



HAL
open science

Dynamic programming using radial basis functions

Oliver Junge, Alex Schreiber

► **To cite this version:**

Oliver Junge, Alex Schreiber. Dynamic programming using radial basis functions. NETCO 2014, 2014, Tours, France. hal-01024655

HAL Id: hal-01024655

<https://inria.hal.science/hal-01024655>

Submitted on 16 Jul 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Dynamic programming using radial basis functions

Oliver Junge

Fakultät für Mathematik
Technische Universität München

joint work with Alex Schreiber

Problem

- ▶ discrete-time control system

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

$f : \Omega \times U \rightarrow \Omega$ continuous

- ▶ $\Omega \subset \mathbb{R}^d$ and $U \subset \mathbb{R}^m$ compact
- ▶ target set $T \subset \Omega$, compact
- ▶ **goal:** construct **feedback** $F : S \rightarrow U$, $S \subset \Omega$, such that for the closed loop system

$$x_{k+1} = f(x_k, F(x_k)), \quad x_k \in S,$$

the target T is asymptotically stable.

Optimal control

- ▶ cost function $c : \Omega \times U \rightarrow [0, \infty)$ continuous, $c(x, u) \geq \delta > 0$ for $x \notin T$ and any $u \in U$.
- ▶ accumulated cost

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

with trajectory $(x_k)_k$ associated to $x_0 \in \Omega$ and $(u_k)_k \in U^{\mathbb{N}}$.

- ▶ optimal value function

$$V(x) = \inf_{(u_k)_k} J(x, (u_k)_k)$$

The Bellman equation

- ▶ V fulfills the Bellman equation

$$\begin{aligned} V(x) &= \inf_{u \in U} \{c(x, u) + V(f(x, u))\} \\ &=: L[V](x) \end{aligned}$$

with boundary condition $V(T) = 0$.

- ▶ optimal feedback

$$F(x) = \operatorname{argmin}_{u \in U} \{c(x, u) + V(f(x, u))\}$$

(whenever the min exists)

Numerical treatment

- ▶ assume $V \in \mathcal{F}$
- ▶ approximation space $\mathcal{A} \subset \mathcal{F}$, $\dim(\mathcal{A}) < \infty$
- ▶ projection $\Pi : \mathcal{F} \rightarrow \mathcal{A}$
- ▶ discretized Bellman operator

$$\Pi \circ L : \mathcal{A} \rightarrow \mathcal{A}$$

- ▶ value iteration: choose $V^{(0)} \in \mathcal{A}$ with $V^{(0)}(T) = 0$,

$$V^{(n+1)} := \Pi \circ L[V^{(n)}], \quad n = 0, 1, \dots$$

- ▶ typical \mathcal{A} : finite differences, finite elements (order p)
- ▶ problem: $\dim(\mathcal{A}) \sim \mathcal{O}(n^d)$ for error $\mathcal{O}(n^{-p})$

Nonlinear approximation

Theorem [Giroso, Anzellotti, '92]

If $f \in H^{s,2}(\mathbb{R}^d)$, $s > d/2$, we can find

- ▶ n coefficients $c_i \in \mathbb{R}$,
- ▶ n centers $x_i \in \mathbb{R}^d$,
- ▶ and n variances $\sigma_i > 0$ such that

$$\left\| f - \sum_{i=1}^n c_i e^{-\frac{\|x-x_i\|^2}{2\sigma_i^2}} \right\|_{\infty}^2 = \mathcal{O}(n^{-1}).$$

Scattered data interpolation

Problem

Given

- ▶ **sites** $X = \{x_1, \dots, x_N\} \subset \Omega \subset \mathbb{R}^d$
- ▶ **data** $f_1, \dots, f_N \in \mathbb{R}$,

find a function $a \in \mathcal{A}$ such that

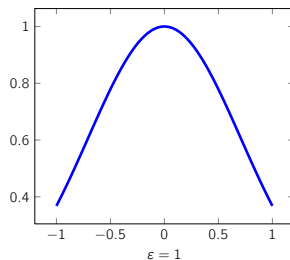
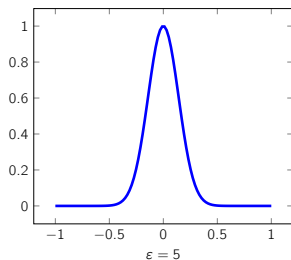
$$a(x_i) = f_i, \quad i = 1, \dots, N.$$

