Received: Added at production | Revised: Added at production | Accepted: Added at production

DOI: xxx/xxxx

RESEARCH ARTICLE

Synthesis of control Lyapunov functions and stabilizing feedback strategies using exit-time optimal control Part II: Numerical approach

Ivan Yegorov*[1] | Peter M. Dower[2] | Lars Grüne[3]

[1]Department of Mathematics, North Dakota State University, North Dakota, USA
[2]Department of Electrical and Electronic Engineering, University of Melbourne, Victoria, Australia
[3]Chair of Applied Mathematics, University of Bayreuth, Bavaria, Germany

Correspondence
*Ivan Yegorov (also known as Egorov).
Email: ivanyegorov@gmail.com

Abstract

This paper continues the study [1] and develops a curse-of-dimensionality-free numerical approach to feedback stabilization, whose theoretical foundation was built in [1] and involved the characterization of control Lyapunov functions (CLFs) via exit-time optimal control. First, we describe an auxiliary linearization based technique for the construction of a local CLF and discuss how to determine its appropriate sublevel set that can serve as the terminal set in the exit-time optimal control problem leading to a global or semi-global CLF. Next, the curse of complexity is addressed with regard to the approximation of CLFs and associated feedback strategies in high-dimensional regions. The goal is to enable for efficient model predictive control implementations with essentially faster (though less accurate) online policy updates than in case of solving direct or characteristics based nonlinear programming problems for each sample instant. We propose a computational approach that combines gradient enhanced modifications of the Kriging and inverse distance weighting frameworks for scattered grid interpolation. It in particular allows for convenient offline inclusion of new data to improve obtained approximations (machine learning can be used to select relevant new sparse grid nodes). Moreover, our method is designed so as to a priori return proper values of the CLF interpolant and its gradient on the entire terminal set of the considered exit-time optimal control problem. Supporting numerical simulation results are also presented.

KEYWORDS:
control Lyapunov functions, feedback stabilization, exit-time optimal control, curse of dimensionality, curse of complexity, method of characteristics, direct collocation, model predictive control.

1 | INTRODUCTION

The previous study [1] has developed theoretical results regarding the use of exit-time optimal control to characterize control Lyapunov functions (CLFs) for deterministic nonlinear systems described by ordinary differential equations. As noted in [1, Conclusion], such characterizations can help to attenuate the curse of dimensionality, but the curse of complexity remains a significant issue for obtaining global or semi-global approximations of CLFs and associated feedback strategies in high-dimensional regions of the state space. This is particularly crucial for model predictive control (MPC) algorithms [2–8].

In general, MPC schemes use measurements or estimates of state variables at sample time instants in order to update control strategies (e. g., via optimization). In standard online (real-time) MPC, an optimal open-loop control problem is numerically solved for each sample instant, and a piece of the approximated optimal open-loop policy is implemented up to the next sample instant, resulting in a quasi-feedback control law. If the sampling time, i. e., the time between two consecutive optimizations, is small, such online updates may not be feasible, because the computational resources available for a real-time control process may not be powerful enough. However, it is more common to have access to powerful computing architectures for preparatory offline model training. A remedy may therefore be provided by methods that precompute additional information (possibly involving costate data) in offline mode and then construct an online feedback control strategy via interpolation or regression [6–8]. They are typically referred to as explicit MPC approaches [9], and one also has to emphasize their connection with the broad areas of dynamic programming, Hamilton–Jacobi–Bellman (HJB) equations, as well as necessary and sufficient conditions for control optimality [10–16]. For high-dimensional problems, these methods are usually demanding in terms of both computational effort and storage requirement, which leads to the curse of dimensionality as well as the curse of complexity (see the related discussion in [1, Introduction]).

It is thus relevant to design approaches that allow for attenuating the aforementioned computational issues and accelerating online control or costate updates in MPC, whilst lowering the accuracy only within acceptable limits. One group of such frameworks involves sparse grids, so that a sparse grid along with suitable data on it is generated beforehand in offline mode, and some interpolation or regression scheme using this data is then executed online. In [6], this has been implemented with Smolyak type sparse grids having predetermined nodes in a specific hierarchical configuration. However, scattered grids with associated interpolation or regression techniques [17–20] are more flexible, since new nodes can be directly included in such a grid to improve the approximation. Furthermore, appropriate new nodes may be selected via machine learning. This is often done in offline mode, while online computations exploit already trained models.

The works [7, 8, 21, 22] propose some other data-driven methods for fast MPC computation that employ neural networks and polynomial regression. These methods have been successfully applied to a number of initial-value problems for HJB equations (those problems describe value functions in case of no terminal state constraints, as opposed to the stabilization problems that we treat via exit-time optimal control). Besides, the neural network characterization results of [21] are established for the special case when the Hamiltonian depends only on the costate, but they work even for some particular Hamiltonians which are neither convex nor concave, allowing some differential games to be handled.

The approach developed in the current paper combines gradient enhanced modifications of the Kriging and inverse distance weighting frameworks [17] for scattered grid interpolation of the value functions in exit-time optimal control problems that can describe CLFs. Note that gradient (costate) enhanced techniques may enable acceptable accuracy with much sparser grids than in case when the gradient information is not taken into account [17–19].

A distinctive feature of our approach is that it constructs multiple gradient enhanced Kriging models on different sparse grids and connects them via a special form of inverse distance weighting (a single interpolation or regression model is expected to work well locally, but it may not provide required performance in a considered semi-global domain of the state space). This in particular guarantees suitable approximations of the value functions (CLFs) and their gradients (costates) on the entire terminal set of cardinality continuum, in addition to desired values at a finite number of sample grid points outside that set, which is substantial for exit-time optimal control problems. Our framework also allows for convenient incorporation of new data in order to improve created interpolants (e. g., within an offline machine learning process). Moreover, other data-driven models, such as neural networks [7, 21, 22], polynomial regression [8], and support vector regression [17], can potentially be used in our general composite model structure instead of Kriging.

