for a counter example see [Fas07, Example 15.10]). For Shepard's method instead of interpolation, though, the exact opposite holds: Non-stationary approximation does not converge (cf. [Fas07, Chapter 24]), while stationary approximation does, as we will show below. In practice, this is an advantage, since stationary approximation allows us to keep the associated matrices sparse.

Assume we are given a Lipschitz continuous function  $g : \Omega \rightarrow \mathbb{R}$ , and a sequence of sets of centres  $X^{(j)} \subset \Omega$  with fill distances  $h^{(j)}$ . We consider the Shepard operators  $S^{(j)}$  associated with the sets  $X^{(j)}$  and shape parameters

$$\sigma^{(j)} := C_1/h^{(j)} \quad (2.9)$$

for some constant  $C_1 > 0$ . Under the assumption that  $h^{(j)} \rightarrow 0$  we get convergence of the respective Shepard approximants  $S^{(j)}g : \Omega \rightarrow \mathbb{R}$  to  $g$  as shown in the following result, which is a minor generalization of the statement preceding Theorem 25.1 in [Fas07] from  $C^1$  functions to Lipschitz functions.

From now on, we consider only shape functions with compact support. Otherwise, the following lemma would require additional assumptions on the sets of centres  $X^{(j)}$ .

**Lemma 2.2.** *Let  $g : \Omega \rightarrow \mathbb{R}$ , be some Lipschitz continuous function with Lipschitz constant  $L_g$ . Then*

$$\|g - S^{(j)}g\|_\infty \leq L_g \frac{\rho}{C_1} h^{(j)}$$

where  $\rho$  is a number such that  $\text{supp}(\varphi) \subset B_\rho(0) = \{x \in \mathbb{R}^s \mid \|x\| \leq \rho\}$ .

*Proof.* By the scaling effect of  $\sigma^{(j)}$  we have

$$\text{supp}(\varphi^{\sigma^{(j)}}) \subset B_{\rho/\sigma^{(j)}}(0)$$

### 2.3. Convergence for decreasing fill distance with discounting

as well as

$$\sup_{x \in B_{\rho/\sigma^{(j)}}(x_0)} |g(x) - g(x_0)| \leq L_g \frac{\rho}{\sigma^{(j)}}, \quad x_0 \in \Omega,$$

and, as a consequence,

$$|g(x_0) - S^{(j)}g(x_0)| \leq L_g \frac{\rho}{\sigma^{(j)}} = L_g \frac{\rho}{C_1} h^{(j)}, \quad x_0 \in \Omega,$$

because the Shepard approximation is given by a convex combination of values of  $g$  inside a  $\rho/\sigma^{(j)}$ -neighbourhood.  $\square$

The analogue result holds true for Lipschitz continuity up to perturbation, as we will show in the next lemma.

A function is called Lipschitz continuous with Lipschitz constant  $L > 0$  up to a constant perturbation  $\varepsilon > 0$ , if

$$|g(x) - g(y)| \leq L\|x - y\| + \varepsilon$$

for all  $x, y \in \Omega$ .

**Lemma 2.3.** *If in the last lemma,  $g$  is only Lipschitz continuous with Lipschitz constant  $L > 0$  up to a constant perturbation  $\varepsilon > 0$ , then*

$$\|g - S^{(j)}g\|_{\infty} \leq L_g \frac{\rho}{C_1} h^{(j)} + \varepsilon.$$

*Proof.* Here, the first inequality from the last proof becomes

$$\sup_{x \in B_{\rho/\sigma^{(j)}}(x_0)} |g(x) - g(x_0)| \leq L_g \frac{\rho}{\sigma^{(j)}} + \varepsilon, \quad x_0 \in \Omega,$$

leading to

$$|g(x_0) - S^{(j)}g(x_0)| \leq L_g \frac{\rho}{\sigma^{(j)}} + \varepsilon = L_g \frac{\rho}{C_1} h^{(j)} + \varepsilon, \quad x_0 \in \Omega.$$

$\square$

## 2.3. Convergence for decreasing fill distance with discounting

In this section we show that the approximate solution  $\tilde{V}$  from Theorem 2.1, which depends on the set of centres  $X$ , converges to the exact solution  $V$  if the fill distance  $h$  of  $X$  converges to zero.

As we show in Section A.1 in the Appendix,  $V$  is Lipschitz continuous for undiscounted control systems.

However, for discounted control systems, one only gets Lipschitz continuity up to an arbitrarily small perturbation:

## 2. The Shepard discretization of the optimality principle

**Theorem 2.4** (Lipschitz continuity of  $V$  up to perturbation). *Let  $V$  be the optimal value function of a discrete-time control system with discount rate  $0 < \beta < 1$ , and let the system dynamics  $f$  and the cost function  $c$  be Lipschitz continuous with Lipschitz constants  $L_f$  and  $L_c$ , respectively.*

*Then  $V$  is Lipschitz continuous with Lipschitz constant  $L = L(\varepsilon) > 0$  up to an arbitrary constant perturbation  $\varepsilon > 0$ . If the discount rate is such that  $\beta L_f < 1$ , then  $V$  is even Lipschitz continuous with Lipschitz constant  $L_c \frac{1}{1 - \beta L_f}$ .*

*Proof.* Let  $M_c := \max_{(x,u) \in \Omega \times U} |c(x,u)|$ . For any  $\varepsilon > 0$ , one can choose  $k \in \mathbb{N}$  such that

$$2((\beta L_f)^k + (\beta L_f)^{k+1} + \dots)M_c < \varepsilon.$$

Then

$$\begin{aligned} |J(x_0, (u_k)) - J(y_0, (u_k))| &= \left| \sum_{k=0}^{\infty} \beta^k c(x_k, u_k) - \sum_{k=0}^{\infty} \beta^k c(y_k, u_k) \right| \\ &\leq \sum_{k=0}^{\infty} |\beta^k c(x_k, u_k) - \beta^k c(y_k, u_k)| \\ &\leq \sum_{k=0}^{\infty} \beta^k L_c L_f^k \|x_0 - y_0\| \\ &\leq \|x_0 - y_0\| \underbrace{L_c(1 + \beta L_f + (\beta L_f)^2 + \dots + (\beta L_f)^{k-1})}_{=:L} + \varepsilon. \end{aligned}$$

Consequently,

$$\begin{aligned} |V(x_0) - V(y_0)| &= \left| \inf_{(u_k)} J(x_0, (u_k)) - \inf_{(u_k)} J(y_0, (u_k)) \right| \\ &\leq \inf_{(u_k)} |J(x_0, (u_k)) - J(y_0, (u_k))| \\ &\leq L \|x_0 - y_0\| + \varepsilon. \end{aligned}$$

If the discount rate is chosen such that  $\beta L_f < 1$ , then

$$\begin{aligned} |J(x_0, (u_k)) - J(y_0, (u_k))| &\leq \sum_{k=0}^{\infty} \beta^k L_c L_f^k \|x_0 - y_0\| \\ &\leq \|x_0 - y_0\| L_c \frac{1}{1 - \beta L_f}, \end{aligned}$$

and so

$$|V(x_0) - V(y_0)| \leq \|x_0 - y_0\| L_c \frac{1}{1 - \beta L_f}.$$

□

The following example shows that, in general,  $V$  is not Lipschitz continuous, so it is in fact necessary to allow for a perturbation.

### 2.3. Convergence for decreasing fill distance with discounting

**Example.** We set  $\Omega = [0, 1]$ ,  $U = \{0\}$ ,  $\beta = \frac{1}{2}$ ,

$$f(x) := f(x, 0) := \begin{cases} 2x - 1 & \text{for } x \geq \frac{2}{3}, \\ 0 & \text{for } x = 0, \\ \text{a smooth monotone interpolation} & \text{for } 0 < x < \frac{2}{3}. \end{cases}$$

and

$$c(x, 0) := x.$$

Then  $V$  is not Lipschitz continuous in a neighbourhood of 1.

*Proof.* The system is deterministic, because it has only one control and so an initial value already determines a trajectory. Define  $(x_k)$  by  $x_0 := 1$  and  $(y_k)$  by  $y_0 := (1 - \frac{1}{2^K})$  for some  $K \in \mathbb{N}$ . Then

$$V(1) = V(x_0) = \sum_{k=0}^{\infty} \beta^k \underbrace{x_k}_{=1} = 2 = 1 + \frac{1}{2} + \frac{1}{4} + \cdots + \frac{1}{2^{K-1}} + \frac{1}{2^K} V(1)$$

and

$$\begin{aligned} V\left(1 - \frac{1}{2^K}\right) &= V(y_0) = \sum_{k=0}^{\infty} \beta^k y_k \\ &= \left(1 - \frac{1}{2^K}\right) + \frac{1}{2} \left(1 - \frac{1}{2^{K-1}}\right) + \frac{1}{4} \left(1 - \frac{1}{2^{K-2}}\right) \\ &\quad + \cdots + \frac{1}{2^{K-1}} \left(1 - \frac{1}{2}\right) + \frac{1}{2^K} V\left(f\left(\frac{1}{2}\right)\right), \end{aligned}$$

where  $\frac{1}{2^K} V(1) > \frac{1}{2^K} V\left(f\left(\frac{1}{2}\right)\right)$  and thus

$$V(1) - V\left(1 - \frac{1}{2^K}\right) > \frac{1}{2^K} + \cdots + \frac{1}{2^K} = K \frac{1}{2^K}$$

and

$$\frac{V(1) - V\left(1 - \frac{1}{2^K}\right)}{1 - \left(1 - \frac{1}{2^K}\right)} = \frac{V(1) - V\left(1 - \frac{1}{2^K}\right)}{\frac{1}{2^K}} > K,$$

so differential quotients of  $V$  can be arbitrarily large in a neighbourhood of 1, and so  $V$  is not Lipschitz continuous.  $\square$

It might appear counterintuitive that for discounted systems one only gets Lipschitz continuity up to perturbation instead of Lipschitz continuity as for undiscounted systems. An explanation is that in the corresponding Theorem A.8, a necessary assumption is that the system is globally controllable or, equivalently, has a finite optimal value function, which is a rather strong condition in the undiscounted case. For discounted systems the finiteness of the optimal value function is already a consequence from the boundedness of the cost function.

## 2. The Shepard discretization of the optimality principle

However, Lipschitz continuity up to arbitrary perturbation suffices to prove the convergence of the approximate solution to the exact solution for the optimal value function for discounted control systems.

**Theorem 2.5.** *Assume a control system with the assumptions from Theorem 2.4 and let  $V$  be Lipschitz continuous with Lipschitz constant  $L_V(\varepsilon) > 0$  up to perturbation  $\varepsilon > 0$ .*

*Assume a sequence of sets of centres  $X^{(j)} \subset \Omega$  with fill distances  $h^{(j)} > 0$ ,  $\sigma^{(j)} = C_1/h^{(j)}$ ,  $C_1 > 0$ , the associated shape parameters,  $S^{(j)} : C^0(\mathbb{R}^s, \mathbb{R}) \rightarrow \mathcal{W}$  the associated Shepard operators and  $\tilde{V}^{(j)} \in \mathcal{W}$  the unique fixed points of  $\tilde{\Gamma}^{(j)} = S^{(j)} \circ \Gamma$ . Moreover, assume  $h^{(j)} \rightarrow 0$  for  $j \rightarrow \infty$ . Then*

$$\|V - \tilde{V}^{(j)}\|_\infty \rightarrow 0.$$

*Proof.* First, we show that

$$\|V - \tilde{V}^{(j)}\|_\infty \leq \frac{1}{1-\beta} \left( \frac{\rho}{C_1} L_V(\varepsilon) h^{(j)} + \varepsilon \right).$$

Let  $e^{(j)}$  be the norm of the residual of  $V$  in the Bellman-Shepard equation, expressly

$$e^{(j)} := \|V - \tilde{\Gamma}^{(j)}(V)\|_\infty = \|V - S^{(j)}V\|_\infty.$$

Then

$$\begin{aligned} \|V - \tilde{V}^{(j)}\|_\infty &\leq \|V - \tilde{\Gamma}^{(j)}(V)\|_\infty + \|\tilde{V}^{(j)} - \tilde{\Gamma}^{(j)}(V)\|_\infty \\ &= e^{(j)} + \|\tilde{\Gamma}^{(j)}(\tilde{V}^{(j)}) - \tilde{\Gamma}^{(j)}(V)\|_\infty \\ &\leq e^{(j)} + \beta \|\tilde{V}^{(j)} - V\|_\infty, \end{aligned}$$

where the last inequality is by Theorem 2.1. Consequently,

$$\|V - \tilde{V}^{(j)}\|_\infty \leq \frac{e^{(j)}}{1-\beta} \leq \frac{1}{1-\beta} \left( \frac{\rho}{C_1} L_V(\varepsilon) h^{(j)} + \varepsilon \right) \rightarrow \frac{\varepsilon}{1-\beta}$$

for  $h^{(j)} \rightarrow 0$ , where the last inequality is by Lemma 2.3. The expression on the right-hand side can be made arbitrarily small, as  $\varepsilon$  can be chosen freely, and hence we have  $\|V - \tilde{V}^{(j)}\|_\infty \rightarrow 0$  for  $j \rightarrow \infty$ .  $\square$

## 2.4. Implications of (non-)discounting for dynamic programming

In Section 2.3 we considered discounted control systems, i.e. that  $0 < \beta < 1$ . While there might be some justification of discounting in applications (in the sense that those costs which have to be paid in the future are not as important as the

## 2.5. Convergence for decreasing fill distance with the Kruřkov transform

immediate costs), in most cases the discount rate is just an artificial modification of the original problem in order to simplify the theory, to force finiteness of the value function and sometimes also to allow for a finite horizon approximation.

Often one is interested in the case that there is no discounting, because the undiscounted value function might be the relevant quantity. In this case one gets a better numerical solution by approximating the undiscounted value function directly, instead of by first modifying the given problem with the introduction of a discount rate.

We recall two crucial differences of undiscounted control problems in comparison with discounted ones:

- The finiteness of the undiscounted value function in certain regions implies the controllability of these regions to the target in the sense of Section 1.1.
- As was explained in section 1.2, if there is no discounting, one needs the Kruřkov transform in order to get a contraction and convergence for the value iteration.

## 2.5. Convergence for decreasing fill distance with the Kruřkov transform

Theorem 2.5 showed that the approximate solution obtained for the optimal value function with the value iteration (2.2) converges to the exact optimal value function if a sequence of sets of centres is considered with fill distance converging to 0. There it was assumed that the discount rate is less than one.

In this section we prove that the corresponding result holds also true if no discount rate is present, but instead the Kruřkov transform is applied.

For the proof of the following theorem, we employ Lemma 2.2 in order to show convergence of the approximate value functions for decreasing fill distance. To this end, we need to assume that the value function  $V$ , and hence also its Kruřkov transform  $v$ , is Lipschitz continuous. This is true, according to Theorem A.8 in the Appendix, under the following assumption:

**Assumption 2.6.** We assume that a discrete-time control system is given that is stabilizable on all of  $\Omega$ , and that there is a feedback so that the closed-loop system has 0 as an asymptotically stable fixed point. Furthermore, we assume that  $f \in C^1(\Omega \times U, \Omega)$ ,  $c \in C^2(\Omega \times U, [0, \infty))$ .

From now on, we need a target set  $T$  as explained in Section 1.2 which yields a contraction rate  $\beta_K$  as in (1.10).

**Theorem 2.7.** Assume a control system with Assumption 2.6 and let  $L_v$  be a Lipschitz constant for  $v$ . Assume a sequence of sets of centres  $X^{(j)} \subset \Omega$  with fill distances  $h^{(j)} > 0$ , let  $\sigma^{(j)} = C_1/h^{(j)}$ ,  $C_1 > 0$ , be the associated shape parameters,

## 2. The Shepard discretization of the optimality principle

$S^{(j)} : C^0(\mathbb{R}^s, \mathbb{R}) \rightarrow \mathcal{W}$  the associated Shepard operators and  $\tilde{v}^{(j)} \in \mathcal{W}$  the unique fixed points of  $\tilde{\Gamma}^{(j)} = S^{(j)} \circ \Gamma$ . Then

$$\|v - \tilde{v}^{(j)}\|_\infty \leq \frac{L_v \rho}{C_1(1 - \beta_K)} h^{(j)}.$$

Moreover, if  $h^{(j)} \rightarrow 0$  for  $j \rightarrow \infty$ , then

$$\|v - \tilde{v}^{(j)}\|_\infty \rightarrow 0.$$

*Proof.* Let  $e^{(j)}$  be the norm of the residual of  $v$  in the Bellman-Shepard equation,

$$e^{(j)} = \|v - \tilde{\Gamma}^{(j)}(v)\|_\infty = \|v - S^{(j)}v\|_\infty.$$

Then

$$\begin{aligned} \|v - \tilde{v}^{(j)}\|_\infty &\leq \|v - \tilde{\Gamma}^{(j)}(v)\|_\infty + \|\tilde{v}^{(j)} - \tilde{\Gamma}^{(j)}(v)\|_\infty \\ &= e^{(j)} + \|\tilde{\Gamma}^{(j)}(\tilde{v}^{(j)}) - \tilde{\Gamma}^{(j)}(v)\|_\infty \\ &\leq e^{(j)} + \beta_K \|\tilde{v}^{(j)} - v\|_\infty. \end{aligned}$$

Consequently,

$$\|v - \tilde{v}^{(j)}\|_\infty \leq \frac{e^{(j)}}{1 - \beta_K} \leq \frac{L_v \rho}{C_1(1 - \beta_K)} h^{(j)},$$

where the last inequality is by Lemma 2.2.

It follows that if  $h^{(j)} \rightarrow 0$  for  $j \rightarrow \infty$ , then

$$\|v - \tilde{v}^{(j)}\|_\infty \rightarrow 0 \quad \text{for } j \rightarrow \infty.$$

□

## Treating the case without target

So far, we have assumed the boundary condition  $V|_T = 0$  for the target set  $T$ , which contains zero as an interior point. On the other hand, for the proof of Theorem A.8 that  $V$  is Lipschitz continuous, it is necessary that  $c(x, u) > 0$  for all  $x \in (\Omega \setminus \{0\}) \times U$ . These assumptions contradict each other.

The boundary condition  $V|_T = 0$  can be relaxed, though. For the existence of the contraction factor  $\beta_K$  in (1.10), it is sufficient that there is a boundary condition  $V|_T = V_T$  (and the function  $V_T : T \rightarrow \mathbb{R}_0^+$  need not be zero), in other words, it suffices that the exact optimal value function is known a priori on a neighbourhood  $T$  of the equilibrium point 0.

However, this assumption is no essential restriction on control problems, as we show now. To this end we prove that we can change a given control system in a target set  $T_2 = T$  such that the value function  $V$  for the new problem is known on  $T$  and deviates from the value function of the original system only

## 2.5. Convergence for decreasing fill distance with the Kruřkov transform

slightly: If a sequence of target sets  $(T^{(j)})_j$  with  $\text{diam}(T^{(j)}) \rightarrow 0$  is given, then  $\|V^{(j)} - V\|_\infty \rightarrow 0$  as we will show next. We extend the control space with a special control  $u^*$ , which allows to move directly to zero, but only at a large cost, increasing fast with distance from zero. Then  $u^*$  is an optimal control only for states  $x$  which are sufficiently close to zero.

Let  $L$  be a Lipschitz constant for  $V$  and  $0 \in T_0 \subset T_1 \subset T_2 = T \subset \Omega$ . Let  $U^* := U \cup \{u^*\}$ ,  $f(x, u^*) := 0$  and

$$c(x, u^*) := \begin{cases} C\|x\|^2 & \text{for } x \in T_1, \\ \text{a } C^2 \text{ interpolation} & \text{for } x \in T_2 \setminus T_1, \\ L\|x\| & \text{for } x \in \Omega \setminus T_2, \end{cases}$$

with a constant  $0 < C < 1$  such that  $c(x, u^*) \leq c(x, u)$  for all  $x \in T_1 \setminus T_0, u \in U$  (This is possible because  $\frac{c(x, u)}{\|x\|^2}$  is a  $C^0(T_1 \setminus T_0)$  function.).

We modify the one-step cost  $c(x, u)$  for  $x \in T_0, u \in U$  into  $\tilde{c}(x, u)$  such that  $c(x, u^*) \leq \tilde{c}(x, u)$  even for all  $x \in T_1, u \in U$ :

$$\tilde{c}(x, u) := \begin{cases} c(x, u) & \text{for } x \in X \setminus T_0, \\ \max\{c(x, u), c(x, u^*)\} & \text{for } x \in T_0. \end{cases}$$

**Proposition 2.8.** *The value function  $\tilde{V}$  of the new problem (with  $u^*$ , and with  $\tilde{c}$  instead of  $c$ ) fulfils*

$$\tilde{V}(x) = C\|x\|^2 \text{ for } x \in T_1.$$

*Proof.* The value function is bounded from below by the cost for the first time step:  $V(x) \geq \min_{u \in U^*} \tilde{c}(x, u) \geq c(x, u^*) = C\|x\|^2$ .

On the other hand, the choice of  $u_0 = u^*$  gives  $x_1 = 0$  and the total cost of  $C\|x\|^2$ . So, in fact,  $V(x) = C\|x\|^2$ .  $\square$

**Theorem 2.9.** *Consider sequences  $0 \in T_0^{(j)} \subset T_1^{(j)} \subset T_2^{(j)}$  with  $\text{diam}(T_2^{(j)}) \rightarrow 0$ . Let  $V^{(j)}$  be the value functions of the new problems (with  $u^*$  and  $\tilde{c}$ ) associated to  $T_i^{(j)}, i = 0, 1, 2$ . Then  $\|V^{(j)} - V\|_\infty \rightarrow 0$ .*

*Proof.* It holds that  $V^{(j)} \leq V + (\text{diam}(T_2^{(j)}))^2$ . To see this, let an optimal trajectory  $(x_n, u_n)$  of the original problem for the initial point  $x_0 \in \Omega$  be given. As  $x_k \rightarrow 0$  for  $k \rightarrow \infty$ , there is a first  $x_n$  along the trajectory with  $x_n \in T_1$ . We consider the control sequence  $(u_0, \dots, u_{n-1}, u^*)$  for the modified control problem with value function  $V^{(j)}$ . Then

$$\begin{aligned} V^{(j)}(x_0) &\leq \sum_{i=0}^{n-1} c(x_i, u_i) + \underbrace{C}_{<1} \|x_n\|^2 \\ &\leq \sum_{i=0}^{n-1} c(x_i, u_i) + (\text{diam}(T_2^{(j)}))^2 \\ &\leq V(x_0) + (\text{diam}(T_2^{(j)}))^2. \end{aligned}$$

## 2. The Shepard discretization of the optimality principle

On the other hand, one can also see that  $V^{(j)} \geq V - L \text{diam}(T_2^{(j)})$ . To this end, we consider an optimal trajectory  $(x_n, u_n)$  for the modified control problem with value function  $V^{(j)}$  for the initial point  $x_0 \in \Omega$ . Then there is a first  $x_n$  with  $x_n \in T_2^{(j)}$ . Now there are two cases:

If  $u_{n-1} \neq u^*$ ,  $V^{(j)}(x_0) \geq \sum_{i=0}^{n-1} c(x_i, u_i)$  and

$$\begin{aligned} V(x_0) &\leq \sum_{i=0}^{n-1} c(x_i, u_i) + V(x_n) \\ &\leq \sum_{i=0}^{n-1} c(x_i, u_i) + L \text{diam}(T_2^{(j)}). \end{aligned}$$

So  $V^{(j)}(x_0) \geq V(x_0) - L \text{diam}(T_2^{(j)})$ .

But if  $u_{n-1} = u^*$ , one gets  $V^{(j)}(x_0) \geq \sum_{i=0}^{n-2} c(x_i, u_i) + L\|x_{n-1}\|$  and  $V(x_0) \leq \sum_{i=0}^{n-2} c(x_i, u_i) + L\|x_{n-1}\|$ . This shows that  $V^{(j)}(x_0) \geq V(x_0)$ . □

## 2.6. Construction of a stabilizing feedback

So far, we were concerned with the approximation of the value function  $V$ , but did not investigate the feedback  $\mathbf{u}$ . The reason is that from a theoretical viewpoint, the construction of one of them yields the other one without much effort: From an optimal feedback  $\mathbf{u}$  one can directly calculate an optimal trajectory  $(x_k)$  by (1.1), and then calculate  $V(x_0)$  from (1.2), because the optimality of the feedback  $\mathbf{u}$ , and hence of the control sequence  $(u_k) = \mathbf{u}(x_k)$  for  $x_0$  has been assumed. The other way round, an optimal feedback can be defined from the optimal value function  $V$  by

$$\mathbf{u}(x) := \operatorname{argmin}_{u \in U} [c(x, u) + V(f(x, u))], \quad x \in \Omega.$$

However, for numerical reasons it is important to analyze the feedback in more detail because the correspondence between optimal value function and optimal feedback is no longer an exact correspondence if these maps are replaced by approximations.

In the preceding section, we analyzed the error that the approximate value function has in comparison with the exact optimal value function. Here, we will show how that implies the stability of the feedback derived from the right-hand side of the optimality principle (1.4).