For $\mathcal{A} = \text{span}\{a_1, \dots, a_N\}$ we get

$$Ac = f, \quad \text{with } A_{ij} = a_j(x_i).$$

Radial basis functions

- ▶ **radial basis functions** $a(\cdot, x_j) = \varphi(\|\cdot - x_j\|_2)$
- ▶ examples:
 - ▶ Gaussian: $\varphi(r) = \exp(-r^2)$,
 - ▶ Wendland function: $\varphi(r) = (1 - r)_+^4 \cdot (4r + 1)$
- ▶ scaling: $a_j = a_j^\varepsilon = \varphi(\varepsilon\|\cdot - x_j\|)$



The Kruzkov transform

- ▶ problem: $V(x)$ increasing, but $\varphi(x)$ decreasing as $\|x\| \rightarrow \infty$
- ▶ Kruzkov transform: $V \mapsto \hat{V} = e^{-V(\cdot)}$
- ▶ Kruzkov-Bellman equation

$$\hat{V}(x) = \sup_{u \in U} \{e^{-c(x,u)} \cdot \hat{V}(f(x,u))\} =: \hat{L}[V](x), \quad x \in \Omega \setminus T$$

with boundary condition $\hat{V}(T) = 1$.

- ▶ under the assumption $c(x,u) \geq \delta > 0$ for $x \notin T$, the Kruzkov-Bellman operator \hat{L} is a contraction on L^∞ .

Dynamic programming using radial basis functions

- ▶ approximation space

$$\mathcal{A} = \mathcal{A}_{X,\varepsilon} = \text{span}\{\varphi(\varepsilon\|\cdot - x\|_2) : x \in X\}$$

- ▶ interpolation operator on X

$$\Pi : \mathcal{F} \rightarrow \mathcal{A}$$

- ▶ discretized Kruzkov-Bellman operator

$$\Pi \circ \hat{L} : \mathcal{A} \rightarrow \mathcal{A}$$

- ▶ value iteration: choose $\hat{V}^{(0)} \in \mathcal{A}$ with $\hat{V}^{(0)}(0) = 1$,

$$\hat{V}^{(n+1)} := \Pi \circ \hat{L}[\hat{V}^{(n)}], \quad n = 0, 1, \dots$$

Weighted least squares

Problem

Given

- ▶ **sites** $X = \{x_1, \dots, x_N\} \subset \Omega \subset \mathbb{R}^d$,
- ▶ **data** $f_1, \dots, f_N \in \mathbb{R}$,
- ▶ **approximation space** $\mathcal{A} = \text{span}\{a_1, \dots, a_m\}$, $m < N$,
- ▶ **weight function** $w : \Omega \rightarrow \mathbb{R}$ with associated scalar product $\langle f, g \rangle_w := \sum_{k=1}^N f(x_k)g(x_k)w(x_k)$ and induced norm

find a function $a \in \mathcal{A}$ such that

$$\|f - a\|_w \stackrel{!}{=} \min$$

Optimal coefficient vector c :

$$Gc = f_{\mathcal{A}}$$

with **Gram matrix** $G = (\langle a_i, a_j \rangle_w)_{ij}$ and $f_{\mathcal{A}} = (\langle f, a_j \rangle_w)_j$.

Moving least squares

Idea

In computing an approximation to the function $f : \Omega \rightarrow \mathbb{R}$ at $x \in \Omega$, only the values at sites $x_j \in X$ **close to x** should play a role.

- ▶ **moving** weight function $w : \Omega \times \Omega \rightarrow \mathbb{R}$
- ▶ $w(x, y)$ small for $\|x - y\|_2$ large
- ▶ inner product: $\langle f, g \rangle_{w(\cdot, x)} := \sum_{k=1}^N f(x_k)g(x_k)w(x_k, x)$
- ▶ **moving least squares approximation** a of data f is

$$a(x) = a^x(x),$$

where $a^x \in \mathcal{A}$ is minimizing $\|f - a^x\|_{w(\cdot, x)}$.

