



Hermite kernel surrogates for the value function of high-dimensional nonlinear optimal control problems

Tobias Ehring¹ · Bernard Haasdonk¹

Received: 13 April 2023 / Accepted: 21 March 2024 / Published online: 29 April 2024
© The Author(s) 2024

Abstract

Numerical methods for the optimal feedback control of high-dimensional dynamical systems typically suffer from the curse of dimensionality. In the current presentation, we devise a mesh-free data-based approximation method for the value function of optimal control problems, which partially mitigates the dimensionality problem. The method is based on a greedy Hermite kernel interpolation scheme and incorporates context knowledge by its structure. Especially, the value function surrogate is elegantly enforced to be 0 in the target state, non-negative and constructed as a correction of a linearized model. The algorithm allows formulation in a matrix-free way which ensures efficient offline and online evaluation of the surrogate, circumventing the large-matrix problem for multivariate Hermite interpolation. Additionally, an incremental Cholesky factorization is utilized in the offline generation of the surrogate. For finite time horizons, both convergence of the surrogate to the value function and for the surrogate vs. the optimal controlled dynamical system are proven. Experiments support the effectiveness of the scheme, using among others a new academic model with an explicitly given value function. It may also be useful for the community to validate other optimal control approaches.

Keywords Optimal feedback control · Kernel methods · Hermite interpolation · Surrogate modeling

Mathematics Subject Classification (2010) 30C40 · 49N35 · 93B52 · 93C15 · 93C10

Communicated by: Tobias Breiten

✉ Tobias Ehring
ehringts@mathematik.uni-stuttgart.de

Bernard Haasdonk
haasdonk@mathematik.uni-stuttgart.de

¹ Institute of Applied Analysis and Numerical Simulation, University of Stuttgart, Pfaffenwaldring 57, Stuttgart 70569, Baden-Württemberg, Germany

1 Introduction

The feedback control of dynamical systems is an important type of automation that is indispensable in modern industrial plants, for example. Furthermore, there are applications in which the control must be optimal to minimize energy consumption or costs. An optimal feedback rule can be based on a mathematical model, an optimal control problem (OCP), which shall be in our consideration of the following form:

$$\min_{\mathbf{u} \in \mathcal{U}_\infty} J(\mathbf{u}) = \min_{\mathbf{u} \in \mathcal{U}_\infty} \int_0^\infty r(\mathbf{x}(s)) + \mathbf{u}(s)^\top R \mathbf{u}(s) \, ds \quad (1)$$

$$\text{subject to } \dot{\mathbf{x}}(s) = f(\mathbf{x}(s)) + g(\mathbf{x}(s))\mathbf{u}(s) \text{ and } \mathbf{x}(0) = x_0 \in \mathbb{R}^N \quad (2)$$

with $\mathcal{U}_\infty := \{\mathbf{u} : [0, \infty) \rightarrow \mathbb{R}^M \mid \mathbf{u} \text{ measurable}\}$ being the space of all possible control signals. Note that we indicate time-dependent scalar or vectorial functions with bold symbols to distinguish them from time-independent quantities. The OCP has an infinite time horizon, which is typical, for example, in mathematical economics when considering economic sustainability or economic growth [1]. Moreover, the dynamics of the trajectory $\mathbf{x} : [0, \infty) \rightarrow \mathbb{R}^N$ is affine with respect to the control signal \mathbf{u} . This is common when the ordinary differential equation (ODE) results from a semi-discretized partial differential equation (PDE) where the controller could be an external force. The function $J : \mathcal{U}_\infty \rightarrow \mathbb{R}_+ \cup \{\infty\}$ is the cost function, which determines the goal of optimal control. It consists of the integral over the running payoff $r \in C(\mathbb{R}^N, \mathbb{R}_+)$ and a quadratic control term, involving a positive definite matrix $R \in \mathbb{R}^{M \times M}$. It is assumed that the preimage of r for 0 is only 0, i.e., $r^{-1}(\{0\}) = \{0\}$ and $f(0) = 0$, leading to the desired target position in the zero state. Furthermore, as the constraints in (2) determine the solution of the dynamical system for a given \mathbf{u} , the cost function depends only on \mathbf{u} . The optimal process of the infinite horizon OCP is denoted by $(\mathbf{x}^*, \mathbf{u}^*)$. For a compact set of initial states $\mathcal{A} \subset \mathbb{R}^N$, it is assumed throughout the paper that for every $x_0 \in \mathcal{A}$, there is an optimal process. Existence results for infinite horizon OCP can be found in [2].

Two possibilities for solving (1)–(2) are given by open-loop and closed-loop control. In an open-loop process, the optimal signal is determined in advance as a function $\mathbf{u} : [0, \infty) \rightarrow \mathbb{R}^M$, and no further adaptation of the signal to the current state is possible in an ongoing process, making it unsuitable for many real-world applications where unpredictable disturbances may occur. This requires a closed-loop that maps the current state to the current optimal control signal, which is in fact a much more general tool since it can be used to solve (1)–(2) for many initial states. The optimal feedback rule is denoted by $\mathcal{K}(\cdot; \nabla v(\cdot)) : \mathbb{R}^N \rightarrow \mathbb{R}^M$. As the notation indicates, the optimal feedback policy depends on the gradient of a function v called the value function (VF). It gives the optimal costs-to-go depending on an initial state. Further, if it is available, the problem of optimal feedback control is solved and the OCP reduces to an ODE. This follows by setting $\mathbf{u}(s) = \mathcal{K}(\mathbf{x}(s); \nabla v(\mathbf{x}(s)))$ and inserting it into (2). From Bellman's dynamic programming principle (DPP), it can be deduced that the VF is the solution of the PDE called the Hamilton-Jacobi-Bellman (HJB) equation. However, the problem with solving the HJB equation is that classical numerical solution algorithms for PDEs,

such as finite differences or finite elements, suffer from the curse of dimensionality [3], making such approaches infeasible for dimensions $N > 6$. Since the ODE in the OCP often originates from a semi-discretized PDE and therefore has a high state dimension, much research has been pursued in recent years to find strategies to overcome this difficulty.

In many methods, a semi-Lagrangian scheme [4] is applied to the HJB equation. Advanced techniques such as spatially adaptive sparse grid [5] are used to avoid an impractical full grid. In the finite horizon case, using a tree structure [6, 7], a reasonable discrete domain can be constructed utilizing the underlying dynamics. This approach can also be combined with model order reduction (MOR) [8]. In the infinite horizon case, from the semi-Lagrangian point of view, but also when (1)–(2) is discretized in time, an iterative fixed-point scheme called value iteration (VI) [9] can be motivated to solve the HJB. Convergence is guaranteed for this method [10], but it comes at a high computational cost and slow convergence, since the contraction constant tends to 1 as the mesh becomes finer. Most methods based on the VI differ in their ansatz to the approximant of the VF. In [11], Shepard's moving least squares approximation method using radial basis functions is utilized to generate an interpolant. A neural network can also be used for this as in [12] and the references therein. Additional references and discussion can be found in [13].

Another notable iterative scheme based on the HJB equation is the policy iteration (PI) [14]. This requires an approximation step for the numerical solvability of the generalized linear HJB equation. In this approximation step, most PI-based methods differ. In [15–17], this is a Galerkin projection of the residual equation using a certain polynomial basis, which is enabled by assuming separability of the data and mitigates the curse of dimensionality. A semi-Lagrangian scheme can also perform the approximation step, as in [18], where MOR then allows applicability to high-dimensional problems. In a finite horizon scenario, the VF, which is then time-dependent, can be approximated for discrete time instances in a tensor train format using a PI on small time intervals [19, 20]. Here, the approximate VF of the previously computed time step is included in the PI to compute the approximate VF of the currently considered time step. To obtain the latter, regression is performed in [19] or the Dirac-Frenkel variational principle is used in [20]. Generally, for the PI, however, convergence is only guaranteed for the stabilizing initial solution of the VF [21], which makes the combination of VI and PI interesting [22].

Furthermore, there are many techniques for generating suboptimal feedback control, such as model predictive control (MPC), which is a very relevant feedback design (see [23] for an introduction). Another common feedback controller can be provided by the linearization of the problem and the solution of the associated algebraic Riccati equation (ARE) [24]. For parameter-dependent problems, this can also be combined with reduced basis methods (see [25]), which is interesting for multi-query scenarios. However, under controllability assumptions, this only stabilizes in a certain neighborhood around the target position. A truncated Taylor expansion can also replace the VF, with the HJB equation and certain regularity assumptions leading to generalized

Lyapunov equations for the higher-order terms [26]. Moreover, it is also possible to solve a state-dependent Riccati equation in each step, provided that the dynamics satisfies the structure $f(x) = A(x)x$ (see [27] for a survey). Here, many AREs have to be solved online, which is costly. In [28], this is avoided by solving many AREs for a set of training states and then fitting a neural network to this data to provide efficient online feedback control.

Furthermore, in the case of finite [29] and infinite horizons [30], the VF and its gradient can be directly approximated by a NN that minimizes an appropriate constrained optimization problem involving trajectories starting from a compact set of initial states. The obtained approximation of the VF then yields an optimal feedback control only on the set of optimal trajectories starting from this set of initial states. Therefore, it is called semi-global. In this respect, this approach is very similar to ours, but differs mainly in the approximation ansatz for the VF and in the way the infinite time horizon is treated. A similar optimization approach to those just discussed is [31] for the finite horizon case and [32] for the infinite horizon case, where instead of a NN, a sparse polynomial ansatz is chosen for the approximate VF. Both papers give a rigorous convergence analysis of their approaches. Furthermore, in [32], the approximate VF is forced to satisfy some known properties of the true VF by an appropriate choice of the ansatz space. We will proceed like this in the current work as well, but go a step further in incorporating structure.

The majority of the techniques mentioned above are based on the DPP. However, there is another class of techniques that are data-driven, including our method. They often rely on an indirect approach that was developed independently of Bellman and goes back to L. S. Pontryagin [34]. There, the first-order necessary condition of the OCP leads to a two-point boundary value problem called Pontryagin's maximum principle (PMP), among whose solutions the optimal solution can be found. This procedure only generates the open-loop solution for one initial state, but with the well-known relation between DPP and PMP, information about the VF and its gradient is provided along this optimal trajectory. Thus, by solving many open-loop controls for different initial states, which can be done in parallel (causal-free) in an offline phase, it is possible to generate a data set that can be used to approximate the VF. An approximate feedback control can then be generated online using a surrogate.

In [35], a neural network is used and the initial states for data generation are first chosen randomly. Later, the data is enriched adaptively based on the current surrogate. Without the latter, a similar approach is taken in [36], where a hyperbolic cross polynomial model is fitted to the VF. Polynomial interpolation is also used in [37], where the VF data is computed on a sparse grid.

The current presentation is based on the idea in [38] and extends [39, 40]. In our approach, the domain of interest is explored by optimal trajectories starting from a problem-dependent set of initial states. This avoids having to specify a domain of possible states, which is generally very difficult. Instead, a hypercube is often used for this purpose, which is a very crude choice since it can contain physically meaningless states in the sense that values in neighboring nodes of the underlying semi-discretized PDE may be uncorrelated. Furthermore, in our data generation method, the selected initial states are chosen adaptively with the advantage that previously computed data

can be used to find new, promising initial states for solving the most informative next open-loop process. This avoids redundant data and keeps it sparse. Hermite kernel interpolation techniques are then applied to this data set to obtain a surrogate that exploits all the available information of the VF. Here, an adapted vectorial kernel orthogonal greedy algorithm (VKOGA) [41] is used to select the interpolation points. In general, these kernel techniques are more robust to the curse of dimensionality, since under certain conditions the convergence rate with respect to the number of interpolation points is partially independent of the dimension [42]. Moreover, they have the advantage of being grid-free and therefore are a perfect choice for scattered data. One difficulty that arises in VKOGA when considering Hermite kernel interpolation is that the surrogate must be evaluated on a big number of inputs, resulting in an extremely large system matrix that cannot be stored or processed. Using a special class of kernels, we overcome this with a matrix-free strategy that can also be used for fast online evaluation of the surrogate on many inputs. Under the (strong) assumption that the VF belongs to the reproducing kernel Hilbert space (RKHS) for the considered kernel, we show the convergence of the surrogate to the VF as well as the convergence of the surrogate-controlled solution to the optimal solution. Later in the numerical experiment, the RKHS will be a Sobolev space, and the requirement that the VF must be in this space is satisfied by assuming that the VF is sufficiently smooth.

Instead of directly approximating the optimal feedback control function, constructing a surrogate for the VF has the advantage that we can incorporate the known structure of the VF and perform a Hermite interpolation using all available data, which may result in a better fit to the underlying true function and thus a better generalization.

The paper is organized as follows. Section 2 provides a brief background on optimal control. Then, in Section 3, the Hermite kernel interpolation is introduced. Additionally, the matrix-free strategy and the construction of the surrogate that satisfies certain properties of the VF are explained. In Section 4, the convergence proofs are given. We perform some numerical tests in Section 5. Finally, conclusions and an outlook are presented in Section 6.

2 Background on optimal control

Optimal closed-loop control

The core of optimal feedback control is the value function, mentioned in the introduction and explicitly given by

$$v(x) = \min_{\mathbf{u} \in \mathcal{U}_\infty} \int_0^\infty r(\mathbf{x}(s)) + \mathbf{u}(s)^\top R \mathbf{u}(s) ds$$

subject to $\dot{\mathbf{x}}(s) = f(\mathbf{x}(s)) + g(\mathbf{x}(s))\mathbf{u}(s)$ and $\mathbf{x}(0) = x$.