The paper is arranged as follows. Section 2 proposes an auxiliary linearization based method to find a quadratic local CLF and discusses how to determine its suitable sublevel set that can then appear as the terminal (target) set in the exit-time optimal control problem leading to a global or semi-global CLF. In Section 3, we construct the aforementioned composite model approach to scattered grid interpolation and describe its use for MPC. Section 4 presents supporting numerical simulation results. The first numerical example is purely illustrative and does not involve MPC. Its purpose is to practically test the theoretical results of the first part [1] as applied to a simple nonlinear control system with two-dimensional state space. This system can be treated analytically to some extent, so that the exact and numerical solutions can be easily compared with each other. The characteristics based numerical method we use in the first example is also illustrative and requires further development for more complex systems, so its description is put in Appendix. The second example is essentially more complicated and considers a nonlinear aircraft dynamic model with six-dimensional state space. We use the numerical frameworks of Sections 2 and 3 to implement

MPC and attenuate the curse of dimensionality with the curse of complexity in this example. Finally, concluding remarks are given in Section 5.

In this paper, we use the same notation and definitions as indicated in [1, Introduction and Subsection 2.1].

2 | CONSTRUCTION OF A LOCAL CLF VIA LINEARIZATION

The theoretical results of [1, Section 2] were obtained in the presence of a local CLF. A typical perspective on local CLF construction is to consider first the ideal case of unconstrained control inputs and search for a related CLF using, e. g., the analytical frameworks of [23, §9.4] or [24, Chapter 5], so that it could then serve as a local CLF in case of pointwise control constraints. This may in general be a challenging task. Moreover, the local or global asymptotic stabilization properties of certain feedbacks for a number of well-recognized mechanical systems are established via nonstrict Lyapunov functions, such that the right-hand sides in the associated infinitesimal decrease conditions vanish not only at the origin [25–30]. However, the strictness appears in the original CLF concept (recall [1, Definition 2.5]) as well as in the sufficient conditions for local asymptotic null-controllability given in [31, Section 2] and used in [1, Remark 2.10].

In this section, we adopt [1, Assumption 2.1] for the system

$$\begin{cases} \dot{x}(t) \;=\; f(x(t),u(t)), \quad t \geqslant 0, \\ x(0) \;=\; x_0 \;\in\; G, \\ u(\cdot) \;\in\; \mathcal{U} \;\stackrel{\text{def}}{=}\; L^{\infty}_{\text{loc}}([0,+\infty),\, U), \end{cases} \tag{1}$$

and propose a linearization based numerical technique to build quadratic local CLFs under some additional conditions, with the considerations of [3, Section 3] serving as an important motivation. Those considerations can also be employed to construct quadratic local CLFs under the same assumptions. Although the technique presented in the current section is less elegant, it is more straightforward to use and does not restrict the right-hand sides in the decrease conditions for the resulting local CLFs necessarily to quadratic functions (in contrast to the approach of [3, Section 3]).

Assumption 2.1. In addition to [1, Assumption 2.1], suppose that $0_m \in \operatorname{int} U$, $f(0_n, 0_m) = 0_n$, the function $f(\cdot,\cdot)$ is continuously differentiable, and the linearization

$$\begin{cases} \dot{x}(t) \;=\; A\,x(t) \;+\; B\,u(t), \quad t \geqslant 0, \\ x(0) \;=\; x_0 \;\in\; G, \\ u(\cdot) \;\in\; \mathcal{U} \;\stackrel{\text{def}}{=}\; L^{\infty}_{\text{loc}}([0,+\infty),\, U), \\ A \;\stackrel{\text{def}}{=}\; \mathrm{D}_x f(0_n,0_m) \;\in\; \mathbb{R}^{n\times n}, \quad B \;\stackrel{\text{def}}{=}\; \mathrm{D}_u f(0_n,0_m) \;\in\; \mathbb{R}^{n\times m}, \end{cases} \tag{2}$$

of the system 1 is asymptotically null-controllable.

A particular case of asymptotic null-controllability is exact null-controllability with the criterion

$$\operatorname{rank}\,\left[B,\, AB,\, A^2B,\, \ldots,\, A^{n-1}B\right] \;=\; n \tag{3}$$

(see, e. g., [32, §3.2, Theorem 3]).

Due to [32, §5.8, Theorem 19], Assumption 2.1 ensures the existence of a positive definite matrix $P \in \mathbb{R}^{n\times n}$ and a matrix $S \in \mathbb{R}^{m\times n}$ such that the functions

$$\check{V}(x) \;=\; \langle Px,\, x\rangle\,, \quad \check{u}(x) \;=\; Sx \quad \forall x \in \mathbb{R}^n \tag{4}$$

are respectively a local quadratic CLF and a locally stabilizing linear feedback for (1) in some neighborhood of the origin 0_n. One can search for such a neighborhood in the form of a sublevel set of $\check{V}(\cdot)$.

In line with [32, §5.8, Proof of Theorem 19], the control matrix S is selected so that $A + BS \in \mathbb{R}^{n\times n}$ is Hurwitz, and P is a unique positive definite solution of the (continuous Lyapunov) matrix equation

$$(A+BS)^{\top} P \;+\; P\,(A+BS) \;=\; -I_{n\times n}. \tag{5}$$

Note that $-I_{n\times n}$ in the right-hand side of (5) can be replaced with any negative definite matrix of the same size.

The gradient of $\check{V}(\cdot)$ is given by

$$\mathrm{D}\check{V}(x) \;=\; 2\,Px \quad \forall x \in \mathbb{R}^n,$$

and the sublevel sets

$$\breve{\Omega}_r \stackrel{\text{def}}{=} \{x \in \mathbb{R}^n : \breve{V}(x) \leqslant r\} \quad \forall r > 0 \tag{6}$$

are closed ellipsoidal domains in $\mathbb{R}^n$. Since $\breve{V}(\cdot)$ is a local CLF for (1), there exists a level $c' > 0$ satisfying

$$\begin{gathered} \breve{\Omega}_{c'} \subset G, \\ \min_{w \in U} \left\langle \mathrm{D}\breve{V}(x),\ f(x,w) \right\rangle < 0 \quad \forall x \in \breve{\Omega}_{c'} \setminus \{0_n\}. \end{gathered} \tag{7}$$