In this section we consider undiscounted problems, i.e.  $\beta = 1$  because for discounted problems the stabilization is not necessary for the finiteness of the value function. As common in dynamic programming, we use the approximate value function  $\tilde{V}(x)$ ,  $x \in \mathcal{S} := \{x \in \Omega : \tilde{V}(x) < \infty\}$ , in order to construct a feedback which stabilizes the closed-loop system on a certain subset of  $\Omega$ .

## 2.6. Construction of a stabilizing feedback

This feedback is (following the exact case from above)

$$\tilde{\mathbf{u}}(x) := \operatorname{argmin}_{u \in U} [c(x, u) + \tilde{V}(f(x, u))], \quad x \in \mathcal{S}.$$

Note that the argmin exists, since  $U$  is compact and  $c, f$  and  $\tilde{V}$  are continuous.

We will give two related, yet independent theorems which state that the exact value function resp. the approximate value function descend along discrete trajectories defined by the feedback as long as the one-step cost is above a certain level. One cannot expect the (exact or approximate) value function to drop according to the one-step cost in each step. The deviation can be bounded, though. In our first theorem this deviation depends only on the approximation error of  $\tilde{V}$  in comparison to  $V$ . We will make frequent use of the minimal one-step cost  $c_0(x) := \min_{u \in U} c(x, u)$  at a point  $x$ .

**Theorem 2.10.** *Assume that the approximate value function fulfils*

$$\|V - \tilde{V}\|_{L^\infty(\Omega)} \leq \varepsilon.$$

*Then we have for the descent in each step*

$$V(f(x, \tilde{\mathbf{u}}(x))) \leq V(x) - c_0(x) + 4\varepsilon.$$

*Proof.* Let  $u_*$  be the optimal control for  $x$ , i.e.

$$V(x) = c(x, u_*) + V(f(x, u_*)).$$

As a consequence,

$$\tilde{V}(x) \geq c(x, u_*) + \tilde{V}(f(x, u_*)) - 2\varepsilon \geq c(x, \tilde{u}) + \tilde{V}(f(x, \tilde{\mathbf{u}}(x))) - 2\varepsilon,$$

implying

$$\tilde{V}(f(x, \tilde{\mathbf{u}}(x))) \leq \tilde{V}(x) - c(x, \tilde{u}) + 2\varepsilon \leq \tilde{V}(x) - c_0(x) + 2\varepsilon,$$

and so

$$V(f(x, \tilde{\mathbf{u}}(x))) \leq V(x) - c_0(x) + 4\varepsilon. \quad \square$$

So the value function decreases along a trajectory of  $\tilde{\mathbf{u}}$  as long as  $c_0(x) > 4\varepsilon$ .

In our next theorem, we will treat the change of the approximate value function instead of the exact value function.

We define the *Bellman residual*

$$e(x) := \inf_{u \in U} [c(x, u) + \tilde{V}(f(x, u))] - \tilde{V}(x), \quad x \in \mathcal{S},$$

and show that  $\tilde{V}$  decreases along a trajectory of the closed-loop system if the set

$$R_\eta := \{x \in \mathcal{S} \mid e(x) \leq \eta \tilde{c}(x)\}, \quad \eta \in (0, 1),$$

## 2. The Shepard discretization of the optimality principle

where the Bellman residual is (at least by a constant rate) smaller than  $\tilde{c}(x) := c(x, \tilde{\mathbf{u}}(x))$ , contains a sublevel set of  $\tilde{V}$  and we choose the initial condition in this set. One can show that the ratio of the descent of the approximate value function and the one-step-cost can be controlled as long as the value function is not too small.

Evidently, the inequality in the definition of  $R_\eta$  can only be checked a posteriori after some approximate optimal value function  $\tilde{V}$  has been computed. The size of the largest sublevel set of  $\tilde{V}$  which is contained in  $R_\eta$  strongly depends on the example and the approximation quality of  $\tilde{V}$  (cf. also examples 3.3 and 3.4).

**Theorem 2.11.** *Suppose that  $D_C = \{x \in \mathcal{S} \mid \tilde{V}(x) < C\} \subset R_\eta$  for some  $C > 0$ . Then for any  $x_0 \in D_C$ , the associated trajectory generated by the closed-loop system  $x_{j+1} = f(x_j, \tilde{\mathbf{u}}(x_j))$ ,  $j = 0, 1, \dots$ , stays in  $D_C$  and satisfies*

$$\tilde{V}(x_\ell) \leq \tilde{V}(x_0) - (1 - \eta) \sum_{j=0}^{\ell-1} c(x_j, \tilde{\mathbf{u}}(x_j)).$$

*Proof.* Since  $e(x) = \tilde{c}(x) + \tilde{V}(f(x, \tilde{\mathbf{u}}(x))) - \tilde{V}(x)$ , we have for  $x_j \in D_C \subset R_\eta$

$$\tilde{V}(x_{j+1}) = \tilde{V}(x_j) - \tilde{c}(x_j) + e(x_j) < \tilde{V}(x_j) < C,$$

thus  $x_{j+1} \in D_C$ , which shows that the closed-loop trajectory stays in  $D_C$ . Further,

$$\begin{aligned} \tilde{V}(x_{j+1}) &= \tilde{V}(x_j) - c(x_j, \tilde{\mathbf{u}}(x_j)) + e(x_j) \\ &\leq \tilde{V}(x_j) - (1 - \eta)c(x_j, \tilde{\mathbf{u}}(x_j)), \end{aligned}$$

which shows the decay property.  $\square$

The strong assumption of Theorem 2.11 is harder to obtain if  $\tilde{c}$  is small. For that reason, we give a variant of the proposition which is similar to Theorem 5 in [GJ05].

To this end, we define the function  $\delta$  which connects  $c_0$  and  $V$  in the sense that

$$\delta(s) := \sup_{x \in \{c_0 \leq s\}} V(x),$$

where  $c_0$  is defined as before by  $c_0(x) := \min_{u \in U} c(x, u)$ , and  $\{c_0 \leq s\}$  is a shorthand notation for  $c_0^{-1}([0, s])$ .

**Theorem 2.12.** *Assume that the approximate value function fulfils*

$$\|V - \tilde{V}\|_{L_\infty(\Omega)} \leq \varepsilon$$

and

$$e(x) \leq \max\{\eta c_0(x), \varepsilon_0\}$$

## 2.6. Construction of a stabilizing feedback

for all  $x \in D \subset \Omega$ , some  $\varepsilon_0 > 0$  and some  $\eta \in (0, 1)$ . Then for each  $x \in D$  and  $x' = f(x, \tilde{\mathbf{u}}(x))$  one has

$$\tilde{V}(x') \leq \tilde{V}(x) - (1 - \eta)c(x, \tilde{\mathbf{u}}(x))$$

or

$$\tilde{V}(x') \leq \delta(\varepsilon_0/\eta) + \varepsilon - \min_{x \in D} c_0(x).$$

*Proof.* If  $\tilde{V}(x') \leq \delta(\varepsilon_0/\eta) + \varepsilon - \min_{x \in D} c_0(x)$ , there is nothing to show.

Otherwise  $\tilde{V}(x') > \delta(\varepsilon_0/\eta) + \varepsilon - \min_{x \in D} c_0(x)$ .

On the other hand,

$$\begin{aligned} V(x) &= \min_{u \in U} [c(x, u) + V(f(x, u))] \\ &\geq \min_{u \in U} [c(x, u) + \tilde{V}(f(x, u))] - \varepsilon \\ &= c(x, \tilde{\mathbf{u}}(x)) + \tilde{V}(f(x, \tilde{\mathbf{u}}(x))) - \varepsilon \\ &= c(x, \tilde{\mathbf{u}}(x)) + \tilde{V}(x') - \varepsilon \\ &\geq c_0(x) + \tilde{V}(x') - \varepsilon \\ &\geq \min_{x \in D} c_0(x) + \tilde{V}(x') - \varepsilon \\ &> \delta(\varepsilon_0/\eta), \end{aligned}$$

so  $V(x) > \delta(\varepsilon_0/\eta)$ , implying  $c_0(x) > \varepsilon_0/\eta$ , i.e.  $\eta c_0(x) > \varepsilon_0$  by definition of  $\delta$ , and so  $e(x) \leq \eta c_0(x)$  by the bound on  $e$ . This in turn implies

$$\begin{aligned} \tilde{V}(x') &= \tilde{V}(x) - c(x, \tilde{\mathbf{u}}(x)) + e(x) \\ &\leq \tilde{V}(x) - c(x, \tilde{\mathbf{u}}(x)) + \eta c_0(x) \leq \tilde{V}(x) - (1 - \eta)c(x, \tilde{\mathbf{u}}(x)). \end{aligned}$$

□

By iteration of the statement of the theorem, one gets the following corollary. If the first case of the theorem occurs in each of the  $l$  steps, the first case in the corollary occurs. If, however, in at least one step the second case of the theorem occurs, the second case in the corollary occurs.

**Corollary 2.13.** *Assume that the approximate value function fulfils*

$$\|V - \tilde{V}\|_{L_\infty(\Omega)} \leq \varepsilon$$

and

$$e(x) \leq \max\{\eta c_0(x), \varepsilon_0\}$$

for all  $x \in D \subset \Omega$ , some  $\varepsilon_0 > 0$  and some  $\eta \in (0, 1)$ . Then for each  $x_0 \in D$  and the corresponding sequence

$$x_{l+1} = f(x_l, \tilde{\mathbf{u}}(x_l))$$

## 2. The Shepard discretization of the optimality principle

one has

$$\tilde{V}(x_l) \leq \tilde{V}(x_0) - (1 - \eta) \sum_{j=0}^{l-1} c(x_j, \tilde{\mathbf{u}}(x_j))$$

or

$$\tilde{V}(x_l) \leq \delta(\varepsilon_0/\eta) + \varepsilon - \min_{x \in D} c_0(x).$$

as long as  $x_0, \dots, x_{l-1} \in D$ .

## 2.7. Summary and outlook

In this chapter, which is central to this thesis, we have constructed the Bellman-Shepard operator as discretization of the Bellman operator and have shown that it is a contraction on the Shepard space. Therefore the value iteration converges to the unique fixed point of the Bellman-Shepard operator. A proof of the convergence of the approximate value function towards the exact one for a sequence of centre sets with decreasing fill distance has been given, both for discounted and undiscounted control systems.

Another aspect was the suitability of the method for the construction of trajectories because an approximate value function gives rise to a corresponding feedback. We have shown conditions for this feedback to induce trajectories along which the value function respectively the approximate value function decreases. These estimates allow for points of the state space to be stabilized, i.e. steered close to the target by the feedback.

For an implementation one has to make some choices, specifically of the shape functions, the set of centres and of a scheme for the specification of the shape parameter for different sets of centres. In Chapter 3 we will give numerical examples and also compare some of the different choices which are possible for the Shepard discretization in a benchmark example.

There are many ways in which the approach presented in this chapter can be extended. One possibility for an extension is not to fix the shape parameter  $\sigma$  at the same value for all basis functions  $\varphi_i$ . One could, e.g. either try to choose  $\sigma$  greedily in an optimal way for each  $\varphi_i$  or to implement a multilevel type scheme which works with a scale of values for  $\sigma$ .

The choice of the centre set can be further investigated. Its only important parameter in the convergence proofs is the fill distance. So one has full flexibility in designing the set of centres. An interesting question is if it is generally better to choose the set of centres in a universal form independent of a specific control problem, to adapt it to individual control problems, or even to use an adaptive algorithm, which adds to an initial set of centres further centres in an improved or even optimal way. A criterion which has been applied successfully in similar situations, e.g. in [Grü97] aims at reducing the residual in the Bellman equation

uniformly, by refining regions with large residual. In Chapter 4 we will present an algorithm which incorporates a corresponding adaptive refinement step.

One drawback of the value iteration used here is that the images of all possible centre-control value pairs have to be computed and the corresponding matrix entries have to be stored. It would be nice to have some sort of “fast marching” type algorithm which needs this reachability information only locally. In Chapter 5 we show a connection between the Bellman-Shepard discretization of control problems and stochastic shortest path problems. This allows for the application of a Dijkstra-like algorithm ([Ber07]), which was originally designed for stochastic shortest path problems, on an equivalent discretized Bellman equation.

As a final remark, the approximation space that we use here is rather smooth—in contrast to the value function which in general is only Lipschitz-continuous. If one considers a *relaxed* version of the optimality principle, e.g. in the sense of [LR06], smoother solutions might exist which can be approximated with higher efficiency. If one aims for higher-order convergence, higher-order moving least squares approximations could then be tried, cf. [Fas07]. We did not find convincing results while investigating in these directions, though.



# 3. Implementation and numerical examples

This chapter is dedicated to the numerical implementation of our method. We start with some details about the implementation, present several numerical examples and compare the impact which different choices for the discretization have. Parts of this chapter have been published in [\[JS15\]](#).

## 3.1. Implementation

A function

$$\tilde{v} \in \mathcal{W} = \text{span}(\psi_1, \dots, \psi_n)$$

is defined by the vector  $\hat{v} = (\hat{v}_1, \dots, \hat{v}_n) \in \mathbb{R}^n$  of its coefficients, i.e.  $\tilde{v} = \sum_{i=1}^n \hat{v}_i \psi_i$ , cf. Section 1.4. We can evaluate  $\tilde{v}$  on an arbitrary set of points  $Y = \{y_1, \dots, y_e\} \subset \Omega$  by the matrix-vector product  $A(Y)\hat{v}$ , where  $A(Y)$  is the  $e \times n$ -matrix with entries  $a_{ij} = \psi_j(y_i)$ .

In order to compute  $\hat{v}^{k+1}$  in the value iteration (2.2), and assuming the Kruřkov transform, we need to evaluate  $\Gamma[\tilde{v}^k]$  on  $X$  as in (2.3), i.e. we have to compute

$$\Gamma[\tilde{v}^k](x_i) = \begin{cases} \sup_{u \in U} [e^{-c(x_i, u)} \tilde{v}^k(f(x_i, u))] & x_i \in X \setminus T, \\ 1 & x_i \in X \cap T. \end{cases}$$

In general, this is a nonlinear optimization problem for each  $x_i$ . For simplicity and computational speed, we choose to solve this by simple enumeration, i.e. choosing a finite set  $\tilde{U} = \{u_1, \dots, u_m\} \subset U$  of control values and approximate

$$\Gamma[\tilde{v}^k](x_i) \approx \max_{j=1, \dots, m} \left\{ e^{-c(x_i, u_j)} \tilde{v}^k(f(x_i, u_j)) \right\} \quad (3.1)$$

for each  $x_i \in X \setminus T$ . This introduces an additional error, which, in principle, could be minimized by using some more sophisticated NLP solver (such as `fminsearch` in MATLAB) at the expense of a drastically increased run time. For a thorough discussion of the effect of the discretization of the control space onto the value function see, e.g. [\[FF94\]](#).

Let  $Y' = \{f(x_i, u_j) \mid i = 1, \dots, n, j = 1, \dots, m\}$ , then the values  $\tilde{v}^k(f(x_i, u_j))$  are given by the matrix-vector product  $A(Y')\hat{v}^k$ . From this, the right-hand side of (3.1) can readily be computed.

### 3. Implementation and numerical examples

**Remark.** 1. We have formulated our approach for a discrete-time control system which may, as mentioned in Section 1.3, be obtained via time-sampling from a continuous-time system. In fact, the systems in the following numerical experiments have been obtained this manner. One of the central questions in this case is the error introduced by the time sampling. We do not consider this problem here and refer instead to [CD83, CDF89, Fal87, FF94].

2. In some cases, our numerical examples are given by restrictions of problems on  $\mathbb{R}^s$  to a compact domain  $\Omega \in \mathbb{R}^s$ . In general the dynamical system on  $\mathbb{R}^s$  is a map  $f_1 : \mathbb{R}^s \times U \rightarrow \mathbb{R}^s$  which does not restrict to a map  $f_2 := f_1|_{\Omega} : \Omega \times U \rightarrow \Omega$ . This can be achieved by replacing  $f_2$  with  $f := \Pi \circ f_2$  where  $\Pi$  is a Lipschitz-continuous projection of  $\mathbb{R}^s$  onto  $\Omega$ . In our numerical experiments, it did not matter whether the projection step was included, which is why we decided to omit it from our implementation.

## Distance matrices and kd-trees

For the numerical realization of our algorithms we have to evaluate radially symmetric functions on large sets of points leading to expressions like  $\varphi^\sigma(\|x_i - y_j\|)$  which have to be evaluated for many pairs of points  $(x_i, y_j) \in X \times Y$  where  $X = \{x_i\} \subset \Omega$  is the set of evaluation points and  $Y = \{y_j\}$  is the set of centres. In order to calculate all of them simultaneously and fast, it is best to construct a *distance matrix* which contains all the pairwise distances  $\|x_i - y_j\|$  for all points  $x_i \in X$  and  $y_j \in Y$ .

However, as we use basis functions with compact support  $U_{1/\sigma}(0)$ , most of these expressions  $\varphi^\sigma(\|x_i - y_j\|)$  equal 0 whenever  $\|x_i - y_j\| \geq \frac{1}{\sigma}$ . Therefore, it is desirable to find for each point  $x_i \in X$  all points  $y_j \in Y$  which are close to  $x_i$ . To that end, we employ a tool called *kd-trees* in the implementation by Guy Shechter, which can be found on ([MCF]). The corresponding algorithm subsequently divides the state space in parts in which one continues to search for nearby points.

## 3.2. Example: A simple 1D example

We begin with the simple one-dimensional system

$$f(x, u) = x + aux$$

on  $\Omega = [0, 1]$ ,  $U = [-1, 1]$ , with parameter  $a = 0.8$  and cost function

$$c(x, u) = ax.$$

Apparently, the optimal feedback is  $\mathbf{u}(x) = -1$ , yielding the optimal value function  $V(x) = x$ . For  $j$  from 10 to 1000, we choose equidistant sets  $X^{(j)} = \{0, 1/j, \dots, 1 - 1/j, 1\}$  of centres,  $\tilde{U} = \{-1, -0.9, \dots, 0.9, 1\}$ ,  $T = [0, 1/(2j)]$  and

### 3.3. Example: Shortest path with obstacles

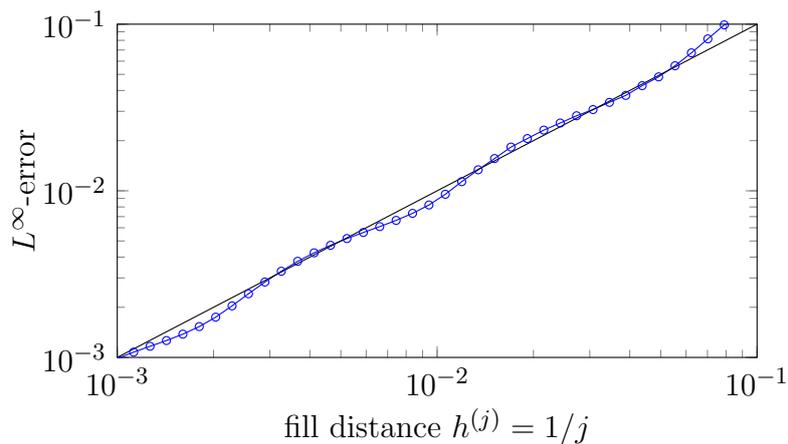


Figure 3.1.:  $L^\infty$ -error of the approximate value function  $\tilde{V}^{(j)} = -\log(\tilde{v}^{(j)})$  in dependence on the fill distance  $1/j$ , where  $j$  is the number of centres.

use the Wendland function  $\varphi^\sigma(r) = \max\{0, (1 - \sigma r)^4(4\sigma r + 1)\}$  as shape function with parameter  $\sigma = j/5$ . In Figure 3.1, we show the  $L^\infty$ -error of the approximate value function  $\tilde{V}^{(j)}$  in dependence on the fill distance  $h^{(j)} = 1/j$  of the set of centres  $X^{(j)}$ . We observe a linear decay of the error in  $h^{(j)}$ .

### 3.3. Example: Shortest path with obstacles

Our next example is supposed to demonstrate that state space constraints can trivially be dealt with, even if they are very irregular: We consider a boat in the Mediterranean Sea surrounding Greece (cf. Fig. 3.2) which moves with constant speed 1. The boat is supposed to reach the harbour of Athens (marked by an x in the map) in shortest time. Accordingly, the dynamics is simply given by

$$f(x, u) = x + hu,$$

where we choose the time step  $h = 0.1$ , with  $U := \{u \in \mathbb{R}^2 : \|u\| = 1\}$ , and the associated cost function by

$$c(x, u) \equiv 1.$$

In other words, we are solving a shortest path problem on a domain with obstacles with complicated shape.

In order to solve this problem by our approach, we choose the set of centres  $X$  as those nodes of an equidistant grid which are placed in the Mediterranean Sea within the rectangle shown in Fig. 3.2, which we normalize to  $[-10, 10]^2$ . We extracted this region from a pixmap of this region with resolution 275 by 257 pixels. The resulting set  $X$  consisted of 50301 centres. We choose  $U = \{\exp(2\pi i j/20) : j = 0, \dots, 19\} \subset \mathbb{C}$  under the identification  $\mathbb{C} \cong \mathbb{R}^2$ . The choice of 20 equidistant points as discretization of the control space does not introduce

### 3. Implementation and numerical examples

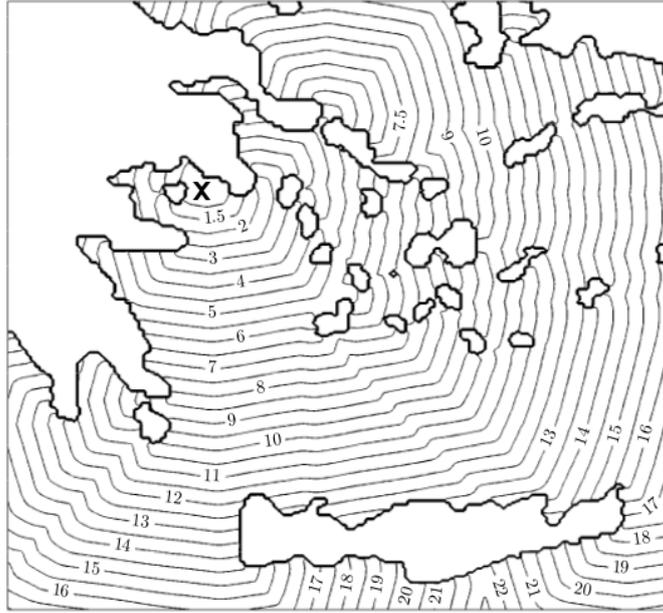


Figure 3.2.: Isolines of the approximate optimal value function for the shortest path example, giving the approximate length of the shortest path from a point in the Mediterranean Sea to the port of Athens (which is marked by an x in the map).

a significant discretization error. In fact, we compared the numerical solution to the ones obtained for much finer control space discretizations and there are no visible differences. The position of Athens in our case is approximately given by  $A = (-4, 4)$  and we choose  $T = A + 0.004 \cdot [-1, 1]^2$ . We again use the Wendland function  $\varphi^\sigma(r) = \max\{0, (1 - \sigma r)^4(4\sigma r + 1)\}$  as shape function with parameter  $\sigma = 10$ .

In Figure 3.2, we show some isolines of the approximate optimal value function. The computation took around 10 seconds on a 2.6 GHz Intel Core i5.

### 3.4. Example: An inverted pendulum on a cart

Our next example is two-dimensional as well, with only box constraints on the states, but the stabilization task is more challenging: We consider balancing a planar inverted pendulum on a cart that moves under an applied horizontal force, cf. [JO04] and Figure 3.3.