Since the initial state is a free variable of the VF, it is denoted by x . Furthermore, in the following, we explicitly indicate the dependence of the trajectory on the initial

state x_0 by $\mathbf{x}(\cdot; x_0)$, if this is not obvious from the context. The VF satisfies Bellman’s optimality principle, which states that any subtrajectory to the target state of an optimal trajectory is also optimal. This is reflected by

$$v(x) = \min_{\mathbf{u} \in \mathcal{U}_\infty} \left\{ \int_0^t r(\mathbf{x}(s; x)) + \mathbf{u}(s; x)^\top \mathbf{R} \mathbf{u}(s; x) \, ds + v(\mathbf{x}(t; x)) \right\}. \tag{3}$$

Next, a precise definition of the domain of interest is given

$$\begin{aligned} \mathcal{T}(\mathcal{A}) := \text{clos} \{ & \mathbf{x}^*(t; x_0) \in \mathbb{R}^N \mid (\mathbf{x}^*(t; x_0), \mathbf{u}^*(t; x_0)) \\ & \text{solves (1)–(2) } \forall x_0 \in \mathcal{A} \text{ and } \forall t \in [0, \infty) \}, \end{aligned}$$

which is the closed set of the states of all optimal trajectories with the initial state in the compact set \mathcal{A} . Here, clos denotes the closure. The main reason why we can mitigate the curse of dimensionality with our data-based approach is that we are only interested in feedback control on $\mathcal{T}(\mathcal{A})$. The latter can be a much lower dimensional manifold since it only contains meaningful states that can occur under optimal control. This is because, for example, values in neighboring nodes of the underlying semi-discrete PDE are typically correlated. Next, we introduce a property that we assume the VF has. A function v is radially unbounded if it satisfies

$$v(x) \longrightarrow \infty \text{ for all } \|x\|_2 \longrightarrow \infty.$$

If this property and the additional continuity on \mathcal{A} hold for the VF, then the set $\mathcal{T}(\mathcal{A})$ is compact:

Lemma 1 *Let $\mathcal{A} \subset \mathbb{R}^N$ be a non-empty compact set. Let further, v be radially unbounded and $v \in C(\mathcal{A})$, then the set $\mathcal{T}(\mathcal{A})$ is bounded.*

Proof We prove this by contradiction. Assume $\mathcal{T}(\mathcal{A})$ is unbounded, and so there exists a sequence $(t_k, x_{0,k})_{k \in \mathbb{N}} \subset [0, \infty) \times \mathcal{A}$ such that

$$\|\mathbf{x}^*(t_k; x_{0,k})\|_2 \longrightarrow \infty.$$

Since the set \mathcal{A} is compact, the continuous VF assumes its maximum on it, i.e., there is a $M \in \mathbb{R}_+$ with

$$M \geq v(x) \text{ for all } x \in \mathcal{A}.$$

Then, Bellman’s principle of optimality leads to

$$\begin{aligned} M \geq v(\mathbf{x}^*(0; x_{0,k})) &= \int_0^{t_k} r(\mathbf{x}^*(s; x_{0,k}), \mathbf{u}^*(s; x_{0,k})) \, ds + v(\mathbf{x}^*(t_k; x_{0,k})) \\ &\geq v(\mathbf{x}^*(t_k; x_{0,k})), \end{aligned}$$

which contradicts the assumption that $v(\mathbf{x}^*(t_k; x_{0,k})) \longrightarrow \infty$ for $k \longrightarrow \infty$, as $\|\mathbf{x}^*(t_k; x_{0,k})\|_2 \longrightarrow \infty$ for $k \longrightarrow \infty$. □

The main assumption of the present study is that the OCP has a VF that fulfills the following assumptions, including those of the latter lemma:

Assumption 1 *The VF v satisfies $v \in C^1(\mathcal{T}(\mathcal{A}))$ and is radially unbounded.*

A comprehensive discussion and conditions for regularity of the VF in the considered infinite horizon setting can be found in [43, 44] and [30, Theorem C.8]. A characterization of the VF as a solution of a PDE can be obtained via (3) by rearranging, dividing by t , and letting t go to 0. It follows

$$\min_{u \in \mathbb{R}^M} \left\{ (f(x) + g(x)u) \cdot \nabla_x v(x) + r(x) + u^\top R u \right\} = 0 \tag{4}$$

for at least all $x \in \mathcal{T}(\mathcal{A})$, which is the famous HJB equation. Due to the structure of the considered problem, the local minimization in (4) can be solved exactly. This yields

$$\nabla v(x)^\top f(x) - \frac{1}{4} \nabla v(x)^\top g(x) R^{-1} g(x)^\top \nabla v(x) + r(x) = 0,$$

where the minimizer is

$$\mathcal{K}(x; \nabla v) := u(x) = -\frac{1}{2} R^{-1} g(x)^\top \nabla v(x). \tag{5}$$

The optimal feedback rule is also given by the latter equation. So if ∇v is known, it can be used to obtain the optimal feedback policy. However, solving the HJB numerically in higher dimensions is not possible with classical numerical solution algorithms, as already discussed. Therefore, in our approach, a data-based surrogate for the VF is constructed using the available data from optimal open-loop controlled trajectories. Since we obtain feedback control only on $\mathcal{T}(\mathcal{A})$ with the resulting VF surrogate, it can be considered semi-global [30].

Optimal open-loop control

There are two conceptually different ways of solving an open-loop control problem [33]. The first is the class of direct methods, where the problem is transformed into a nonlinear finite-dimensional minimization problem by discretization. This can be seen as “first discretize, then optimize.” However, this does not provide any additional information about the system, in particular about the gradient of the VF. This requires indirect methods using PMP-like conditions, which can be understood as “first optimize, then discretize.” Since we are interested in this additional information for the Hermite interpolation, we present the indirect ansatz below. For what follows, we need another mild assumption. This is a summary and adaptation of Assumptions (A0) and (A1) in [45].

Assumption 2 Let $r, f, g \in C^1(\mathbb{R}^N)$ and let further for all $x_0 \in \mathcal{A}$ exist a continuous function $\varphi_1 : [0, 1) \rightarrow (0, 1)$ and a locally integrable function $\varphi_2 : [0, 1) \rightarrow \mathbb{R}$, such that for almost every $t \geq 0$ it holds

$$\max_{\bar{x} : \|\bar{x} - \mathbf{x}^*(t)\|_2 \leq \varphi_1(t)} \left\{ \sum_{i=1}^N \|\nabla_x (f(\bar{x}) + g(\bar{x})\mathbf{u}^*(t))\|_i + \|\nabla_x r(\bar{x})\|_2 \right\} \leq \varphi_2(t)$$

with $(\mathbf{x}^*, \mathbf{u}^*)$ being the optimal process corresponding to x_0 .

This makes it possible to specify the necessary conditions that an optimal process must fulfill:

Theorem 1 Let Assumption 2 be satisfied and let further $(\mathbf{x}^*, \mathbf{u}^*)$ be an optimal process, then there is a $q_0 \geq 0$ and locally absolutely continuous co-state $\mathbf{p}^* : [0, \infty) \rightarrow \mathbb{R}^N$ such that

$$\dot{\mathbf{x}}^*(s) = \nabla_p H(\mathbf{x}^*(s), q_0, \mathbf{p}^*(s), \mathbf{u}^*(s)), \quad \mathbf{x}^*(0) = x, \tag{6}$$

$$\dot{\mathbf{p}}^*(s) = -\nabla_x H(\mathbf{x}^*(s), q_0, \mathbf{p}^*(s), \mathbf{u}^*(s)), \tag{7}$$

$$\mathbf{u}^*(s) = \arg \min_{u \in \mathbb{R}^M} \{H(\mathbf{x}^*(s), q_0, \mathbf{p}^*(s), u)\} \tag{8}$$

are valid for $s \in [0, \infty)$, where $H(x, q, p, u) = (f(x) + g(x)u) \cdot p + q(r(x) + u^\top Ru)$ is the control Hamiltonian.

The proof for this theorem can be found in [45]. These equations are the core conditions of the PMP, since they determine the dynamics of the system. We extend them by an additional transversality condition

$$\lim_{t \rightarrow \infty} \mathbf{p}^*(t) = 0. \tag{9}$$

to further restrict the solution space. It is motivated by the finite horizon OCP, where the PMP equations are given by the upper core condition (6)–(8) with the additional transversality condition $\mathbf{p}_T^*(T) = 0$. Here, $T > 0$ is the finite time horizon. However, under the current assumption, (9) need not be satisfied as T goes to infinity, as the famous Halkin counterexample shows. Therefore, we pose another assumption and refer the reader to [45, 46] where additional conditions implying (9) are given.

Assumption 3 Let the co-state satisfy (9) and let $q_0 = 1$.

In the latter, we also include the assumption that we do not consider abnormal cases, i.e., $q_0 = 0$. In all other cases, we have, without loss of generality, that $q_0 = 1$ (see [45] for instance). Overall, under Assumptions 2 and 3, the PMP conditions for the infinite horizon OCP are given by

$$\dot{\mathbf{x}}^*(s) = \nabla_p H(\mathbf{x}^*(s), \mathbf{p}^*(s), \mathbf{u}^*(s)), \quad \mathbf{x}^*(0) = x, \tag{10}$$

$$\dot{\mathbf{p}}^*(s) = -\nabla_x H(\mathbf{x}^*(s), \mathbf{p}^*(s), \mathbf{u}^*(s)), \quad \lim_{t \rightarrow \infty} \mathbf{p}^*(t) =: \mathbf{p}^*(\infty) = 0, \tag{11}$$

$$\mathbf{u}^*(s) = \arg \min_{u \in \mathbb{R}^M} \{H(\mathbf{x}^*(s), \mathbf{p}^*(s), u)\} \tag{12}$$

for $s \in [0, \infty)$ and with the abbreviation $H(x, p, u) := H(x, q = 1, p, u) = (f(x) + g(x)u) \cdot p + r(x) + u^\top Ru$. With respect to our data-based Hermite interpolation approach, the next well-known relationship (see [43, 45, 46]) between the co-state and the gradient of the VF is very important:

$$\mathbf{p}^*(s) = \nabla_x v(\mathbf{x}^*(s)) \tag{13}$$

It is valid with Assumption 1. One can also determine the values of the VF along the optimal trajectories by an additional function $\mathbf{v}^* : [0, \infty) \rightarrow \mathbb{R}_+$ which satisfies

$$\dot{\mathbf{v}}^*(s) = -r(\mathbf{x}^*(s)) - \mathbf{u}^*(s)^\top R \mathbf{u}^*(s) \text{ with } \lim_{t \rightarrow \infty} \mathbf{v}^*(t) =: \mathbf{v}^*(\infty) = 0. \tag{14}$$

Note that $\mathbf{v}^*(t) = v(\mathbf{x}^*(t))$. The conditions in (10)–(12) and (14) can be formulated as an inhomogeneous Dirichlet two-point boundary value problem, among whose solutions the optimal one can be found. The vector

$$\mathbf{z}^*(s) := [\mathbf{x}^*(s)^\top \mathbf{p}^*(s)^\top \mathbf{v}^*(s)]^\top$$

is introduced for this purpose and the right-hand side

$$F(\mathbf{z}^*) := \begin{bmatrix} \nabla_p H(\mathbf{x}^*, \mathbf{p}^*, \mathbf{u}^*(\mathbf{x}^*, \mathbf{p}^*)) \\ -\nabla_x H(\mathbf{x}^*, \mathbf{p}^*, \mathbf{u}^*(\mathbf{x}^*, \mathbf{p}^*)) \\ -r(\mathbf{x}^*) - (\mathbf{u}^*(\mathbf{x}^*, \mathbf{p}^*))^\top R (\mathbf{u}^*(\mathbf{x}^*, \mathbf{p}^*)) \end{bmatrix}$$

with $\mathbf{u}^*(\mathbf{x}^*(s), \mathbf{p}^*(s)) := \arg \min_{u \in \mathbb{R}^M} \{H(\mathbf{x}^*(s), \mathbf{p}^*(s), u)\}$. The boundary conditions in (10), (11) and (14) can be compactly represented by

$$b(\mathbf{z}^*(0), \mathbf{z}^*(\infty)) := \begin{bmatrix} I_N & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \mathbf{z}^*(0) + \begin{bmatrix} 0 & 0 & 0 \\ 0 & I_N & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{z}^*(\infty) - \begin{bmatrix} x_0 \\ 0 \\ 0 \end{bmatrix} = 0.$$

Here, $I_N \in \mathbb{R}^{N \times N}$ is the $N \times N$ identity matrix. In total,

$$\dot{\mathbf{z}}^*(s) = F(\mathbf{z}^*(s)) \text{ with } b(\mathbf{z}^*(0), \mathbf{z}^*(\infty)) = 0, \tag{15}$$

needs to be solved, where the infinity part of the boundary condition has to be understood in the limit sense. Note that (15) are necessary conditions, i.e., they only provide candidates for the optimal solution. Depending on the problem, this has to be taken into account when working with this PMP-based approach. Later in the numerical experiments, we will see that for the considered model problems, the satisfaction of the HJB on the computed trajectories is extremely good, which is a strong indication that we are computing the optimal trajectories. Conditions under which (15) is also sufficient to deliver an optimal solution can be found in [47].

With the basic idea of [48], such problems with an infinite time horizon can be treated numerically. That article also contains a list of references to other works that

analyze this technique theoretically. We refrain from such considerations in the current work. In general, the technique is to map the interval $[0, 1)$ to the interval $[0, \infty)$ using a function such as $\Phi(\tau) = \frac{\tau}{1-\tau}$. The transformed system $\bar{\mathbf{z}}^*(\tau) := \mathbf{z}(\Phi(\tau))$ on the finite interval $[0, 1)$ with the new time variable τ is then

$$\dot{\bar{\mathbf{z}}}^*(\tau) = \Phi'(\tau)F(\bar{\mathbf{z}}^*(\tau)) \text{ with } b(\bar{\mathbf{z}}^*(0), \bar{\mathbf{z}}^*(1)) = 0. \tag{16}$$

However, even for this finite time system, solvers for inhomogeneous Dirichlet two-point boundary value problems such as *bvp5c* from Matlab or *solve_bvp* from SciPy cannot be used. These evaluate the right-hand side of (16) at the singularity $\tau = 1$. Similar to [49], this can be avoided by using one step of an explicit Euler method. An approximation to the solution $\bar{\mathbf{z}}^*(1)$ on the right boundary can be generated as follows:

$$\bar{\mathbf{z}}^*(1) \approx \bar{\mathbf{z}}^*(1 - \Delta\tau) + \Delta\tau\Phi'(1 - \Delta\tau)F(\bar{\mathbf{z}}^*(1 - \Delta\tau)). \tag{17}$$

The system which is now on the finite time interval $[0, 1 - \Delta\tau]$ without singularity problems and approximates (15) is

$$\begin{aligned} \dot{\bar{\mathbf{z}}}^*(\tau) &= \Phi'(\tau)F(\bar{\mathbf{z}}^*(\tau)) & (18) \\ \text{with } b(\bar{\mathbf{z}}^*(0), \bar{\mathbf{z}}^*(1 - \Delta\tau) + \Delta\tau\Phi'(1 - \Delta\tau)F(\bar{\mathbf{z}}^*(1 - \Delta\tau))) &= 0. & (19) \end{aligned}$$

Overall, solving (18)–(19) allows information to be gathered about both the VF and its gradient along optimal trajectories. Here, it does not matter whether we consider the data on the transformed time interval or not, since the original system is autonomous and therefore the time values are not included in the construction of the surrogate. At this point, however, it is not yet clear which optimal trajectories are chosen. This is realized using the greedy exploration algorithm that we introduced in [40] which builds a data set $\mathcal{D} \subset \mathcal{T}(\mathcal{A})$ that approximates $\mathcal{T}(\mathcal{A})$. Subsequently, on \mathcal{D} a feedback rule is generated. The idea of this algorithm is to compute the optimal open-loop trajectories only for particularly promising initial states of \mathcal{A} in order to expand the entire data set. It is given by the Algorithm 1.