- ▶ given by solving the Gram system $G^x c^x = f_{\mathcal{A}}^x$

Shepard's method

D. Shepard, *A two dimensional interpolation function for irregularly spaced data*, Proc. 23rd Nat. Conf. ACM, 1968.

- ▶ simply choose $\mathcal{A} = \text{span}\{1\}$
- ▶ Gram matrix $G^x = \langle 1, 1 \rangle_{w(\cdot, x)} = \sum_{i=1}^N w(x_i, x)$
- ▶ right hand side $f_{\mathcal{A}}^x = \langle f, 1 \rangle_{w(\cdot, x)} = \sum_{i=1}^N f(x_i)w(x_i, x)$
- ▶ thus we get

$$c^x = f^x / G^x = \sum_{i=1}^N f(x_i) \underbrace{\frac{w(x_i, x)}{\sum_{i=1}^N w(x_i, x)}}_{=: a_i(x)}$$

- ▶ and so the **Shepard approximant** is

$$Sf(x) = c^x \cdot 1 = \sum_{i=1}^N f(x_i) a_i(x)$$

- ▶ advantage: Shepard approximation requires **no matrix solve**

Shepard discretization of the Bellman equation

- ▶ approximation space

$$\mathcal{A} = \text{span} \left\{ \frac{w(x_i, \cdot)}{\sum_{i=1}^N w(x_i, x)}, x_i \in X \right\}$$

- ▶ Shepard approximation operator

$$S : \mathcal{F} \rightarrow \mathcal{A}$$

- ▶ discretized Kruzkov-Bellman operator

$$S \circ \hat{L} : \mathcal{A} \rightarrow \mathcal{A}$$

- ▶ value iteration as usual

Convergence of the value iteration

- ▶ $f \mapsto Sf$ is linear,
- ▶ for each $x \in \Omega$, $Sf(x)$ is a convex combination of the values $f(x_1), \dots, f(x_n)$, therefore
- ▶ the Shepard operator $S : (L^\infty, \|\cdot\|_\infty) \rightarrow (\mathcal{A}, \|\cdot\|_\infty)$ has norm 1,

thus we get

Lemma

Value iteration with the discretized Kruzkov-Bellman operator $S \circ \hat{L} : (\mathcal{A}, \|\cdot\|_\infty) \rightarrow (\mathcal{A}, \|\cdot\|_\infty)$ converges to the unique fixed point of $S \circ \hat{L}$.

Convergence for fill distance $\rightarrow 0$

fill distance of $X \subset \Omega$

$$h = h(X, \Omega) = \sup_{x \in \Omega} \min_{x_j \in X} \|x - x_j\|_2$$

If $f : \Omega \rightarrow \mathbb{R}$ is Lipschitz continuous with constant L then

$$\|f - Sf\|_\infty \leq CLh$$

for some constant $C > 0$.

Convergence for fill distance $\rightarrow 0$

- ▶ sequence $(X_n)_n$ of nodes sets, $X_n \subset \Omega$, fill distances h_n , Shepard operators S_n ,
- ▶ $K < 1$ contraction constant of \hat{L} ,
- ▶ \hat{V} fixed point of \hat{L} , \hat{V}_n fixed point of $S_n \circ \hat{L}$

Theorem

If \hat{V} is Lipschitz continuous, then

$$\|\hat{V} - \hat{V}_n\|_\infty \leq \frac{CL}{1-K}h$$

Example 1

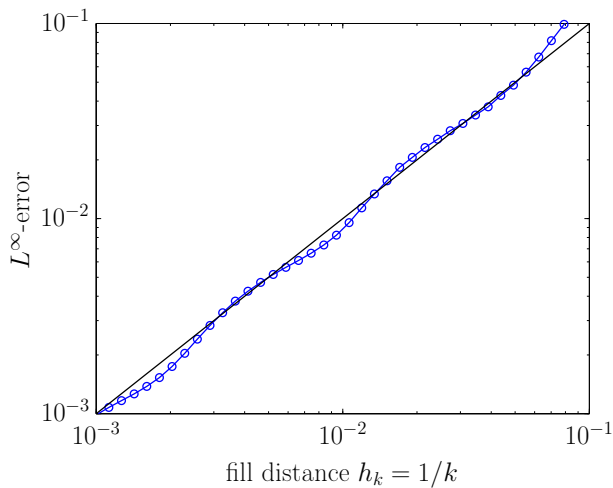
A simple 1D example

$$f(x, u) = aux, \quad c(x, u) = ax, \quad x \in [0, 1], u \in [-1, 1]$$

- ▶ optimal feedback $u(x) = -1$
- ▶ optimal value function $V(x) = x$
- ▶ nodes X_k equidistant with spacing $1/k$
- ▶ $T = [0, 1/(2k)]$
- ▶ $\tilde{U} = -1 : 0.1 : 1$
- ▶ ϕ^σ : Wendland function of order 4, $\sigma = k/5$

Example 1

A simple 1D example



Example 2

shortest path, geometrically complicated state constraints

We consider a boat in the mediteranian sea surrounding Greece which moves with constant speed 1, i.e.