Then one can take

$$\Omega = \operatorname{int} \breve{\Omega}_{c'}, \quad c \in (0, c'), \quad \Omega_c = \breve{\Omega}_c \tag{8}$$

and select a local CLF and a locally stabilizing feedback as

$$\begin{aligned} &V_{\mathrm{loc}}(x) = \breve{V}(x), \\ &u_{\mathrm{loc}}(0_n) = 0_m, \\ &u_{\mathrm{loc}}(x) = \breve{u}(x) = Sx \quad \text{if} \quad Sx \in U \quad \text{and} \quad \langle \mathrm{D}V_{\mathrm{loc}}(x),\ f(x,Sx) \rangle < 0, \\ &u_{\mathrm{loc}}(x) \in \operatorname{Arg}\min_{w \in U} \langle \mathrm{D}V_{\mathrm{loc}}(x),\ f(x,w) \rangle \quad \text{otherwise}, \\ &\forall x \in \Omega, \end{aligned} \tag{9}$$

so that [1, Assumption 2.8] is satisfied. The greater such a level c, the wider the target set Ω_c, and, hence, the easier to numerically solve the exit-time optimal control problem [1, (23)]. This leads to the problem of finding the supremum $c_{\sup}$ of all suitable levels, which can be practically treated by testing the nodes of a uniform grid on the interval $[0, \tilde{c}]$ with a sufficiently small stepsize and a sufficiently large right endpoint $\tilde{c} > 0$. For each node, an appropriate finite-dimensional optimization problem should be numerically solved. Finally, it is reasonable to select c somewhat lower than $c_{\sup}$, so that $\min_{w \in U} \langle \mathrm{D}V_{\mathrm{loc}}(x),\ f(x,w) \rangle$ is not very close to zero at states x near the boundary $l_c = \partial\Omega_c$.

Note also that the lengths of the principal axes of the ellipsoid (6) with matrix P and level r are equal to $\sqrt{r/\lambda_i(P)}$, $i = \overline{1,n}$, where $\lambda_i(P)$, $i = \overline{1,n}$, are the eigenvalues of P (which are all real and positive due to the positive definiteness of P). This can be verified via the principal axis theorem (see, e. g., [34, p. 120] and [35, §3.2]).

Remark 2.2. The matrix $P \in \mathbb{R}^{n \times n}$ in (4) can be chosen as a unique positive definite solution of the algebraic Riccati equation

$$A^\top P + PA - PBR^{-1}B^\top P + Q = 0_{n \times n} \tag{10}$$

with arbitrary positive definite matrices $Q \in \mathbb{R}^{n \times n}$, $R \in \mathbb{R}^{m \times m}$, and (4) with $S = -R^{-1}B^\top P$ gives the value function and optimal feedback strategy for the infinite-horizon linear-quadratic optimal control problem

$$\begin{cases} \dot{x}(t) = A\,x(t) + B\,u(t), \quad t \geqslant 0, \\ x(0) = x_0 \in \mathbb{R}^n, \\ u(\cdot) \in L^\infty_{\mathrm{loc}}([0,+\infty), \mathbb{R}^m), \\ \displaystyle\int_0^{+\infty} (\langle Q\,x(t),\ x(t) \rangle + \langle R\,u(t),\ u(t) \rangle)\,\mathrm{d}t \longrightarrow \min, \end{cases} \tag{11}$$

with unconstrained control inputs. In order to verify the existence and uniqueness of such a nonnegative definite matrix P, it suffices to use [33, Part III, Theorem 1.4] and to note that the conditions of both items of [33, Part III, Theorem 1.5] are satisfied (indeed, the pair (A, B) is stabilizable, Q is positive definite, $Q^{1/2}$ is nondegenerate, and the pair $\left(A, Q^{1/2}\right)$ is observable). The positive definiteness of P immediately follows from [32, §5.7, Theorem 18] if one rewrites the algebraic Riccati equation (10) as

$$\left(A - BR^{-1}B^\top P\right)^\top P + P\left(A - BR^{-1}B^\top P\right) = -Q - PBR^{-1}B^\top P$$

and uses the fact that $A + BS = A - BR^{-1}B^\top P$ is Hurwitz, Q, R are positive definite, and $PBR^{-1}B^\top P$ is nonnegative definite.

It is typical to choose the matrices Q, R in the diagonal form $Q = \varkappa_1 I_{n \times n}$, $R = \varkappa_2 I_{m \times m}$, $\varkappa_1 > 0$, $\varkappa_2 > 0$. Each pair $(\varkappa_1, \varkappa_2) \in (0, +\infty)^2$ then determines a unique positive definite solution $P_{\varkappa_1,\varkappa_2}$ of (10). A certain pair $(\varkappa_1, \varkappa_2)$ can be selected as follows:

- first, take a finite number (grid) of sample pairs $(\varkappa_1, \varkappa_2)$;

- out of this collection, pick the pair that maximizes the minimal principal axis

$$\sqrt{\frac{\gamma_{\sup,x_1,x_2}}{\max\limits_{i\in\{1,2,\ldots,n\}}\lambda_i(P_{x_1,x_2})}} \tag{12}$$

of the ellipsoid with matrix P_{x_1,x_2} and level

$$\gamma_{\sup,x_1,x_2} = \sup\Big\{\gamma>0:\ \text{the ellipsoid with matrix } P_{x_1,x_2} \text{ and level } \gamma \text{ is contained in } G, \text{ and } \min_{w\in U}\big\langle P_{x_1,x_2}x,\ f(x,w)\big\rangle < 0 \ \text{ for all } x\neq 0_n \text{ lying in this ellipsoid}\Big\} \tag{13}$$

(this is the supremum of all acceptable levels in terms of the CLF decrease condition).

Maximization of the minimal principal axis means that one wants to increase the closest distance from the boundary of the target ellipsoid to the origin. Other selection criteria may also be used. For example, one can maximize the ellipsoidal volume which is proportional to the product of the principal axes lengths [35, §3.2]. □

3 | COMPOSITE MODEL APPROACH TO SCATTERED GRID INTERPOLATION AND ITS USE FOR MPC

This section develops a composite model approach to scattered grid interpolation with the aim of attenuating the curse of dimensionality as well as the curse of complexity for MPC application to exit-time problems that characterize CLFs.

Consider an exit-time optimal control problem that is stated either as in [1, Section 2] (with a local CLF involved) or as in [1, Section 3] (without using any local CLF). We adopt also the following notation:

- $V(\cdot)$ is the value function;
- $\Phi = \{x\in\mathbb{R}^n:\ \varphi(x)\leqslant\gamma\}$ is the terminal (target) set;
- $\partial\Phi = \{x\in\mathbb{R}^n:\ \varphi(x)=\gamma\}$ is the boundary of Φ;
- $V_{\partial\Phi}$ is the constant value of $V(\cdot)$ on $\partial\Phi$.