Algorithm 1 Greedy exploration algorithm.

Require: $\mathcal{A}, \mathcal{X} := \emptyset, \mathcal{D} = \{0\}$ and $\epsilon_{\text{tol,d}}$

- 1: $\epsilon := \epsilon_{\text{tol,d}} + 1$
- 2: **while** $\epsilon > \epsilon_{\text{tol,d}}$ **do**
- 3: $x_a := \arg \max_{x \in \mathcal{A}} \min_{y \in \mathcal{D}} \|x - y\|_2$
- 4: $\epsilon := \min_{y \in \mathcal{D}} \|x_a - y\|_2$
- 5: $[\mathbf{x}(\cdot; x_a), v(\mathbf{x}(\cdot; x_a)), \nabla v(\mathbf{x}(\cdot; x_a))] := \text{solveOpenLoop}(x_a)$
- 6: $\mathcal{D} := \mathcal{D} \cup \{\mathbf{x}(\cdot; x_a)\}$
- 7: **end while**

The rule in step 3 is used to find new starting positions x_a . The `solveOpenLoop` function in step 5 outputs the optimal trajectory and the corresponding value of the VF and its gradient for the initial state x_a . The data set \mathcal{D} is then expanded by that trajectory in step 6. This process is performed until the cover distance in step 4 is smaller than a given tolerance $\epsilon_{\text{tol,d}}$. Here, the quantity ϵ behaves like $n^{-\frac{1}{N}}$ in the

worst case, where n is the number of iterations [50]. This is due to the fact that the fill distance is an upper bound of the cover distance. However, since we choose \mathcal{A} as a set of lower dimensional structures with physically meaningful initial states, we expect this rate to be much better, as in [40]. The outputs of this algorithm are the sets \mathcal{D} , $v(\mathcal{D})$ and $\nabla v(\mathcal{D})$. In the numerical implementation, the set of initial states \mathcal{A} will be discrete and \mathcal{D} will consist of time-discrete trajectories that we get from the two-point boundary value problem solver we use. Step 3 can be implemented efficiently by storing a list of minimum distances from each element in \mathcal{A} to the current set \mathcal{D} . When a new trajectory is added to the set \mathcal{D} , only this list needs to be updated by considering the distance of the elements of \mathcal{A} to the new trajectory. Then, the maximum of the list is the desired result. Thus, step 3 is in the complexity of the number of elements in \mathcal{A} times the number of elements of a new trajectory.

3 On Hermite kernel surrogates

We recall some basic information about kernels. A symmetric function $k : \Omega \times \Omega \rightarrow \mathbb{R}$ for a non-empty set Ω is called a kernel. If for each finite pairwise distinct set $X_n := \{x_1, \dots, x_n\} \subset \Omega$, the Gram matrix $(\mathcal{K}_{X_n})_{i,j} = k(x_i, x_j)$ is positive semidefinite, then the kernel is called positive definite (p.d.), and if all such Gram matrices are positive definite, then the kernel is called strictly positive definite (s.p.d.). Obviously, all s.p.d kernels are also p.d kernels, the latter being of particular interest since each of them can be associated with a unique reproducing kernel Hilbert space $\mathcal{H}_k(\Omega)$ (RKHS), where k is the corresponding reproducing kernel. A RKHS is a Hilbert space of functions $f : \Omega \rightarrow \mathbb{R}$ with the property that there exists a kernel $k : \Omega \times \Omega \rightarrow \mathbb{R}$, such that $k(x, \cdot) \in \mathcal{H}_k(\Omega)$ for all $x \in \Omega$ and $\langle f, k(x, \cdot) \rangle_{\mathcal{H}_k(\Omega)} = f(x)$ for all $f \in \mathcal{H}_k(\Omega)$. The latter is called reproducing property. An introduction can be found in [51, Chapter 10].

Hermite kernel interpolation

We continue with the introduction of generalized kernel interpolation as discussed in [51, Chapter 16], which also covers the Hermite kernel interpolation. For a p.d. kernel, the interpolation conditions for an unknown interpolant $s_f^n \in \mathcal{H}_k(\Omega)$ in a generalized setting are given by

$$\lambda_i(s_f^n) = f_i, \tag{20}$$

where $\lambda_1, \dots, \lambda_n \in \mathcal{H}_k(\Omega)'$ are linear functionals contained in the dual space of the RKHS and $f_1, \dots, f_n \in \mathbb{R}$ are some target values. From Theorem 16.1 in [51], it follows that (20) can be solved uniquely if the $\lambda_1, \dots, \lambda_n \in \mathcal{H}_k(\Omega)'$ are linearly independent. Furthermore, the interpolant is given by

$$s_f^n = \sum_{i=1}^n \alpha_i v_i,$$

where $v_i \in \mathcal{H}_k(\Omega)$ is the Riesz representer of λ_i and the coefficients $\alpha_1, \dots, \alpha_n \in \mathbb{R}$ are determined via the interpolation condition in (20). The considered Hermite

interpolation problem is of the following type: It is assumed that for a set $\mathcal{D} \subset \Omega$, the values of the VF $v(\mathcal{D})$ and of its gradient $\nabla v(\mathcal{D})$ are given. For a p.d. kernel $k \in C^2(\Omega \times \Omega)$ and a finite pairwise distinct set $X_n = \{x_1, \dots, x_n\} \subset \mathcal{D}$, the goal is to find a surrogate such that

$$\partial^a s_v^n(x_i) = \delta_{x_i} \partial^a s_v^n = \delta_{x_i} \partial^a v = \partial^a v(x_i) \tag{21}$$

for all multiindices $a \in \mathbb{N}_0^N$ with $\sum_{l=1}^N (a)_l \leq 1$ and $i = 1, \dots, n$. Note that due to the assumed regularity of the kernel, both the point evaluation functional δ_{x_i} and the combination with the partial derivative $\delta_{x_i} \partial^a$ are in the dual space $\mathcal{H}_k(\Omega)'$. This results from the reproducing property. The Riesz representers of these functionals are given by $\partial_1^a k(x_i, \cdot)$ since $\delta_{x_i} \partial^a g = \langle \partial_1^a k(x_i, \cdot), g \rangle_{\mathcal{H}_k(\Omega)}$ for all $g \in \mathcal{H}_k(\Omega)$ (see Theorem 10.45 in [51]). Here, the 1 in the subscript of ∂_1^a denotes that it acts on the first input variable of the kernel. Overall, the surrogate can be represented by

$$s_v^n(x) = \sum_{i=1}^n \alpha_i k(x_i, x) + \langle \beta_i, \nabla_1 k(x_i, x) \rangle_2. \tag{22}$$

The condition from (21) for determining the coefficients $\{\alpha_i\}_{i=1}^n \subset \mathbb{R}$ and $\{\beta_i\}_{i=1}^n \subset \mathbb{R}^N$ can be written compactly with matrices as follows:

$$[\mathcal{K}_{X_n} \ \mathcal{B}_{X_n}] \begin{bmatrix} \underline{\alpha} \\ \underline{\beta} \end{bmatrix} = [v(x_1) \ \dots \ v(x_n)]^\top =: \underline{v(X_n)},$$

where

$$\begin{aligned} \mathcal{K}_{X_n} &:= \begin{bmatrix} k(x_1, x_1) & \dots & k(x_n, x_1) \\ \vdots & \ddots & \vdots \\ k(x_1, x_n) & \dots & k(x_n, x_n) \end{bmatrix} \in \mathbb{R}^{n \times n}, \\ \mathcal{B}_{X_n} &:= \begin{bmatrix} \nabla_1 k(x_1, x_1)^\top & \dots & \nabla_1 k(x_n, x_1)^\top \\ \vdots & \ddots & \vdots \\ \nabla_1 k(x_1, x_n)^\top & \dots & \nabla_1 k(x_n, x_n)^\top \end{bmatrix} \in \mathbb{R}^{n \times nN}, \\ \underline{\alpha} &:= [\alpha_1 \ \dots \ \alpha_n]^\top \in \mathbb{R}^n \text{ and } \underline{\beta} := [\beta_1^\top \ \dots \ \beta_n^\top]^\top \in \mathbb{R}^{nN} \end{aligned}$$

for the part of the conditions involving the values of the VF. For the derivative part, we first take a look at the gradient of the surrogate, which is

$$\nabla s_v^n(x) = \sum_{i=1}^n \alpha_i \nabla_2 k(x_i, x) + \mathcal{E}_k(x_i, x) \beta_i$$

with

$$\mathcal{E}_k(x_i, x) := \begin{bmatrix} \partial_{2,1} \nabla_1 k(x_i, x)^\top \\ \vdots \\ \partial_{2,N} \nabla_1 k(x_i, x)^\top \end{bmatrix} \in \mathbb{R}^{N \times N}.$$

Here, ∇_2 denotes the gradient that acts on the second input variable of the kernel and $\partial_{2,i}$ the partial derivative that acts on the i -th component of the second input variable. A compact matrix notation of the second part then results in

$$[\mathcal{B}_{\mathcal{X}}^\top \mathcal{C}_{\mathcal{X}}] \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = [\nabla v(x_1)^\top \dots \nabla v(x_n)^\top]^\top =: \underline{\nabla v(X_n)},$$

where

$$\mathcal{C}_{\mathcal{X}} = \begin{bmatrix} \mathcal{E}_k(x_1, x_1) & \dots & \mathcal{E}_k(x_n, x_1) \\ \vdots & \ddots & \vdots \\ \mathcal{E}_k(x_1, x_n) & \dots & \mathcal{E}_k(x_n, x_n) \end{bmatrix} \in \mathbb{R}^{nN \times nN}.$$

In doing so, we have taken advantage of the fact that $\nabla_2 k(x_i, x_j) = \nabla_1 k(x_j, x_i)$. Ultimately, the system of linear equations for determining the coefficients for the surrogate is

$$\underbrace{\begin{bmatrix} \mathcal{K}_{X_n} & \mathcal{B}_{X_n} \\ \mathcal{B}_{X_n}^\top & \mathcal{C}_{X_n} \end{bmatrix}}_{=: \mathcal{M}_{X_n}} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} v(X_n) \\ \underline{\nabla v(X_n)} \end{bmatrix}. \tag{23}$$

By construction, the matrix $\mathcal{M}_{X_n} \in \mathbb{R}^{N(n+1) \times N(n+1)}$ is symmetric. Furthermore, it can be shown that it is positive definite for a suitable choice of kernel, and therefore, the system is uniquely solvable.

Proposition 1 *Let $k(x, y) = \Phi(x - y)$ be a s.p.d kernel with $\Phi \in C^2(\Omega) \cap L^1(\Omega)$, then the matrix \mathcal{M}_{X_n} is symmetric positive definite for all pairwise distinct $X_n \subset \mathbb{R}^N$.*

Proof First, we recall Theorem 16.4 in [51], which states that under the assumed condition the set

$$\{\delta_x \partial^a \mid x \in \Omega \text{ and } (a_i)_{i=1}^N \in \mathbb{N}_0^N \text{ with } \sum_{i=1}^N a_i \leq m\} \subset \mathcal{H}_k(\Omega)'$$

is linearly independent. So, since

$$\partial_{1,i} k(y, x) = \langle \partial_{1,i} k(y, \cdot), k(x, \cdot) \rangle_{\mathcal{H}_k(\Omega)}$$

and

$$\partial_{2,j} \partial_{1,i} k(y, x) = \langle \partial_{1,i} k(y, \cdot), \partial_{1,j} k(x, \cdot) \rangle_{\mathcal{H}_k(\Omega)}$$

for all $\alpha \in \mathbb{R}^{N(n+1)} \setminus \{0\}$, it follows that

$$\begin{aligned} a^\top \mathcal{M}_{X_n} a &= \left\| \sum_{i=1}^n a_i k(x_i, \cdot) + \sum_{i=1}^n \sum_{j=1}^N a_{(iN+j)} \partial_{1,j} k(x_i, \cdot) \right\|_{\mathcal{H}_k(\Omega)}^2 \\ &= \left\| \sum_{i=1}^n a_i \delta_{x_i} + \sum_{i=1}^n \sum_{j=1}^N a_{(iN+j)} \delta_{x_i} \partial_{1,j} \right\|_{\mathcal{H}_k(\Omega)'}^2 \neq 0, \end{aligned}$$

which shows that the matrix is positive definite. □

A greedy selection criterion for the interpolation points

At this point, we have not yet clarified how the centers that represent the interpolation points will be selected. This is determined by an iterative scheme that successively increases the set of centers through a greedy strategy. The algorithm is a version based on the VKOGA from [41], which by Algorithm 2 has been adapted to the Hermite interpolation.

Algorithm 2 Hermite VKOGA.