The configuration of the pendulum is given by the offset angle  $\varphi$  from the vertical upright position. We do not include the position or motion of the cart in the state space. Correspondingly, the state of the system is  $x = (x_1, x_2) := (\varphi, \dot{\varphi}) \in \mathbb{R}^2$ . The equation of motion becomes

$$\left(m_r \cos^2(\varphi) - \frac{4}{3}\right) \ddot{\varphi} - \frac{1}{2} m_r \sin(2\varphi) \dot{\varphi}^2 + \frac{g}{\ell} \sin(\varphi) - \frac{m_r}{m \ell} \cos(\varphi) u = 0, \quad (3.2)$$

### 3.4. Example: An inverted pendulum on a cart

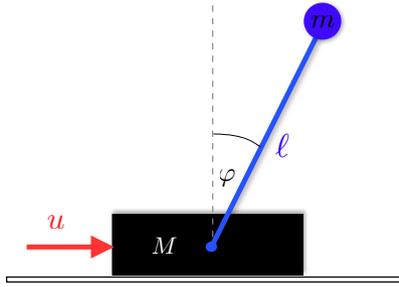


Figure 3.3.: Model of the inverted pendulum on a cart.

where  $M = 8\text{ kg}$  is the mass of the cart,  $m = 2\text{ kg}$  the mass of the pendulum and  $\ell = 0.5\text{ m}$  is the distance of the centre of mass from the pivot. We use  $m_r = m/(m + M)$  for the mass ratio and  $g = 9.8\text{ m/s}^2$  for the gravitational constant. The stabilization of the pendulum is subject to the cost

$$c(x, u) = c((\varphi, \dot{\varphi}), u) = \frac{1}{2} (0.1\varphi^2 + 0.05\dot{\varphi}^2 + 0.01u^2). \quad (3.3)$$

For our computations, we need to obtain a discrete-time control system. To this end, we consider the time sampled system with sampling period  $h = 0.1$  and keep the control  $u(t)$  constant during this period. The time sampling map has been computed via five steps of the classical Runge-Kutta scheme of order 4 with step size 0.02. We choose  $\Omega = [-8, 8] \times [-10, 10]$  as the region of interest,  $T = \{0\}$  as the target set and the set  $X$  of centres as an equidistant  $128 \times 128$  grid (cf. Code 1), and  $\tilde{U} = \{-128, 120, \dots, 120, 128\}$ . The parameters and also the control space were chosen as in [JO04] to allow for a comparison. We again use the Wendland function  $\varphi^\sigma(r) = \max\{0, (1 - \sigma r)^4(4\sigma r + 1)\}$  as shape function, the shape parameter  $\sigma$  is chosen such that the support of each  $\varphi_i$  overlaps with the supports of roughly 20 other  $\varphi_i$ 's, i.e.  $\sigma \approx 2.22$  here.

In Figure 3.4, we show the behaviour of the (relative)  $L^\infty$ -approximation error during the value iteration. We observe geometric convergence as expected by Theorem 2.1. The computation of the optimal value function took 13 seconds.

In Figure 3.5, we compare the resulting value function (left) to the one computed by the method from [JO04] (right) on a partition of  $2048 \times 2048$  boxes. Note that the latter one is a pointwise lower bound on the true value function. In our experiments, the functions on partitions with fewer elements took considerably smaller values and we therefore believe that the Shepard-RBF approximation is accurate.

In Figure 3.6, some isolines of the approximate value function are shown, together with the complement of the set  $R_1$  (cf. Theorem 2.11) as well as the first 50 points of a trajectory of the closed-loop system with the optimal feedback. Note that the discrete-time system has been derived from a controlled ordinary differential equation and that we work with a fixed time step, leading to the oscillatory behaviour of the feedback trajectory close to the origin.

### 3. Implementation and numerical examples

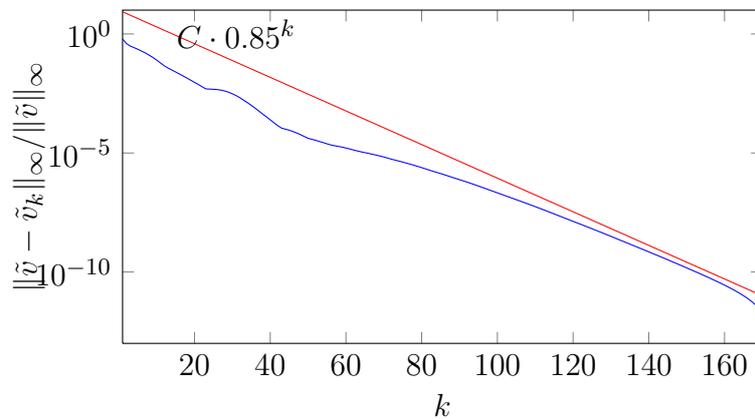


Figure 3.4.: Inverted pendulum: Relative  $L^\infty$ -error of  $\tilde{v}_k$  in dependence on the number of iterations  $k$  in the fixed point iteration (2.2). Here, we used the iterate  $\tilde{v}_{173}$  as an approximation to the true fixed point  $\tilde{v}$ , since the error is approximately at machine precision then.

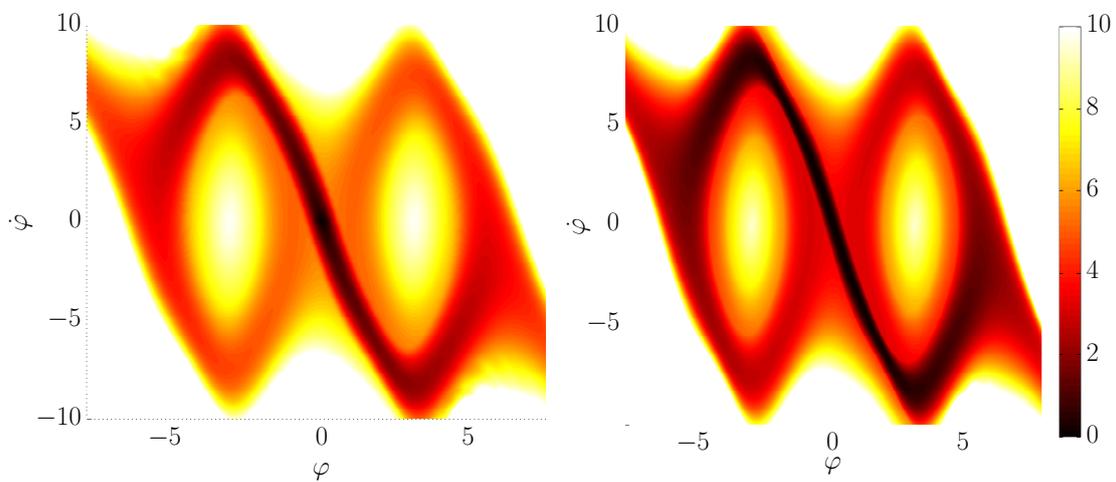


Figure 3.5.: Inverted pendulum: Approximate optimal value function on a  $128 \times 128$  grid using the Shepard-RBF method described here (left) and the graph based method from [JO04] on a partition of  $2048 \times 2048$  boxes (right).

3.4. Example: An inverted pendulum on a cart

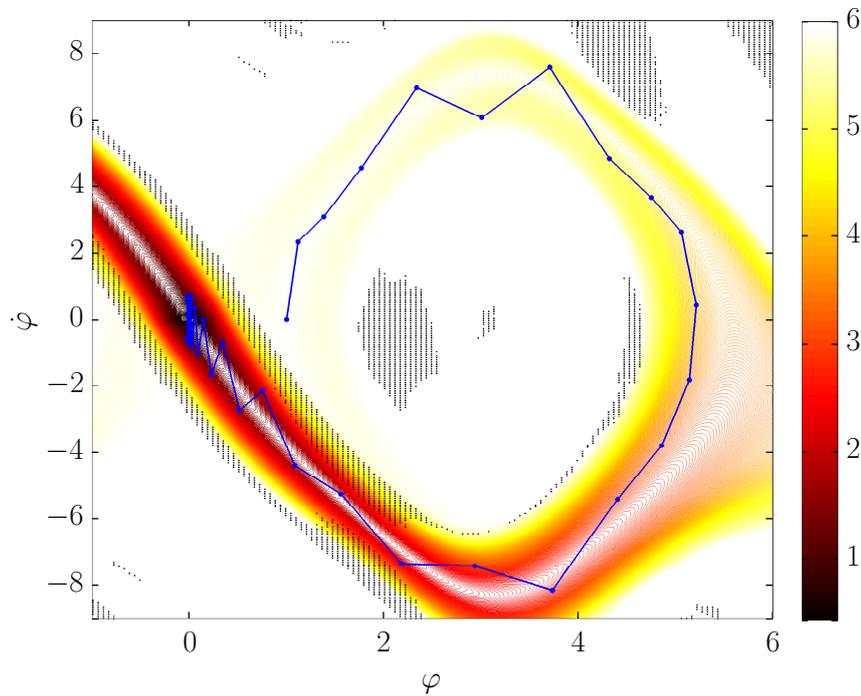


Figure 3.6.: Inverted pendulum: Approximate optimal value function  $\tilde{V} = -\log(\tilde{v}(\cdot))$  (isolines colour coded), together with the set  $\Omega \setminus R_1$ , where the Bellman residual  $e$  is larger than  $\tilde{c}$  (black dots) and a trajectory of the closed-loop system starting at the initial value  $(1, 0)$  (blue).

### 3. Implementation and numerical examples

```

1 %% an inverted pendulum on a cart
2 m = 2; M = 8; l = 0.5; g = 9.8; q1 = 0.1; q2 = 0.05;
3 r0 = 0.01; m_r = m/(m+M);
4 vf = @(x,u) [ x(:,2), ... % vector field
5             (g/l*sin(x(:,1)) - 0.5*m_r*x(:,2).^2.*sin(2*x(:,1)) - ...
6             m_r/(m*l)*u.*cos(x(:,1)))/(4.0/3.0-m_r*cos(x(:,1)).^2), ...
7             0.5*( q1*(x(:,1).^2) + q2*(x(:,2).^2) + r0*u.^2 )];
8 h = 0.02; steps = 5; % step size
9 f = @(x,u) rk4u(vf,[x zeros(size(x,1),1)],u,h,steps); % control system
10 phi = @(r) max(spones(r)-r,0).^4.*(4*r+spones(r)); % Wendland func.
11 T = [0 0]; v_T = 1; % boundary cond.
12 shepard = @(A) spdiags(1./sum(A'),' ,0,size(A,1),size(A,1))*A; % Shep.op.
13 S = [8,10]; % radii of domain
14 L = 33; U = linspace(-128,128,L)'; % control values
15 N = 128; X1 = linspace(-1,1,N);
16 [XX,YY] = meshgrid(X1*S(1),X1*S(2)); X = [XX(:) YY(:)]; % nodes
17 XU = kron(X,ones(size(U,1),1)); UX = kron(ones(size(X,1),1),U);
18 ep = 1/sqrt((4*prod(S)*20/N^2)/pi); % shape parameter
19 fcXU = f(XU,UX);
20 A = shepard(phi(ep*sdistm(fcXU(:,1:2),[T;X],1/ep))); % Shepard matrix
21 C = exp(-fcXU(:,3)); % one step costs
22
23 %% value iteration
24 v = zeros(N^2+1,1); v0 = ones(size(v)); TOL = 1e-12;
25 while norm(v-v0,inf)/norm(v,inf) > TOL
26     v0 = v;
27     v = [v_T; max(reshape(C.*(A*v),L,N^2))']; % Bellman op.
28 end

```

Code 1: MATLAB code for the inverted pendulum example. Here,  $A = \text{sdistm}(X,Y,r)$  is the sparse matrix of pairwise Euclidean distances between the points in the rows of  $X$  and  $Y$  not exceeding distance  $r$ . An implementation is available on the webpage [1] of the authors of [JS15].

Finally, the behaviour of the  $L^\infty$ -error of the approximate optimal value function in dependence on the fill distance  $h$  is shown in Figure 3.7. Here, we used the value function with fill distance  $h = 0.02$  as an approximation to the true one. Again, we observe an essentially linear decay of the error. The corresponding MATLAB code for this example is given in Code 1.

## 3.5. Example: Magnetic wheel

Lastly, we consider an example with three-dimensional state space: The stabilization of a *magnetic wheel*, used in magnetic levitation trains, cf. [GMM79] and Figure 3.8.

A point in state space is given by the gap  $s$  (in meters) between the magnet and the track, its change rate  $\dot{s} = v$  (in  $m/s$ ) and the electrical current  $J$  (in Ampere) through the magnet. The control is the voltage  $u$  applied to the circuit.

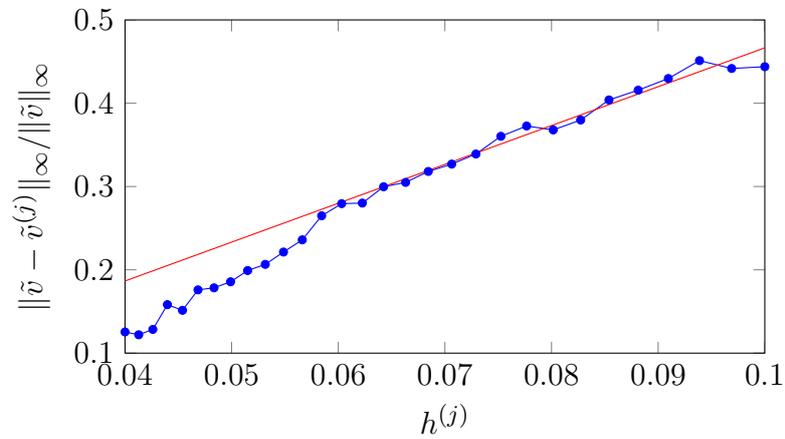


Figure 3.7.: Inverted pendulum: Relative  $L^\infty$ -error of the approximate optimal value function  $\tilde{v}^{(j)}$  in dependence on the fill distance of the centres. Here, we used the value function for fill distance 0.02 as an approximation to the true one.

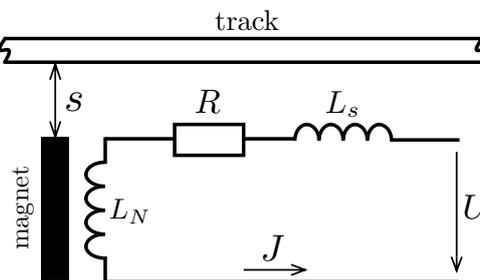


Figure 3.8.: Model of the magnetic wheel.

### 3. Implementation and numerical examples

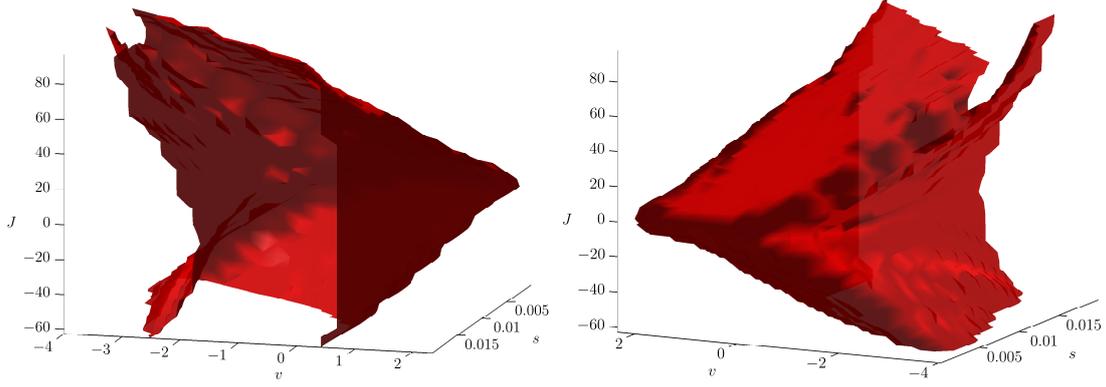


Figure 3.9.: Magnetic wheel example: The set  $\{x \in \Omega \mid \tilde{v}(x) > 10^{-20}\}$  from two perspectives.

The dynamics is given by

$$\begin{aligned} \dot{s} &= v, \\ \dot{v} &= \frac{CJ^2}{m_m 4s^2} - \mu g, \\ \dot{J} &= \frac{1}{L_s + \frac{C}{2s}} \left( -RJ + \frac{C}{2s^2} Jv + u \right), \end{aligned}$$

where  $C = L_N 2s_0$ , the target gap  $s_0 = 0.01$ , the inductance  $L_N = 1$  of the magnet, the magnet mass  $m_m = 500$ , the ratio of the total mass and the magnet mass  $\mu = 3$ , the resistance  $R = 4$ , the leakage inductance  $L_s = 0.15$  and the gravitational constant  $g = 9.81$ . The system is subject to the cost function

$$c(x, u) = c((s, v, J), u) = \frac{1}{2} (100(s - s_0))^2 + v^2 + 0.002u^2. \quad (3.4)$$

We consider the time sampled system with sampling period  $h = 0.001$ , approximated by one explicit Euler step and keep the control  $u(t)$  constant during the sampling period. The model has an unstable equilibrium at approximately  $x_0 := (s_0, v_0, J_0) = (0.01, 0, 17.155)$ , which we would like to stabilize by an optimal feedback. We choose  $\Omega = [0, 0.02] \times [-4, 4] \times [J_0 - 80, J_0 + 80]$  as state space,  $T = \{x_0\}$  as the target set,  $\tilde{U} = \{6 \cdot 10^3 a^3 \mid a \in \{-1, -0.99, \dots, 0.99, 1\}\}$  as the set of controls, an equidistant grid  $X$  of  $30 \times 30 \times 30$  centres in  $\Omega$ , the Wendland function  $\varphi^\sigma(r) = \max\{0, (1 - \sigma r)^4(4\sigma r + 1)\}$  as shape function with shape parameter  $\sigma = 11.2$ , such that the support of each  $\varphi_i$  overlaps with the supports of roughly 10 other  $\varphi_i$ 's. The computation of the value function took around 60 seconds. In Figure 3.9, we show a subset of the stabilizable subset of  $\Omega$ , i.e. we show the set  $\{x \in \Omega \mid \tilde{v}(x) > 10^{-20}\}$ .

### 3.6. Dependence on the shape parameter

In this and the following two sections, we explore the impact of different choices for some numerical aspects of our method. These include the shape parameter, the set of RBF centres and the shape functions.

As a benchmark example, we use the inverted pendulum from Section 3.4.

The basis functions depend heavily on the shape parameter. As long as the set of centres is chosen as a regular grid, it is usually best to choose a uniform shape parameter for all basis functions.

Unfortunately, the choice of this shape parameter is not obvious.

We suggested to use stationary approximation in Section 2.2, meaning that for changing sets of centres, the shape parameter is chosen inversely proportional to the fill distance of the centre set, explicitly  $\sigma^{(j)} := C_1/h^{(j)}$  in (2.9) for a constant ratio  $C_1 > 0$ . For our experiments we chose the shape parameter such that about 10 other centres are in the support of each basis function. This specification is consistent with the existence of such a constant ratio  $C_1 > 0$  which, for the pendulum example and equidistant centre sets is  $C_1 = 0.32$ .

We compared this choice to  $C_1 = 0.32 \cdot C_2$  for some numbers  $C_2$  close to 1.

Figure 3.10 shows

- the relative  $L^\infty$ -approximation error for the first 100 value iterations for a relatively dense centre set ( $h = 0.0315$ ), and
- the error of the numerical solution (estimated by the comparison with the numerical solution for a fine centre set) for centre sets with different fill distance  $h^{(j)}$ ,

where the five graphs correspond to the scaling parameter values  $C_2 = 0.5, 0.7$ , etc.

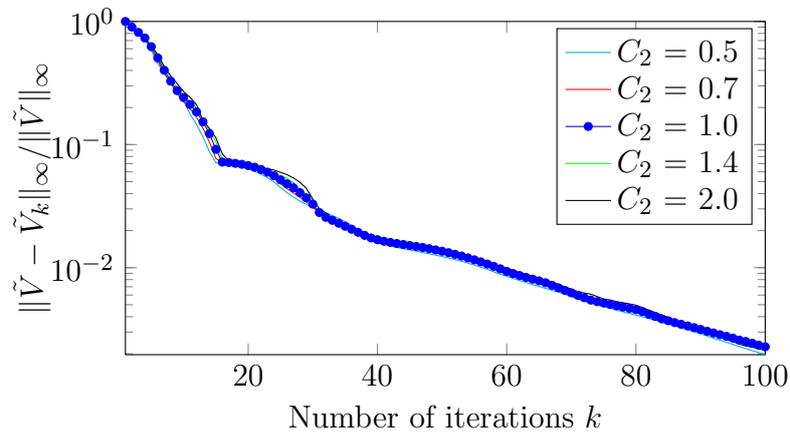
In (a) we see that the shape parameter is irrelevant for the convergence of the value iteration; in  $L^\infty$ -norm the convergence corresponds to the worst case estimate for the contraction in all five cases.

More interesting is (b), the comparison of the errors of the limit functions from the value iteration. Our choice of  $C_2 = 1$ , i.e.  $C_1 = 0.32$  appears to be reasonable: The asymptotic behaviour of the  $L_\infty$ -error compares well to other  $C_2$ -values.

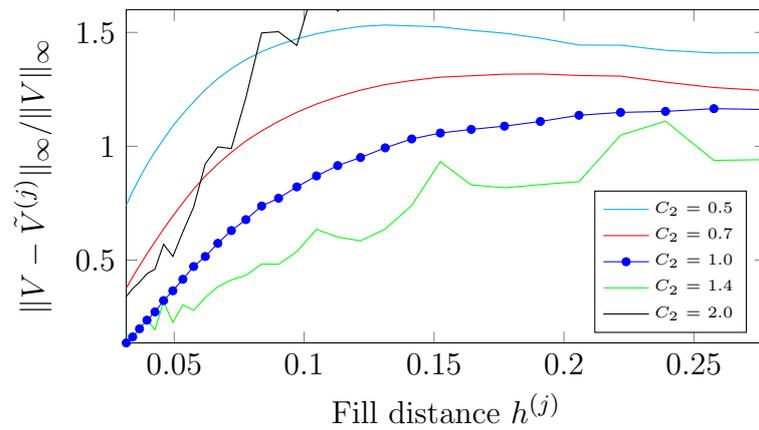
### 3.7. Dependence on the set of centres

So far, we used equidistant sets of centres for the Shepard discretization. In this section, we compare the equidistant choice with sets of centres consisting of random points and two (related) types of so-called pseudo-random points, which we briefly introduce here. More details can be found in [KN74] or [Nie92].

### 3. Implementation and numerical examples



(a)



(b)

Figure 3.10.: (a) Relative  $L^\infty$ -error in dependence on the number of iterations of the value iteration. (b) Relative  $L^\infty$ -error of the approximate value function in dependence on the fill distance.

Both figures have five graphs for different scaling factors  $C_2$  for the shape parameter.

**Definition 3.1** (pseudo-random sequence, [Nie92]). A pseudo-random (or low-discrepancy) sequence in  $[0, 1]^s$  is a sequence of points  $x_1, \dots$  whose initial subsequences  $X_N := (x_1, \dots, x_N)$  have low discrepancy  $D_N(X_N) := \sup_{B \in \mathbb{B}} |\lambda_N(B) - \lambda_s(B)|$  where  $\lambda_s$  is the  $s$ -dimensional Lebesgue measure,  $\lambda_N$  is  $\frac{1}{N}$  times the point measure induced by  $X_N$  and  $\mathbb{B}$  is the set of cuboids in  $[0, 1]^s$  which are parallel to the axes.

This is only a “pseudo-definition” as it is not universally agreed what a “low” discrepancy is. Anyway, the best known sequences fulfil the inequality

$$D_N(X_N) \leq C(s) \frac{(\ln N)^s}{N}, \quad \text{for all } N \in \mathbb{N},$$

with a constant  $C(s) > 0$  which depends only on the dimension  $s$ . The sequences to be defined below fulfil this asymptotic behaviour, although with constants  $C(s)$  which are larger than the smallest known ones.

The name “pseudo-random” refers to the fact that pseudo-random sequences share similarities with random sequences. Specifically, both have low discrepancy.

A classic example for pseudo-random sequences in one dimension are the van der Corput sequences.

**Definition 3.2** (Van der Corput sequence, [Nie92]). The van der Corput sequence for some integer  $b \geq 2$  is given by  $x_n = \sum_{j=0}^{\infty} a_j(n)b^{-j-1}$ ,  $n = 1, 2, \dots$  for the base- $b$  representation  $n = \sum_{j=0}^{\infty} a_j(n)b^j$ .

The points of this sequence lie in the interval  $[0, 1]$ .

For the multidimensional case one can combine van der Corput sequences for different bases to define Halton points.

**Definition 3.3** (Halton points, [Nie92]). For the dimension  $s$  and pairwise relative prime integers (e.g. pairwise disjoint prime numbers)  $p_1, \dots, p_s$ , we have  $s$  van der Corput sequences  $x_1^k, x_2^k, \dots$  for the bases  $p_k$ , for  $k = 1, 2, \dots, s$ . The corresponding Halton sequence in  $[0, 1]^s$  is defined as

$$(x_1^1, \dots, x_1^s), (x_2^1, \dots, x_2^s), \dots$$

In order to avoid that some special property of Halton points have an effect for the Shepard discretization, we also use another multidimensional pseudo-random sequence for comparison. Namely, the multidimensional van der Corput sequence is constructed from the one-dimensional one by using the standard bijection between  $\mathbb{R}^s$  and  $\mathbb{R}$  (or  $[0, 1]^s$  and  $[0, 1]$ ) which combines digits in the sense that, e.g. for dimension  $s = 2$  and base  $b = 10$  the point  $(0.a_1^1 a_2^1 a_3^1 \dots, 0.a_1^2 a_2^2 a_3^2 \dots)$  is mapped to  $0.a_1^1 a_1^2 a_2^1 a_2^2 a_3^1 a_3^2 \dots$  and correspondingly for  $s \geq 3$  and other bases  $b$ . More formally, we have the following definition.

### 3. Implementation and numerical examples

**Definition 3.4.** Let  $\varphi_s : \mathbb{R}^s \rightarrow \mathbb{R}$  be given by, using the representations  $x^k = \sum_{j=0}^{\infty} a_j^k(n)b^{-j-1}$ ,

$$\varphi_s(x) := \sum_{k=1}^s \sum_{j=0}^{\infty} a_j^k(n)b^{-(s(j+1)+k)}.$$

We use the inverse of  $\phi_s$  to map a van der Corput sequence into  $[0, 1]^s$ .