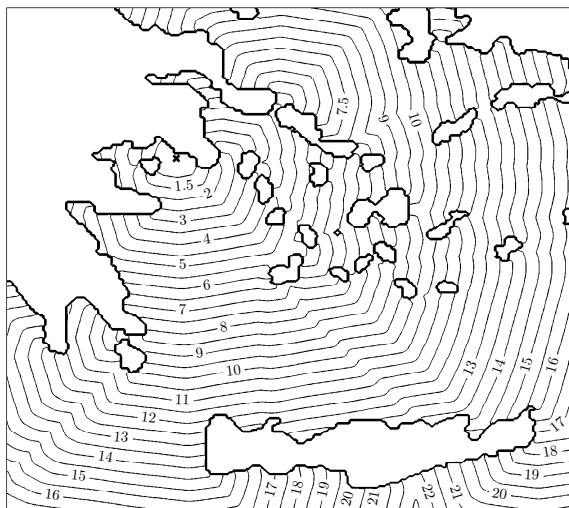
$$f(x, u) = x + hu, \quad c(x, u) \equiv 1, \quad x \in \text{neighborhood of Greece}$$

with time step $h = 0.1$ and $u \in \{u \in \mathbb{R}^2 : \|u\| = 1\}$

- ▶ T = neighborhood of the harbour of Athens
- ▶ X = equidistant in the sea on a 275×257 grid (50301 nodes)
- ▶ $\tilde{U} = \{\exp(2\pi ij/20) : j = 0, \dots, 19\}$
- ▶ ϕ^σ : Wendland function of order 4, $\sigma = 10$
- ▶ CPU time: 6 secs

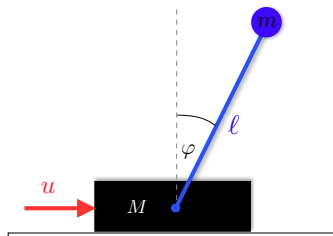
Example 2

shortest path, geometrically complicated state constraints



Example 3

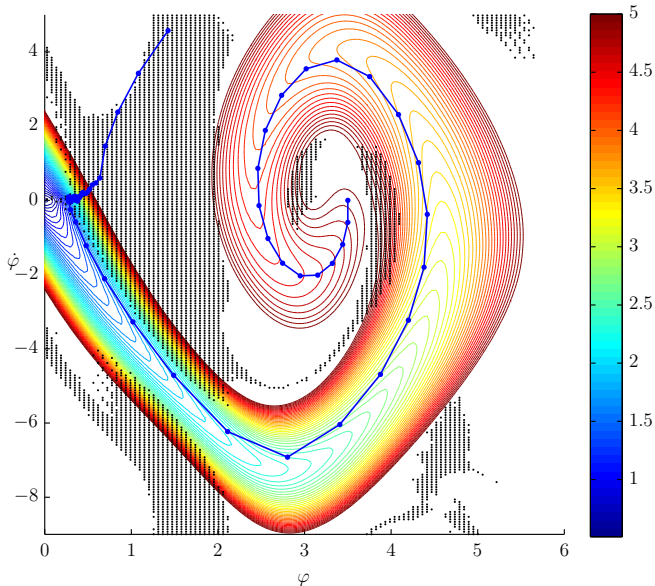
inverted pendulum, highly nonlinear dynamics



- ▶ f : equations of the forced pendulum
- ▶ c : quadratic deviation from the origin + quadratic in control
- ▶ T : neighborhood of the origin
- ▶ X : equidistant grid of 100×100 nodes
- ▶ $\tilde{U} = -128 : 8 : 128$
- ▶ ϕ^σ : Wendland function of order 4, $\sigma = 2.22$
- ▶ CPU time: 7 secs