- Require:** $\mathcal{D}, v(\mathcal{D}), \nabla v(\mathcal{D}), X := \emptyset, s_v := 0$ and $\epsilon_{\text{tol},f}$
- 1: **while** $\max_{x \in \mathcal{D} \setminus X} (|v(x) - s_v(x)| + \|\nabla v(x) - \nabla s_v(x)\|_2) > \epsilon_{\text{tol},f}$ **do**
 - 2: $x := \arg \max_{x \in \mathcal{D} \setminus X} (|v(x) - s_v(x)| + \|\nabla v(x) - \nabla s_v(x)\|_2)$
 - 3: $X := X \cup \{x\}$
 - 4: $s_v := \text{interpolant}(X, \nabla v(X))$
 - 5: **end while**
-

The procedure is performed until the interpolation error is less than $\epsilon_{\text{tol},f}$. At each iteration step in the loop, the algorithm selects the state that has the greatest deviation between the current interpolation and the output data in the sense that both the function value and the value of the gradient are included. In the numerical implementation, this means finding the maximum of a list, since \mathcal{D} is then discrete. The picked state is added to the set of centers in step 3. Thus, the resulting Hermite interpolant has no error there. The determination of the surrogate in each step can be performed by incrementally constructing the inverse of the Cholesky factorization of the system matrix \mathcal{M}_{X_n} . The complexity of each step is determined by a Cholesky factorization of a matrix of size $(N + 1) \times (N + 1)$ and by the product of a lower triangular matrix of size $Nn \times Nn$ with a dense matrix of size $Nn \times (N + 1)$, where n is the current iteration step. The output of the algorithm is a surrogate $s_v \approx v$. Theoretical results on the observed excellent convergence rates of the VKOGA can be found in [42]. A difficulty in the implementation of VKOGA for the Hermite interpolation arises in steps 1 and 2. In general, a matrix of size $|\mathcal{D}|N \times (n + 1)N$, where $|\mathcal{D}|$ is the number of samples the input data set, must be constructed to evaluate the surrogate on the data set, i.e.,

$$\begin{bmatrix} \mathcal{K}_{\mathcal{D}, X_n} & \mathcal{B}_{\mathcal{D}, X_n} \\ \mathcal{B}_{X_n, \mathcal{D}}^\top & \mathcal{C}_{\mathcal{D}, X_n} \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} =: \begin{bmatrix} s_v^n(\mathcal{D}) \\ \nabla s_v^n(\mathcal{D}) \end{bmatrix}$$

with $(\mathcal{K}_{\mathcal{D}, X_n})_{j,i}^{|\mathcal{D}|,n} = k(y_j, x_i)$ for $x_i \in \{x_1, \dots, x_n\} =: X_{\tilde{n}}, y_j \in \{y_1, \dots, y_{|\mathcal{D}|}\} =: \mathcal{D}$ and $\mathcal{B}_{\mathcal{D}, X_n}, \mathcal{C}_{\mathcal{D}, X_n}$ accordingly. This is practically infeasible for even moderately large state dimension and data set \mathcal{D} . For the special class of kernels, however, this is not necessary, as the next section shows.

Matrix-free approach

An explicit representation of the matrix used to evaluate the surrogate on the data is not needed for some kernels. All that is required here is the implementation of a matrix–vector multiplication. By restricting the consideration to a special class of radial kernels, the matrix–vector multiplication becomes efficient. The kernels in this class are of the form $k(x, y) = \Phi(\|x - y\|_2^2)$ with $\Phi \in C^2(\mathbb{R}_0^+, \mathbb{R})$. To see that this choice is beneficial, we look at $\mathcal{E}_k(x, y)$:

$$\mathcal{E}_k(x, y) = -2\Phi'(\|x - y\|_2^2)\mathbb{I}_{N \times N} + 4\Phi''(\|x - y\|_2^2)(x - y)(y - x)^\top.$$

Thus, the matrix $\mathcal{E}_k(x, y)$ consists of two scalar factors, a unit matrix $\mathbb{I}_{N \times N} \in \mathbb{R}^{N \times N}$ and the rank-one matrix $(x - y)(y - x)^\top$. Both the unit matrix and the latter matrix are advantageous in terms of matrix–vector multiplication, since

$$\mathcal{C}_{\mathcal{D}, X_n} \underline{\beta} = \begin{bmatrix} \sum_{i=1}^n -2\Phi'(\|x_i - y_1\|_2^2)\beta_i + 4\Phi''(\|x_i - y_1\|_2^2)(x_i - y_1)(y_1 - x_i, \beta_i)_{\mathbb{R}^N} \\ \vdots \\ \sum_{i=1}^n -2\Phi'(\|x_i - y_{|\mathcal{D}|}\|_2^2)\beta_i + 4\Phi''(\|x_i - y_{|\mathcal{D}|}\|_2^2)(x_i - y_{|\mathcal{D}|})(y_{|\mathcal{D}|} - x_i, \beta_i)_{\mathbb{R}^N} \end{bmatrix},$$

which reduces the total number of operations from $O(N|\mathcal{D}|n^2)$ to $O(|\mathcal{D}|n^2)$. Since N is potentially very high, only such a matrix-free strategy is capable of realizing the VKOGA algorithm for Hermite interpolation.

A structured surrogate for the value function

Now, we describe a strategy for structuring the surrogate of the VF in a way that incorporates three known properties of the VF. This makes the surrogate more context-aware. The first property is the fact that $v(0) = 0$ and $\nabla v(0) = 0$, since the zero state is the target state and therefore we have no cost there. By using a kernel k which satisfies $k(x, 0) = 0, \partial_{1,i}k(x, 0) = 0$ and $\partial_{2,j}\partial_{1,i}k(x, 0) = 0$ for all $x \in \Omega$ and all $i, j = 1, \dots, n$, one can ensure that the surrogate also fulfills this. Such a kernel can be generated, for example, by multiplying any kernel k' by $\langle x, y \rangle^2$. If k' is p.d., then so is the resulting kernel. Furthermore, we then remove the state 0 from the data set \mathcal{D} . Another possibility would be to force $s_v^n(0) = 0$ via an interpolation condition, but this would be incompatible with the realization of the next property. The second property to consider is that the VF can be very well represented locally near zero as a quadratic function. For example, if $(J_f(0), g(0))$ is controllable (see [52]), it can be determined by the linearized OCP. Here, J_f is the Jacobian matrix of f . The VF of the linearized problem can be generated by solving the corresponding algebraic Riccati equation. It

has the form $v_{\text{local}}(x) = x^\top Qx$, where $Q \in \mathbb{R}^{n \times n}$ is a positive definite matrix. This is added to the surrogate so that only a correction term is determined by the data-based strategy. The third property that the surrogate should adapt is positivity, since the costs considered must always be positive. This is achieved by squaring. Altogether, the structured surrogate is given by

$$s_{\text{Str},v}^n(x) = \left(\sqrt{x^\top Qx} + \sum_{i=1}^n \alpha_i k(x_i, x) + \langle \beta_i, \nabla_1 k(x_i, x) \rangle_2 \right)^2.$$

As in the penultimate subsection, the interpolation condition for determining the coefficients $\{\alpha_i\}_{i=1}^n \subset \mathbb{R}$ and $\{\beta_i\}_{i=1}^n \subset \mathbb{R}^N$ are

$$s_{\text{Str},v}^n(x_j) = v(x_j) \text{ and } \nabla s_{\text{Str},v}^n(x_j) = \nabla v(x_j) \text{ for all } j = 1, \dots, n. \tag{24}$$

These conditions are not linear with respect to $\{\alpha_i\}_{i=1}^n$ and $\{\beta_i\}_{i=1}^n$, but linear conditions implying them can be found. Since the VF is always positive,

$$\sum_{i=1}^n \alpha_i k(x_i, x_j) + \langle \beta_i, \nabla_1 k(x_i, x_j) \rangle_2 = \sqrt{v(x_j)} - \sqrt{x_j^\top Qx_j} \text{ for all } j = 1, \dots, n \tag{25}$$

implies the first equation in (24). Furthermore, this linear condition can be written compactly in matrix form as follows:

$$[\mathcal{K}_{X_n} \ \mathcal{B}_{X_n}] \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \left[\sqrt{v(x_1)} - \sqrt{x_1^\top Qx_1} \ \dots \ \sqrt{v(x_n)} - \sqrt{x_n^\top Qx_n} \right]^\top.$$

The second interpolation condition can also be expressed by an equivalent linear system of equations. To see this, we first need to look at the derivative of the structured surrogate:

$$\nabla s_{\text{Str},v}^n(x) = 2\sqrt{s_{\text{Str},v}^n(x)} \left(\frac{Qx}{\sqrt{x^\top Qx}} + \sum_{i=1}^n \alpha_i \nabla_2 k(x_i, x) + \mathcal{E}_k(x_i, x) \beta_i \right)$$

and with the interpolation conditions in (24), we get

$$\nabla v(x_j) = \nabla s_{\text{Str},v}^n(x_j) \stackrel{25}{=} 2\sqrt{v(x_j)} \left(\frac{Qx_j}{\sqrt{x_j^\top Qx_j}} + \sum_{i=1}^n \alpha_i \nabla_2 k(x_i, x_j) + \mathcal{E}_k(x_i, x_j) \beta_i \right)$$

for all $j = 1, \dots, n$. The case $v(x_j) = 0$ cannot appear, as this is equivalent to $x_j = 0$ which we have excluded from the data set \mathcal{D} as explained at the beginning of the current subsection. Therefore, with the upper equation, the linear condition implying the second interpolation condition in (24) for all x_j with $v(x_j) \neq 0$ becomes

$$\frac{\nabla v(x_j)}{2\sqrt{v(x_j)}} - \frac{Qx_j}{\sqrt{x_j^\top Qx_j}} = \sum_{i=1}^n \alpha_i \nabla_2 k(x_i, x_j) + \mathcal{E}_k(x_i, x_j) \beta_i \text{ for all } j = 1, \dots, n,$$

which can be compactly formulated with matrices by

$$[\mathcal{B}_{X_n}^\top \ C_{X_n}] \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \left[\left(\frac{\nabla v(x_1)}{2\sqrt{v(x_1)}} - \frac{Qx_1}{\sqrt{x_1^\top Qx_1}} \right)^\top \cdots \left(\frac{\nabla v(x_n)}{2\sqrt{v(x_n)}} - \frac{Qx_n}{\sqrt{x_n^\top Qx_n}} \right)^\top \right]^\top .$$

Therefore, the system of linear equations to obtain the surrogate is similar to (23), only the right-hand side is changed. It is remarkable that all three properties can be included natively in the surrogate without any additional effort in the solution step, as long as the matrix Q is known.

4 Two convergence results for the Hermite kernel method

We present two convergence results for the (unstructured) Hermite surrogate and for the surrogate-controlled trajectories. Another assumption combining the conditions from Sections 2 and 3 is required at this point.

Assumption 4 For a compact set $\Omega \subset \mathbb{R}^N$, let $k \in C^4(\text{conv}(\Omega) \times \text{conv}(\Omega))$ be a s.p.d. kernel. Further, let the VF v be radially unbounded and $v \in \mathcal{H}_k(\Omega)$.

Note that this assumption implies $v \in C^2(\Omega)$ (also with Theorem 10.45 in [51]), which is a higher regularity requirement than that of Section 2. However, this leads to a continuous dependence of the optimal trajectories on the initial data, which is essential for the subsequent proof of convergence. For brevity, we will use the abbreviations $\|\cdot\|_k := \|\cdot\|_{\mathcal{H}_k(\Omega)}$ for the norm induced by the inner product of the RKHS and $\langle \cdot, \cdot \rangle_k := \langle \cdot, \cdot \rangle_{\mathcal{H}_k(\Omega)}$ for the inner product of the RKHS in the following.

Lemma 2 Under the Assumption 4, it holds

$$|s_v^n(x) - s_v^n(y)| + \|\nabla s_v^n(x) - \nabla s_v^n(y)\|_2 \leq L \|x - y\|_2 \tag{26}$$

for all $x, y \in \Omega$, where the constant L is independent of the respective Hermite kernel surrogate for the VF.

Proof First, we need to see that the RKHS norm of a surrogate is bounded by the RKHS norm of the VF independent of n :

$$\begin{aligned} \|v\|_k \|s_v^n\|_k &\geq \langle v, s_v^n \rangle_k + \sum_{i=1}^n \alpha_i \underbrace{\langle s_v^n(x_i) - v(x_i) \rangle}_{=0} + \langle \beta_i, \underbrace{\nabla s_v^n(x_i) - \nabla v(x_i)}_{=0} \rangle_2 \\ &= \langle v, s_v^n \rangle_k + \langle s_v^n - v, s_v^n \rangle_k = \|s_v^n\|_k^2 \end{aligned}$$

For the first inequality, we used the Cauchy-Schwarz inequality and inserted a zero, which is possible because of the interpolation conditions. Using the reproducing

property and the sum representation of the Hermite interpolant, we have reformulated the zero term. So, overall, we see that $\|v\|_k \geq \|s_v^n\|_k$. This property is the reason why later the constant L depends only on $\|v\|_k$. To show (26), we consider

$$\begin{aligned} |s_v^n(x) - s_v^n(y)|^2 &= (s_v^n, k(x, \cdot) - k(y, \cdot))_k^2 \\ &\leq \|s_v^n\|_k^2 \|k(x, \cdot) - k(y, \cdot)\|_k^2 \\ &\leq \|v\|_k^2 (k(x, x) - k(x, y) + k(y, y) - k(y, x)) \\ &= \|v\|_k^2 (\nabla_2 k(x, \xi_1)(x - y) + \nabla_2 k(y, \xi_2)(y - x)) \\ &\leq \|v\|_k^2 \|x - y\|_2 \|\nabla_2 k(x, \xi_1) - \nabla_2 k(y, \xi_2)\|_2 \\ &= \|v\|_k^2 \|x - y\|_2 \|\nabla_2 k(x, \xi_1) - \nabla_2 k(x, \xi_2) + \nabla_2 k(x, \xi_2) - \nabla_2 k(y, \xi_2)\|_2 \\ &\leq C_1 \|v\|_k^2 \|x - y\|_2 (\|x - y\|_2 + \|\xi_1 - \xi_2\|_2) \\ &\leq 2C_1 \|v\|_k^2 \|x - y\|_2^2 \end{aligned}$$

with $\xi_1 = (1 - t_1)y + t_1x$ for a $t_1 \in [0, 1]$ and $\xi_2 = (1 - t_2)x + t_2y$ for a $t_2 \in [0, 1]$. For the first inequality, we utilized the Cauchy-Schwarz inequality, and for the second inequality, we used the first partial result of this proof. The third and fourth inequalities follow from the mean value theorem, with a zero term inserted for the latter. Hence,

$$C_1 := \sqrt{N} \max_{\bar{x}, \bar{y} \in \text{conv}(\Omega)} \max_{i=1, \dots, N} \max\{\|\partial_{i,1} \nabla_2 k(\bar{x}, \bar{y})\|_2, \|\partial_{i,2} \nabla_2 k(\bar{x}, \bar{y})\|_2\},$$

because the maximum is reached since $\text{conv}(\Omega)$ is compact and $k \in C^4(\text{conv}(\Omega) \times \text{conv}(\Omega))$ with the Assumption 4. The last inequality follows from the fact that

$$\|\xi_1 - \xi_2\|_2 = \|(1 - t_1)y + t_1x - (1 - t_2)x - t_2y\|_2 = |1 - t_1 - t_2| \|y - x\|_2 \leq \|x - y\|_2,$$

which is a rough estimate. This shows that s_v^n is Lipschitz continuous, where the Lipschitz constant is independent of the respective Hermite kernel surrogate for the VF. For the partial derivative, very similar steps can be performed:

$$\begin{aligned} &|\partial_i s_v^n(x) - \partial_i s_v^n(y)|^2 \\ &= |(s_v^n, \partial_{i,1} k(x, \cdot) - \partial_{i,1} k(y, \cdot))_k|^2 \\ &\leq \|s_v^n\|_k^2 \|\partial_{i,1} k(x, \cdot) - \partial_{i,1} k(y, \cdot)\|_k^2 \\ &\leq \|v\|_k^2 (\partial_{i,2} \partial_{i,1} k(x, x) - \partial_{i,2} \partial_{i,1} k(x, y) + \partial_{i,2} \partial_{i,1} k(y, y) - \partial_{i,2} \partial_{i,1} k(y, x)) \\ &\leq \|v\|_k^2 \|x - y\|_2 \left\| \nabla_2 \partial_{i,2} \partial_{i,1} k(x, \tilde{\xi}_1) - \nabla_2 \partial_{i,2} \partial_{i,1} k(y, \tilde{\xi}_2) \right\|_2 \\ &= \|v\|_k^2 \|x - y\|_2 \\ &\quad \cdot \left\| \nabla_2 \partial_{i,2} \partial_{i,1} k(x, \tilde{\xi}_1) - \nabla_2 \partial_{i,2} \partial_{i,1} k(x, \tilde{\xi}_2) + \nabla_2 \partial_{i,2} \partial_{i,1} k(x, \tilde{\xi}_2) - \nabla_2 \partial_{i,2} \partial_{i,1} k(y, \tilde{\xi}_2) \right\|_2 \\ &\leq 2C_2 \|v\|_k^2 \|x - y\|_2^2, \end{aligned}$$

where

$$C_2 := \sqrt{N} \max_{\bar{x}, \bar{y} \in \text{conv}(\Omega)} \max_{j,i=1, \dots, N} \max\{\|\partial_{j,1} \nabla_2 \partial_{i,2} \partial_{i,1} k(\bar{x}, \bar{y})\|_2, \|\partial_{j,2} \nabla_2 \partial_{i,2} \partial_{i,1} k(\bar{x}, \bar{y})\|_2\}.$$