**Definition 3.5** (Multidimensional van der Corput sequence). Let  $(x_n)$  be the van der Corput sequence for basis  $b \geq 2$ . Then we define the multidimensional van der Corput sequence by

$$x^s(n) := \varphi_s^{-1}(x_n).$$

We give a compact implementation for the construction of this sequence in Code 2.

```

1 function X = md_vdC_sequence(n,s)
2 %     n = Number of points, s = dimension, b = Basis of b-ary numbers = 2
3 b = 2;
4 dig = floor(log(n)/log(b)/s)+1;
5 [Dig,N] = meshgrid(0:s*dig-1,1:n);
6 A = mod(floor(N./b.^Dig),b);
7 B = kron(b.^(1:dig)', eye(s));
8 X = A*B;

```

Code 2: MATLAB code for the construction of the multidimensional van der Corput sequence.

For the application for the pendulum example, one has to rescale the square  $[0, 1]^2$  with the set of centres to the rectangular state space.

In Figure 3.12 we have drawn error plots corresponding to the ones in Figure 3.10. This time the convergence of the value iteration is somewhat different for the four centre sets, but not significantly. The convergence behaviour of the limiting functions suggests that equidistant centre sets behave much better than random or pseudo-random sets of centres.

## 3.8. Dependence on the shape function

A crucial aspect of RBF methods is the question which shape function is used. In most cases we use the Wendland function  $\varphi_{3,1}(r) := (1-r)_+^4(4r+1)$ .

Here, we test and compare different shape functions for the inverted pendulum example.

As mentioned in Section 1.4, the Wendland functions ([Wen95, Fas07]) are defined by  $\varphi(r) = \max\{0, P(r)\}$ , where  $P$  is an appropriate polynomial which

### 3.8. Dependence on the shape function

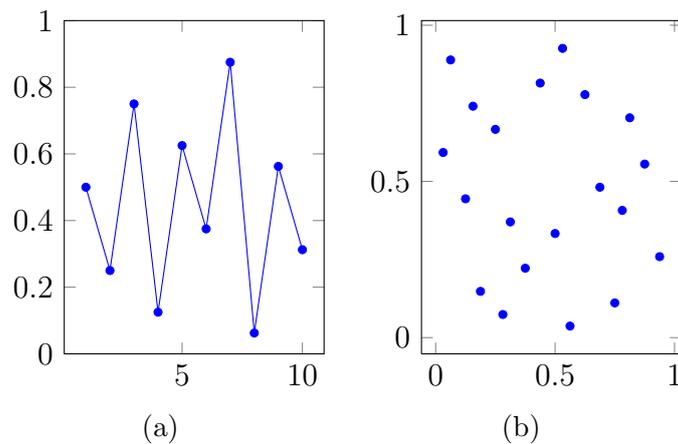


Figure 3.11.: The first points of Halton sequences: Left: The first few members of the one-dimensional Halton sequence in their order of appearance. Right: The first few unordered members of the two-dimensional Halton sequence.

leads to  $\varphi$  being strictly positive definite, having a certain smoothness and among these having the minimal possible degree. For their construction, the following integral operator is applied.

**Definition 3.6.** Let

$$(I\varphi)(r) := \int_r^\infty t\varphi(t)dt, \quad r \geq 0,$$

assuming the integral exists and is finite.

Obviously, this operator maps polynomials to polynomials and the same holds true for piecewise polynomials. The Wendland functions  $\varphi_{s,k}$  are defined as follows.

**Definition 3.7.** One sets  $f_+ := \max(0, f)$ ,

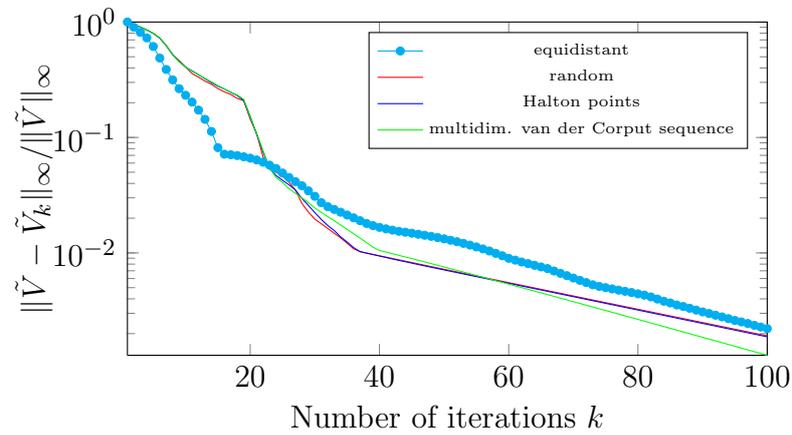
$$\varphi_l(r) = (1-r)_+^l \text{ and}$$

$$\varphi_{s,k} := I^k \varphi_{[s/2]+k+1}.$$

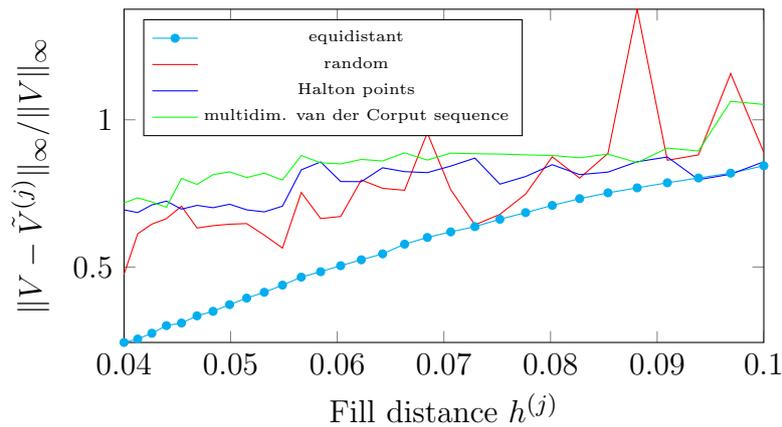
By definition of the Gauss bracket in  $[s/2]$ , we have that  $\varphi_{2,k} = \varphi_{3,k}$  for all  $k \in \mathbb{N}$ . Consequently, the Wendland functions in dimensions 2 and 3 agree. The first few are given by

$$\begin{aligned} \varphi_{3,0}(r) &= (1-r)_+^2, \\ \varphi_{3,1}(r) &= (1-r)_+^4(4r+1), \\ \varphi_{3,2}(r) &= (1-r)_+^6(35r^2+18r+3) \text{ and} \\ \varphi_{3,3}(r) &= (1-r)_+^8(32r^3+25r^2+8r+1). \end{aligned}$$

### 3. Implementation and numerical examples



(a)



(b)

Figure 3.12.: (a) Relative  $L^\infty$ -error in dependence on the number of iterations of the value iteration. (b) Relative  $L^\infty$ -error of the approximate value function in dependence on the fill distance.

Both figures have four graphs for equidistant (cyan), random (red), Halton points (blue) and multidimensional van der Corput (green) sets of centres.

### 3.8. Dependence on the shape function

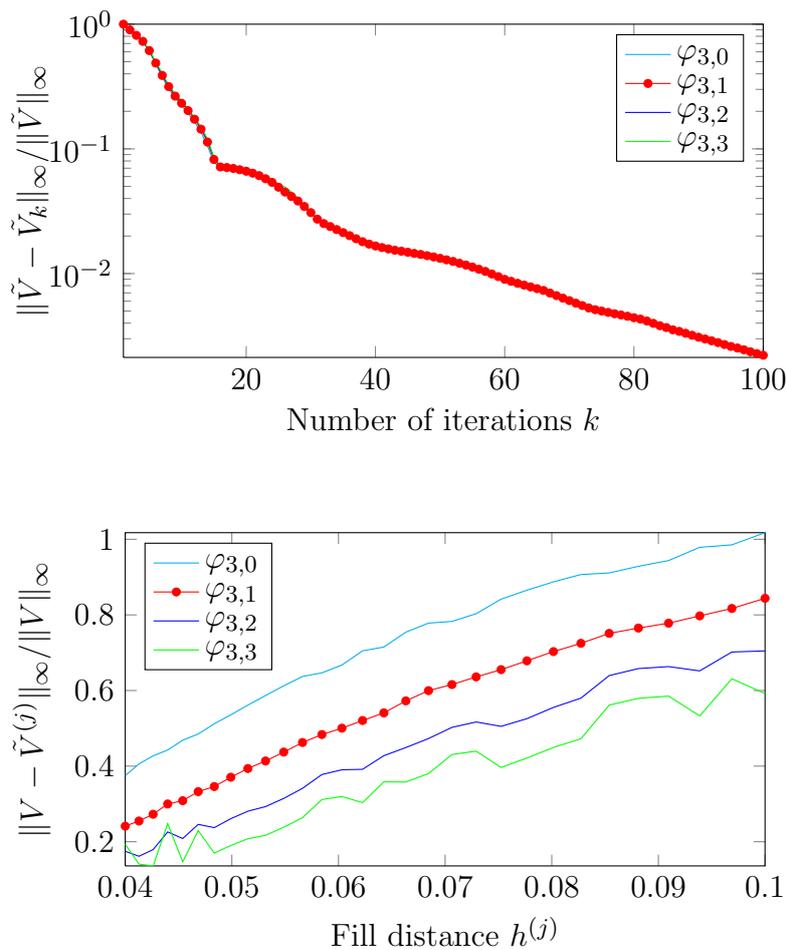


Figure 3.13.: (a) Relative  $L^\infty$ -error in dependence on the number of iterations of the value iteration. (b) Relative  $L^\infty$ -error of the approximate value function in dependence on the fill distance.

Both figures have four graphs for different shape functions.

We have already plotted these functions in Figure 1.1, right, in Chapter 1.

In Figure 3.13 we have drawn error plots corresponding to the ones in Figure 3.10. The convergence of the value iteration does not depend on the shape function; with the exception of the first few iterations, the convergence corresponds to the worst case for all shape functions.

One sees that those Wendland functions which are defined by higher-order polynomials have the tendency to lead to somewhat smaller errors of the approximate solution. This is at the expense of a slightly increased numerical effort, though.



## 4. Adaptive choice of the centres

Discretizations with radial basis functions allow for the choice of diverse centre sets. In order to take advantage of this flexibility, we compared different options in Section 3.7. It turned out that the RBF discretization with random or pseudo-random sets of centres behaves significantly worse than with equidistant sets of centres. This experience calls for a more sophisticated way of selecting the centres for the basis functions.

In this chapter we develop an algorithm for adaptively adding new centres to an existing set of centres. As the aim is to achieve a small (global) error of the approximate solution, we try to add centres in regions with comparatively large (local) error. An estimate between the error and the *residual* allows for finding the regions with large error, because the calculation of the residual is possible without knowledge of the exact solution.

We remind the reader that in Section 1.4, we specified to use radial basis functions for designing the approximation space  $\mathcal{A}$ , i.e. functions  $\varphi_i : \Omega \subset \mathbb{R}^s \rightarrow \mathbb{R}$  of the form  $\varphi_i(x) = \varphi_i^\sigma(x) = \varphi(\sigma\|x - x_i\|_2)$  on some set  $X = \{x_1, \dots, x_n\} \subset \Omega$  of centres. In this chapter we use the notation  $\varphi_i(x; \sigma)$  instead of  $\varphi_i^\sigma(x)$  to make the notation more readable.

The shape parameter  $\sigma$ , which controls the “width” of the radial basis functions, had, up to now, to be chosen globally for a set of centres  $X$ . In this chapter we allow to choose it individually for the different basis functions.

To simplify matters we only consider discounted control systems here, so we always have  $0 < \beta < 1$ . In particular, this will be required already in Lemma 4.1.

### 4.1. Adaptive construction of the centres

In Theorem 2.5 we showed the convergence of the approximate value function towards the true value function for any sequence  $(X^{(j)})_j$  of centre sets for which the fill distance decreases to zero. In practice, it would be desirable to construct this sequence such that, e.g. the error decreases as fast as possible.

To this end, we are now going to describe a construction of such a sequence based on an adaptive refinement of  $X^{(j)}$  in state regions where the Bellman residual

$$e^{(j)}(x) := \inf_{u \in U} [c(x, u) + \beta \tilde{V}^{(j)}(f(x, u))] - \tilde{V}^{(j)}(x), \quad x \in X^{(j)},$$

is large. Here,  $\tilde{V}^{(j)}$  is the approximate value function corresponding to the centre set  $X^{(j)}$ . More precisely, in order to construct  $X^{(j+1)}$ , we are going to refine those

#### 4. Adaptive choice of the centres

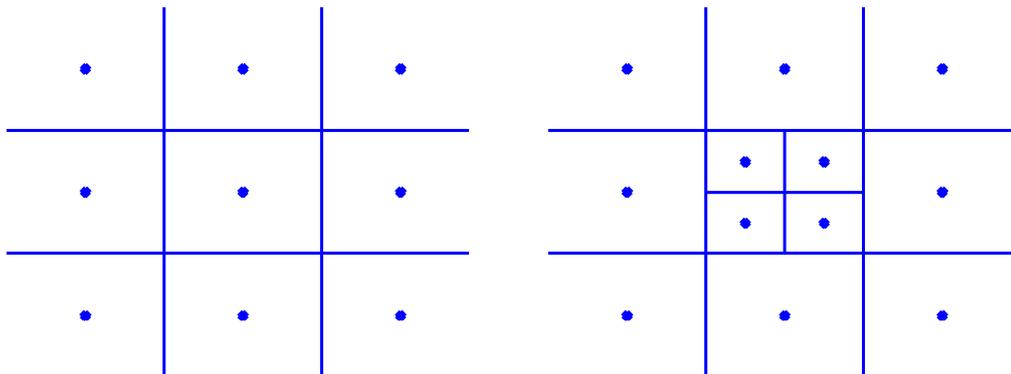


Figure 4.1.: Subdivision: One centre from the left is assumed to be in  $\hat{X}$  and thus subdivided to get the figure on the right.

centres  $x \in X^{(j)}$  for which  $|e^{(j)}(x)|$  is larger than a fixed fraction of the maximal residual  $\max_{x \in X^{(j)}} |e^{(j)}(x)|$ .

Instead of using the same shape parameter  $\sigma^{(j)}$  for all basis functions in  $\mathcal{A}^{(j)}$  we are going to work with a *shape vector* of individual  $\sigma_i^{(j)}$  for each centre  $x_i \in X^{(j)}$ . For each  $x_i$ , we choose the largest  $\sigma_i^{(j)} > 0$  such that the support of the associated basis function  $\varphi_i(x; \sigma_i^{(j)})$  contains at least a fixed number  $n_c \in \mathbb{N}$  of other centres from  $X^{(j)}$ .

Our procedure for choosing centres and regions for refinement resembles the one in [DH07].

During the adaptive refinement, the set of centres will have varying “fineness” in different regions. In order to keep track of this, we imagine the state space being partitioned into disjoint boxes with centres  $x_i \in X^{(j)}$  and radii (vectors of half edge lengths)  $h_i \in (\mathbb{R}^+)^s$ . In each iteration of the adaptive refinement of  $X^{(j)}$ , some centres  $x_i$  will be refined by replacing  $x_i$  by the vertices of the box with centre  $x_i = (x_i^1, \dots, x_i^s)$  and radius  $h_i/2$ , i.e. by the  $2^s$  points

$$\text{vert}(x_i) = \left( x_i^j \pm \frac{h_i^j}{2} \right)_{j=1, \dots, s}.$$

In Figure 4.1 an example of a refinement of a centre in the two-dimensional case is given.

The adaptive algorithm reads as follows:

**Algorithm 4.1** (Adaptive Dynamic Programming). *Fix  $\theta \in [0, 1]$  and some initial centre set  $X^{(0)} \subset \Omega$ . For  $j = 0, 1, 2, \dots$ ,*

a) *compute  $\tilde{V}^{(j)}$  and  $e^{(j)}(x)$  for all  $x \in X^{(j)}$ ,*

- b) set  $\hat{X}^{(j)} := \{\hat{x} \in X^{(j)} \mid |e^{(j)}(\hat{x})| > \theta \max_{x \in X^{(j)}} |e^{(j)}(x)|\}$ ,
- c) set  $X^{(j+1)} := (X^{(j)} \setminus \hat{X}^{(j)}) \cup \bigcup_{\hat{x} \in \hat{X}^{(j)}} \text{vert}(x)$ .

The calculation of  $\text{vert}(x)$  contains the implicit task of updating the  $h_i$  for new centres in the course of the algorithm.

We terminate the algorithm as soon as  $\sup_{x \in X^{(j)}} |e^{(j)}(x)| < TOL$  for a given tolerance  $TOL > 0$ .

## 4.2. Convergence of the algorithm

Grüne ([Grü97]) proposed an adaptive finite difference scheme for the solution of the Bellman equation and showed the convergence of the approximate solution to the exact solution under the algorithm. Due to the similarity of our algorithm, some of the results can be restated in our context.

First, there are estimates between the error and the residual of an approximate solution. The first inequality is easier to prove, but the second one is more important to use because it implies that a sequence of approximate solutions converges to the exact solution if the sequence of corresponding residuals converges to zero.

**Lemma 4.1.** (*Estimate between error and residual*) *We have*

$$\frac{1}{2} \max_{x \in \Omega} |e^{(j)}(x)| \leq \sup_{x \in \Omega} |V(x) - \tilde{V}^{(j)}(x)| \leq \frac{1}{1 - \beta} \max_{x \in \Omega} |e^{(j)}(x)|.$$

*Proof.* This works exactly as the proof for Theorem 2.2 in [Grü97]. □

These estimates can be localized in the following sense.

**Lemma 4.2.** (*Estimate between error and residual, local version*) *It also holds*

$$\frac{1}{2} e^{(j)}(x) \leq \sup_{y \in B_{M_f}(x)} |V(y) - \tilde{V}^{(j)}(y)|$$

where  $M_f := \max_{(x,u) \in \Omega \times U} \|f(x, u)\|$  and

$$\sup_{x \in K} |V(x) - \tilde{V}^{(j)}(x)| \leq \frac{1}{1 - \beta} \max_{x \in K} e^{(j)}(x)$$

if  $K \subset \Omega$  fulfils  $f(x, u^*) \in K$  and  $f(x, \tilde{u}) \in K$  for all  $x \in K$ , where  $u^*$  and  $\tilde{u}$  are the optimal control values with respect to  $V$  and  $\tilde{V}^{(j)}$ , respectively.

*Proof.* See [Grü97], Theorem 2.3. □

From now on, we make the following assumptions. Of special importance is d), which assures that the shape parameters  $\sigma_i^{(j)}$  do not deviate too much locally from each other.

#### 4. Adaptive choice of the centres

**Assumption 4.3.** For a point  $x \in \Omega$  we define  $I(x) := I(x)^{(j)} := \{i | x \in \text{supp } \varphi_i(\cdot, \sigma_i^{(j)})\}$ ,  $\bar{\sigma}^{(j)}(x) := \max_{i \in I(x)} \sigma_i^{(j)}$  and  $\underline{\sigma}^{(j)}(x) := \min_{i \in I(x)} \sigma_i^{(j)}$ .

- a) At each point  $x \in \Omega$  the supports of at most  $\hat{n}_c \in \mathbb{N}$  basis functions  $\varphi_i$  intersect, where  $\hat{n}_c$  is independent of  $j$ .
- b) There is some  $\varepsilon > 0$  such that  $\sum_{i=1}^n \varphi_i(x, \sigma_i^{(j)}) \geq \varepsilon$  for all  $x \in \Omega$  and all  $j$ .
- c) The unscaled shape function  $\varphi$  is Lipschitz continuous with constant  $L_1 > 0$ .
- d) There is a  $\hat{\sigma} > 0$  such that  $\frac{\bar{\sigma}^{(j)}(x)}{\underline{\sigma}^{(j)}(x)} \leq \hat{\sigma}$  for all  $x \in \Omega$  and all  $j$ .
- e) The map  $f$  and the cost function  $c$  satisfy the Lipschitz conditions

$$\|f(x, u) - f(y, u)\| \leq L_f \|x - y\|, \quad |c(x, u) - c(y, u)| \leq L_c \|x - y\|$$

for all  $x, y \in \Omega, u \in U$  and some constants  $L_f > 0$  and  $L_c > 0$ .

We will need the next two lemmas as prerequisite for Lemma 4.6.

**Lemma 4.4.** *The basis functions  $\psi_i^{(j)}(x) = \varphi_i(x, \sigma_i^{(j)}) / \sum_{k \in I(x)} \varphi_k(x, \sigma_k^{(j)})$  are locally Lipschitz continuous with Lipschitz constant  $2\hat{n}_c \bar{\sigma}^{(j)} L_1 / \varepsilon^2$ .*

*Proof.* We have

$$\begin{aligned} \left| \frac{d}{dx} \psi_i^{(j)}(x) \right| &= \left| \frac{\varphi_i'(x, \sigma_i^{(j)}) \sum_{k \in I(x)} \varphi_k(x, \sigma_k^{(j)}) - \varphi_i(x, \sigma_i^{(j)}) \sum_{k \in I(x)} \varphi_k'(x, \sigma_k^{(j)})}{(\sum_{k \in I(x)} \varphi_k(x, \sigma_k^{(j)}))^2} \right| \\ &\leq \frac{L_1 \sigma_i^{(j)} \hat{n}_c + 1 \sum_{k \in I(x)} L_1 \sigma_k^{(j)}}{\varepsilon^2} \\ &\leq \frac{2\hat{n}_c L_1 \bar{\sigma}^{(j)}}{\varepsilon^2}. \end{aligned}$$

□

For the next lemma, we need the Shepard operator  $S$  from (1.15).

**Lemma 4.5.** *Assume that  $f : \Omega \rightarrow \mathbb{R}$  is Lipschitz continuous with constant  $L_0 > 0$ . Then  $Sf$  is Lipschitz continuous with Lipschitz constant*

$$L = \frac{2\hat{n}_c^2 L_1 L_0 \hat{\sigma}}{\varepsilon^2}.$$

*Proof.* We have

$$\begin{aligned} \frac{d}{dx} Sf(x) &= \sum_{i \in I(x)} f(x_i) \frac{d}{dx} \psi_i^{(j)}(x) \\ &= \sum_{i \in I(x)} (f(x_i) - f(x_k)) \frac{d}{dx} \psi_i^{(j)}(x) + f(x_k) \underbrace{\sum_{i \in I(x)} \frac{d}{dx} \psi_i^{(j)}(x)}_{=0} \end{aligned}$$

## 4.2. Convergence of the algorithm

since  $\sum_{i \in I(x)} \psi_i^{(j)}(x) = \sum_i \psi_i^{(j)}(x) = 1$  for all  $x \in \Omega$ , where  $k$  is some arbitrary index from  $I(x)$ . So

$$\left| \frac{d}{dx} S f(x) \right| \leq \left( \sum_{i \in I(x)} L_0 \underbrace{\|x_i - x_k\|_2}_{\leq 1/\underline{\sigma}^{(j)}(x)} \right) \frac{2\hat{n}_c L_1 \bar{\sigma}^{(j)}(x)}{\varepsilon^2} \leq \frac{2\hat{n}_c^2 L_1 L_0 \bar{\sigma}^{(j)}(x)}{\varepsilon^2 \underline{\sigma}^{(j)}(x)} \leq \frac{2\hat{n}_c^2 L_1 L_0 \hat{\sigma}}{\varepsilon^2}.$$

□

From now on, let  $N_j$  be the number of centres in  $X^{(j)}$  and  $\hat{v}^{(j)} \in \mathbb{R}^{N_j}$  be the coefficients of  $\tilde{V}^{(j)}$  with respect to the basis  $\psi_1^{(j)}, \dots, \psi_{N_j}^{(j)}$  of  $\mathcal{W}^{(j)}$ , i.e.

$$\tilde{V}^{(j)}(x) = \sum_{i=1}^{N_j} \hat{v}_i^{(j)} \psi_i^{(j)}(x).$$

**Lemma 4.6.** (*Hölder continuity*) For any set  $X^{(j)} \subset \Omega$  of centres which satisfies Assumption 4.3, any two centres  $x_m, x_n \in X^{(j)}$  and any two points  $x, y \in \Omega$ , the inequalities

$$|\hat{v}_m^{(j)} - \hat{v}_n^{(j)}| \leq H \|x_m - x_n\|^\gamma \text{ and } |\tilde{V}^{(j)}(x) - \tilde{V}^{(j)}(y)| \leq H \|x - y\|^\gamma$$

hold for constants  $H > 0$  and  $0 < \gamma < 1$  independent of  $X^{(j)}$ .