In the setting of [1, Section 2], $V(\cdot)$ has the same meaning, $\Phi=\Omega_c$, $\varphi(\cdot)=V_{\mathrm{loc}}(\cdot)$, and $\gamma=V_{\partial\Phi}=c$. In the setting of [1, Section 3], one has $V(\cdot)=V_\delta(\cdot)$, $\Phi=\overline{\mathrm{B}}_\delta(0_n)$, $\varphi(x)=\|x\|^2$, $\gamma=\delta^2$, and $V_{\partial\Phi}=0$.

The goal is to design a method that reasonably approximates the value function and its gradient (costate) in a relatively large bounded domain containing the terminal set and works substantially faster (though less accurately) in real-time MPC implementations than online solution of direct or characteristics based nonlinear programming problems. The latter then have to be treated in real time only when a current state leaves this bounded domain (which is in general not invariant with respect to an implemented MPC strategy).

We suppose that the conditions of [1, Theorems 2.21, 2.29] or [1, Theorems 3.9, 3.13] are satisfied (in the second case, the Petrov condition in [1, Item 4 of Theorem 3.9] is also assumed to hold). This in particular ensures that the value function is locally Lipschitz continuous in $\mathcal{D}_0$ and hence differentiable almost everywhere in $\mathcal{D}_0$ (with respect to the Lebesgue measure in $\mathbb{R}^n$). For instance, the value function is typically not differentiable at states on the boundary of the terminal set. Some manifolds of nondifferentiability with Lebesgue measure zero may also appear outside the terminal set.

Furthermore, let $\varphi(\cdot)$ be a continuously differentiable and convex function. Then its sublevel set Φ is convex. One also has $\mathrm{D}\varphi(x)\neq 0_n$ for any $x\in\partial\Phi$, and $\partial\Phi$ is therefore a connected regular hypersurface in $\mathbb{R}^n$ (recall the properties of a local CLF in [1, Assumption 2.8] as well as the choice of the equivalents of Φ, $\varphi(\cdot)$ and γ in the setting of [1, Section 3]).

If the value function $V(\cdot)$ grows rather fast when moving away from the terminal set, the corresponding approximation (interpolation) errors may also rise dramatically. In order to mitigate this negative effect, it may be reasonable to consider a transformation $\tilde{V}(\cdot)$ of $V(\cdot)$ with limited growth. A possible choice is $\tilde{V}(\cdot)=\ln(1+V(\cdot))$ (recall that $V(\cdot)$ is nonnegative by definition, since the associated running cost and $V_{\partial\Phi}$ are nonnegative).

3.1 | Model generation and implementation

Kriging can in general be understood as a form of Gaussian process regression, and it can perform interpolation using data on scattered grids [17–20]. In particular, the gradient enhanced Kriging method of [18, 19] incorporates the gradient as a covariable and uses a Bayesian framework so that variograms (i. e., spatial covariance models) and relations between values and gradients are incorporated in the prior, while observation errors are contained in the likelihood.

For Kriging models of scattered grid interpolation, it is often recommended to take quasi–Monte Carlo grids that reduce the clumping (discrepancy) present in standard Monte Carlo grids and can be based on Halton, Sobol, or Faure quasi-random number sequences [20]. Since a single Kriging model may not be enough to give an acceptable performance, and it is also relevant to incorporate gradient (costate) data, we propose to generate multiple gradient enhanced Kriging models [17–19] and combine them via inverse distance weighting. Moreover, the resulting model should properly estimate the value function and costate on the boundary of the terminal set.

Our composite model is constructed as follows.

First, generate a number of mutually disjoint quasi–Monte Carlo grids $\Sigma_i,\ i = \overline{1, N}$, in subregions of a certain bounded domain of the state space. The grids have to lie outside the terminal set Φ; if one generates a grid node in Φ, it has to be excluded. For each remaining grid node, compute the value function $V(\cdot)$ and its gradient (costate) using, e. g., a direct collocation method [36–41]. When solving related nonlinear programming problems, it may be reasonable to approximate a suitable transformation $\tilde{V}(\cdot) = \zeta(V(\cdot))$ of $V(\cdot)$ as noted above. The function $\zeta(\cdot)$ is supposed to have a positive and continuous derivative on $[0, +\infty)$; $\zeta(\rho) = \ln(1 + \rho)$ is a possible transformation with limited growth. In this case, the costate is accordingly transformed:

$$\tilde{V}(\cdot) = \zeta(V(\cdot)), \quad \mathrm{D}\tilde{V}(\cdot) = \zeta'(V(\cdot))\,\mathrm{D}V(\cdot),$$
$$V(\cdot) = \zeta^{-1}\left(\tilde{V}(\cdot)\right), \quad \mathrm{D}V(\cdot) = \frac{1}{\zeta'\left(\zeta^{-1}\left(\tilde{V}(\cdot)\right)\right)}\,\mathrm{D}\tilde{V}(\cdot);$$

for example, if $\tilde{V}(\cdot) = \ln(1+V(\cdot))$, then $\mathrm{D}\tilde{V}(\cdot) = (1/(1+V(\cdot)))\,\mathrm{D}V(\cdot)$, $V(\cdot) = \exp\left(\tilde{V}(\cdot)\right)-1$, and $\mathrm{D}V(\cdot) = \exp\left(\tilde{V}(\cdot)\right)\,\mathrm{D}\tilde{V}(\cdot)$. If such a transformation of $V(\cdot)$ is not needed, we simply set $\zeta(\cdot) \equiv 1$ and $\tilde{V}(\cdot) = V(\cdot)$. One also has to remove the grid nodes for which the output of the numerical optimization process (i. e., estimation of the value function and costate) does not satisfy specified tolerance requirements. For convenience, adopt the same notation $\Sigma_i,\ i = \overline{1, N}$, for the resulting quasi–Monte Carlo grids and let

$$\Sigma_i = \left\{ x^{(ij)} : j = \overline{1, N_i} \right\}, \quad i = \overline{1, N}. \tag{14}$$

Next, for every $i = \overline{1, N}$, construct a smooth and gradient enhanced Kriging model $K_i(\cdot)$ such that $K_i(x)$ and $\mathrm{D}K_i(x)$ interpolate respectively $\tilde{V}(x)$ and $\mathrm{D}\tilde{V}(x)$ at all nodes $x \in \Sigma_i$; interpolation error tolerances can be specified for both $K_i(\cdot)$ and $\mathrm{D}K_i(\cdot)$ when using a gradient enhanced Kriging software package such as [19].