Overall, the statement follows with $L = \sqrt{2}(\sqrt{C_1} + \sqrt{NC_2}) \|v\|_k$. □

Now, we can give a convergence result related to the VF and its derivative:

Theorem 2 *Let Assumption 4 with $\Omega := \mathcal{T}(\mathcal{A})$ and additionally $f, g \in C^1(\mathcal{T}(\mathcal{A}))$ hold, then for every $T > 0$ there is a constant $C_T > 0$ independent of $\epsilon_{tol,d}$, $\epsilon_{tol,f}$ and the respective Hermite kernel interpolant s_v^n for v such that*

$$\begin{aligned} \max_{(t,x_0) \in [0,T] \times \mathcal{A}} & (|v(\mathbf{x}^*(t; x_0)) - s_v^n(\mathbf{x}^*(t; x_0))| \\ & + \|\nabla v(\mathbf{x}^*(t; x_0)) - s_v^n(\mathbf{x}^*(t; x_0))\|_2) \leq \epsilon_{tol,d} C_T + \epsilon_{tol,f} \end{aligned}$$

if Algorithm 1 with $\epsilon_{tol,d}$ and Algorithm 2 with $\epsilon_{tol,f}$ have terminated.

Proof First, we fix an $\mathbf{x}^*(t_1; x_{0,1}) \in \mathcal{T}(\mathcal{A})$ with $t_1 \leq T$ and $x_{0,1} \in \mathcal{A}$. Due to the selection criterion in Algorithm 1, we know that for the state $x_{0,1}$ in \mathcal{A} , there is a trajectory $\mathbf{x}^*(\cdot; x_{0,2}) \subset \mathcal{D}$ in the available data set and a time $t_2 \geq 0$ with

$$\|x_{0,1} - \mathbf{x}^*(t_2; x_{0,2})\|_2 \leq \epsilon_{tol,d}. \tag{27}$$

Next, we define $h(x) := f(x) - \frac{1}{2}g(x)R^{-1}g(x)^\top \nabla v(x)$, which is Lipschitz continuous with constant L_h on the compact set $\mathcal{T}(\mathcal{A})$ as f, g and ∇v are differentiable. Thus, the ODE of the optimal trajectory in integral form for any $x_0 \in \mathcal{A}$ becomes

$$\mathbf{x}^*(t; x_0) = x_0 + \int_0^t h(\mathbf{x}^*(s; x_0)) ds.$$

Using this, we get

$$\begin{aligned} & \|\mathbf{x}^*(t_1; x_{0,1}) - \mathbf{x}^*(t_1 + t_2; x_{0,2})\|_2 \\ & \leq \|x_{0,1} - \mathbf{x}^*(t_2; x_{0,2})\|_2 + \int_0^{t_1} \|h(\mathbf{x}^*(s; x_{0,1})) - h(\mathbf{x}^*(s + t_2; x_{0,2}))\|_2 ds \\ & \leq \|x_{0,1} - \mathbf{x}^*(t_2; x_{0,2})\|_2 + L_h \int_0^{t_1} \|\mathbf{x}^*(s; x_{0,1}) - \mathbf{x}^*(s + t_2; x_{0,2})\|_2 ds, \end{aligned}$$

which leads with Grönwall’s lemma, Equation (27) and $t_1 \leq T$ to

$$\|\mathbf{x}^*(t_1; x_{0,1}) - \mathbf{x}^*(t_1 + t_2; x_{0,2})\|_2 \leq \|x_{0,1} - \mathbf{x}^*(t_2; x_{0,2})\|_2 e^{L_h t_1} \leq \epsilon_{tol,d} e^{L_h T}. \tag{28}$$

Furthermore, after the termination of Algorithm 2, we know that

$$\begin{aligned} & |v(\mathbf{x}^*(t_1 + t_2; x_{0,2})) - s_v^n(\mathbf{x}^*(t_1 + t_2; x_{0,2}))| \\ & + \|\nabla v(\mathbf{x}^*(t_1 + t_2; x_{0,2})) - \nabla s_v^n(\mathbf{x}^*(t_1 + t_2; x_{0,2}))\|_2 \leq \epsilon_{tol,f} \end{aligned} \tag{29}$$

as $\mathbf{x}^*(\cdot; x_{0,2}) \subset \mathcal{D}$ is in the available data set. So, overall, we get

$$\begin{aligned}
 & |v(\mathbf{x}^*(t_1; x_{0,1})) - s_v^n(\mathbf{x}^*(t_1; x_{0,1}))| + \|\nabla v(\mathbf{x}^*(t_1; x_{0,1})) - \nabla s_v^n(\mathbf{x}^*(t_1; x_{0,1}))\|_2 \\
 = & |v(\mathbf{x}^*(t_1; x_{0,1})) - v(\mathbf{x}^*(t_1 + t_2; x_{0,2})) + v(\mathbf{x}^*(t_1 + t_2; x_{0,2})) - s_v^n(\mathbf{x}^*(t_1; x_{0,1}))| \\
 & + \|\nabla v(\mathbf{x}^*(t_1; x_{0,1})) - \nabla v(\mathbf{x}^*(t_1 + t_2; x_{0,2})) \\
 & + \nabla v(\mathbf{x}^*(t_1 + t_2; x_{0,2})) - \nabla s_v^n(\mathbf{x}^*(t_1; x_{0,1}))\|_2 \\
 \leq & (L_v + L_{\nabla v})e^{L_h T} \epsilon_{tol,d} \\
 & + |v(\mathbf{x}^*(t_1 + t_2; x_{0,2})) - s_v^n(\mathbf{x}^*(t_1; x_{0,1}))| \\
 & + \|\nabla v(\mathbf{x}^*(t_1 + t_2; x_{0,2})) - \nabla s_v^n(\mathbf{x}^*(t_1; x_{0,1}))\|_2 \\
 = & (L_v + L_{\nabla v})e^{L_h T} \epsilon_{tol,d} \\
 & + |v(\mathbf{x}^*(t_1 + t_2; x_{0,2})) - s_v^n(\mathbf{x}^*(t_1 + t_2; x_{0,2})) \\
 & + s_v^n(\mathbf{x}^*(t_1 + t_2; x_{0,2})) - s_v^n(\mathbf{x}^*(t_1; x_{0,1}))| \\
 & + \|\nabla v(\mathbf{x}^*(t_1 + t_2; x_{0,2})) - \nabla s_v^n(\mathbf{x}^*(t_1 + t_2; x_{0,2})) \\
 & + \nabla s_v^n(\mathbf{x}^*(t_1 + t_2; x_{0,2})) - \nabla s_v^n(\mathbf{x}^*(t_1; x_{0,1}))\|_2 \\
 \leq & (L_v + L_{\nabla v})e^{L_h T} \epsilon_{tol,d} + \epsilon_{tol,f} \\
 & + |s_v^n(\mathbf{x}^*(t_1 + t_2; x_{0,2})) - s_v^n(\mathbf{x}^*(t_1; x_{0,1}))| \\
 & + \|\nabla s_v^n(\mathbf{x}^*(t_1 + t_2; x_{0,2})) - \nabla s_v^n(\mathbf{x}^*(t_1; x_{0,1}))\|_2 \\
 \leq & \underbrace{(L_v + L_{\nabla v} + L)}_{=: C_T} e^{L_h T} \epsilon_{tol,d} + \epsilon_{tol,f},
 \end{aligned}$$

where we inserted zero terms for the first inequality and then used the triangle inequality, the Lipschitz continuity of v (Lipschitz constant L_v) and ∇v (Lipschitz constant $L_{\nabla v}$ and Equation (28) to obtain the first inequality. For the second equality, we inserted other zero terms and then utilized the triangle inequality, Lemma 2, and Equation (28) to get the second inequality. The last inequality results from (29). Since $\mathbf{x}^*(t_1; x_{0,1})$ was chosen arbitrarily, the above inequality also holds for the maximum, and so the statement follows. \square

Note that the latter theorem is valid for any fixed finite time horizon T , but not in the limit $T \rightarrow \infty$, since the constants involved also go to infinity.

The next result concerns the convergence of the surrogate-controlled trajectory \mathbf{x}_s to the optimal trajectory \mathbf{x}^* . In order to establish the convergence, it is required that the regularity of the VF applies to a slightly larger domain. Therefore, for a fixed $\delta > 0$, we define the dilated set $\mathcal{T}_\delta(\mathcal{A}) := \mathcal{T}(\mathcal{A}) + \overline{B}_\delta$, where \overline{B}_δ is the closed ball centered at zero with radius δ . By the addition of two sets, we mean the set that results from adding arbitrary elements of both sets.

Theorem 3 *Let Assumption 4 with $\Omega := \mathcal{T}_\delta(\mathcal{A})$ hold for a $\delta > 0$ and additionally $f, g \in C^1(\mathcal{T}_\delta(\mathcal{A}))$, then for every $T > 0$, there are two constants $\overline{C}_T > 0$ and*

$\tilde{C}_T > 0$ independent of $\epsilon_{tol,d}$, $\epsilon_{tol,f}$ and the respective Hermite kernel interpolant s_v^n for v such that

$$\max_{(t,x_0) \in [0,T] \times \mathcal{A}} \|\mathbf{x}^*(t; x_0) - \bar{\mathbf{x}}_s(t; x_0)\|_2 \leq \epsilon_{tol,d} \bar{C}_T + \epsilon_{tol,f} \tilde{C}_T,$$

if Algorithm 1 with $\epsilon_{tol,d}$ and Algorithm 2 with $\epsilon_{tol,f}$ have terminated. Furthermore, it is assumed that the thresholds $\epsilon_{tol,d}$, $\epsilon_{tol,f}$ are so small that $\epsilon_{tol,d} \bar{C}_T + \epsilon_{tol,f} \tilde{C}_T < \delta$ holds.

Proof We start by defining a function that summarizes the dynamics of the ODE systems under consideration for a given VF surrogate $e(x)$:

$$h_e(x) := f(x) - \frac{1}{2}g(x)R^{-1}g(x)^\top \nabla e(x)$$

Note that the function $h_{s_v^n}$ is Lipschitz continuous on $\mathcal{T}_\delta(\mathcal{A})$ where the Lipschitz constant L_{h_s} is independent of the respective interpolant of the VF. This follows with the differentiability of f , g , and ∇v . The independence of the constant from the surrogate follows as in Lemma 2. Due to Kirszbraun’s theorem [53], there is a globally Lipschitz continuous function $\bar{h}_{s_v^n} : \mathbb{R}^N \rightarrow \mathbb{R}^N$ with the same Lipschitz constant L_{h_s} and $\bar{h}_{s_v^n}(x) = h_{s_v^n}(x)$ for $x \in \mathcal{T}_\delta(\mathcal{A})$. Next, an auxiliary trajectory $\bar{\mathbf{x}}_s(t; x_0)$ is defined which satisfies the ODE

$$\dot{\bar{\mathbf{x}}}_s(t; x_0) = \bar{h}_{s_v^n}(\bar{\mathbf{x}}_s(t; x_0))$$

with $\bar{\mathbf{x}}_s(0; x_0) = x_0$. With the Picard-Lindelöf theorem, such a trajectory exists for all time intervals, due to the global Lipschitz continuity of $\bar{h}_{s_v^n}$. In a further step, we show that the auxiliary trajectory and the optimal trajectory are getting closer to each other as $\epsilon_{tol,d}$ and $\epsilon_{tol,f}$ decrease. Let us also use the definition $G(x) = -\frac{1}{2}g(x)R^{-1}g(x)^\top$ here to simplify. For a fixed $(t; x_0)$ with $t \leq T$, it holds

$$\begin{aligned} & \|\mathbf{x}^*(t; x_0) - \bar{\mathbf{x}}_s(t; x_0)\|_2 \\ & \leq \int_0^t \|h_v(\mathbf{x}^*(t; x_0)) - \bar{h}_{s_v^n}(\bar{\mathbf{x}}_s(t; x_0))\|_2 dt \\ & = \int_0^t \|h_v(\mathbf{x}^*(t; x_0)) - \bar{h}_{s_v^n}(\mathbf{x}^*(t; x_0)) + \bar{h}_{s_v^n}(\mathbf{x}^*(t; x_0)) - \bar{h}_{s_v^n}(\bar{\mathbf{x}}_s(t; x_0))\|_2 dt \\ & \leq \int_0^t \|h_v(\mathbf{x}^*(t; x_0)) - \bar{h}_{s_v^n}(\mathbf{x}^*(t; x_0))\|_2 dt + \int_0^t \|\bar{h}_{s_v^n}(\mathbf{x}^*(t; x_0)) - \bar{h}_{s_v^n}(\bar{\mathbf{x}}_s(t; x_0))\|_2 dt \\ & \leq \int_0^t \underbrace{\|h_v(\mathbf{x}^*(t; x_0)) - \bar{h}_{s_v^n}(\mathbf{x}^*(t; x_0))\|_2}_{=\|G(\mathbf{x}^*(t; x_0))(\nabla v(\mathbf{x}^*(t; x_0)) - \nabla s_v^n(\mathbf{x}^*(t; x_0)))\|_2} dt + L_{h_s} \int_0^t \|\mathbf{x}^*(t; x_0) - \bar{\mathbf{x}}_s(t; x_0)\|_2 dt \\ & \leq (\epsilon_{tol,d} C_T + \epsilon_{tol,f}) \max_{x \in \mathcal{T}(\mathcal{A})} \|G(x)\|_2 T + L_{h_s} \int_0^t \|\mathbf{x}^*(t; x_0) - \bar{\mathbf{x}}_s(t; x_0)\|_2 dt, \end{aligned}$$

where we utilized Theorem 2 for the last inequality. So with the Grönwall’s inequality [54], the definitions $\bar{C}_T := C_T \max_{x \in \mathcal{T}(\mathcal{A})} \|G(x)\|_2 T e^{L_{h_s} T}$ and $\tilde{C}_T := \max_{x \in \mathcal{T}(\mathcal{A})} \|G(x)\|_2 T e^{L_{h_s} T}$, it follows

$$\|\mathbf{x}^*(t; x_0) - \bar{\mathbf{x}}_s(t; x_0)\|_2 \leq \epsilon_{tol,d} \bar{C}_T + \epsilon_{tol,f} \tilde{C}_T \leq \delta.$$