*Proof.* The proof works along the lines of Theorem 2.9 in [Grü97]. The only step which has to be adapted is the inductive proof that the iterates of the value iteration

$$v^{(j),0} \equiv 0, \quad v^{(j),k+1} = \tilde{\Gamma} v^{(j),k}$$

are Lipschitz continuous with Lipschitz constants  $L_k > 0$  which do not depend on the set of centres  $X^{(j)}$ : Assume  $v^{(j),k-1}$  is Lipschitz continuous with Lipschitz constant  $L_{k-1}$ . Then

$$|(\Gamma v^{(j),k-1})(x_m) - (\Gamma v^{(j),k-1})(x_n)| \leq \beta L_{k-1} L_f \|x_m - x_n\| + L_c \|x_m - x_n\|$$

and so  $\Gamma v^{(j),k-1}$  is also Lipschitz continuous with a Lipschitz constant  $\hat{L}_k = \beta L_{k-1} L_f + L_c$ . By Lemma 4.5,  $v^{(j),k} = \tilde{\Gamma} v^{(j),k-1} = S(\Gamma v^{(j),k-1})$  is also Lipschitz continuous with Lipschitz constant  $L_k := L \hat{L}_k$  for the constant  $L > 0$  from Lemma 4.5 which is independent of  $k$  and  $X^{(j)}$ .

On the other hand, we have  $\|v^{(j),k} - \tilde{V}^{(j)}\| \leq \beta^k \|v^{(j),0} - \tilde{V}^{(j)}\|$  because  $\tilde{\Gamma}^{(j)}$  is a  $\beta$ -contraction.

Now, the remaining part of the proof of Theorem 2.9 in [Grü97] can be applied to see that  $\tilde{V}^{(j)}$  is Hölder continuous.

From the equation  $\hat{v}_m^{(j)} = \inf_{u \in U} [c(x_m, u) + \beta \tilde{V}^{(j)}(f(x_m, u))]$  one gets that a Hölder condition is also fulfilled for the coefficients  $\hat{v}_m^{(j)}$ . □

The residual at centre points is not necessarily zero because the Shepard approximation is not an interpolation but it converges to zero for a local fill distance shrinking to zero.

#### 4. Adaptive choice of the centres

**Lemma 4.7.** Let  $h_i^{(j)} = \max_{k \in I(x_i)} \|x_k - x_i\|_2$  for  $x_i \in X^{(j)}$ . Then

$$|e^{(j)}(x_i)| \leq H(h_i^{(j)})^\gamma$$

with the constants  $H$  and  $\gamma$  from Lemma 4.6.

*Proof.* From  $v_i^{(j)} = \inf_{u \in U} [c(x_i, u) + \beta \tilde{V}^{(j)}(f(x_i, u))]$  one gets

$$\begin{aligned} |e^{(j)}(x_i)| &= \left| \inf_{u \in U} [c(x_i, u) + \beta \tilde{V}^{(j)}(f(x_i, u))] - \tilde{V}(x_i) \right| = |\hat{v}_i^{(j)} - \tilde{V}(x_i)| \\ &= \left| \hat{v}_i^{(j)} - \sum_{k \in I(x_i)} \hat{v}_k^{(j)} \psi_k^{(j)}(x_i) \right| = \left| \sum_{k \in I(x_i)} (\hat{v}_i^{(j)} - \hat{v}_k^{(j)}) \psi_k^{(j)}(x_i) \right| \leq H(h_i^{(j)})^\gamma. \end{aligned}$$

□

The last two lemmas immediately yield

**Theorem 4.8.** For all  $x, y \in \Omega$ ,

$$|e^{(j)}(x) - e^{(j)}(y)| \leq H_e \|x - y\|^\gamma \quad \text{and} \quad |e^{(j)}(x)| \leq (H + H_e) \max\{h(x), h(x_i)\}^\gamma$$

for some constant  $H_e > 0$ ,  $h(x)^{(j)} = \max_{x_k \in I(x)} \|x_k - x\|_2$  and the constant  $\gamma$  from Lemma 4.6.

*Proof.* The first inequality follows from

$$|\tilde{V}^{(j)}(x) - \tilde{V}^{(j)}(y)| \leq H \|x - y\|^\gamma$$

in Lemma 4.6:

$$\begin{aligned} &|e^{(j)}(x) - e^{(j)}(y)| \\ &= \left| \left( \inf_{u \in U} [c(x, u) + \beta \tilde{V}^{(j)}(f(x, u))] - \tilde{V}(x) \right) - \left( \inf_{u \in U} [c(y, u) + \beta \tilde{V}^{(j)}(f(y, u))] - \tilde{V}(y) \right) \right| \\ &\leq L_c \|x - y\| + \beta H \|x - y\|^\gamma L_f + H \|x - y\|^\gamma \\ &\leq H_e \|x - y\|^\gamma \end{aligned}$$

for some constant  $H_e > 0$ . For, the second inequality we need a centre  $x_i$  close to  $x$  and get from the first inequality of this theorem and Lemma 4.7

$$\begin{aligned} |e^{(j)}(x)| &\leq |e^{(j)}(x_i)| + |e^{(j)}(x_i) - e^{(j)}(x)| \\ &\leq H(h_i^{(j)})^\gamma + H_e \|x_i - x\|^\gamma \\ &\leq (H + H_e) (\max\{h(x), h(x_i)\})^\gamma. \end{aligned}$$

□

The algorithm terminates as soon as  $\sup_{x \in X^{(j)}} |e^{(j)}(x)| < TOL$ , but only refines centres  $x$  with  $|e^{(j)}(x)| > \theta \max_{x \in X^{(j)}} |e^{(j)}(x)|$ . This implies that centres with

$|e^{(j)}(x)| \leq \theta \text{ TOL}$  are never refined and consequently also no centres for which  $(H + H_e) \max\{h(x), h(x_i)\}^\gamma < \theta \text{ TOL}$ . So the local fill distance is bounded from below and, consequently, the algorithm eventually terminates. The error is bounded by

$$\sup_{x \in \Omega} |V(x) - \tilde{V}^{(j)}(x)| \leq \frac{1}{1 - \beta} \text{ TOL}$$

according to Lemma 4.1.

## 4.3. Implementation

In comparison to Section 3.1 some additional points have to be addressed in the case of the adaptive algorithm.

We implemented a function `shapes` which calculates the shape parameters just small enough for the individual basis functions to fulfil the condition that each one leads to the support of the associated basis function just large enough to contain at least  $n_c$  other centres.

A crucial point is the “subdivision” of centres with relatively large residuals, i.e. their replacement by new centres as described in Section 4.1. To this purpose it is necessary to save the radii  $h_i \in (\mathbb{R}^+)^s$  with each centre  $x_i \in \mathbb{R}^s$ . The shape parameters  $\sigma_i \in \mathbb{R}^+$  which are also associated with the individual centres can be saved, too, but have to be recalculated with the function `shapes` after each subdivision step.

In order that Assumption 4.3 d) be fulfilled we slightly modified Algorithm 4.1 in our implementation in such a way that more centres are refined than specified in the original form in order to get more moderate transitions between fine and coarse regions of the set of centres. Theorem 4.8 and the lemmas preceding it continue to hold true in this case.

## 4.4. Numerical examples

### 4.4.1. A 1D example with non-smooth solution

We begin with Test 2 from [FF94]. Let

$$V(x) := \left| \frac{1}{e} - e^{-x^2} \right|.$$

Assume a continuous-time control problem on  $\Omega = [-2, 2]$  with  $U = [0, 1]$ ,

$$\begin{aligned} f(x, u) &= u(x - 2)(x + 2), \\ c(x, u) &= -u(x - 2)(x + 2)V'(x) - (u^2 - 2)V(x) \end{aligned}$$

#### 4. Adaptive choice of the centres

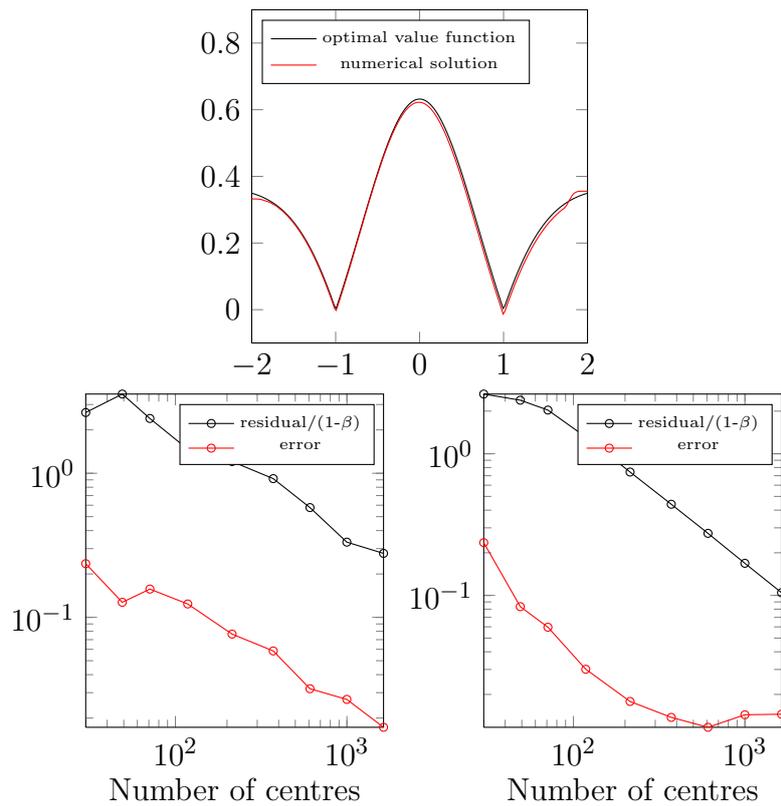


Figure 4.2.: 1D example: Comparison of exact and approximate optimal value function (above) and error and residual in dependence on the number of centres during the adaptive algorithm (below left) and for sets of equidistant centres (below right).

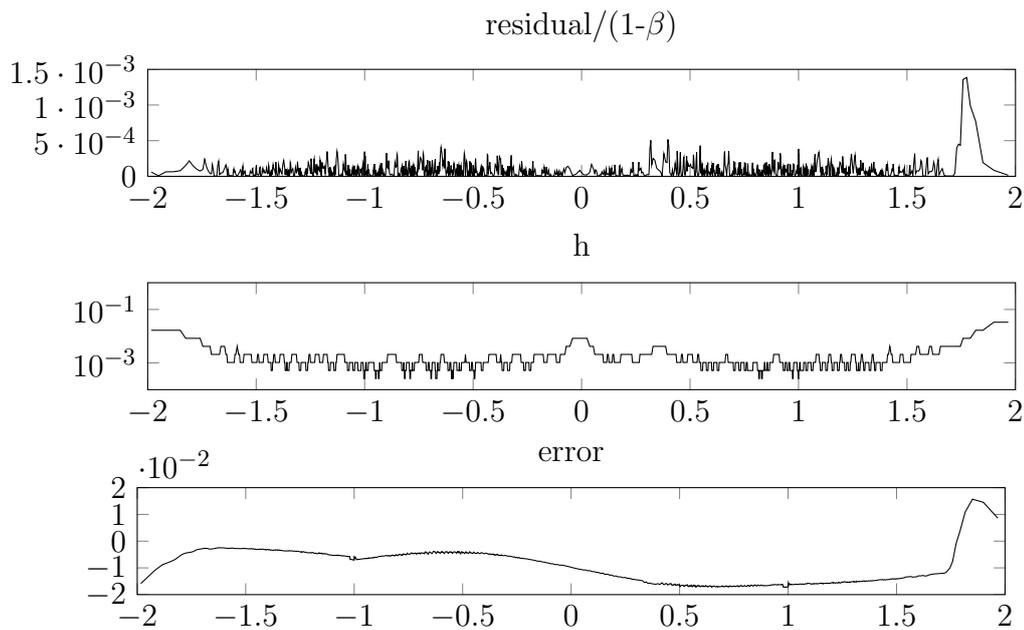


Figure 4.3.: 1D example: Distribution of local residuals, fineness and the errors in the last iteration step.

and continuous discount rate  $\lambda = 1$ . It turns out that  $V$  is the optimal value function. We use a discrete Euler step with step size  $h = 0.005$  to get a discrete-time control system. The corresponding discrete-time discount rate is  $\beta = 1 - h\lambda$ .

In Figure 4.2, above, we plot the value function, once the exact function and once a numerical approximation, which we got from the method in Chapter 2, i.e. with a prescribed, in this case equidistant, set of centres.

In Figure 4.2, below left, we show the  $L^\infty$ -error and the  $L^\infty$ -residual, rescaled by the factor  $\frac{1}{1-\beta}$  of the approximate value function  $\tilde{V}^{(j)}$  in dependence on the number of centres in each iteration step. In accordance with Lemma 4.1 the error is smaller than the rescaled residual. In the below right figure we see the corresponding plot for the non-adaptive procedure, as in Chapter 2, with comparable numbers of equidistant centres in each step. It turns out that in this example, the adaptive procedure does not give an advantage.

In Figure 4.3 one sees the local residuals, fineness and the errors in the last iteration step of the adaptive algorithm.

#### 4.4.2. A basic growth model with explicit solution

Next, we consider Example 4.1 from [GS04]. Its map

$$f(x, u) = u$$

#### 4. Adaptive choice of the centres

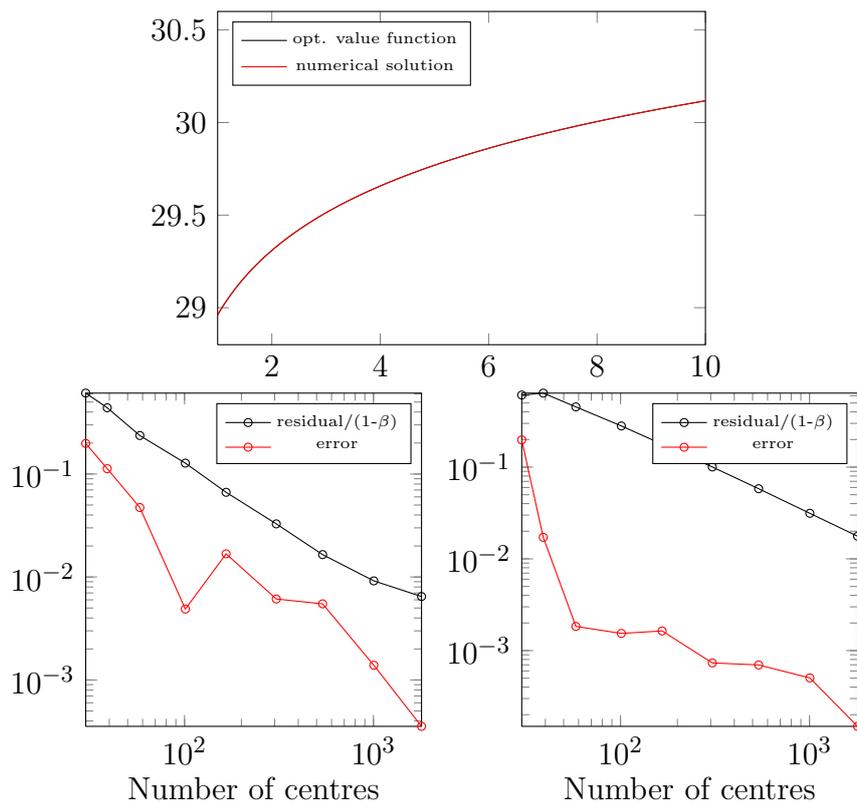


Figure 4.4.: Growth example: Comparison of exact and approximate optimal value function (above) and error and residual in dependence on the number of centres during the adaptive algorithm (below left) and for sets of equidistant centres (right).

allows the controller at each time step the choice what the next state is and its cost function is given by

$$c(x, u) = \ln(Ax^\alpha - xu(t))$$

with constants  $A > 0$  and  $0 < \alpha < 1$ . This example is different from the others in the sense that there is no equilibrium point  $x_0$  with  $f(x_0, 0) = c(x_0, 0) = 0$ . The exact solution of this problem is known and is given by

$$V(x) = B + C \ln x$$

with  $C = \frac{\alpha}{1-\alpha\beta}$  and

$$B = \frac{\ln((1-\alpha\beta)A) + \alpha\beta/(1-\alpha\beta) \ln(\alpha\beta A)}{1-\beta}.$$

In Figure 4.4, above, we plot the exact and the numerical solution for the value function. In Figure 4.4, below left, we again draw the  $L^\infty$ -error and the rescaled  $L^\infty$ -residual of the approximate value function  $\tilde{V}^{(j)}$  in dependence on the number

of centres in each iteration step and as comparison the corresponding plot for equidistant sets of centres. This time, it turns out that the adaptive algorithm manages to shrink the residual, and thus an upper bound for the error slightly faster than the equidistant algorithm. The actual error itself is slightly larger than in the non-adaptive setting, though.

### 4.4.3. An inverted pendulum on a cart

We revisit the planar inverted pendulum from Section 3.4.

We use an initial grid of  $N = 70 \cdot 70$  centres, yielding the approximate value function shown in Fig. 4.5 (a). We then perform three steps of Algorithm 4.1, yielding the approximate value functions shown in Fig. 4.5 (b)-(d). Also shown are the centre sets  $X^{(0)}, \dots, X^{(3)}$  by black dots.

As in the previous examples, we compare in Figure 4.6 the  $L^\infty$ -error and the  $L^\infty$ -residual, rescaled by the factor  $\frac{1}{1-\beta}$ , both for the adaptive algorithm, and the non-adaptive algorithm with equidistant sets of centres. This time the adaptive implementation of the algorithm gives a slight improvement for both the residual and the error.

### 4.4.4. An office robot

We finally investigate a variant of the shortest path example in Section 3.3. We consider an imaginary robot which is used in an office building (cf. Fig. 4.7). The robot is supposed to drive from the office (red) to the coffee machine in the common room (yellow) below (and back). The aim is to reach the coffee machine in shortest possible time. The speed  $v(x)$  of the robot varies in different rooms according to the colour in the image – it is 3 in the corridor (green) while it is 2 and 1 in the common room and the office, resp. Thus, the dynamics is simply given by

$$f(x, u) = x + v(x)hu,$$

where we choose the time step  $h = 0.1$ , with  $U := \{u \in \mathbb{R}^2 : \|u\| = 1\}$ , and the associated cost function by

$$c(x, u) \equiv \begin{cases} 0 & \text{at the target set (coffee machine),} \\ 1 & \text{otherwise.} \end{cases}$$

In order to solve this problem by our approach, we choose as initialization the set of centres  $X$  as those nodes of an equidistant grid which are in one of the rooms (or the corridor) outside of the obstacles. The position of the coffee machine in our case is approximately at the origin.

In Figure (4.7), right, we show some isolines of the approximate optimal value function. Again, we compare in Figure 4.8 the  $L^\infty$ -error and the rescaled  $L^\infty$ -residual for the adaptive and non-adaptive algorithm with equidistant sets of

#### 4. Adaptive choice of the centres

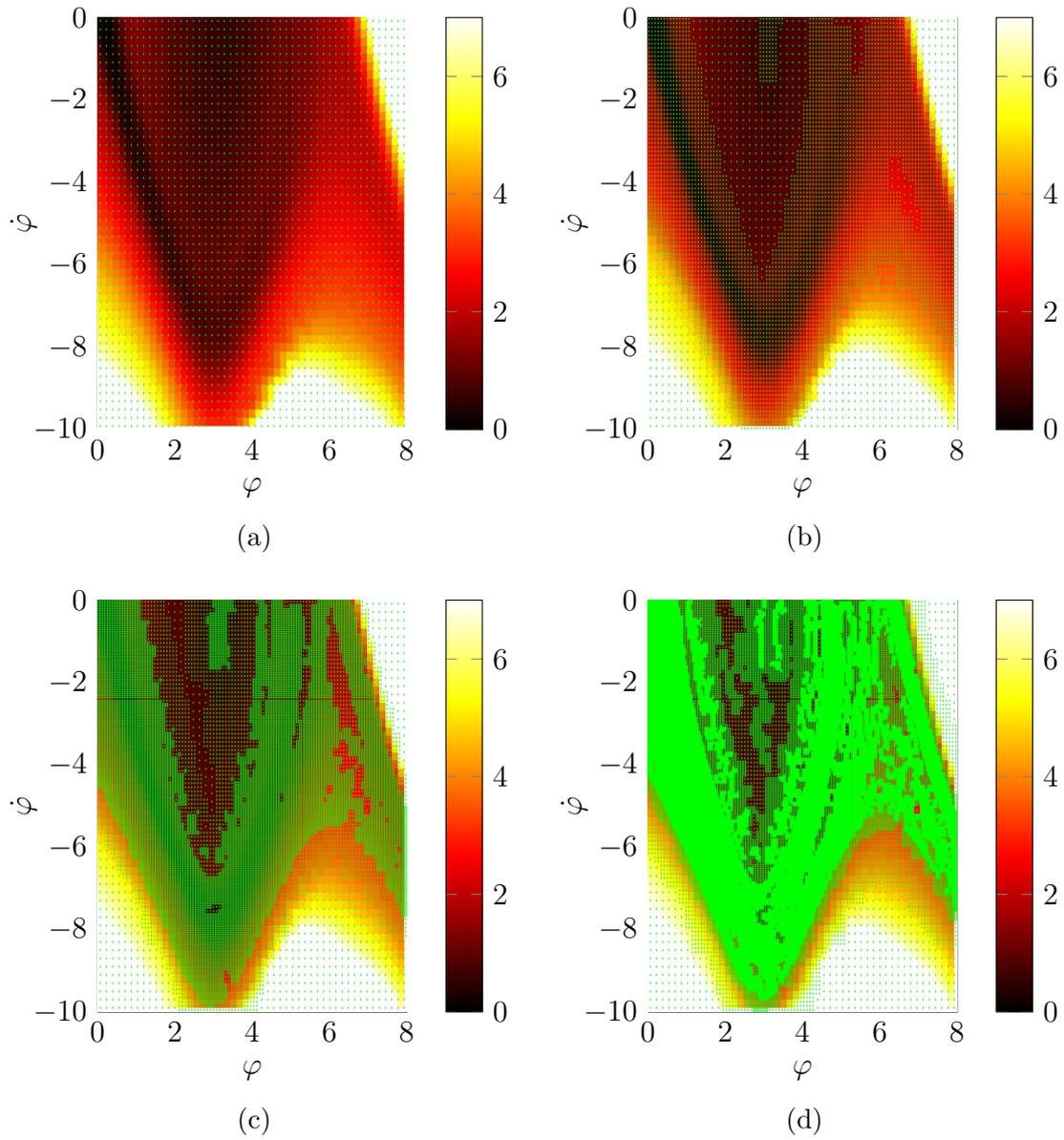


Figure 4.5.: Inverted pendulum: Approximate optimal value function and refined centres (green points) in four steps of the algorithm.

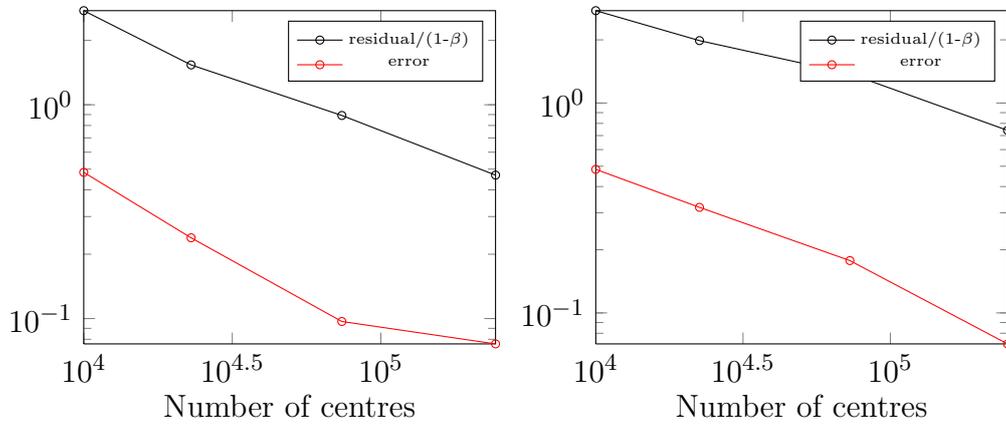


Figure 4.6.: Inverted pendulum: Error and residual in dependence on the number of centres during the adaptive algorithm (left) and for sets of equidistant centres (right).

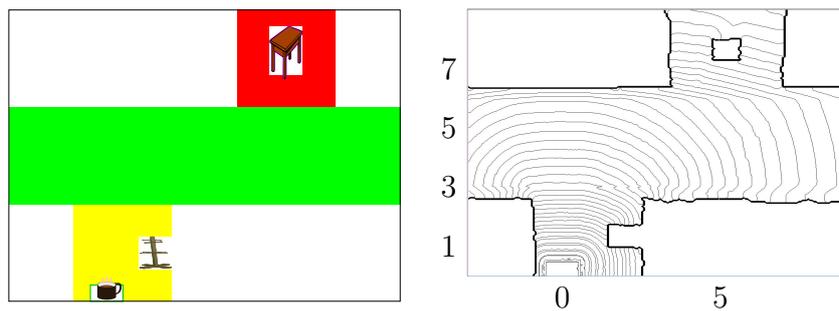


Figure 4.7.: Office robot: Left: Model of the office robot problem. Right: Isolines of the approximate optimal value function.