The sparse grids $\Sigma_i,\ i = \overline{1, N}$, and Kriging models $K_i(\cdot),\ i = \overline{1, N}$, can be built in offline mode, i. e., before a real-time MPC implementation. For the further use of a certain form of inverse distance weighting [17, §5], some more data has to be generated in offline mode. Namely, fix a parameter $\nu \in \mathbb{N}$ and determine an influence radius R_{ij} for the j-th node $x^{(ij)}$ of the grid Σ_i so that the closed ball with center at $x^{(ij)}$ and radius R_{ij} contains exactly $\nu + 1$ nodes (including the center) from the union $\bigcup_{k=1}^{N} \Sigma_k$ of all considered grids, $j = \overline{1, N_i},\ i = \overline{1, N}$. Obtain also an influence radius R_0 around the entire terminal set Φ: the closed R_0–neighborhood of Φ should contain exactly ν nodes from $\bigcup_{k=1}^{N} \Sigma_k$. One can select $R_{ij},\ j = \overline{1, N_i},\ i = \overline{1, N}$, and R_0 as the minimal influence radii that satisfy the mentioned conditions.

It is also reasonable to specify the minimal and maximal allowed radii $R_{\min}$ and $R_{\max}$, so that a radius of influence is reset to $R_{\min}$ if it is less than $R_{\min}$ and to $R_{\max}$ if it exceeds $R_{\max}$.

Let us now describe how online (real-time) estimation of $\tilde{V}(\cdot)$ and $\mathrm{D}\tilde{V}(\cdot)$ can be performed.

In addition to the Kriging models $K_i(\cdot),\ i = \overline{1, N}$, one requires a separate model to compute the value function and costate on the boundary $\partial\Phi = \{x \in \mathbb{R}^n : \varphi(x) = \gamma\}$ of the terminal set. Due to the terminal costate and Hamiltonian vanishing conditions in Pontryagin's principle (see [1, Theorem 2.26 and Remark 3.12]), one can estimate the costate at $x \in \partial\Phi$ as $\eta^*(x)\,\mathrm{D}\varphi(x)$, where $\eta^*(x) \in (0, +\infty)$ is the minimal root of the Hamiltonian based function

$$[0, +\infty) \ni \eta \mapsto \mathcal{H}(x,\ \eta\,\mathrm{D}\varphi(x),\ 1) = \min_{u \in U} \{\eta\ \langle \mathrm{D}\varphi(x),\ f(x,u)\rangle + g(x,u)\}, \quad x \in \partial\Phi.$$

This function is positive at $\eta = 0$ (in line with [1, Item 4 of Assumption 2.16 and Item 2 of Assumption 3.5]) and tends to $-\infty$ as $\eta \to +\infty$ (by virtue of the Petrov condition in [1, Proposition 2.12 and Item 4 of Theorem 3.9] as well as the boundedness of $g(\cdot,\cdot)$ on the bounded set $\Phi \times U$), so at least one root exists. There may be multiple roots, but we select only the minimal

one from a heuristic and practical perspective. An open theoretical problem is to derive easily verifiable sufficient conditions for the root uniqueness or, more generally, for the property that the minimal root indeed corresponds to the optimal costate on $\partial\Phi$. Furthermore, we assume that $\eta^*(\cdot)$ is Lipschitz continuous on $\partial\Phi$, even though rigorous verification of this property also remains an open problem.

Consider the boundary conditions

$$\begin{aligned} & V_{\text{appr}}(x) \;=\; V_{\partial\Phi}, \\ & \partial V_{\text{appr}}(x;\xi) \;=\; \langle \eta^*(x)\,\mathrm{D}\varphi(x),\; \xi\rangle \\ & \forall\, \xi \;\in\; \mathbb{R}^n \setminus (\operatorname{int} \mathrm{T}(x;\Phi)) \quad \forall x \in \partial\Phi \end{aligned} \tag{15}$$

for a value function approximation. The second relation in (15) means that $\eta^*(x)\,\mathrm{D}\varphi(x)$ is the outer gradient of $V_{\text{appr}}(\cdot)$ at a boundary point $x \in \partial\Phi$ of Φ. Here $\partial V_{\text{appr}}(x;\xi)$ is the directional derivative of $V_{\text{appr}}(\cdot)$ along a vector $\xi \in \mathbb{R}^n$ at a point $x \in \partial\Phi$, and $\mathrm{T}(x;\Phi)$ is the tangent cone to Φ at x. Since Φ is a closed convex set with a connected regular boundary, it admits a tangent supporting hyperplane at any $x \in \partial\Phi$, and this hyperplane is the boundary of $\mathrm{T}(x;\Phi)$, i. e., $\mathrm{T}(x;\Phi)$ is a closed half-space. Hence, $\mathbb{R}^n \setminus (\operatorname{int} \mathrm{T}(x;\Phi))$ is the set of all vectors that do not point strictly inside Φ when starting at $x \in \partial\Phi$.

Based on (15), we introduce the local model

$$\begin{aligned} & V_{\text{appr, loc}}(x) \;=\; V_{\partial\Phi} \;+\; \eta^*(\operatorname{proj}_\Phi x)\,(\varphi(x) - \gamma), \\ & \tilde{V}_{\text{appr, loc}}(x) \;=\; \zeta(V_{\text{appr, loc}}(x)) \\ & \forall\, x \;\in\; G \setminus (\operatorname{int}\Phi), \end{aligned} \tag{16}$$

where $\operatorname{proj}_\Phi x$ is an orthogonal projection of x onto Φ. Since Φ is a closed convex set, $\operatorname{proj}_\Phi x$ is unique for any $x \in \mathbb{R}^n$. For example, if $\varphi(\cdot)$ is a positive definite quadratic function, then Φ is an ellipsoid and one can therefore use the algorithm of [42] to compute the projection $\operatorname{proj}_\Phi x$ and distance $\operatorname{dist}(x,\Phi)$ for any selected state x. It is not difficult to show that (15) holds when written with $V_{\text{appr, loc}}(\cdot)$ instead of $V_{\text{appr}}(\cdot)$ (using the continuity of the orthogonal projection onto Φ, the continuity of $\eta^*(\cdot)$ on $\partial\Phi$, and the fact that $\varphi(x) = \gamma$ for all $x \in \partial\Phi$).