Here, the latter inequality comes from the assumption. The crucial point now is that because of the estimate $\|\mathbf{x}^*(t; x_0) - \bar{\mathbf{x}}_s(t; x_0)\|_2 \leq \delta$, it holds that $\bar{\mathbf{x}}_s(t; x_0) \in \mathcal{T}_\delta(\mathcal{A})$ for all $t \in [0, T]$ and therefore $\bar{\mathbf{x}}_s(t; x_0) = \mathbf{x}_s(t; x_0)$ as $\bar{h}_{s_v}^n(x) = h_{s_v}^n(x)$ for $x \in \mathcal{T}_\delta(\mathcal{A})$. Thus, the statement follows as $(t; x_0)$ was chosen arbitrarily. \square

From a practical point of view, the last theorem is very interesting because it gives a guarantee that the approximate optimal feedback control will provide near optimal trajectories if Algorithms 1 and 2 have terminated for sufficiently small thresholds $\epsilon_{tol,d}$ and $\epsilon_{tol,f}$.

5 Numerical experiments

Three model problems are considered in this section. For each of them, a training data set is computed using Algorithm 1. Because the optimal open-loop solution becomes slightly less accurate near zero, as indicated by the first model problem, all trajectories are truncated after time T . Algorithm 2 generates the Hermite surrogate and the structured Hermite surrogate, adapting step 4 for the latter. For the former surrogate, we use the Wendland kernel $k(x, y) = \Phi_{N,m}(\gamma \|x - y\|_2)$ with

$$\Phi_{N,m} = (1 - r)_+^{l+2} \left[(l^2 + 4l + 3)r^2 + (3l + 6)r + 3 \right],$$

$l := \lfloor N/2 \rfloor + 3$ and $(1 - r)_+ := \max\{1 - r, 0\}$ from Corollary 9.14 in [51], which exactly satisfies the regularity assumptions in Assumption 4. Here, $1/\gamma > 0$ is the kernel width. We multiply this Wendland kernel by $\langle x, y \rangle^2$ as mentioned in Section 3 to obtain the kernel for the structured Hermite surrogate. The VF surrogates are assessed with respect to a test error for the approximation of the VF and the quality of the resulting surrogate-controlled trajectories. The required test set consists of 20 open-loop trajectories for which the initial states $\{x_{0,1}, \dots, x_{0,20}\}$ are selected in \mathcal{A} using a so-called geometric greedy procedure [50]. In our case, this test set size is about 10% of the training set size, which seems reasonable. To determine the quality of the feedback realized by the surrogate, we use a mean relative L^2 -error

$$MRL^2 Error(\mathbf{x}_{OL}, \mathbf{x}_{SR}) := \frac{1}{20} \sum_{j=1}^{20} \sqrt{\frac{\sum_{i=0}^{N_j} \|\mathbf{x}_{OL}(t_{j,i}; x_{0,j}) - \mathbf{x}_{SR}(t_{j,i}; x_{0,j})\|_2^2}{\sum_{i=0}^{N_j} \|\mathbf{x}_{OL}(t_{j,i}; x_{0,j})\|_2^2}}$$

for partitionings $0 = t_{j,0} < t_{j,1} < \dots t_{j,N_j} = T$, which are determined adaptively by the solver `solve_bvp` from SciPy, for $j = 1, \dots, 20$. It compares the optimal open-loop

controlled trajectories \mathbf{x}_{OL} of the test set with the surrogate-controlled trajectories \mathbf{x}_{SR} . The latter is computed by solving the approximate optimal feedback control in closed-loop form

$$\begin{aligned} \dot{\mathbf{x}}_{SR}(s; x_{0,j}) &= f(\mathbf{x}_{SR}(s; x_{0,j})) \\ &\quad - \frac{1}{2} g(\mathbf{x}_{SR}(s; x_{0,j})) R^{-1} g(\mathbf{x}_{SR}(s; x_{0,j}))^\top \nabla_{s_v} v(\mathbf{x}_{SR}(s; x_{0,j})) \end{aligned}$$

with $\mathbf{x}(0) = x_{0,j}$ for all $j = 1, \dots, 20$ using SciPy’s *solve_ivp*. This error is also used to determine the hyperparameter γ , which is realized with a s -fold cross validation, where the training set of the trajectories is divided into s sets of complete trajectories. To allow a better comparison of the model problems with each other, we introduce the quantities $C_{\max, \mathcal{A}} := \max_{x \in \mathcal{A}} \|x\|_2$ and $C_{\max, v} := \max_{x \in \mathcal{D}} (\|v(x)\|_2 + \|\nabla v(x)\|_2)$ to specify a relative cover distance and training error in the following. All runtimes of the numerical experiments refer to a laptop with an AMD Ryzen 9 5900HX CPU and 16GB RAM.

An academic model problem

The first model problem is an academic model problem (AMP), which is not application-motivated but dimension-variable and has the advantage that the VF is known analytically, allowing exact error assessment. It has the form

$$\min_{\mathbf{u} \in \mathcal{U}_\infty} J(\mathbf{u}) = \min_{\mathbf{u} \in \mathcal{U}_\infty} \int_0^\infty \alpha e^{\|\mathbf{x}(s)\|_2^2} \langle \mathbf{x}(s), \mathbf{x}(s) \rangle_2^2 + \beta (\mathbf{u}(s))^2 \, ds \quad (30)$$

$$\text{subject to } \dot{\mathbf{x}}(s) = \|\mathbf{x}(s)\|_2^2 \mathbf{x}(s) + e^{-\frac{\|\mathbf{x}(s)\|_2^2}{2}} \mathbf{x}(s) \mathbf{u}(s) \text{ and } \mathbf{x}(0) = x_0 \in \mathbb{R}^N \quad (31)$$

with controller dimension $M = 1$. This model problem may also be useful for other studies. The HJB equation for this problem is

$$\langle \nabla v(x), x \rangle_2 \|x\|_2^2 - \frac{e^{-\|x\|_2^2}}{4\beta} \langle \nabla v(x), x \rangle_2^2 + \alpha e^{\|x\|_2^2} \|x\|_2^4 = 0. \quad (32)$$

For the VF, we chose the ansatz $v(x) = C e^{\|x\|_2^2} - C$ and thus $\nabla v(x) = 2C e^{\|x\|_2^2} x$. Inserting this into (32) gives

$$\left(2C - C^2 \frac{1}{\beta} + \alpha \right) e^{\|x\|_2^2} \langle x, x \rangle_2^2 = 0,$$

which is solved for $C_{1\setminus 2} := \beta(1 \pm \sqrt{1 + \alpha/\beta})$. But only $C := \beta(1 + \sqrt{1 + \alpha/\beta})$ makes sense, because the VF must always be positive. The optimal closed-loop system in terms of (30) is therefore

$$\begin{aligned} \dot{\mathbf{x}}^*(s) &= \left(\|\mathbf{x}^*(s)\|_2^2 - \frac{1}{2\beta} e^{-\|\mathbf{x}^*(s)\|_2^2} \langle \mathbf{x}^*(s), \nabla v(x) \rangle_2 \right) \mathbf{x}^*(s) \\ &= -\sqrt{1 + \frac{\alpha}{\beta}} \|\mathbf{x}^*(s)\|_2^2 \mathbf{x}^*(s), \end{aligned}$$

which is globally asymptotically stable, since a Lyapunov function is given by v (see [55, Chapter 4] for more background information on Lyapunov stability). The standard linearization approach to get an approximation of the VF in a neighborhood of zero, as required for the structured Hermite surrogate, is not possible for this problem, since the Jacobian matrix of f holds $J_f(0) = 0$ and $g(0) = 0$. Nevertheless, a quadratic approximation can also be obtained by truncating the Taylor series of the true VF:

$$v(x) \approx v(0) + \nabla v(0)(x - 0) + (x - 0)^\top J_{\nabla v}(0)(x - 0) = 2C \|x\|_2^2$$

The approximate optimum closed-loop system is therefore

$$\dot{\mathbf{x}}^*(s) = \left(1 - \frac{2C}{\beta} e^{-\|\mathbf{x}^*(s)\|_2^2} \right) \|\mathbf{x}^*(s)\|_2^2 \mathbf{x}^*(s),$$

which is only guaranteed to be asymptotically stable if

$$\left(1 - \frac{2C}{\beta} e^{-\|\mathbf{x}^*(s)\|_2^2} \right) < 0,$$

which leads to

$$\|\mathbf{x}^*(s)\|_2 < \sqrt{\log \left(2 + 2\sqrt{1 + \frac{\alpha}{\beta}} \right)} := r_1,$$

since the VF v is again a Lyapunov function on the Ball $B_{r_1}(0)$.

Note that the construction of the model problem works for arbitrary state dimension N . For the following experiments, we set $N = 2$ and sample the initial states \mathcal{A} uniformly from $[-1, 1]^N$ with 200 points in each direction. Although the model is dimensionally scalable, we choose a very small dimension for this first example. This is because our method is expected to perform well only for problems with a low-dimensional solution manifold, which is not the case for the academic model problem in higher dimensions with a hypercube as the set of initial states. This is due to the non-physical origin and the resulting unclear choice of the set of initial states. Moreover, the design parameters of the problem are chosen to be $\alpha = 10^5$ and $\beta = 1$, such that the high ratio $\frac{\alpha}{\beta}$ ensures that the system quickly approaches zero. Figure 1 shows the training (turquoise) and test (orange) data sets for the academic model problem, using 100 trajectories for the former, which then has a cover distance

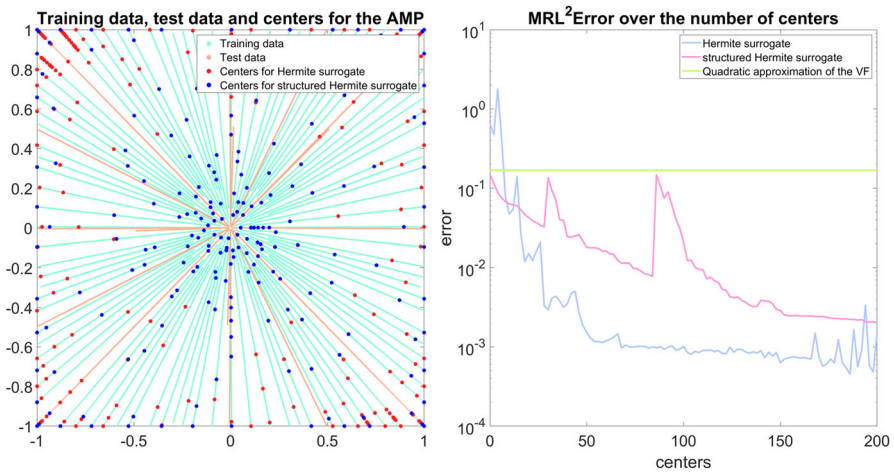


Fig. 1 The left diagram shows the training data, the test data, and the centers chosen by Algorithm 2 for the Hermite and the structured Hermite surrogate. On the right diagram, we present a semilog plot of the MRL^2Error for the three types of surrogate over the number of centers selected

smaller than $\epsilon_{tol,d}/C_{max,\mathcal{A}} = 1.3 \cdot 10^{-3}$ in Algorithm 1. The relative deviation of the computed data to the values of the true VF and its gradient is smaller than 10^{-7} . Without truncating the trajectories to $T = 99$, which corresponds to $\tau = 0.99$ on the transformed interval, it would be 10^{-5} . Since only here the true VF is accessible, we keep this value for T also for the other model problems. Another indicator of data quality is the fulfillment of the HJB equation along the trajectories. Here, the largest deviation is smaller than 10^{-12} . Since this is close to machine precision, the Hermite kernel interpolation can then also be interpreted as a collocation method of the HJB equation. The average time taken to compute an optimal open-loop trajectory is approximately 42 seconds. For cross validation, $s = 10$ was used to determine the kernel shape parameter. This yields $\gamma_H = 0.04$ for the Hermite surrogate. With a total number of 200, the centers are scattered like the red dots in Fig. 1 (left). Using Algorithm 2, the selection took 940 seconds and resulted in a relative training error smaller than $\epsilon_{tol,f}/C_{max,v} = 5.089 \cdot 10^{-4}$. For the structured Hermite surrogate, the cross validation leads to $\gamma_{SH} = 0.4$. Here, the 200 centers are selected as the blue dots in Fig. 1 (left), which took 893 seconds. In this, the relative training error is smaller than $\epsilon_{tol,f}/C_{max,v} = 6.798 \cdot 10^{-4}$.