#### 4. Adaptive choice of the centres

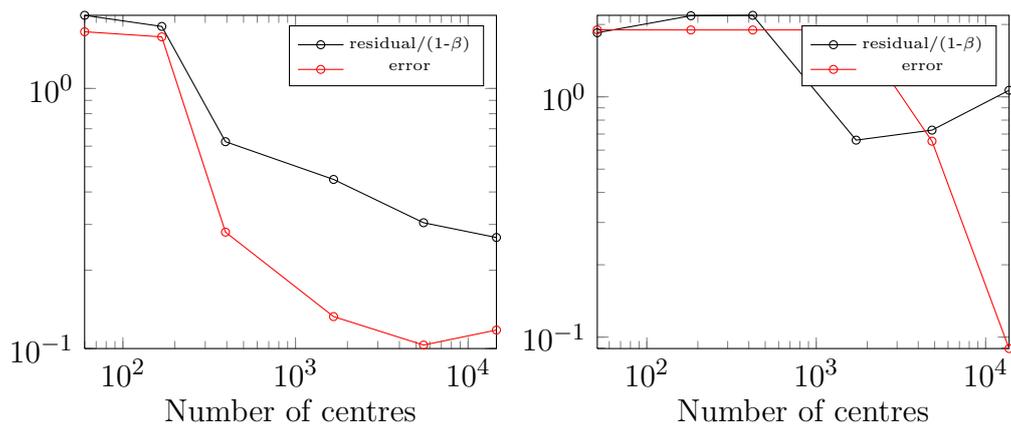


Figure 4.8.: Office robot: Error and residual in dependence on the number of centres during the adaptive algorithm (left) and for sets of equidistant centres (right).

centres. The adaptive procedure leads to a somewhat smaller residual and error for corresponding numbers of centres. In the non-adaptive case the scaled residual sometimes appears to be smaller than the error. This is no contradiction to Lemma 4.1 because the actual  $L^\infty$ -norm of the residual is larger than the estimate by calculating the maximum of it on finitely many centres.

## 5. Shepard value iteration and stochastic shortest path problems

The value iteration introduced in Chapter 2 provides an efficient method to approximate the optimal value function of a control problem. For the value iteration, however, it is not only necessary to calculate the dynamics and the costs for all centre-control pairs, but also to store this information and revisit it for each iteration. One could increase the efficiency of the method if one could achieve that the information about the respective dynamics and costs is only used locally in each step, in a region where the value function is updated.

To this end, we note that there is a connection between the Shepard discretization of the optimality principle for undiscounted control systems, and *stochastic shortest path problems*, which Bertsekas considers in [Ber07, Chapter 2]. The Shepard discretization relies on a finite subset, the set of centres, of the state space whereas stochastic shortest path problems are defined on a finite graph. It turns out, however, that the Shepard discretization gives rise to the construction of a graph, whose vertices correspond to the RBF centres, such that the explicit formula (2.7) for the fixed point of the Bellman-Shepard operator  $\tilde{\Gamma} = S \circ \Gamma$  from (2.1) is equivalent to the equation one gets from the dynamic programming principle for a related stochastic shortest path problem.

From this observation stems the appealing idea to approximate the optimal value function under Shepard discretization not by value iteration as we have done in Chapters 2-4 but by solving the corresponding stochastic shortest path problem instead. In some cases, this can be done by a Dijkstra-like algorithm from [Ber07]. Bertsekas assumes a so-called consistently improving policy as a sufficient condition to apply the algorithm. In this case one gets the same solution as with Shepard value iteration, but with less numerical effort. The existence of a consistently improving policy is a rather strong condition, though. Moreover, it cannot be checked a priori.

For that reason we do not use this assumption, but note that without it the application of the algorithm might introduce an additional error on the approximate solution, besides the approximation error from the spatial discretization with Shepard's method and possibly from the discretization in time of a continuous-time control system.

We give an introduction to shortest path problems in Section 5.1, to stochastic shortest path problems in Section 5.2, present the Dijkstra-like algorithm and its properties in Section 5.3 and conclude the chapter with some notes on the

application on continuous-time control systems.

## 5.1. Shortest path problems and Dijkstra's algorithm

In this section we give a short introduction to shortest path problems in a directed graph with a target and Dijkstra's algorithm ([Dij59]) to solve them efficiently. Note that this introduction only refers to deterministic problems and gives prerequisites for the remainder of this chapter in which we consider stochastic shortest path problems and a stochastic version of Dijkstra's algorithm.

Let  $G = (X, E)$  be a directed graph with a positive length  $c(i, j) > 0$  assigned to each edges  $(i, j)$ . We define  $c(i, j) = \infty$  if  $(i, j)$  is no edge. Furthermore, we assume a special vertex  $0 \in X$ , the *target*. We denote the set of vertices with  $X$ —not with  $V$  which is more usual in the literature. The reason is that it corresponds to the set of centres we use in this thesis.

The aim is to find paths of minimal length from any vertex to the target where the length of a path is the sum of the lengths of all the edges along the path. The distance between a vertex and the target is defined as the minimal length of such a path. So “distance” is used here in an asymmetric way: Only paths to the target and not from the target are considered. By definition, it is infinite for a vertex, if there is no path from the vertex to the target.

**Algorithm 5.1** (Dijkstra's algorithm, [Dij59]). *We use a subset of vertices  $P \subset X$ , a function  $V : X \rightarrow [0, \infty)$  and a map  $\mathbf{u} : X \rightarrow X$  which all change in the course of the algorithm.*

*$P$  is the set of accepted vertices at each stage,  $V(i)$  assigns to each vertex  $i$  the (current) approximate distance from the target and  $\mathbf{u}(i)$  the next vertex of an (current) approximate optimal path from  $i$  to the target.*

*As initialization, one sets  $P := \{0\}$  and*

$$V(i) := \begin{cases} 0 & \text{if } i = 0, \\ \infty & \text{if } i \neq 0, \end{cases}$$

*whereas  $\mathbf{u}(\cdot)$  can be initialized arbitrarily.*

*In each iteration,*

*a) the algorithm terminates as soon as  $\min_{i \in X \setminus P} V(i) = \infty$ , i.e.*

- $P = X$  or
- $V(i) = \infty$  for all  $i \in X \setminus P$ .

*b) the vertex  $i \in X \setminus P$  with minimal  $V(i)$  is added to  $P$ .*

c) one updates

$$\mathbf{u}(i) := \begin{cases} \mathbf{u}(i) & \text{if } V(i) \leq \min_{j \in P}(c(i, j) + V(j)), \\ \operatorname{argmin}_{j \in P}(c(i, j) + V(j)) & \text{otherwise,} \end{cases}$$

and

$$V(i) := \min \left( V(i), \min_{j \in P}(c(i, j) + V(j)) \right).$$

Usually, the algorithm is given in a form where the optimal successor vertex  $\mathbf{u}(i)$  for each vertex is not saved, but only the function  $V$  is constructed. In this case the map  $u$  can be reconstructed from  $V$  by  $\mathbf{u}(i) := \operatorname{argmin}_{j \in X}[c(i, j) + V(j)]$ .

**Theorem 5.1** ([Dij59]). *The maps  $V : X \rightarrow [0, \infty)$  and  $\mathbf{u} : X \rightarrow X$  resulting from the algorithm fulfil the following.  $X(i)$  equals the distance of  $i$  to the target 0.*

We remark that a path of minimal length from a vertex  $i \in X$  to the target 0 is given by  $i, \mathbf{u}(i), \mathbf{u}(\mathbf{u}(i)), \mathbf{u}(\mathbf{u}(\mathbf{u}(i))), \dots$

## 5.2. Stochastic shortest path problems

Before we turn our focus to stochastic shortest path problems, we notice that a deterministic shortest path problem can be regarded as an optimal control problem. For this, we assume a directed graph with labelled edges where we identify the vertex set both with the state space and the control space. The control system is then defined by  $f(i, j) = j$  and

$$c(i, j) = \begin{cases} \text{the label of the edge } ij & \text{if this edge exists,} \\ \infty & \text{otherwise,} \end{cases}$$

for all  $i, j \in X$ .

Throughout this chapter, we assume that as long as a path is outside the target, some positive cost is being added to the total cost:  $c(i, u) > 0$  for any  $i > 0$  and  $u \in U$ .

This control problem can be interpreted in the following way: An imaginary player is given a vertex to start with; he can choose at each time step an edge which begins at the current vertex. The vertex in which this edge ends defines the next state for the player.

In a stochastic shortest path problem the player cannot choose the following vertex, but instead, he chooses at each step a control  $u \in U$  from a finite control space. The vertex  $i$  and the control  $u$  together impose a one-step cost  $c(i, u)$  and a probability distribution over all possible successor vertices  $j$ , where  $p_{ij}(u)$  is the probability of moving from state  $i$  to state  $j$ . The target 0 has the properties that it is stable and does not impose any cost, i.e.  $p_{00}(u) = 1$  and  $c(0, u) = 0$  for any  $u \in U$ .

## Feedbacks

A feedback—called a stationary (time-independent) policy in [Ber07], but we follow here and in other cases our previous terminology—, namely the choice of a control  $u_i \in U$  for each state  $i$ , induces an expected value of the total cost

$$\sum_{k=0}^{\infty} c(i_k, u_k) = \sum_{k=0}^{K-1} c(i_k, u_k), \quad K = \min\{k \geq 0 \mid i_k = 0\} \quad (5.1)$$

for each initial state  $i = i_0$  to get to the target 0.

The aim is to minimize this total cost for each starting vertex, usually by the choice of a feedback.

Note that the concept of a control sequence  $(u_k)_k$  as in (1.1) does not make much sense in the stochastic setting because a reasonable player would not choose all the controls of a path from the start, but choose them in dependence on the actual vertex at each time. In theory, it would be possible to use a strategy other than a feedback, but we do not consider them here because it would not give any advantage to the player if he used different controls for the same vertex at different times.

For the choice of a feedback, there is an associated directed graph, possibly with loops and double-edges, in the following way. The vertices of the graph are identified with the states  $0, 1, 2, \dots, n$  as before, and there is an edge from  $i$  to  $j$  exactly if  $p_{ij}(u_i) > 0$ .

## Optimal feedbacks

By definition, there is an optimal feedback in dependence on the initial vertex  $i \in X$ : It is a feedback for which the expected total cost (5.1) is minimal. It should not come as a surprise that the dependence on the initial vertex is not necessary.

**Proposition 5.2** ([Ber07]). *For a stochastic shortest path problem there is an optimal feedback.*

This can be proved by a stochastic version of the Bellman equation for stochastic shortest path problems, which reads

$$V(i) = \min_{u \in U} [c(i, u) + \sum_{j=0}^n p_{ij}(u) V(j)], \quad i = 0, \dots, n. \quad (5.2)$$

It turns out that this formula is essentially the same as the explicit formula for the Shepard-Bellman equation (2.7) if one sets  $i = x$ ,  $j = x_l$  and  $p_{ij}(u) := \psi_j(f(x, u))$ , as we assume here that  $\beta = 1$  and that  $U$  is finite. Thus the Shepard-Bellman operator can be interpreted as the Bellman operator for a stochastic

shortest path problem and one can try to approximate the solution of optimal control problems with the method in this chapter.

The method relies on ideas in [Ber07] where it is proposed to use a stochastic version of Dijkstra's algorithm from Section 5.1.

## Acyclic and monotone feedbacks

In the following definition, we call a feedback  $\mathbf{u}$  *acyclic* if any path which can be induced by it with positive probability, i.e. every sequence of vertices  $i_0, i_1, \dots$  with  $p_{i_k, i_{k+1}}(\mathbf{u}(i_k)) > 0$  for all  $k \geq 0$ , can visit every vertex only once, with the exception of the target vertex.

**Definition 5.3** (acyclic feedback). A feedback is *acyclic* if the associated directed graph as defined after (5.1) is acyclic.

By relaxing this condition to allow immediate repetitions of vertices, one gets the following definition.

**Definition 5.4** (weakly acyclic feedback). A feedback is *weakly acyclic* if the associated graph is acyclic, with the exception that loops, i.e. edges  $(i, i)$  of vertices  $i$ , are allowed, as long as the corresponding probability  $p_{ii}(\mathbf{u}(i))$  of staying on the vertex is less than one for  $i \neq 0$ .

In general, acyclic feedbacks do not exist for a given stochastic shortest path problem. If not, one can still try to find an acyclic feedback on a maximal subset of the vertex set.

These definitions are independent of the cost function, unlike the following stronger conditions.

**Definition 5.5** (strictly monotone feedback). A feedback is *strictly monotone* if its induced expected value  $V(i)$  of the total costs as in (5.1) has the property that it drops with each transition with probability one: If  $p_{ij}(\mathbf{u}(i)) > 0$ , then  $V(i) > V(j)$ .

A strictly monotone feedback which is also an optimal feedback is called a *consistently improving* feedback in [Ber07]. Again, we use a relaxed form.

**Definition 5.6** (weakly monotone feedback). A feedback is *weakly monotone* if for  $V$  as before it holds: If  $p_{ij}(\mathbf{u}(i)) > 0$ , then  $V(i) > V(j)$  or  $i = j$ . For each  $i \in X$ ,  $p_{ii}(\mathbf{u}(i)) < 1$ .

**Remark.** A strictly monotone feedback is acyclic. A weakly monotone feedback is weakly acyclic.

### 5.3. A Dijkstra-like algorithm for stochastic shortest path problems

We use the following modification of Bertsekas' ([Ber07]) algorithm for stochastic shortest path problems, which itself is a generalization of Algorithm 5.1 (Dijkstra's Algorithm).

**Algorithm 5.2.** *We use a subset of vertices  $P \subset X$ , a function  $V : X \rightarrow [0, \infty)$  and a map  $\mathbf{u} : X \rightarrow U$ , which all change in the course of the algorithm.*

*$P$  is the set of accepted vertices at each stage,  $V(i)$  assigns to each vertex  $i$  the (current) approximate minimal expected distance from the target and  $\mathbf{u}(i)$  a (current) approximate optimal control for  $i$ .*

*As initialization, one sets  $P := \{\}$  and*

$$V(i) := \begin{cases} 0 & \text{if } i = 0, \\ \infty & \text{if } i \neq 0, \end{cases}$$

*whereas  $\mathbf{u}(\cdot)$  can be initialized arbitrarily.*

*In each iteration,*

*a) the algorithm terminates as soon as  $\min_{i \in X \setminus P} V(i) = \infty$ , i.e.*

- $P = X$  or
- $V(i) = \infty$  for all  $i \in X \setminus P$ .

*b) the vertex  $j \in X \setminus P$  with minimal  $V(j)$  is added to  $P$ .*

*c) for all  $i \in X \setminus P$  one defines*

$$U(i) := \{u \in U \mid p_{ij}(u) > 0 \text{ and } p_{ik}(u) = 0 \text{ for all } k \in X \setminus (P \cup \{i\})\}$$

*and updates*

$$\mathbf{u}(i) := \begin{cases} \mathbf{u}(i) & \text{if } V(i) \leq \min_{u \in U(i)} \frac{c(i,u) + \sum_{j \in P} p_{ij}(u)V(j)}{1 - p_{ii}(u)}, \\ \operatorname{argmin}_{u \in U(i)} \frac{c(i,u) + \sum_{j \in P} p_{ij}(u)V(j)}{1 - p_{ii}(u)} & \text{otherwise} \end{cases}$$

*and*

$$V(i) := \min \left( V(i), \min_{u \in U(i)} \frac{c(i,u) + \sum_{j \in P} p_{ij}(u)V(j)}{1 - p_{ii}(u)} \right).$$

Note that for the vertex  $i$  considered in part c) of the algorithm and the vertex  $j$  just added to  $P$  in the same step of the algorithm, we always have  $j \neq i$  and thus from  $p_{ij}(u) > 0$  it follows  $p_{ii}(u) = 1 - \sum_{k \neq i} p_{ik}(u) < 1$  and  $1 - p_{ii}(u) \neq 0$ , so all the fractions are well-defined.

In comparison to [Ber07] we have changed the algorithm in some respects.

### 5.3. A Dijkstra-like algorithm for stochastic shortest path problems

- Bertsekas states as an explicit requirement for his algorithm that an optimal feedback exists which is strictly monotone. This is actually a sufficient condition for the optimal value function to be found by the algorithm.

As our application of the algorithm is for value functions under the Shepard discretization, which already introduces an approximation error, we do not require the algorithm to solve the shortest path problem exactly. So we skip the assumption Bertsekas poses that there be a strictly monotone optimal feedback and instead try to solve the problem exactly on a preferably large subset of the state space. In any case the algorithm gives an upper estimate to the value function

Actually, it is already true for a subset of the state space that for the optimal value function to be found by the algorithm it is necessary for the value function to be induced by an acyclic feedback and sufficient to be induced by a strictly monotone feedback. While in general the obtained value function might not be optimal, the constructed subset  $P \subset X$  is exactly the set where a stabilizing feedback exists, i.e. a feedback which moves any initial state of  $P$  with probability 1 to the target without any repetition of states (with the exception described in the next paragraph).

- We relax the algorithm slightly to adapt it to feedbacks which are only weakly monotone resp. weakly acyclic instead of strictly monotone resp. acyclic. If the graph associated to the feedback is considered, the requirement of a weakly acyclic feedback would amount to allowing feedbacks which induce directed graphs that are acyclic with the exception that loops, i.e. edges with the same start and end vertex are allowed. The same exception applies for weakly monotone feedbacks.

Although this might seem only a minor generalization, it leads to the advantage that the relaxed version of the algorithm is then applicable in more cases, especially for more time-discretizations of continuous-time control systems, as considered in the next section.

- Bertsekas uses an additional set  $L$  for vertices in  $X \setminus P$  with (already) finite  $V$  which is, in principle, not necessary.
- The feedback is not saved in the original version, so they only construct the function  $V$ , not  $\mathbf{u}$ . If the constructed value function is optimal, the feedback can be reconstructed with the Bellman equation by defining

$$\mathbf{u}(i) := \operatorname{argmin}_{u \in U} [c(i, u) + \sum_{j \in X} p_{ij}(u) V(j)].$$

For the general case, where the constructed  $V_d$  might not be optimal, we prefer saving the feedback as well.

## 5. Shepard value iteration and stochastic shortest path problems

The next theorem gives an overview in the general situation where we do not pose additional assumptions but define subsets of the set of centres and show a chain of inclusions between them. It can be considered a generalization of Bertsekas claim that if a strictly monotone optimal feedback exists, the optimal value function is found by the algorithm. With the notation of the theorem, this claim could be stated as follows (with the difference of weak instead of strict monotonicity): If  $X_m = X$ , then  $X = X_d = X_a = X_m$ .

**Theorem 5.7.** *Assume a stochastic shortest path problem. Let  $V$  be the optimal value function,  $u^*$  an optimal feedback, and  $V_d$  resp.  $\mathbf{u}$  the approximate value function resp. the feedback constructed by Algorithm 5.2.*

*Let  $X_a$  resp.  $X_m$  be the set of vertices  $i \in X$  for which an optimal weakly acyclic resp. optimal weakly monotone feedback exists.*

*Let  $X_d$  be the set of vertices  $i \in X$  where the algorithm gives finite and optimal values, i.e.  $V_d(i) = V(i) < \infty$ .*

*Then it follows that*

- a)  $V_d$  is the cost function induced from the feedback  $\mathbf{u}$ .
- b)  $X_m \subset X_d$ .
- c)  $\mathbf{u}$  is weakly acyclic on the subset  $P \subset X$  where  $V_d$  is finite. Furthermore, we have that  $X_d \subset X_a$ .
- d)  $P$  is maximal in the following sense:  $i \in P$  if and only if there is a feedback which steers  $i$  to 0 with probability one without revisiting a vertex which has already been left (so only immediate loops are allowed). Furthermore, we have that  $X_a \subset P$ .

The theorem shows (amongst others) that there is the chain of inclusions

$$X_m \subset X_d \subset X_a \subset P \subset X.$$

*Proof.* a) We have to show that  $V_d$  is the cost function induced from the feedback  $\mathbf{u}$ , specifically that

$$V_d(i) = [c(i, \mathbf{u}(i)) + \sum_{j \in X} p_{ij}(\mathbf{u}(i))V_d(j)] \text{ for all } i \in P. \quad (5.3)$$

Actually, this holds not only at the end, but also at each stage of the algorithm (if the updates of  $V$  and  $\mathbf{u}$  in step c) of the algorithm are applied at once). For distinction, we use the notation  $V_t$  and  $\mathbf{u}_t$  for the temporary maps during the algorithm, while  $V_d$  and  $\mathbf{u}$  still refer to the maps at the end of the algorithm.

Whenever the first case for an  $i \in X \setminus P$  in step c) of the algorithm occurs, namely

$$V_t(i) \leq \min_{u \in U(i)} \frac{c(i, u) + \sum_{j \in P} p_{ij}(u)V_t(j)}{1 - p_{ii}(u)},$$

### 5.3. A Dijkstra-like algorithm for stochastic shortest path problems

no update of  $\mathbf{u}_t(i)$  and  $V_t(i)$  takes place, so all instances of (5.3) remain valid. Otherwise,  $\mathbf{u}_t(i)$  and  $V_t(i)$  are updated such that

$$V_t(i) = \frac{c(i, \mathbf{u}_t(i)) + \sum_{j \in P} p_{ij}(\mathbf{u}_t(i)) V_t(j)}{1 - p_{ii}(u)},$$

so

$$\begin{aligned} V_t(i) &= p_{ii}(u) V_t(i) + c(i, \mathbf{u}_t(i)) + \sum_{j \in P} p_{ij}(\mathbf{u}_t(i)) V_t(j) \\ &= c(i, \mathbf{u}_t(i)) + \sum_{j \in P \cup \{i\}} p_{ij}(\mathbf{u}_t(i)) V_t(j) \\ &= c(i, \mathbf{u}_t(i)) + \sum_{j \in X} p_{ij}(\mathbf{u}_t(i)) V_t(j). \end{aligned}$$

This shows that (5.3) remains valid for  $i$ .

However, one has to check that from the change of  $V_t(i)$  and  $\mathbf{u}_t(i)$  all the other instances of equation (5.3) remain valid.  $\mathbf{u}_t(i)$  does not occur in any of them, so only the change of  $V_t(i)$  could be relevant. Actually there are no other instances which are affected by the update of  $V_t(i)$  because from  $i \in X \setminus P$  it follows that  $i$  has not been in  $P$  up to this phase of the algorithm, and consequently for all  $i' \in X$  the choice of  $U(i')$  and hence  $\mathbf{u}_t(i')$  was such that  $p_{i'i} = 0$  so  $V_t(i)$  does not appear on the right-hand side of (5.3) for  $j$ .

This shows the claim that the feedback and the value function constructed by the algorithm fit together as in (5.3).

- b) Now for the inclusion  $X_m \subset X_d$ . Assume an optimal weakly monotone feedback is given on  $X_m$  and the corresponding value function  $V$  with vertices ordered accordingly:

$$0 = V(0) \leq V(1) \leq \dots \leq V(K-1)$$

where  $K$  is the number of elements of  $X_m$ . We show by induction that the algorithm adds in step  $i+2$  a vertex (usually  $i+1$ ) with value  $V(i+1)$  for  $i = -1, 0, 1, \dots$

Assume that the states  $0, 1, \dots, i$  have been added to  $P$  and that  $V(\cdot) = V_d(\cdot)$  for these states. Then from the vertex  $i+1$  with the control  $u^*(i+1)$  only vertices in  $\{0, \dots, i+1\}$  are reached because by definition the feedback  $u^*$  is assumed to be weakly monotone. So  $u^*(i+1)$  is a possible choice for state  $i+1$  in step  $i+1$ , i.e.  $u^*(i+1) \in U(i+1)$ .

On the other hand, one has  $V(k) \leq V_d(k)$  for all  $k$  because  $V$  is the optimal value function, and consequently

$$V(i) = \frac{c(i, u^*(i)) + \sum_{j \in P} p_{ij}(u^*(i)) V(j)}{1 - p_{ii}(u^*(i))} \leq \frac{c(i, u) + \sum_{j \in P} p_{ij}(u) V(j)}{1 - p_{ii}(u)}$$

## 5. Shepard value iteration and stochastic shortest path problems

for any  $u \in U$ .

So for the vertex  $i + 1$  one will have  $V_d(i + 1) = V(i + 1)$  and  $u^*(i + 1)$  or an equivalent choice (such that the right-hand side of (5.3) takes the same value). In part a) of the next step ( $i + 3$ ) the vertex  $i + 1$  can be added to  $P$  (or another vertex  $k$  and  $u^*(k)$  with  $U(i + 1) = U(k)$  for a  $k > i + 1$ ; but then the order of the vertices could be changed; so, without loss of generality,  $i + 1$  has actually been chosen.). This concludes the inductive step and the proof of the inclusion  $X_m \subset X_d$ .