The following boundary conditions should thus be fulfilled for a sought-after approximation $\tilde{V}_{\text{appr}}(\cdot)$ of $\tilde{V}(\cdot)$:

$$\begin{aligned} & \tilde{V}_{\text{appr}}(x) \;=\; \tilde{V}_{\text{appr, loc}}(x) \;=\; \zeta(V_{\partial\Phi}), \\ & \partial\tilde{V}_{\text{appr}}(x;\xi) \;=\; \partial\tilde{V}_{\text{appr, loc}}(x;\xi) \\ & \qquad =\; \zeta'(V_{\partial\Phi})\,\partial V_{\text{appr, loc}}(x;\xi) \;=\; \zeta'(V_{\partial\Phi})\,\langle \eta^*(x)\,\mathrm{D}\varphi(x),\; \xi\rangle \\ & \forall\, \xi \;\in\; \mathbb{R}^n \setminus (\operatorname{int} \mathrm{T}(x;\Phi)) \quad \forall x \in \partial\Phi. \end{aligned} \tag{17}$$

The constructed models $K_i(\cdot),\ i = \overline{1,N},$ and $\tilde{V}_{\text{appr, loc}}(\cdot)$ can then be combined according to

$$\begin{aligned} & \tilde{V}_{\text{appr}}(x) \;=\; \hat{W}_0(x)\,\tilde{V}_{\text{appr, loc}}(x) \;+\; \sum_{j=1}^{N} \hat{W}_j(x)\,K_j(x), \\ & \hat{W}_i(x) \;=\; \lim_{\xi \to x,\ \xi \in G \setminus (\operatorname{int}\Phi)} \frac{W_i(\xi)}{\sum_{j=0}^{N} W_j(\xi)}, \quad i = \overline{0,N}, \\ & \forall\, x \;\in\; G \setminus (\operatorname{int}\Phi), \end{aligned} \tag{18}$$

where the functions $W_i(\cdot),\ i = \overline{0,N},$ are selected in line with an inverse distance weighting methodology [17, §5] to ensure the relations (17) and

$$\begin{aligned} & \tilde{V}_{\text{appr}}(x) \;=\; K_i(x), \quad \mathrm{D}\tilde{V}_{\text{appr}}(x) \;=\; \mathrm{D}K_i(x) \\ & \forall x \in \Sigma_i, \quad i = \overline{1,N}, \end{aligned} \tag{19}$$

while $\hat{W}_i(\cdot),\ i = \overline{0,N},$ are the normalized weights. The definition of the latter involves limits to account for the situation when $W_0(x)$ tends to $+\infty$ as $\operatorname{dist}(x,\Phi) \to 0$ and when $W_i(x)$ tends to $+\infty$ as $x \to x^{(ij)} \in \Sigma_i,\ j = \overline{1,N_i},\ i = \overline{1,N}$. The exact interpolation requirement may be relaxed by incorporating appropriate tolerances for the relations (19).

We specify the weight $W_0(\cdot)$ as

$$W_0(x) \;=\; \left(\max\left\{0,\; \frac{1}{\operatorname{dist}(x,\Phi)} - \frac{1}{R_0}\right\}\right)^2 \quad \forall x \in G \setminus \Phi. \tag{20}$$

This function tends to $+\infty$ as $\operatorname{dist}(x, \Phi) \to 0$ and vanishes outside the open R_0–neighborhood of Φ. Then the normalized weight $\hat{W}_0(\cdot)$ is continuously differentiable on $G \setminus \Phi$ and equals 1 on $\partial\Phi$ (since Σ_i, $i = \overline{1, N}$, do not contain any points from Φ and the weights $W_i(\cdot)$, $i = \overline{1, N}$, should hence be bounded near Φ), and the conditions (17) are satisfied.

Next, let us list several options to determine the weights $W_i(\cdot)$, $i = \overline{1, N}$:

1) fix a small constant $\varepsilon > 0$ and take

$$W_i(x) = \sum_{j=1}^{N_i} \left(\left(\max \left\{ 0, \frac{1}{\left\| x - x^{(ij)} \right\|} - \frac{1}{R_{ij}} \right\} \right)^2 + \varepsilon \right) \quad \forall\, x \in G \setminus ((\operatorname{int} \Phi) \cup \Sigma_i), \quad i = \overline{1, N} \tag{21}$$

(see [17, §5] and note that ε is added to prevent vanishing of the denominators in the representations of the normalized weights in (18)), so that $W_i(x)$ tends to $+\infty$ as x approaches a node of Σ_i and equals ε when x lies outside all of the influence neighborhoods for the nodes of Σ_i, $i = \overline{1, N}$;

2) fix a parameter $\varkappa > 0$ and modify the previous relations according to

$$W_i(x) = \sum_{j=1}^{N_i} \frac{1}{\left\| x - x^{(ij)} \right\|^2 \exp\left(\varkappa \left\| x - x^{(ij)} \right\|^2 \right)} \quad \forall\, x \in G \setminus ((\operatorname{int} \Phi) \cup \Sigma_i), \quad i = \overline{1, N}, \tag{22}$$

with no radii of influence involved (the presence of the exponent accelerates the decay of the weights when moving away from the grids);

3) introduce a smooth minimum function approximation, such as

$$F_{\text{smooth min}}(\xi_1, \xi_2, \ldots, \xi_l) = \frac{\sum_{i=1}^{l} \xi_i \exp(-\lambda \xi_i)}{\sum_{i=1}^{l} \exp(-\lambda \xi_i)} = \frac{\sum_{i=1}^{l} \xi_i \exp\left(-\lambda \left(\xi_i - \min\{\xi_1, \xi_2, \ldots, \xi_l\}\right)\right)}{\sum_{i=1}^{l} \exp\left(-\lambda \left(\xi_i - \min\{\xi_1, \xi_2, \ldots, \xi_l\}\right)\right)} \quad \forall\, (\xi_1, \xi_2, \ldots, \xi_l) \in \mathbb{R}^l \quad \forall l \in \mathbb{N} \tag{23}$$

with a parameter $\lambda > 0$ [43], and take

$$W_i(x) = \frac{1}{\theta_i(x) \exp(\varkappa\, \theta_i(x))}, \quad \theta_i(x) = F_{\text{smooth min}} \left\{ \left\| x - x^{(ij)} \right\|^2 : j = \overline{1, N_i} \right\} \quad \forall\, x \in G \setminus (\operatorname{int} \Phi), \quad i = \overline{1, N}, \tag{24}$$

with a parameter $\varkappa > 0$ (here $\theta_i(x)$ is a smooth approximation of the minimal distance from x to Σ_i).