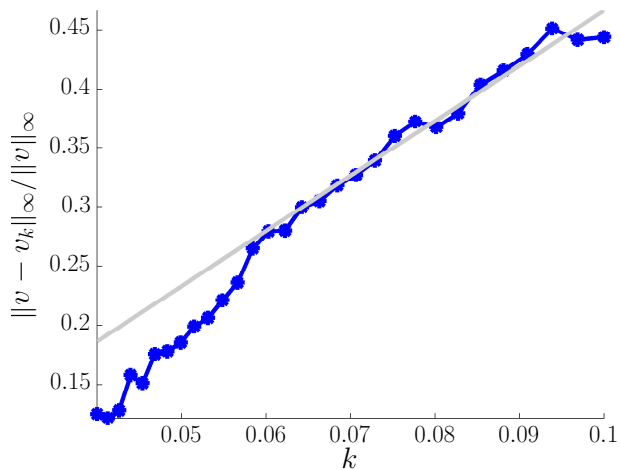
Example 3

inverted pendulum, highly nonlinear dynamics



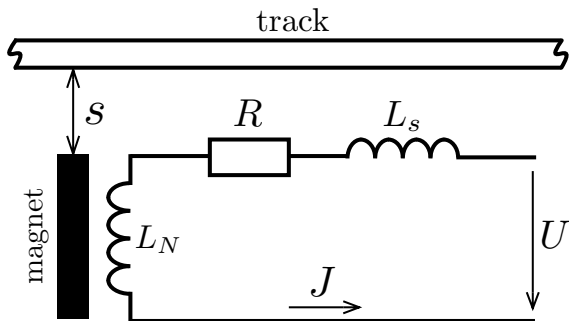
Example 3

inverted pendulum, highly nonlinear dynamics



Example 4

magnetic wheel, 3D example



Example 4

magnetic wheel, 3D example

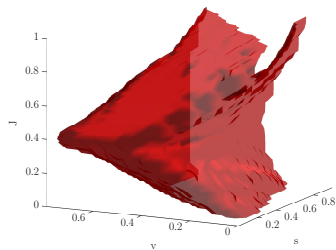
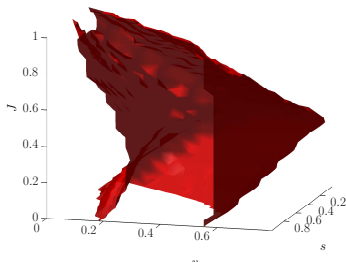
$$\begin{aligned}\dot{s} &= v, \\ \dot{v} &= \frac{CJ^2}{m_m 4s^2} - \mu g, \\ \dot{j} &= \frac{1}{L_s + \frac{C}{2s}} \left(-RJ + \frac{C}{2s^2} Jv + U \right),\end{aligned}$$

cost function c quadratic in s , v and u

- ▶ Ω : suitably chosen box
- ▶ $\tilde{U} = \{6 \cdot 10^3 u^3 \mid u \in \{-1, -0.99, \dots, 0.99, 1\}\}$
- ▶ T : neighborhood of the equilibrium $(0.01, 0, 17.155)$
- ▶ X : equidistant grid of $30 \times 30 \times 30$ nodes
- ▶ ϕ^σ : Wendland function of order 4, $\sigma = 11.2$
- ▶ CPU time: 60 secs

Example 4

magnetic wheel, 3D example



Matlab code template

```
f = @(x,u) ...
c = @(x,u) ...
phi = @(r) max(spones(r)-r,0).^4.*(4*r+spones(r));
T = [0 0]; v_T = 1;
shepard = @(A) spdiags(1./sum(A')',0,size(A,1),size(A,1))*A
S = [8,10];
L = 33; U = linspace(-128,128,L)';
N = 100; X1 = linspace(-1,1,N);
[XX,YY] = meshgrid(X1*S(1),X1*S(2)); X = [XX(:) YY(:)];
ep = 1/sqrt((4*prod(S)*20/N^2)/pi);
A = shepard(phi(ep*sdistm(f(X,U),[T;X],1/ep)));
C = c(X,U);

v = zeros(N^2+1,1); v0 = ones(size(v)); TOL = 1e-12;
while norm(v-v0,inf)/norm(v,inf) > TOL
    v0 = v;
    v = [v_T; max(reshape(C.*(A*v),L,N^2))'];
end
contour(...
```

Conclusion

Pros

- ▶ simple convergence theory
- ▶ simple implementation, independent of state dimension
- ▶ easy to incorporate complicated state constraints

Cons

- ▶ delicate choice of the shape parameter
- ▶ does not solve the curse of dimension ;-)