- c) The algorithm constructs a weakly acyclic feedback (on all of  $P$ ):  $\mathbf{u}(i)$  is updated always in such a way that  $p_{ij} > 0$  only if  $j$  is already in  $P$  or  $i = j$ . So the order in which the vertices are added to  $P$  is an order which is acknowledged by all pairs  $(i, j)$  with  $p_{ij}(\mathbf{u}(i)) > 0$ .  $j$  has to be added to  $P$  before  $i$  or  $i = j$  in order that  $p_{ij}(\mathbf{u}(i)) > 0$  is possible. The feedback from the algorithm is thus weakly acyclic and by definition optimal on  $X_d$ . On the other hand,  $X_a$  was defined as the subset of states where a weakly acyclic optimal feedback exists. So  $X_d \subset X_a$ .
- d) The inclusion  $X_a \subset P$  will be clear if we show that  $P$  is the set of all vertices which have finite induces total costs for a weakly acyclic feedback. To that end, we are going to show that for any vertex  $i$  the algorithm finds a finite value  $V_d(i)$  for  $i$  exactly if there is a weakly acyclic feedback which controls  $i$ . The direction “ $\Rightarrow$ ” is true because we have already shown in the last paragraph that the algorithm constructs an acyclic feedback on  $P$ .

Now for the direction “ $\Leftarrow$ ”:

If the algorithm terminates with  $V_d(i) = \infty$  for a state  $i$  then there is no way to control  $i$  with a weakly acyclic feedback, because with any control  $\mathbf{u}(i)$  it holds either that  $p_{ii}(\mathbf{u}(i)) = 1$  or that there is a transition from  $i$  to a  $j \neq i$ ,  $p_{ij}(\mathbf{u}(i)) > 0$  with  $V_d(j) = \infty$ , otherwise  $i$  would have been accepted with  $\mathbf{u}(i)$  or a better control at some stage of the algorithm and  $V_d(i)$  would be finite. In the first case, it is obvious that with this  $\mathbf{u}(i)$  the vertex  $i$  cannot be controlled towards the target, while in the latter case, with any control  $\mathbf{u}(j)$  there is likewise a transition  $j \rightarrow k$  with  $V_d(k) = \infty$  if not  $p_{jj}(\mathbf{u}(j)) = 1$  and so on. So there remains a path of positive probability which never leaves the area where  $V_d(\cdot) = \infty$ , contradicting the assumption that there were a acyclic feedback to stabilize  $i$ , concluding the proof. □

**Remark.** We get from Algorithm 5.2 a weakly acyclic feedback and a value function  $V_d$  which is optimal in the following sense: If the vertices are numbered according to the steps in which they are added to  $P$ , a feedback which is weakly acyclic and respects this ordering of vertices, induces a value function  $V$  which fulfils  $V_d \leq V$ .

To check for optimality of the approximate value function, one can apply value iteration once and see where the value function changes. If it changes at a vertex  $j \in P$ , it has not been optimal there and neither at other vertices  $i \in P$  from which one can move to  $j$ , i.e.  $p_{ij}(\mathbf{u}(i))$ .

By repeating this method recursively, one can construct the set of vertices where the value function has not been optimal.

## 5.4. Application on continuous-time control systems

As explained in Section 5.2, a discrete-time optimal control problem can be treated with the methods of this chapter because the Bellman-Shepard discretization (2.7) leads to the same equation 5.2 as an equivalent stochastic shortest path problem.

In this section we consider optimal control problems which are originally defined in continuous time, so that one must also choose the discretization in time. For a fixed time discretization, e.g. the explicit Euler method, and a chosen method to construct centre sets of prescribed fill distance for the Shepard discretization, these discretizations depend only on two parameters,  $h > 0$  and  $\tau > 0$  for space and time discretization, respectively.

The question is if one can choose  $h$  and  $\tau$  such that the resulting stochastic shortest path problem has large subsets  $X_m$  resp.  $X_d$  of  $X$  if those are defined as in Theorem 5.7. This is more likely for small  $h$  and large  $\tau$  because, in general, a large  $\tau$  implies relatively large jumps  $\|f(x, u) - x\|$ , and with a small parameter  $h$  it might be possible to avoid cycles in the associated graph.

### Event-based time discretization and variants

Algorithm 5.2 often does not allow for constructing the optimal value function on a large subset of the state space because the underlying graph of an optimal feedback is not acyclic, usually because the system dynamics are “too slow”.

A way to handle the different lengths of the continuous-time dynamics  $f(x, u)$  is to use  $U \times \mathbb{R}_\tau$  for a finite  $\mathbb{R}_\tau \subset \mathbb{R}^+$  as an extended control space, in order to explicitly allow different time steps for the time discretization. However, here the numerical effort increases significantly because the control space gets larger. For this reason we have not investigated this idea any further.

A similar approach is to use varying (sampled) time steps for the time discretization, but with an a priori choice instead of an extension of the control space. These time steps can be specially designed to force relatively large sets  $X_m$  resp.  $X_d$  as in Theorem 5.7. A possible way to do that is motivated by so-called event-based methods ([AB02, GJJ<sup>+</sup>10]). The name event-based derives from the fact that in a mesh-based space discretization of control problems the transition of the state between different cells is considered to be an event. Usually, only such events trigger new calculations of controls.

## 5. Shepard value iteration and stochastic shortest path problems

For radial basis or other meshfree methods this approach is not well-suited because these events require a cell decomposition of the state space. A modified, more empirical possibility which does not require regular updates is to use different time steps which force “space steps” of similar lengths. The way to adjust the time steps is to rescale in the following way:

**Definition 5.8.** For continuous-time maps  $f : \Omega \times U \rightarrow \mathbb{R}^s$  and  $c : \Omega \times U \rightarrow [0, \infty)$ , we define their rescalings

$$\tilde{f}(x, u) := \frac{f(x, u)}{\|f(x, u)\|} \quad \text{and} \quad \tilde{c}(x, u) := \frac{c(x, u)}{\|f(x, u)\|}.$$

These modified continuous-time maps  $f$  and  $c$  are then discretized with, e.g. the explicit Euler method.

So the dynamics are discretized such that one gets jumps of constant length instead of jumps which correspond to equal time steps. In this way, the control problem becomes more like a shortest-path problem, in the sense that it can be modelled on an Euclidean graph.

### Comparison of the Dijkstra-like algorithm with Shepard value iteration

The use of Algorithm 5.2 introduces in general an additional error on the approximate value function one gets from the Shepard discretization. As a compensation, it may bring a speed-up compared to value iteration because each “edge” is considered just once for this algorithm, but several times (the number of iterations) for value iteration.

However, one has to consider the following aspects:

- Usually the number of iterations for value iteration is not large; numerical experiments suggest: It does not depend much on the fill distance, for smaller fill distance it seems to be a bit smaller. It seems to be approximately inversely proportional to the time step, though. This is plausible because for smaller time steps a larger number of steps is needed to move a far-away point to the equilibrium.
- For higher-dimensional systems, the role of the number of iterations is, in general, even less important because the numerical effort for value iteration depends for higher-dimensional systems more on the state space than on the number of iterations.
- The matrix-vector-multiplication which is used for value iteration is fast because it is parallelized (in contrast to the rather sequential Dijkstra-like algorithm); its time consumption does not depend much on the number of centres.

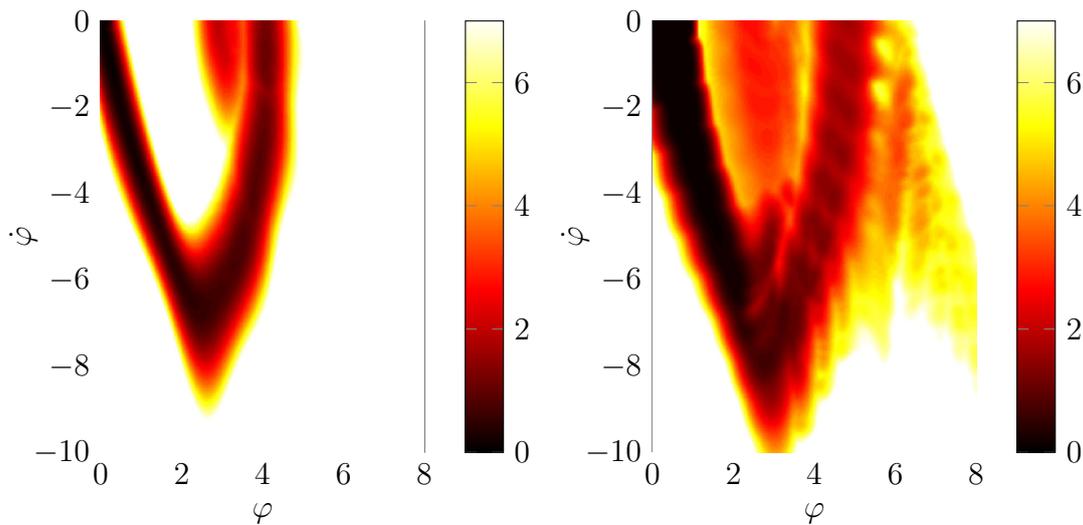


Figure 5.1.: Inverted pendulum: Approximate optimal value function for the Dijkstra-like algorithm. Left: With Algorithm 5.2. Right: With Algorithm 5.2 and rescaling according to Definition 5.8.

- Algorithm 5.2 requires some additional operations, e. g. calculating submatrices. These can be implemented with sparse matrices, though.

### Numerical example: An inverted pendulum on a cart

We again consider the inverted pendulum from Section 3.4 as an example to apply Algorithm 5.2. In Figure 5.1 we have plotted the approximate value function in the lower right quarter of the state space, once the approximation we got from Algorithm 5.2 and once with the rescaled  $\tilde{f}$  and  $\tilde{c}$  from Definition 5.8. In the latter case, the subset where the approximate value function is finite is considerably larger as is clearly visible from the figure. As the approximate value function is in both cases an upper bound to the correct value function, the rescaling of  $f$  and  $c$  seems to be an improvement in this example.



# A. Regularity of the value function without discounting

In this Appendix we investigate the regularity of the optimal value function of undiscounted optimal control problems. In Section A.1 we show its Lipschitz continuity (which is required in Section 2.5) and in Section A.2 we prove that it is generally not differentiable, by providing a counter example. There are similar results in [Vel97, BCD97, RZS09]. However, the assumptions are different from ours and to our best knowledge, the literature does not answer the question whether the optimal value function is Lipschitz continuous or even differentiable in our undiscounted setting.

## A.1. Lipschitz continuity of the value function

The regularity proof for the value function without discounting is considerably more complicated than the corresponding result, Theorem 2.4, in the discounted case.

We assume that a discrete-time control problem is given that is stabilizable on all of  $\Omega$ , and that there is a feedback so that the closed-loop system has 0 as an asymptotically stable fixed point. Furthermore, we assume that  $f \in C^1(\Omega \times U, \Omega)$ ,  $c \in C^2(\Omega \times U, [0, \infty))$ .

Let  $L_f$  be a Lipschitz constant for  $f$  w.r.t.  $x$  and  $L_c$  and  $L_u$  be Lipschitz constants for  $c$  w.r.t.  $x$  resp.  $u$ .

A crucial idea of the proof that  $V$  is Lipschitz continuous is that the number of time steps needed to steer an arbitrary starting point into a neighbourhood of the equilibrium point, turns out to be bounded. The proof will consist of two parts, namely

1. finding a neighbourhood of the equilibrium point where  $V$  can be shown to be Lipschitz continuous and
2. using the Lipschitz constants of  $f$  and  $c$  and the previously mentioned bound on the number of steps needed to control arbitrary initial points into the neighbourhood of the equilibrium point and extending the proof of Lipschitz continuity from the neighbourhood to the whole state space.

## A. Regularity of the value function without discounting

### Local Lipschitz continuity

In the following, we consider the approximation by the linear-quadratic (LQ) system

$$\bar{f}(x, u) = Ax + Bu, \quad \bar{c}(x, u) = x^T Qx + u^T Ru$$

where

$$A = \frac{\partial}{\partial x} f(0, 0), \quad B = \frac{\partial}{\partial u} f(0, 0), \quad Q = \frac{1}{2} \frac{\partial^2}{\partial x^2} c(0, 0), \quad R = \frac{1}{2} \frac{\partial^2}{\partial u^2} c(0, 0).$$

Under the assumption that the original system is controllable, this linearized system is controllable as well. Note that  $c$  is semi-convex by definition. Thus the LQR system is well-posed and its optimal feedback is given by a linear-quadratic regulator (LQR), which is a linear map  $\mathbf{U}(x) = \mathbf{U}x$ , where  $\mathbf{U}$  is a  $d \times s$  matrix, see, e.g. [Son98].

The optimal feedback  $\mathbf{u}(x) = \mathbf{U}x$  for this LQ system is not an optimal feedback for the original system, but it is still a locally stabilizing one as we will see; let  $\bar{V}$  be the derived (in general not optimal) value function

$$\bar{V}(x) = \sum_{k=0}^{\infty} c(x_k, \mathbf{U}x_k), \quad x_{k+1} = f(x_k, \mathbf{U}x_k)$$

of this feedback  $\mathbf{U}$  for the original system.

Consider the matrix-valued map

$$M : \mathbb{R}^s \times \mathbb{R}^d \rightarrow \mathbb{R}^{s \times s}, \quad M(x, u) := f_x(x, u) + f_u(x, u) \cdot \mathbf{U}.$$

The right-hand side is actually  $\frac{\partial}{\partial x}(f(x, \mathbf{U}x))$ .

The spectral radius fulfils  $\rho(M(0, 0)) < 1$  because the original system has 0 as an asymptotically stable fixed point. So there is a norm  $\|\cdot\|_a$  on  $\mathbb{R}^s$  and an  $\varepsilon_a > 0$  with  $\|M(0, 0)\|_a < 1 - 2\varepsilon_a$ . Let  $0 \in \Omega_1 \subset \Omega$  and  $0 \in U_1 \subset U$  be open neighbourhoods such that

$$\|M(x, u)\|_a \leq 1 - \varepsilon_a. \tag{A.1}$$

for all  $(x, u) \in \Omega_1 \times U_1$ .

We choose even smaller open sets  $0 \in \Omega_2 \subset \Omega_1, 0 \in U_2 \subset U_1$  by further requiring

- $\Omega_2$  to be a open ball relative to the norm  $\|\cdot\|_a$ ,
- $\mathbf{U}x \in U_2$  for all  $x \in \Omega_2$  and
- $u + \mathbf{U}(\tilde{x} - x) \in U_1$  for all  $x, \tilde{x} \in \Omega_2$  and  $u \in U_2$ .

Now we show

### A.1. Lipschitz continuity of the value function

**Lemma A.1.** *Let  $(x, u) \in (\Omega_2 \times U_2)$ . Then*

$$\phi : \Omega_2 \rightarrow \Omega, \quad \phi(\tilde{x}) := f(\tilde{x}, u + \mathbf{U}(\tilde{x} - x)).$$

*is a contractive relative to the norm  $\|\cdot\|_a$  with contraction factor  $1 - \varepsilon_a$ .*

Note that in general  $\phi$  does not map  $\Omega_2$  to itself.

*Proof.* One has

$$\begin{aligned} \left\| \frac{\partial}{\partial \tilde{x}} \phi(\tilde{x}) \right\|_a &= \|f_x(\tilde{x}, u + \mathbf{U}(\tilde{x} - x)) + f_u(\tilde{x}, u + \mathbf{U}(\tilde{x} - x)) \cdot \mathbf{U}\|_a \\ &= \|M(\tilde{x}, u + \mathbf{U}(\tilde{x} - x))\|_a \leq 1 - \varepsilon_a \end{aligned}$$

by (A.1) and, consequently, by the mean value theorem,  $\phi$  is a contractive.  $\square$

By setting  $(x, u) = (0, 0)$  one gets the following corollary. Noting that  $\Omega_2$  is an open ball relative to  $\|\cdot\|_a$ , this time the map has images in  $\Omega_2$ .

**Corollary A.2.**  *$f(\cdot, \mathbf{U}(\cdot))$  is a contraction on  $\Omega_2$  relative to the norm  $\|\cdot\|_a$ .*

This allows us to show that  $\bar{V}$  is continuous on a neighbourhood of 0.

**Corollary A.3.**  *$\bar{V}$  is continuous on  $\Omega_2$ , even Lipschitz continuous.*

*Proof.*  $c$  was assumed to be Lipschitz continuous w.r.t.  $x$  resp.  $u$  with Lipschitz constants  $L_c$  resp.  $L_u$ . By the equivalence of norms on finite-dimensional vector spaces,  $c$  is also Lipschitz continuous w.r.t.  $x$  resp.  $u$  with some Lipschitz constants  $L_a$  and  $L_{au}$  relative to the norm  $\|\cdot\|_a$ . Consequently,  $x \mapsto c(x, \mathbf{U}x)$  is Lipschitz continuous with Lipschitz constant  $L_a + L_{au}\|\mathbf{U}\|_a$  relative to  $\|\cdot\|_a$ .

Now one can show with the help of Lemma A.1 that  $\bar{V}$  is Lipschitz continuous with Lipschitz constant  $\frac{L_a + L_{au}\|\mathbf{U}\|_a}{\varepsilon_a}$  relative to the norm  $\|\cdot\|_a$ . This can be seen by considering two points  $x_0, \tilde{x}_0 \in \Omega_2$  and comparing their trajectories  $(x_k)$  and  $(\tilde{x}_k)$  under the feedback  $\mathbf{U}$ . Their mutual distances  $x_k - \tilde{x}_k$  relative to  $\|\cdot\|_a$  develop at most like a geometric sequence with contraction factor  $1 - \varepsilon_a$ . Then

one gets

$$\begin{aligned} \bar{V}(x) - \bar{V}(\tilde{x}) &\leq \sum_{k=0}^{\infty} |c(x_k) - \tilde{c}(x_k)| \\ &\leq \sum_{k=0}^{\infty} (1 - \varepsilon_a)^k (L_a + L_{au}\|\mathbf{U}\|_a) \\ &= \frac{L_a + L_{au}\|\mathbf{U}\|_a}{\varepsilon_a}. \end{aligned}$$

$\square$

In the next step we choose an even smaller open neighbourhood  $0 \in \Omega_3 \subset \Omega_2$  according to the following lemma.

A. Regularity of the value function without discounting

**Lemma A.4.** *There is a neighbourhood  $0 \in \Omega_3 \subset \Omega_2$  such that each optimal feedback of the original system on  $\Omega_3$  has values in  $U_2$ .*

*Proof.* One has

$$\min_{x \in \Omega, u \in U \setminus U_2} c(x, u) = \delta_2 > 0$$

because of the compactness of  $\Omega$  and  $U$ , the fact that  $U_2$  is open and  $c(x, u) > 0$  for  $u \neq 0$ . One can choose  $\Omega_3$  such that

$$\sup_{x \in \Omega_3} \bar{V}(x) < \delta_2,$$

because  $\bar{V}(0) = 0$ , and consequently

$$\sup_{x \in \Omega_3} V(x) \leq \sup_{x \in \Omega_3} \bar{V}(x) < \delta_2$$

because  $v$  is the optimal value function. Now, for points in  $\Omega_3$ , the optimal controls are in  $U_2$ .  $\square$

We choose  $\Omega_3$  according to this lemma, but with the additional condition that  $\bar{\Omega}_3 \subset \Omega_2$ . So the Hausdorff distance relative to the norm  $\|\cdot\|_a$  between the sets,  $d_a(\Omega_2, \Omega_3)$  is positive. We choose a neighbourhood  $0 \in \Omega_4 \subset \Omega_3$  such that all optimal trajectories starting in  $\Omega_4$  stay in  $\Omega_3$  for all times. This is the case if  $\sup_{\Omega_4} V \leq \sup_{\Omega_4} \bar{V} \leq \min_{x \in \Omega \setminus \Omega_3, u \in U} c(x, u)$ . The last term is positive because of compactness of  $(\Omega \setminus \Omega_3) \times U$ . In the proof of the following lemma we explicitly need the cost functional

$$J(x_0, (u_k)) := \sum_{k=0}^{\infty} c(x_k, u_k)$$

from Chapter 1 which maps an initial point  $x_0$  and a control sequence  $(u_k)$  onto the total cost along the induced trajectory  $x_{k+1} = f(x_k, u_k)$ ,  $k = 0, 1, 2, \dots$

**Theorem A.5.**  *$V$  is Lipschitz continuous on the neighbourhood  $\Omega_4 \subset \Omega$  of 0.*

*Proof.* We show the Lipschitz continuity for pairs of points which are close to each other. The obtained Lipschitz constant will be independent of this choice of a pair of points, though, implying the Lipschitz continuity on the whole set. For given  $x_0, \tilde{x}_0 \in \Omega_4$  with  $\|x_0 - \tilde{x}_0\|_a < d_a(\Omega_2, \Omega_3)$  we choose  $(u_k)$  as an almost optimal control sequence for  $x_0$ :  $J(x_0, (u_k)) \leq V(x_0) + (L_a + L_{au} \|\mathbf{U}\|_a) \frac{1}{\varepsilon_a} \|\tilde{x}_0 - x_0\|_a$  and such that  $(x_k)$  stays in  $\Omega_3$ . The latter is possible because  $\Omega_3$  is open, so almost optimality is enough for a trajectory to remain in  $\Omega_3$ . Note that optimality of the control sequence cannot be assumed at this point, as it is not yet clear that  $V$  is continuous. From the continuity of  $V$  and the Bellman equation the existence of optimal controls will follow, but as long as the continuity of  $V$  has not been shown only almost optimal controls can be assumed, i.e. control sequences which induce a value at a point arbitrarily close to the optimal value function at this point.

### A.1. Lipschitz continuity of the value function

The sequence  $(u_k)$  gives us a sequence  $(x_k)$ . For the construction of the sequence  $\tilde{x}_k$  from the point  $\tilde{x}_0$ , we define iteratively  $\tilde{x}_k := f(\tilde{x}_{k-1}, \tilde{u}_{k-1})$  with

$$\tilde{u}_k := u_k + \mathbf{U}(\tilde{x}_k - x_k),$$

which can be considered a “linear correction” of  $u_k$  with the feedback one has from the linear-quadratic approximation system.

Consider

$$\phi_k(\tilde{x}) := f(\tilde{x}, u_k + \mathbf{U}(\tilde{x} - x_k)),$$

so by Lemma A.1

$$\|x_{k+1} - \tilde{x}_{k+1}\|_a \leq (1 - \varepsilon_a)\|x_k - \tilde{x}_k\|_a$$

and iteratively one sees that  $(\tilde{x}_k)$  stays in  $\Omega_2$  because of the condition  $\|x_0 - \tilde{x}_0\|_a < d_a(\Omega_2, \Omega_3)$ .

Consequently,

$$\begin{aligned} |J(x_0, (u_k)) - J(\tilde{x}_0, (\tilde{u}_k))| &\leq (L_a + L_{au}\|\mathbf{U}\|_a) \frac{1}{1 - (1 - \varepsilon_a)} \|x_0 - \tilde{x}_0\|_a \\ &= (L_a + L_{au}\|\mathbf{U}\|_a) \frac{1}{\varepsilon_a} \|x_0 - \tilde{x}_0\|_a \end{aligned}$$

and thus

$$\begin{aligned} V(\tilde{x}_0) &\leq J(\tilde{x}_0, (\tilde{u}_k)) \leq J(x_0, (u_k)) + (L_a + L_{au}\|\mathbf{U}\|_a) \frac{1}{\varepsilon_a} \|x_0 - \tilde{x}_0\|_a \\ &\leq V(x_0) + (L_a + L_{au}\|\mathbf{U}\|_a) \frac{2}{\varepsilon_a} \|x_0 - \tilde{x}_0\|_a. \end{aligned}$$

Changing the roles of  $x_0$  and  $\tilde{x}_0$ , and noting that the two norms  $\|\cdot\|$  and  $\|\cdot\|_a$  on a finite-dimensional space are equivalent, we conclude

$$|V(x_0) - V(\tilde{x}_0)| \leq L_{\text{loc}}\|x_0 - \tilde{x}_0\|$$

for some  $L_{\text{loc}} > 0$  independent of  $x_0$  and  $\tilde{x}_0$ , which concludes the proof of Lipschitz continuity on  $\Omega_4$ .  $\square$

For the continuation in the next section we choose  $\varepsilon_1 > 0$  with  $U_{2\varepsilon_1}(0) \subset \Omega_4$ .

## Global Lipschitz continuity

The generalization of the local Lipschitz continuity on  $U_{2\varepsilon_1}(0)$  to global Lipschitz continuity on  $\Omega$  is essentially based on the compactness of  $\Omega$  as we will see in the following.

**Lemma A.6.** *V is bounded on  $\Omega$ .*

### A. Regularity of the value function without discounting

*Proof.* Let  $x_0 \in \Omega$  and  $(u_k)_k$  a stabilizing control sequence for  $x_0$ . Therefore the sequence defined by  $x_{k+1} = f(x_k, u_k)$  converges to 0 and there is a  $k_1$  such that  $x_{k_1} \in U_{\varepsilon_1}(0)$ . From the continuity of the system dynamics  $f$  it follows that there is a neighbourhood  $N(x_0)$  such that each  $\tilde{x} \in N(x_0)$  is steered to  $U_{2\varepsilon_1}(0)$  in  $k_1$  steps if the same control sequence  $(u_k)_k$  is applied.