In all of the three options, the normalized weights $\hat{W}_i(\cdot)$, $i = \overline{1, N}$, are continuously differentiable on $G \setminus (\operatorname{int} \Phi)$. In options 1 and 2, $\hat{W}_i(x)$ is equal to 1 for $x \in \Sigma_i$, $i = \overline{1, N}$, and the conditions (19) are fulfilled exactly. In option 3, the conditions (19) do not hold exactly, but one would expect a reasonable accuracy if λ is large enough; however, large λ make the minimum function approximation rather sharp, so there is a trade-off in the choice of λ.

A noticeable issue is that the gradients of the generated Kriging models may practically vanish when the distance from each of the corresponding grid nodes is relatively large (for example, this is a constructive feature of the gradient enhanced Kriging implementation of [19]). To avoid the vanishing of the gradient of the resulting extrapolant outside a certain compact domain D_1 of the state space (D_1 is supposed to satisfy $D_1 \subset G$, $\Phi \subset \operatorname{int} D_1$, $\bigcup_{i=1}^{N} \Sigma_i \subset D_1$) and therefore to expand the practical region of asymptotic null-controllability, it may be reasonable to use a proper and locally Lipschitz continuous extrapolation $\tilde{V}_{\text{appr, extrapol}} : G \to [0, +\infty)$ outside D_1. Namely, the resulting model can then be specified as

$$\tilde{V}_{\text{appr, updated}}(x) = \tilde{V}_{\text{appr}}(x) + \omega(x) \left(\tilde{V}_{\text{appr, extrapol}}(x) - \tilde{V}_{\text{appr}}(x) \right) \quad \forall x \in G, \tag{25}$$

where $\omega : G \to [0,1]$ is a continuously differentiable function satisfying

$$\begin{gathered} \omega(x) = 0 \quad \forall x \in D_2, \\ D_2 \text{ is a compact domain such that } D_2 \subset \operatorname{int} D_1 \text{ and } \Phi \subset \operatorname{int} D_2, \\ \omega(x) = 1 \quad \forall\, x \in G \setminus D_1, \\ \mathrm{D}\omega(x) = 0_n \quad \forall\, x \in \partial D_1 \cup \partial D_2; \end{gathered} \tag{26}$$

hence, $\tilde{V}_{\text{appr, updated}}(\cdot)$, $\mathrm{D}\tilde{V}_{\text{appr, updated}}(\cdot)$ coincide respectively with $\tilde{V}_{\text{appr}}(\cdot)$, $\mathrm{D}\tilde{V}_{\text{appr}}(\cdot)$ in D_2 and with $\tilde{V}_{\text{appr, extrapol}}(\cdot)$, $\mathrm{D}\tilde{V}_{\text{appr, extrapol}}(\cdot)$ in $G \setminus D_1$. For instance, if

$$D_1 = \overline{\mathrm{B}}_{r_1}(0_n), \quad D_2 = \overline{\mathrm{B}}_{r_2}(0_n), \quad 0 < r_2 < r_1, \tag{27}$$

one can take

$$\omega(x) = -2\left(\frac{\|x\| - r_2}{r_1 - r_2}\right)^3 + 3\left(\frac{\|x\| - r_2}{r_1 - r_2}\right)^2 \quad \forall\, x \in D_1 \setminus D_2 \tag{28}$$

(in D_2 and $G \setminus D_1$, $\omega(\cdot)$ is defined according to (26)). In Subsection 4.2, the extrapolant is selected as

$$\tilde{V}_{\text{appr, extrapol}}(x) = \ln\left(1 + \left\langle P_{\text{extrapol}}\, x,\, x\right\rangle\right) \quad \forall x \in \mathbb{R}^n, \tag{29}$$

where $P_{\text{extrapol}} \in \mathbb{R}^{n\times n}$ is a positive definite matrix that is obtained by solving an auxiliary nonlinear programming problem in offline mode. A more accurate but more computationally expensive approach is to use direct collocation based estimates [36–41] for control or costate extrapolation in online mode.

In order to estimate a current costate in an online MPC implementation, one can exploit the proposed model (18) and perform numerical differentiation. Typically, gradient enhanced Kriging software, such as [19], allows for fast evaluation of the gradient of an interpolant, and it is worth taking this into account so as to differentiate the weighted combination (18) in a semi-analytical fashion. The costate estimate can then serve to retrieve control actions within a sampling time interval. Each sampling time interval should start with updating the costate estimate. Associated delays in a real-time physical control process may be attenuated if the state $x(t_{k+1})$ at the end of a current sampling time interval $[t_k, t_{k+1}]$ is predicted in the beginning of this interval using a measurement based estimate of $x(t_k)$ and numerical MPC simulations with the previously evaluated costate $p(t_k)$, and if the costate $p(t_{k+1})$ is then obtained from the simulation based estimate of $x(t_{k+1})$ via our interpolation model. If the interpolation is fast enough, an estimate of $p(t_{k+1})$ should be ready before $t = t_{k+1}$. However, if a current state leaves an a priori selected bounded domain that contains the sparse grids, it may be required to significantly increase the sampling time interval and compute the control or costate in real time via, e. g., direct collocation.

3.2 | Model training

In contrast to online (real-time) MPC evaluation, much more time and computational resources are typically available for preparatory offline model training (machine learning). This training can be iterative and based on solving a large number of nonlinear programming problems for a model such as that built in the previous subsection. The purpose is to improve the grids iteratively. Each iteration has to return a point in the state space that will be involved on the next iteration either as a new node in an existing grid or as the center of a new grid to be included in the model.