In terms of the MRL^2Error (see Fig. 1 right), it can be seen that both data-based Hermite surrogates produce smaller errors than the quadratic approximation, whose MRL^2Error is around $1.688 \cdot 10^{-1}$. For a few centers, the structured Hermite surrogate is the best, as it gives a MRL^2Error of $6.757 \cdot 10^{-2}$ after 8 selected centers. However, it is quickly caught up by the Hermite surrogate, which then overall results in a lower MRL^2Error . Its minimum is at 186 centers with a value of $4.556 \cdot 10^{-4}$. The minimal MRL^2Error for the structured Hermite surrogate is at 200 centers with a value of $2.033 \cdot 10^{-3}$. It is also clearly visible in Fig. 1 (right) that for both surrogates the error is beginning to stagnate, which is because there are trajectories in the test

data that are not seen in the training (see Fig. 1 left). Therefore, the error could only be reduced by having data with a smaller $\epsilon_{\text{tol,d}}$, which fits to the convergence results in Theorem 3. The calculation of a surrogate control trajectory for an initial state in the test data set took on average 0.96 s for the Hermite surrogate and 1.16 s for the structured Hermite surrogate. So, here, we have a considerable advantage compared to the runtime of 42 s mentioned above for an optimal open-loop solution, while achieving almost the same result. Without reporting detailed results, we would like to remark that we also performed experiments in the eight-dimensional case using a very coarse grid in the eight-dimensional hypercube as the initial state and 250 trajectories. However, the Hermite surrogate with 1000 centers was unable to provide feedback control that stabilized the system. On the other hand, the structured Hermite surrogate was able to do so, although the quadratic approximation part itself does not stabilize on some parts of the unit hypercube in this eight-dimensional case. This suggests that the structured Hermite surrogate generalizes better.

Gripper-Soft-Tissue

The second model problem is the Gripper-Soft-Tissue (GST). We only give a very concise description of the model; for more details, we refer to [39]. The GST is a two-dimensional physical domain model that describes a gripper that has gripped a soft tissue, such as a fruit or a piece of meat, and brings it to a prescribed target position while avoiding an obstacle. Figure 2 shows a schematic representation. The governing equation of a linear elastic body is used to model the displacement field of the soft tissue. The gripper is modeled by a point mass whose displacement is described by Newton's second law. An external force $\mathbf{u}(s)$ controls the gripper in this process. Consequently, the controller dimension is $M = 2$. A semi-discretized version of the coupled system can be represented by an ODE system utilizing finite element methods. For this problem, we choose $N = 36$. The target position is the state with no displacement and no velocity. The state constraints for the obstacle placed in

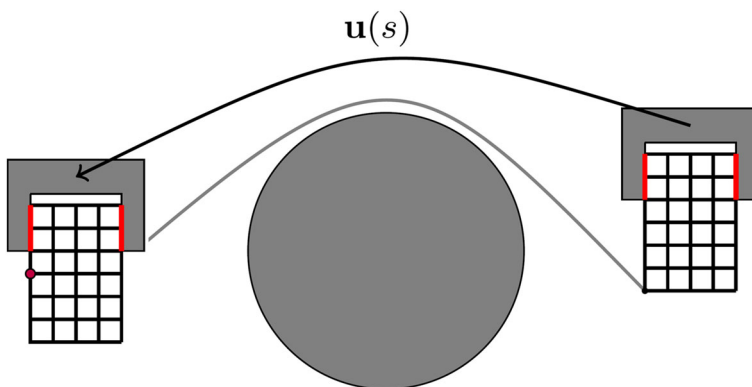


Fig. 2 A gripper that transports soft tissue to a target point while avoiding an obstacle. The purple dot indicates the node which is zero when the system is in a position with no displacement

(4, 0) with a radius of 1.8 are handled by an external penalty method. This means that an additional term is introduced into the running payoff that makes states inside the obstacle very expensive and thus the optimal trajectory bypasses the obstacle. The OCP reads

$$\min_{\mathbf{u} \in \mathcal{U}_\infty} \int_0^\infty \mathbf{x}(s)^\top \mathbf{Q} \mathbf{x}(s) + \mathbf{x}(s)^\top \mathbf{x}(s) \sum_{i=1}^c e^{3.5 C_i(\mathbf{x}(s))} + \mathbf{u}(s)^\top \mathbf{R} \mathbf{u}(s) \, ds$$

subject to $E \dot{\mathbf{x}}(s) = \mathbf{A} \mathbf{x}(s) + \mathbf{D} \mathbf{u}(s)$ and $\mathbf{x}(0) = \mathbf{x}_0$,

where the functions $(C_i)_{i=1}^c$ are positive if the i th-node is in the obstacle and negative otherwise. Note that the problem without obstacle is a standard linear-quadratic regulator (LQR) whose solution can be computed by solving the algebraic Riccati equation. This was used to compute the quadratic approximation of the VF.

For reasons of dimensionality reduction, the set of initial states is chosen to contain only states with constant displacement and velocity. So there is no initial state with a deformed soft tissue. This results in four degrees of freedom for displacement and velocity in x and y directions, which are taken from a $250 \times 120 \times 5 \times 5$ uniform grid in $[-2, 10] \times [0, 6] \times [-2, 2] \times [-2, 2]$ to obtain the set of initial states \mathcal{A} . States touching or inside the obstacle are removed. Note that only initial states that have their optimal path above the obstacle are included in this set. If we would allow trajectories below the obstacle, there would be unique problems when computing the optimal solution with the necessary PMP conditions. Figure 3 illustrates the training (turquoise) and test (orange) data set, showing only the position of the purple node from Fig. 2. The training data set consists of 250 trajectories, which have a cover distance smaller than $\epsilon_{\text{tol,d}}/C_{\text{max},\mathcal{A}} = 1.2 \cdot 10^{-2}$ in Algorithm 1. The maximum deviation from the HJB equation

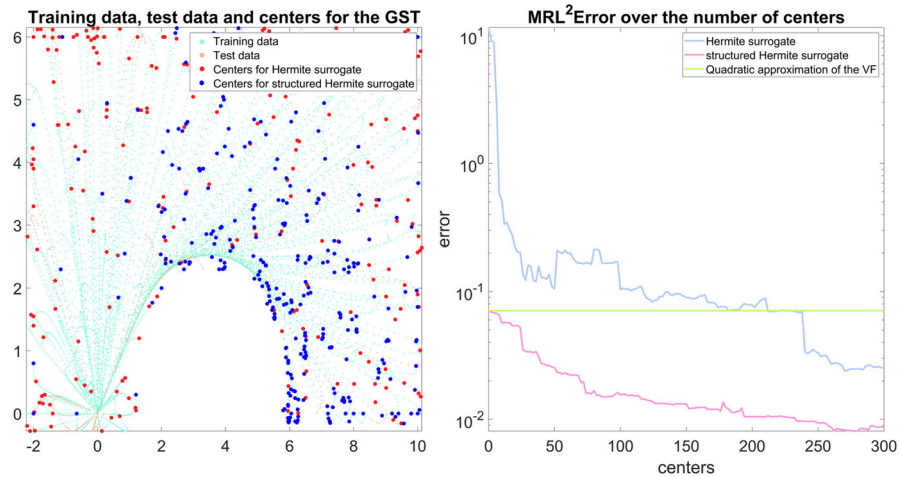


Fig. 3 The left diagram shows the training data, the test data, and the centers chosen by Algorithm 2 for the Hermite and the structured Hermite surrogate. Here, the respective state is indicated by the displacement of the lower left corners of the soft tissue. The right diagram is a semilogy plot of the MRL^2 Error for the three types of surrogate over the number of centers selected

on the trajectories in the data set is smaller than 10^{-7} and the calculation of an optimal trajectory takes about 110 s. For cross validation, $s = 10$ results in $\gamma_H = 0.003$ for the Hermite surrogate and $\gamma_{SH} = 0.015$ for the structured Hermite surrogate. For both, 300 centers were selected, distributed as illustrated by the blue dots in Fig. 3 (left) for the Hermite surrogate and by the red dots for the structured Hermite surrogate. The training of the first surrogate took 413 seconds resulting in a relative training error less than $\epsilon_{\text{tol},f}/C_{\text{max},v} = 1.466 \cdot 10^{-2}$. The second surrogate was trained in 431 seconds and yields a relative training error less than $\epsilon_{\text{tol},f}/C_{\text{max},v} = 5.155 \cdot 10^{-3}$.

Looking at the $MRL^2 Error$, we see that both surrogates can beat the control of the quadratic approximation, as the latter leads to an error of $7.088 \cdot 10^{-2}$. As this quadratic VF is the solution of the OCP without penalty term, the trajectories controlled by this surrogate do not avoid the obstacle. What is striking about this model problem is that the structured Hermite surrogate with a minimum $MRL^2 Error$ of $8.099 \cdot 10^{-3}$ clearly outperforms the Hermite surrogate, which at its best has $2.399 \cdot 10^{-2}$. By design, the former starts with a much better error. In addition, very few centers are selected near the zero state as expected. Again, the error begins to stagnate for both surrogates, which can only be improved by reducing $\epsilon_{\text{tol},d}$. Here, the computation of a surrogate-controlled trajectory with an initial state in the test data set took on average 7.6 s for the Hermite surrogate and 2.8 s for the structured Hermite surrogate. Thus, also for this model problem, computing a trajectory based on the surrogate is much faster than computing an optimal open-loop solution with the PMP conditions, which took about 110 s as mentioned above.

Nonlinear heat equation

The third model problem we consider is a nonlinear heat equation (NHE) of Zeldovich type [56, p. 2], which we adapted from [57]. It is of the form

$$\dot{y}(\xi, t) = \alpha \Delta y(\xi, t) + \beta(y^2(\xi, t) - y^3(\xi, t)) + \Xi(\xi)u(\xi, t)$$

for $(\xi, t) \in \Omega \times [0, \infty)$ with the boundary conditions

$$\partial_n y(\xi, t) = 0 \text{ for } (\xi, t) \in \partial\Omega \times [0, \infty) \text{ and } y(\xi, 0) = y_0(\xi) \text{ for } \xi \in \Omega,$$

$\Omega = (0, 1) \times (0, 1)$, $\alpha = 5$ and $\beta = 0.5$. Here, Ξ is the indicator function of the set $[0.25, 0.75] \times [0.25, 0.75]$. The finite difference method can be used to semi-discretise the NHE. This leads to an ODE system involving the matrix $A \in \mathbb{R}^{N \times N}$, which is the discretized version of the operator $\alpha \Delta$ and the matrix $B \in \mathbb{R}^{N \times M}$, that is the discretized version of Ξ . The quadratic and cubic terms become a component-wise application of this operation on the state. For the experiments, we choose $N = 100$. This gives $M = 36$. The system has a stable equilibrium point in the constant one

function and an unstable equilibrium point in the constant zero function. The goal of the control is to optimally steer the system to zero. We fix the following OCP:

$$\min_{\mathbf{u} \in \mathcal{U}_\infty} \int_0^\infty \|\mathbf{x}(s)\|_2^2 + 10^{-3} \|\mathbf{u}(s)\|_2^2 ds$$

subject to $\dot{\mathbf{x}}(s) = \mathbf{A}\mathbf{x}(s) + \beta(\mathbf{x}^2(s) - \mathbf{x}^3(s)) + \mathbf{B}\mathbf{u}(s)$ and $\mathbf{x}(0) = \mathbf{x}_0$.

We choose the space-continuous version of the set of initial states as

$$\{a \sin(b\pi \xi_1)^2 \sin(c\pi \xi_2)^2 + d \sin(e\pi \xi_1^2)^2 \sin(f\pi \xi_2)^2 \mid a, d \in [-0.25, 0.5] \text{ and } c, b, e, f \in \{1, 2\}\},$$

since the functions satisfy the Neumann boundary condition. We have to make this restriction for dimensionality reasons. Then, the set \mathcal{A} is given by the spatially discretized version of these functions according to the finite difference grid, where the parameters a and d are uniformly sampled from $[-0.25, 0.5]^2$ with 25 points in each direction. A meaningful visualization of the training and test data sets, as well as the centers selected for each of the two Hermite surrogates, is difficult for this model problem. For illustrative purposes, Fig. 4 (left) only shows the course of the first optimal trajectory in the training data set. The latter consists of 250 trajectories, with a cover distance smaller than $\epsilon_{\text{tol,d}}/C_{\text{max},\mathcal{A}} = 3.956 \cdot 10^{-2}$. Computing an optimal trajectory with PMP conditions took about 344 s. The maximum deviation of the optimal trajectory from the HJB equation is smaller than 10^{-10} . The cross validation with $s = 10$ resulted in $\gamma_H = 0.002$ for the Hermite surrogate and $\gamma_{SH} = 0.02$ for the structured Hermite surrogate. Again, we use 200 centers for both model problems. For the Hermite kernel surrogate, the training phase took 570 s for the parameter γ_H and the relative training error fell below $\epsilon_{\text{tol,f}}/C_{\text{max},v} = 7.326 \cdot 10^{-4}$. For the structured

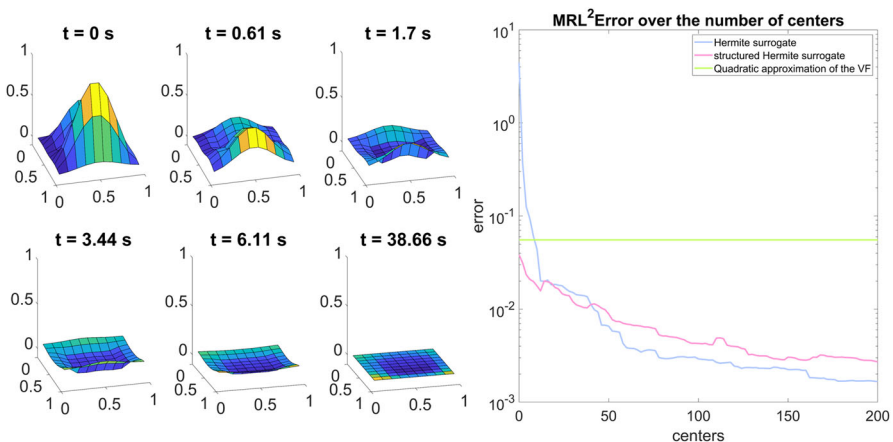


Fig. 4 The left diagram shows the first trajectory in the training data set on six time instants. The right diagram presents a semilog plot of the MRL^2 Error for the three types of surrogate over the number of centers selected

Hermite surrogate, the relative training error is less than $\epsilon_{\text{tol,f}}/C_{\text{max},v} = 2.119 \cdot 10^{-3}$. This took 542 s to train.