By Lemma A.5  $V$  is Lipschitz continuous and hence bounded on  $U_{2\varepsilon_1}(0)$ , so it is also bounded on  $N(x_0)$  because  $c$  is bounded: For any  $\tilde{x} \in N(x_0)$  it holds that

$$\begin{aligned} V(\tilde{x}) &\leq \sum_{k=0}^{k_1-1} c(\tilde{x}_k, u_k) + V(\tilde{x}_{k_1}) \\ &\leq k_1 \sup_{\Omega \times U} c(x, u) + L_{\text{loc}} \text{diam}(U_{2\varepsilon_1}(0)) \\ &= k_1 \sup_{\Omega \times U} c(x, u) + L_{\text{loc}} 2\varepsilon_1. \end{aligned}$$

Now by compactness of  $\Omega$ , finitely many such sets  $N(x_0)$  cover  $\Omega$ . By taking the supremum of the finitely many integers  $k_1(x)$ ,  $V$  is also bounded on  $\Omega$  by

$$\sup_{x \in \Omega} k_1(x) \sup_{\Omega \times U} c(x, u) + L_{\text{loc}} 2\varepsilon_1.$$

□

Let  $\Delta := \sup_{x \in \Omega} V(x)$  which is finite as was just shown. Let the cost function  $c$  be bounded from below by  $\delta_3 > 0$  outside of  $U_{\varepsilon_1}(0)$ . A positive lower bound exists because of the compactness of  $\Omega \setminus U_{\varepsilon_1}(0) \times U$ . Let  $k_0$  be an integer with  $k_0 \geq \frac{\Delta}{\delta_3}$  and  $\varepsilon_0 := \varepsilon_1 / L_f^{k_0}$ . The definition of  $k_0$  is such that for any point  $x \in \Omega$  it is possible to reach  $U_{\varepsilon_1}(0)$  in at most  $k_0$  step, because as long as a trajectory stays outside of  $U_{\varepsilon_1}(0)$  in each step a cost of at least  $\delta_3$  is added, so in  $k_0 + 1$  steps at least  $(k_0 + 1)\delta_3 > \Delta$ , so  $U_{\varepsilon_1}(0)$  can be reached earlier if the control sequence is chosen almost optimal.

From now on, let  $F : \Omega_4 \rightarrow U$  be an optimal feedback which exists on  $\Omega_4$  because  $V$  is continuous on  $\Omega_4$  by Lemma A.5 and so the right-hand side of the Bellman equation depends continuously on  $u \in U$  and  $U$  is compact such that an optimal  $u \in U$  can always be chosen.

With the next lemma we show Lipschitz continuity of  $V$  locally, for pairs of nearby points.

**Lemma A.7.** *For any  $x_0 \in \Omega$  there is a neighbourhood  $x_0 \in A \subset \Omega$  such that any  $\tilde{x}_0 \in A$  fulfils the following statement. Let  $(u_k)$  a control sequence which steers  $x_0$  in  $k_0$  steps to  $U_{\varepsilon_1}(0)$  and then continues as given by the optimal feedback  $F$  and let*

$$(\tilde{u}_k) := (u_0, \dots, u_{k_0}, F(\tilde{x}_{k_0+1}), F(\tilde{x}_{k_0+2}), \dots)$$

*be a control sequence for  $\tilde{x}_0$ . Then*

$$|J(x_0, (u_k)) - J(\tilde{x}_0, (\tilde{u}_k))| \leq L_1 \|x_0 - \tilde{x}_0\|$$

*with a constant  $L_1 > 0$  independent of the points  $x_0$  and  $\tilde{x}_0$ .*

### A.1. Lipschitz continuity of the value function

*Proof.* For the points  $x_0, \tilde{x}_0$  with  $d := \|x_0 - \tilde{x}_0\| < \varepsilon_0$  we consider the trajectories  $(x_k)$  and  $(\tilde{x}_k)$  of  $f$  starting at  $x_0, \tilde{x}_0$  for the control sequences  $(u_k)$  and  $(\tilde{u}_k)$ , respectively. By the choice of  $(u_k)$  and the definition of  $k_0$ , one has  $x_{k_0} \in U_{\varepsilon_1}(0)$ .

In addition, one has  $\|x_{k_0} - \tilde{x}_{k_0}\| \leq \varepsilon_0 L_f^n = \varepsilon_1$ , so  $\tilde{x}_{k_0} \in U_{2\varepsilon_1}(0)$ . It follows

$$\begin{aligned}
|J(x_0, (u_k)) - J(\tilde{x}_0, (\tilde{u}_k))| &\leq |c(x_0, u_0) - c(\tilde{x}_0, u_0)| + |c(x_1, u_1) - c(\tilde{x}_1, u_1)| \\
&\quad + \dots + |c(x_{k_0-1}, u_{k_0-1}) - c(\tilde{x}_{k_0-1}, u_{k_0-1})| \\
&\quad + |V(x_{k_0}) - V(\tilde{x}_{k_0})| \\
&\leq L_c d + L_c d L_f + \dots + L_c d L_f^{n-1} + d L_f^n L_{\text{loc}} \\
&= d \underbrace{(L_c + L_c L_f + \dots + L_c L_f^{n-1} + L_f^n L_{\text{loc}})}_{=: L_1} \\
&= \|x_0 - \tilde{x}_0\| L_1
\end{aligned}$$

from Theorem A.5. □

**Theorem A.8** (Lipschitz continuity of  $V$ ). *Let  $V$  be the optimal value function of a discrete-time control problem that is stabilizable on the compact state space  $\Omega$  and with the compact control space  $U$ , where the system dynamics  $f$  is in  $C^1(\Omega \times U, \Omega)$ , the cost function  $c$  is in  $C^2(\Omega \times U, \mathbb{R}^+)$ , and that has a feedback whose closed-loop system has 0 as an asymptotically stable fixed point. Then  $V$  is Lipschitz continuous.*

*Proof.* Let  $x_0$  and  $\tilde{x}_0$  be two points in  $\Omega$  with  $\|x_0 - \tilde{x}_0\| \leq \varepsilon_0$  and  $(u_k)$  a control sequence such that the induced trajectory for the point  $x_0$  fulfils

$$V(\tilde{x}_0) \leq J(\tilde{x}_0, (\tilde{u}_k)) + L_1 \|x_0 - \tilde{x}_0\|$$

From the preceding lemma, we get

$$\begin{aligned}
V(\tilde{x}_0) &\leq J(\tilde{x}_0, (\tilde{u}_k)) + L_1 \|x_0 - \tilde{x}_0\| \\
&\leq J(x_0, (u_k)) + 2L_1 \|x_0 - \tilde{x}_0\| \\
&= V(x_0) + 2L_1 \|x_0 - \tilde{x}_0\|.
\end{aligned}$$

Changing the roles of  $x_0$  and  $\tilde{x}_0$ , we conclude

$$|V(x_0) - V(\tilde{x}_0)| \leq 2L_1 \|x_0 - \tilde{x}_0\|.$$

Now we can skip the assumption  $\|x_0 - \tilde{x}_0\| \leq \varepsilon_0$ , because a local Lipschitz constant which does not depend from the choice of the pair of points is also a global Lipschitz constant, showing the global Lipschitz continuity of  $V$ . □

**Corollary A.9.** *Under the assumptions of Theorem A.8 the Kružkov transformed optimal value function (1.6) is also Lipschitz continuous.*

## A. Regularity of the value function without discounting

*Proof.* By the definition of  $v$  as  $v(\cdot) = \exp(-V(\cdot))$  and Lipschitz continuity of the exponential function on  $(-\infty, 0]$ , the function  $v$  is also Lipschitz continuous.  $\square$

In general,  $V$  and  $v$  can not be expected to be differentiable as we will see in the next section.

We remark that one can get as a corollary the corresponding statement for continuous-time systems. By the considerations of Section 1.3 continuous-time control problems can be considered a special case of discrete-time control problems because of its continuous evolution. The infinite dimensionality of the control space turns out not to be a fundamental problem. For the feedback  $F : \Omega_4 \rightarrow U$  optimality can no longer be assured because  $U$  is no longer compact but for the proof it is possible to take a feedback which is sufficiently close to optimality.

## A.2. Nondifferentiability of the value function

In Section A.1 we showed the Lipschitz continuity of optimal value functions for undiscounted discrete-time control problems under certain assumptions. A natural question is whether this result can be strengthened towards differentiability or even smoothness of the optimal value function if the maps  $f$  and  $c$  have sufficient regularity.

In general this is not possible. The main obstruction is that the definition of the value function as well as the Bellman equation have an infimum on the right-hand side taken over all possible controls  $u \in U$ , which in general leads to a non-differentiable function  $V$  even if the single functions for the different  $u \in U$  is smooth.

At some distance from the point of equilibrium there can be topological limitations if branches of different solutions intersect each other. Perhaps the easiest example for such a case is given by the shortest path problem on the circle  $S^1 = \{z \in \mathbb{C} \mid |z| = 1\} \subset \mathbb{C}$ . If the target set  $\{1\}$  is considered, then the distance function on  $S^1$  is piecewise linear (along the curve, locally parametrized according to path length) and has two points of non-differentiability, namely 1 and -1. The non-differentiability at the point of equilibrium can be explained by the jump in the cost function which is 0 in 1, and 1 elsewhere. However, the other irregularity at -1 is a consequence of the branching between the paths which go along the upper respectively the lower semicircle (in other words, paths with positive respectively negative imaginary parts).

In this section we present an example which shows that an optimal value function does not need to be differentiable, not even in a neighbourhood of an equilibrium point and not even in the case that the dynamics and cost function are smooth. The example will work both in the case of continuous-time and discrete-time systems with almost no modification.

## A.2. Nondifferentiability of the value function

We will need the well-known smooth function

$$\mathbb{R} \rightarrow \mathbb{R}, x \mapsto \begin{cases} 0 & x \leq 0, \\ e^{-\frac{1}{x}} & x > 0, \end{cases}$$

from which we derive the odd function

$$\phi : [-1, 1] \rightarrow [-1, 1], x \mapsto \begin{cases} -e \cdot e^{\frac{1}{x}} & x < 0, \\ 0 & x = 0, \\ e \cdot e^{-\frac{1}{x}} & x > 0, \end{cases}$$

which is a smooth bijection of  $[-1, 1]$  onto itself.

**Example.** First, we consider the discrete-time control system where

$$\Omega = [-1, 1], U = [-1, 1],$$

$$f(x, u) = x + \phi(u)$$

and

$$c(x, u) = x^2 + \phi(|u|).$$

**Proposition A.10.** *The optimal value function for this control system is  $V(x) = |x| + x^2$  and the optimal feedback is  $\mathbf{u}(x) = \phi^{-1}(-x)$ .*

*Proof.* First, we show that  $V \geq |x| + x^2$ . To this aim, note that for any initial point  $x_0$  and control sequence  $(u_k)$  we have  $x_k = x_0 + \sum_{i=0}^k u_i$  and for the total cost

$$\begin{aligned} J(x_0, (u_k)) &= \sum_{k=0}^{\infty} c(x_k, u_k) \\ &= \sum_{k=0}^{\infty} (x_k^2 + \phi(|u_k|)) \\ &= \sum_{k=0}^{\infty} (x_k^2 + |\phi(u_k)|) \\ &\geq \sum_{k=0}^{\infty} x_k^2 + \left| \lim_{k \rightarrow \infty} \sum_{i=0}^k \phi(u_i) \right| \\ &= \sum_{k=0}^{\infty} x_k^2 + \left| \lim_{k \rightarrow \infty} x_k - x_0 \right| \\ &\geq x_0^2 + |x_0|. \end{aligned}$$

The last inequality holds true because  $\lim_{k \rightarrow \infty} x_k = 0$  or otherwise  $\sum_{k=0}^{\infty} x_k^2 = \infty$  such that the inequality holds trivially.

### A. Regularity of the value function without discounting

Thus, we have shown that  $V \geq |x| + x^2$ .

On the other hand, the trajectory can be forced to be  $x_1 = x_2 = \dots = 0$  by choosing the control sequence with

$$u_0 = \phi^{-1}(-x_0) \Leftrightarrow \phi(u) = -x_0 \Leftrightarrow f(x_0, u) = 0$$

and  $u_1 = u_2 = \dots = 0$ , leading to

$$J(x_0, (u_k)) = x_0^2 + \phi(|u_0|) = x_0^2 + |x_0|,$$

which shows  $V(x_0) \leq |x_0| + x_0^2$ .

Together, from both parts, one gets  $V(x) = |x| + x^2$ . □

One could ask why we did not just take the cost function  $c(x, u) = \phi(|u|)$  instead of  $c(x, u) = x^2 + \phi(|u|)$ . The reason is that then the incentive to steer the system to 0 would be missing, formally contradicting the requirement that  $c(x, u) > 0$  for  $x$  outside the target and any  $u \in U$ . The control function  $u \equiv 0$  for any initial point  $x_0$  would show that the optimal value function is actually  $V \equiv 0$ .

From this discrete-time control system it is possible to derive a corresponding continuous-time example. We present it here for completeness, although continuous-time control systems as such are not within the scope of this thesis.

**Example.** Consider the continuous-time control system defined by

$$\Omega = [-1, 1], U = [-1, 1],$$

$$f(x, u) = \phi(u)$$

and

$$c(x, u) = x^2 + \phi(|u|).$$

**Proposition A.11.** *The optimal value function for this control system is*

$$V(x) = |x| + \frac{1}{3}|x|^3.$$

*Proof.* We begin with the lower bound  $V(x) \geq |x| + \frac{1}{3}|x|^3$ .

For an initial point  $x_0$ , a piecewise continuous control function  $u : [0, \infty) \rightarrow U$  and the induced trajectory  $x : [0, \infty) \rightarrow \Omega$ , one has

$$x(t) = x_0 + \int_0^t f(x(t), u(t))dt = x_0 + \int_0^t \phi(u(t))dt$$

## A.2. Nondifferentiability of the value function

and thus, along the trajectory, the associated cost

$$\begin{aligned}
 J(x_0, u(\cdot)) &= \int_0^\infty c(x(t), u(t)) dt \\
 &= \int_0^\infty (x(t)^2 + \phi(|u(t)|)) dt \\
 &= \int_0^\infty x(t)^2 dt + \lim_{t \rightarrow \infty} \int_0^t \phi(|u(t)|) dt \\
 &= \int_0^\infty x(t)^2 dt + \lim_{t \rightarrow \infty} \int_0^t |\phi(u(t))| dt \\
 &\geq \int_0^\infty x(t)^2 dt + \limsup_{t \rightarrow \infty} \left| \int_0^t \phi(u(t)) dt \right| \\
 &= \int_0^\infty x(t)^2 dt + \limsup_{t \rightarrow \infty} |x(t) - x_0| \\
 &\geq \int_0^\infty x(t)^2 dt + |x_0| \\
 &\geq |x_0| + \frac{1}{3}|x_0|^3,
 \end{aligned}$$

where the last inequality is true because  $\max |\phi| = 1 \Rightarrow |x'(t)| \leq 1$  implies

$$\int_0^\infty x(t)^2 dt \geq \int_0^\infty (|x_0 - t|_+)^2 dt = \frac{1}{3}|x_0|^3,$$

and the second to last inequality,

$$\int_0^\infty x(t)^2 dt + \limsup_{t \rightarrow \infty} |x(t) - x_0| \geq \int_0^\infty x(t)^2 dt + |x_0|, \quad (\text{A.2})$$

is valid because  $\lim_{t \rightarrow \infty} x(t) = 0$  or otherwise there would be a sequence  $0 < t_1 < t_2, \dots$  with  $t_{k+1} > t_k + 1$  and  $|x(t_k)| \geq \varepsilon$  for all  $k$  and a  $0 < \varepsilon < 1$ , and consequently the intervals  $[t_k, t_k + 1]$  would be pairwise disjoint and so, from  $|x'(t)| \leq 1$ ,

$$\int_0^\infty x(t)^2 dt \geq \sum_{k=1}^\infty \int_{t_k}^{t_{k+1}} x(t)^2 dt \geq \sum_{k=1}^\infty \underbrace{\int_0^1 (|\varepsilon - t|_+)^2 dt}_{=\varepsilon^3/3} = \infty,$$

so  $\int_0^\infty x(t)^2 dt = \infty$  and (A.2) holds again.

For the upper bound,  $V(x) \leq |x| + \frac{1}{3}|x|^3$ , we consider the feedback

$$\mathbf{u}(x) = \begin{cases} 1 & x < 0, \\ 0 & x = 0, \\ -1 & x > 0, \end{cases}$$

A. *Regularity of the value function without discounting*

which yields the trajectory  $x(t) = (x_0 - t)_+$  if  $x_0 \geq 0$  and  $x(t) = -(-x_0 - t)_+$  if  $x_0 < 0$  and the cost

$$\begin{aligned}
 J(x_0, \mathbf{u}(x(\cdot))) &= \int_0^\infty c(x(t), u(t)) dt \\
 &= \int_0^\infty (x(t)^2 + \phi(|u(t)|)) dt \\
 &= \int_0^{x_0} (((|x_0| - t)_+)^2 + 1) dt + \int_{x_0}^\infty \underbrace{(x(t))^2}_{=0} + \phi(\underbrace{|u(t)|}_{=0}) dt \\
 &= \frac{1}{3}|x_0|^3 + |x_0|,
 \end{aligned}$$

consequently  $V(x) \leq |x| + \frac{1}{3}|x|^3$ .

Altogether, we have shown  $V(x) = |x| + \frac{1}{3}|x|^3$ .

□

Finally we want to remark that if the aim had just been to give a counterexample to the smoothness (rather than differentiability) of  $V$  then it would have been sufficient to take  $\Omega = U = [-1, 1]$ ,  $f(x, u) = u$ ,  $c(x, u) = x^2$  leading to  $V(x) = \frac{1}{3}|x|^3$ .

# Bibliography

- [1] [http://www-m3.ma.tum.de/Allgemeines/JuSch\\_DP\\_RBF](http://www-m3.ma.tum.de/Allgemeines/JuSch_DP_RBF).
- [AB02] K. J. Åström and B. M. Bernhardsson. Comparison of Riemann and Lebesgue sampling for first order stochastic systems. In *Proceedings of the 41st IEEE Conference on Decision and Control*, volume 2, pages 2011–2016, Dec 2002.
- [AWJR12] H. Alwardi, S. Wang, L. S. Jennings, and S. Richardson. An adaptive least-squares collocation radial basis function method for the HJB equation. *J. Glob. Opt.*, 52(2):305–322, 2012.
- [BCD97] M. Bardi and I. Capuzzo-Dolcetta. *Optimal control and viscosity solutions of Hamilton-Jacobi-Bellman equations*. Birkhäuser, Boston, Boston, MA, 1997.
- [Bel54] R. Bellman. Dynamic programming and a new formalism in the calculus of variations. *Proceedings of the National Academy of Sciences of the United States of America*, 40(4):pp. 231–235, 1954.
- [Bel57] R. Bellman. *Dynamic programming*. Princeton University Press, Princeton, NJ, 1957.
- [Ber05] D. P. Bertsekas. *Dynamic Programming and Optimal Control*. Number Bd. 1 in Athena Scientific optimization and computation series. Athena Scientific, 2005.
- [Ber07] D. P. Bertsekas. *Dynamic Programming and Optimal Control, Vol. II*. Athena Scientific, 3rd edition, 2007.
- [CD83] I. Capuzzo-Dolcetta. On a discrete approximation of the Hamilton-Jacobi equation of dynamic programming. *Appl. Math. Optim.*, 10(1):367–377, 1983.
- [CDF89] I. Capuzzo-Dolcetta and M. Falcone. Discrete dynamic programming and viscosity solutions of the Bellman equation. *Ann. Inst. H. Poincaré Anal. Non Linéaire*, 6(suppl.):161–183, 1989.
- [Cla04] F. Clarke. Lyapunov functions and feedback in nonlinear control. In *Optimal control, stabilization and nonsmooth analysis*, volume 301 of

## Bibliography

- Lecture Notes in Control and Inform. Sci.*, pages 267–282. Springer, Berlin, 2004.
- [CLSW98] F. Clarke, Y. Ledyev, R. Stern, and P. Wolenski. *Nonsmooth analysis and control theory*, volume 178 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 1998.
- [CQO04] T. Cecil, J. Qian, and S. Osher. Numerical methods for high dimensional Hamilton-Jacobi equations using radial basis functions. *J. Comp. Phys.*, 196(1):327 – 347, 2004.
- [DH07] T. A. Driscoll and Alfa R. H. Heryudono. Adaptive residual subsampling methods for radial basis function interpolation and collocation problems. *Comput. Math. Appl.*, 53(6):927–939, 2007.
- [Dij59] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1):269–271, 1959.
- [Fal87] M. Falcone. A numerical approach to the infinite horizon problem of deterministic control theory. *Appl. Math. Optim.*, 15(1):1–13, 1987.
- [Fas07] G. E. Fasshauer. *Meshfree Approximation Methods with Matlab*. World Scientific, 2007.
- [FF94] M. Falcone and R. Ferretti. Discrete time high-order schemes for viscosity solutions of Hamilton-Jacobi-Bellman equations. *Numer. Math.*, 67(3):315–344, 1994.
- [GJ05] L. Grüne and O. Junge. A set oriented approach to optimal feedback stabilization. *Sys. Contr. Lett.*, 54(2):169 – 180, 2005.
- [GJJ<sup>+</sup>10] L. Grüne, S. Jerg, O. Junge, D. Lehmann, J. Lunze, F. Müller, and M. Post. Two complementary approaches to event-based control (Zwei komplementäre Zugänge zur ereignisbasierten Regelung). *Automatisierungstechnik*, 58(4):173–183, 2010.
- [GMM79] E. Gottzein, R. Meisinger, and L. Miller. Anwendung des ”Magnetischen Rades” in Hochgeschwindigkeitsmagnetschwebbahnen. *ZEV-Glasers Annalen*, 103, 1979.
- [Grü97] L. Grüne. An adaptive grid scheme for the discrete Hamilton-Jacobi-Bellman equation. *Numer. Math.*, 75(3):319–337, 1997.
- [GS04] L. Grüne and W. Semmler. Using dynamic programming with adaptive grid scheme for optimal control problems in economics. *J. Econom. Dynam. Control*, 28(12):2427–2456, 2004.

- [HWCL06] C. S. Huang, S. Wang, CS Chen, and Z. C. Li. A radial basis collocation method for Hamilton-Jacobi-Bellman equations. *Automatica*, 42(12):2201–2207, 2006.
- [JO04] O. Junge and H. M. Osinga. A set oriented approach to global optimal control. *ESAIM Control Optim. Calc. Var.*, 10(2):259–270, 2004.
- [JS15] O. Junge and A. Schreiber. Dynamic programming using radial basis functions. *Discrete and Continuous Dynamical Systems*, 35(9):4439–4453, 2015.
- [KN74] L. Kuipers and H. Niederreiter. *Uniform distribution of sequences*. Wiley-Interscience [John Wiley & Sons], New York-London-Sydney, 1974. Pure and Applied Mathematics.
- [Kru75] S. N. Kružkov. Generalized solutions of Hamilton-Jacobi equations of eikonal type. I. *Mat. Sb. (N.S.)*, 98(140):450–493, 1975.
- [LR06] B. Lincoln and A. Rantzer. Relaxing dynamic programming. *IEEE Trans. Auto. Ctrl.*, 51(8):1249–1260, 2006.
- [MCF] <http://www.mathworks.com/matlabcentral/fileexchange/>.
- [Nie92] H. Niederreiter. *Random Number Generation and Quasi-Monte Carlo Methods*. Society for Industrial and Applied Mathematics, 1992.
- [Pon87] L. S. Pontryagin. *Mathematical Theory of Optimal Processes*. Classics of Soviet Mathematics. Taylor & Francis, 1987.
- [RZS09] J. P. Rincón-Zapatero and M. S. Santos. Differentiability of the value function without interiority assumptions. *J. Econom. Theory*, 144(5):1948–1964, 2009.
- [Sch95] R. Schaback. Creating surfaces from scattered data using radial basis functions. In *in Mathematical Methods for Curves and Surfaces*, pages 477–496. University Press, 1995.
- [She68] D. Shepard. A two-dimensional interpolation function for irregularly-spaced data. In *Proc. 23rd ACM*, pages 517–524. ACM, 1968.
- [Son98] E. D. Sontag. *Mathematical Control Theory: Deterministic Finite Dimensional Systems*. Springer, 1998.
- [SW00a] R. Schaback and H. Wendland. Adaptive greedy techniques for approximate solution of large rbf systems. *Numerical Algorithms*, 24(3):239–254, 2000.

## Bibliography

- [SW00b] R. Schaback and H. Wendland. Numerical techniques based on radial basis functions. Technical report, DTIC Document, 2000.
- [Vel97] V. M. Veliov. Lipschitz continuity of the value function in optimal control. *J. Optim. Theory Appl.*, 94(2):335–363, 1997.
- [Wen95] H. Wendland. Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. *Adv. Comp. Math.*, 4:389–396, 1995.
- [Wen04] H. Wendland. *Scattered Data Approximation*. Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, 2004.