In terms of the MRL^2Error , both surrogates quickly give better results than the quadratic approximation to the VF, which delivers $5.556 \cdot 10^{-2}$. In this model problem, the latter is constructed by linearizing the ODE and solving the resulting algebraic Riccati equation. As with all model problems, the structured Hermite surrogate is better at the beginning, but similar to the first model problem, it is caught up by the Hermite surrogate. However, the structured Hermite surrogate gives a similar MRL^2Error curve whose minimum value is $2.713 \cdot 10^{-3}$. The minimum MRL^2Error value for the Hermite surrogate is $1.658 \cdot 10^{-3}$. It took on average 3.5 s to compute a surrogate-controlled trajectory with an initial state in the test data set for the Hermite surrogate and 3.6 s for the structured Hermite surrogate, which again is much less than the 344 s for computing an optimal open-loop controlled solution.

6 Conclusion

A key element of our data-based approach is the generation of high-quality VF data. This is achieved by a method that solves the infinite time horizon problem by transforming the integration domain and adjusting the boundary conditions accordingly. An indicator that this method works very well is the small deviation of the data from the HJB equation of the infinite time horizon OCP. In fact, this was so small for some model problems that one could also interpret it as a collocation approach. Another aspect that makes the VKOGA for Hermite interpolation possible in the first place is the matrix-free approach. Here, it should be emphasized that the matrix–vector multiplication can be reduced from $O(N|\mathcal{D}|n^2)$ to $O(|\mathcal{D}|n^2)$. This allows medium-dimensional problems, such as the NHE model problem, to be considered and the use of the selection criterion in the VKOGA algorithm, without incurring extreme runtimes. As outlook, to work with high-dimensional problems, one could use MOR. In the numerical experiments, we mainly discussed the quality of control by the surrogate via the MRL^2Error . It should be noted that the error between the real VF and the surrogate could accumulate at each control step in the surrogate closed-loop control. However, this is not observed in the numerical experiments, where not only the stability of this closed-loop is seen, but also that the solutions are nearly optimal. Moreover, these surrogate-controlled solutions can be computed much faster than optimal open-loop solutions when the surrogate is available, as the runtimes show, leading to a significant offline-online decomposition. Furthermore, the numerical experiments show that the use of the context-aware structured Hermite surrogate is worth trying when a quadratic approximation to the VF is available. This is motivated by the result for the GST model problem. In general, this data-based approach could be extended to OCPs with finite time horizons, such as those found in MPC. In this, the VF is dependent on the initial time, which raises the question of how the surrogate should be designed to deal with this additional time variable.

Funding Open Access funding enabled and organized by Projekt DEAL. The authors gratefully acknowledge the financial support of this project by the International Research Training Group 2198 (IRTG) “Soft

Tissue Robotics.” Further, we thank the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) for supporting this work by funding—EXC2075—390740016 under Germany’s Excellence Strategy.

Declarations

Conflict of interest The authors declare no competing interests.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Sethi, S.P.: Optimal control theory: applications to management science and economics. Springer (2021). <https://doi.org/10.1007/978-3-030-91745-6>
2. Dmitruk, A.V., Kuz’kina, N.V.: Existence theorem in the optimal control problem on an infinite time interval. *Math. Notes* **78**(3), 466–480 (2005). <https://doi.org/10.1007/s11006-005-0147-3>
3. Bellman, R.E.: Adaptive control processes: a guided tour. Princeton University Press (1961)
4. Falcone, M., Ferretti, R.: Semi-Lagrangian approximation schemes for linear and Hamilton—Jacobi equations. Society for Industrial and Applied Mathematics, Philadelphia, PA (2013). <https://doi.org/10.1137/1.9781611973051>
5. Bokanowski, O., Garcke, J., Griebel, M., Klompaker, I.: An adaptive sparse grid semi-Lagrangian scheme for first order Hamilton-Jacobi Bellman equations. *J. Sci. Comput.* **55**(3), 575–605 (2013). <https://doi.org/10.1007/s10915-012-9648-x>
6. Alla, A., Falcone, M., Saluzzi, L.: An efficient DP algorithm on a tree-structure for finite horizon optimal control problems. *SIAM J. Sci. Comput.* **41**(4), 2384–2406 (2019). <https://doi.org/10.1137/18M1203900>
7. Alla, A., Falcone, M., Saluzzi, L.: A tree structure algorithm for optimal control problems with state constraints. *Rendiconti di Matematica e delle sue Applicazioni* **5**, 193–221 (2020)
8. Alla, A., Saluzzi, L.: A HJB-POD approach for the control of nonlinear PDEs on a tree structure. *Applied Numerical Mathematics* **155** (2019). <https://doi.org/10.1016/j.apnum.2019.11.023>
9. Bellman, R.: A Markovian decision process. *Journal of Mathematics and Mechanics* **6**(5), 679–684 (1957)
10. Falcone, M.: A numerical approach to the infinite horizon problem of deterministic control theory. *Appl. Math. Optim.* **15**(1), 1–13 (1987). <https://doi.org/10.1007/BF01442644>
11. Alla, A., Oliveira, H., Santin, G.: HJB-RBF based approach for the control of PDEs. *J. Sci. Comput.* **96**(1) (2023). <https://doi.org/10.1007/s10915-023-02208-3>
12. Heydari, A.: Revisiting approximate dynamic programming and its convergence. *IEEE Trans. Cybern.* **44**(12), 2733–2743 (2014). <https://doi.org/10.1109/TCYB.2014.2314612>
13. Kamalapurkar, R., Walters, P., Rosenfeld, J., Dixon, W.: Reinforcement learning for optimal feedback control: a Lyapunov-based approach (communications and control engineering). Springer (2018). 3319783831
14. Bellman, R.E.: A dynamic programming. Princeton University Press (1957)
15. Kalise, D., Kunisch, K.: Polynomial approximation of high-dimensional Hamilton-Jacobi-Bellman equations and applications to feedback control of semilinear parabolic PDEs. *SIAM J. Sci. Comput.* **40**(2), 629–652 (2018). <https://doi.org/10.1137/17M1116635>
16. Kalise, D., Kundu, S., Kunisch, K.: Robust feedback control of nonlinear PDEs by numerical approximation of high-dimensional Hamilton-Jacobi-Isaacs equations. *SIAM J. Appl. Dyn. Syst.* **19**(2), 1496–1524 (2020). <https://doi.org/10.1137/19M1262139>

17. Dolgov, S., Kalise, D., Kunisch, K.: Tensor decomposition methods for high-dimensional Hamilton-Jacobi-Bellman equations. *SIAM J. Sci. Comput.* **43**(3), 1625–1650 (2021). <https://doi.org/10.1137/19M1305136>
18. Alla, A., Haasdonk, B., Schmidt, A.: Feedback control of parametrized PDEs via model order reduction and dynamic programming principle. *Adv. Comput. Math.* **46**(1), 9 (2020). <https://doi.org/10.1007/s10444-020-09744-8>
19. Oster, M., Sallandt, L., Schneider, R.: Approximating optimal feedback controllers of finite horizon control problems using hierarchical tensor formats. *SIAM J. Sci. Comput.* **44**(3), 746–770 (2022). <https://doi.org/10.1137/21m1412190>
20. Eigel, M., Schneider, R., Sommer, D.: Dynamical low-rank approximations of solutions to the Hamilton–Jacobi–Bellman equation. *Num. Linear Algebra Appl.* **30**(3) (2022). <https://doi.org/10.1002/nla.2463>
21. Saridis, G.N., Lee, C.G.: An approximation theory of optimal control for trainable manipulators. *IEEE Trans. Syst. Man Cybern.* **9**(3), 152–159 (1979). <https://doi.org/10.1109/TSMC.1979.4310171>
22. Alla, A., Falcone, M., Kalise, D.: An efficient policy iteration algorithm for dynamic programming equations. *SIAM J. Sci. Comput.* **37**(1), 181–200 (2015). <https://doi.org/10.1137/130932284>
23. Grüne, L., Pannek, J.: Nonlinear model predictive control: theory and algorithms. Springer, Communications and Control Engineering (2011)
24. Freeman, R.A., Kokotovic, P.V.: Optimal nonlinear controllers for feedback linearizable systems. In: Proceedings of the American Control Conference, vol. 4, pp. 2722–2726 (1995). IEEE Inc
25. Schmidt, A., Haasdonk, B.: Reduced basis approximation of large scale parametric algebraic Riccati equations. *ESAIM: Control Optim. Calculus Var.* **24**(1), 129–151 (2018)
26. Breiten, T., Kunisch, K., Pfeiffer, L.: Taylor expansions of the value function associated with a bilinear optimal control problem. *Annales de l’Institut Henri Poincaré C, Analyse non linéaire* **36**(5), 1361–1399 (2019). <https://doi.org/10.1016/j.anihpc.2019.01.001>
27. Çimen, T.: State-dependent Riccati equation (SDRE) control: a survey. *IFAC Proc. Vol.* **41**(2), 3761–3775 (2008). <https://doi.org/10.3182/20080706-5-KR-1001.00635>. 17th IFAC World Congress
28. Albi, G., Bicego, S., Kalise, D.: Gradient-augmented supervised learning of optimal feedback laws using state-dependent Riccati equations. *IEEE Control Syst. Lett.* **6**, 836–841 (2022). <https://doi.org/10.1109/LCSYS.2021.3086697>
29. Kunisch, K., Walter, D.: Optimal feedback control of dynamical systems via value-function approximation (2023). <https://doi.org/10.48550/ARXIV.2302.13122>
30. Kunisch, K., Walter, D.: Semiglobal optimal feedback stabilization of autonomous systems via deep neural network approximation. *ESAIM: Control Optim. Calculus Var.* **27**, 16. <https://doi.org/10.1051/cocv/2021009>
31. Kunisch, K., Vásquez-Varas, D.: Optimal polynomial feedback laws for finite horizon control problems (2023). <https://doi.org/10.48550/ARXIV.2302.09878>
32. Kunisch, K., Vásquez-Varas, D., Walter, D.: Learning optimal feedback operators and their polynomial approximation (2022). <https://doi.org/10.48550/ARXIV.2208.14120>
33. Rao, A.V.: A survey of numerical methods for optimal control. *Adv. Astronaut. Sci.* **135**(1), 497–528 (2009)
34. Pontryagin, L.S., Boltyanskii, V.G., Gamkrelidze, R.V., Mishchenko, E.F.: The mathematical theory of optimal processes. Interscience (1962)
35. Nakamura-Zimmerer, T., Gong, Q., Kang, W.: Adaptive deep learning for high-dimensional Hamilton-Jacobi-Bellman equations. *SIAM J. Sci. Comput.* **43**(2), 1221–1247 (2021). <https://doi.org/10.1137/19M1288802>
36. Azmi, B., Kalise, D., Kunisch, K.: Optimal feedback law recovery by gradient-augmented sparse polynomial regression. *J. Mach. Learn. Res.* **22**(48), 1–32 (2021)
37. Kang, W., Wilcox, L.C.: Mitigating the curse of dimensionality: sparse grid characteristics method for optimal feedback control and HJB equations. *Comput. Optim. Appl.* **68**(2), 289–315 (2017). <https://doi.org/10.1007/s10589-017-9910-0>
38. Schmidt, A., Haasdonk, B.: Data-driven surrogates of value functions and applications to feedback control for dynamical systems. *IFAC-PapersOnLine* **51**(2), 307–312 (2018). 9th Vienna International Conference on Mathematical Modelling
39. Ehring, T., Haasdonk, B.: Feedback control for a coupled soft tissue system by kernel surrogates. In: Proceedings of COUPLED 2021. Scipedia (2021). <https://doi.org/10.23967/coupled.2021.026>

40. Ehring, T., Haasdonk, B.: Greedy sampling and approximation for realizing feedback control for high dimensional nonlinear systems. *IFAC-PapersOnLine* **55**(20), 325–330 (2022). <https://doi.org/10.1016/j.ifacol.2022.09.116>. 10th Vienna International Conference on Mathematical Modelling MATHMOD 2022
41. Wirtz, D., Haasdonk, B.: A vectorial kernel orthogonal greedy algorithm. *Dolomites Res. Note Approximation* **6**, 83–100 (2013)
42. Wenzel, T., Santin, G., Haasdonk, B.: Analysis of target data-dependent greedy kernel algorithms: convergence rates for f - f - P - and f/P -greedy. *Constructive Approximation*, 1–30 (2022)
43. Bardi, M., Capuzzo-Dolcetta, I.: *Optimal control and viscosity solutions of Hamilton-Jacobi-Bellman equations*. Modern Birkhäuser Classics, Birkhäuser Boston (2008)
44. Benveniste, L.M., Scheinkman, J.A.: On the differentiability of the value function in dynamic models of economics. *Econometrica* **47**(3), 727–732 (1979)
45. Aseev, S.M., Veliov, V.M.: Another view of the maximum principle for infinite-horizon optimal control problems in economics. *Russ. Math. Surv.* **74**(6), 963 (2019)
46. Michel, P.: On the transversality condition in infinite horizon optimal problems. *Econometrica: Journal of the Econometric Society*, 975–985 (1982)
47. Seierstad, A., Sydsaeter, K.: Sufficient conditions in optimal control theory. *International Economic Review* **18**(2), 367. <https://doi.org/10.2307/2525753>
48. Fahroo, F., Ross, I.M.: Pseudospectral methods for infinite-horizon nonlinear optimal control problems. *J. Guid. Control. Dyn.* **31**(4), 927–936 (2008)
49. Garg, D., Hager, W.W., Rao, A.V.: Pseudospectral methods for solving infinite-horizon optimal control problems. *Automatica* **47**(4), 829–837 (2011)
50. De Marchi, S., Schaback, R., Wendland, H.: Near-optimal data-independent point locations for radial basis function interpolation. *Adv. Comput. Math.* **23**, 317–330 (2005)
51. Wendland, H.: *Scattered data approximation vol. 17*. Cambridge university press (2004)
52. Sontag, E.D.: *Mathematical control theory: deterministic finite dimensional systems vol. 6*. Springer (2013)
53. Kirszbraun, M.: Über die zusammenziehende und Lipschitzsche Transformationen. *Fundam. Math.* **22**(1), 77–108 (1934)
54. Gronwall, T.H.: Note on the derivatives with respect to a parameter of the solutions of a system of differential equations. *Annals of Mathematics*, 292–296 (1919)
55. Khalil, H.: *Nonlinear systems: Pearson New*, vol. 3, International Pearson Education, Limited, Harlow (2013)
56. Gilding, B.H., Kersner, R.: *Travelling waves in nonlinear diffusion-convection reaction vol. 60*. Springer (2004)
57. Alla, A., Kalise, D., Simoncini, V.: State-dependent Riccati equation feedback stabilization for nonlinear PDEs. *Adv. Comput. Math.* **49**(1), 9 (2023). <https://doi.org/10.1007/s10444-022-09998-4>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.