One possible approach to the arrangement of model training iterations is to search for the minimal level of a current CLF approximation at which one of the following two properties holds: 1) the infinitesimal decrease condition starts to be violated at a point in a certain compact domain E of the state space; 2) E is left by the related state trajectories while the infinitesimal decrease condition is fulfilled. Here E is assumed to satisfy $E \subset G$ and $\Phi \subset \operatorname{int} E$. Namely, the goal is to find

$$\begin{aligned} &\min(\gamma_1, \gamma_2), \\ &\gamma_1 = \inf\left\{\tilde{V}_{\text{appr}}(x) : \; x \in E \setminus \Phi, \; \left|\inf_{u\in U} \partial^- \tilde{V}_{\text{appr}}(x;\, f(x,u))\right| \leqslant \tilde{\varepsilon}\right\}, \\ &\gamma_2 = \inf\left\{\tilde{V}_{\text{appr}}(x) : \; x \in \partial E, \; \inf_{u\in U} \partial^- \tilde{V}_{\text{appr}}(x;\, f(x,u)) \leqslant -\tilde{\varepsilon}, \right. \\ &\qquad \left. f(x,\, \tilde{u}_{\text{appr}}(x)) \text{ does not point strictly inside } E \text{ when starting at } x\right\}, \end{aligned} \tag{30}$$

where $\tilde{\varepsilon} > 0$ is a small tolerance parameter, and $\tilde{u}_{\text{appr}}(\cdot)$ is a feedback control strategy that is associated with $\tilde{V}_{\text{appr}}(\cdot)$ and can be determined by

$$\tilde{u}_{\text{appr}}(x) \in \operatorname{Arg}\min_{u\in U} \partial^- \tilde{V}_{\text{appr}}(x;\, f(x,u)) \tag{31}$$

or by some regularized modification of this inclusion. We suppose $\tilde{u}_{\text{appr}}(\cdot)$ to be defined on $E \setminus \Phi$ so that the corresponding state trajectories are correctly determined until reaching Φ or exiting from E. For $x \in \Phi$, $\tilde{u}_{\text{appr}}(x)$ relates to a local CLF in the setting of [1, Section 2] and can be arbitrarily chosen from U in the setting of [1, Section 3]. Note also the representation

$$\begin{aligned} &\partial^- \tilde{V}_{\text{appr}}(x;\xi) \;=\; \left\langle \mathrm{D}\tilde{V}_{\text{appr}}(x),\, \xi \right\rangle \\ &\forall\, x \;\in\; \left\{ x' \in \mathbb{R}^n \,:\; \tilde{V}_{\text{appr}}(\cdot) \text{ is differentiable at } x' \right\} \quad \forall \xi \in \mathbb{R}^n, \end{aligned} \tag{32}$$

which can be practically used in computations for any considered x, since $\tilde{V}_{\text{appr}}(\cdot)$ is locally Lipschitz continuous in $G \setminus (\text{int}\,\Phi)$ (according to the assumptions adopted in the previous subsection) and consequently differentiable almost everywhere in this region (with respect to the Lebesgue measure in $\mathbb{R}^n$).

A state obtained by solving the problem (30) should then be used to update the model for the next iteration (e. g., in one of the two ways mentioned in the first paragraph of this subsection). This problem is in general essentially multi-extremal, and one therefore has to run a numerical optimization algorithm multiple times with different initial guesses. The latter can be randomly generated in $E \setminus \Phi$ for the γ_1–subproblem and on ∂E for the γ_2–subproblem. One has to select only the initial guesses that fulfill the constraints. Moreover, computations for different initial guesses can be parallelized.

Due to the statement of the problem (30), the set

$$\left\{ x \in E \,:\; \tilde{V}_{\text{appr}}(x) \;<\; \min(\gamma_1, \gamma_2),\;\; \inf_{u \in U} \partial^- \tilde{V}_{\text{appr}}(x;\, f(x,u)) \;<\; -\tilde{\varepsilon} \right\} \tag{33}$$

gives an inner estimate of the region of attraction to the target set Φ when using the feedback control law $\tilde{u}_{\text{appr}}(\cdot)$. However, this estimate is likely to be rather conservative. Indeed, a state trajectory governed by $\tilde{u}_{\text{appr}}(\cdot)$ may eventually reach the target set even if the infinitesimal decrease condition for $\tilde{V}_{\text{appr}}(\cdot)$ does not hold on some subarcs.

It may hence be reasonable to replace the nonlinear programming problem (30) with the following one (although its numerical solution is much more computationally expensive as compared to (30)):

$$\begin{aligned} \gamma_3 \;=\; \inf \big\{ &\tilde{V}_{\text{appr}}(x_0) \,:\; x_0 \in E \setminus \Phi, \\ &\text{the state trajectory with } x(0) = x_0 \text{ constructed via the MPC simulation scheme} \\ &\text{with costate updates from the current CLF interpolation model (18) every } \Delta \text{ time units} \\ &\text{does not reach the target set } \Phi \text{ during the whole time interval } \left[0, T_{\max}\right] \big\}. \end{aligned} \tag{34}$$

Here $\Delta > 0$ is a fixed sampling time, $T_{\max} > 0$ is a fixed upper bound for terminal times, and, for training purposes, only the current CLF interpolation model (18) is used to update the costate, without the modifications indicated in the end of the previous subsection (it is nonetheless worth incorporating one of these modifications in real-time MPC with the resulting interpolation model). An inner estimate for the region of attraction to Φ for $\tilde{u}_{\text{appr}}(\cdot)$ is then taken as

$$\left\{ x \in E \,:\; \tilde{V}_{\text{appr}}(x) \;<\; \gamma_3 \right\}. \tag{35}$$

Even though this estimate seems less conservative than (33), the actual region of attraction to Φ for an implemented MPC strategy is still expected to be markedly larger, especially if one uses a modified model (25), (26) or incorporates direct collocation (as mentioned in the end of the previous subsection).

3.3 | Summary

Let us briefly list the key stages of the numerical approach described in the previous two subsections.

The offline (preparatory) computation involves the following steps:

1) select a function $\varphi(\cdot)$ and a level γ that determine the terminal (target) set Φ in the studied exit-time optimal control problem (as indicated in the beginning of Section 3, this is rather straightforward for the setting of [1, Section 3], while the setting of [1, Section 2] requires a local CLF to be constructed, and the considerations of Section 2 may help in that regard);

2) generate initial sparse grids together with value function and costate data on them, and build the corresponding Kriging models (as described in the beginning of Subsection 3.1);

3) train and update the composite interpolation model (18) (as discussed in Subsection 3.2);

4) construct an appropriate extrapolation (25), (26) if necessary (in the numerical example of Subsection 4.2, a special form of this extrapolation model with (27)–(29) is used, and the matrix P_{extrapol} in (29) is obtained by solving the nonlinear programming problem (66) with a certain collection of initial guesses).

The online (real-time) control process can be arranged as outlined in the last paragraph of Subsection 3.1; see Subsection 4.2 for more specific details.

Furthermore, a typical scheme to obtain an online control action from a current state and a transformed costate estimate is also indicated in Subsection 4.2 (see the end of the first paragraph that starts after (62)).

4 | NUMERICAL SIMULATIONS

The aim of the first example in this section is to numerically test the theoretical results of the previous study [1] with regard to a simple nonlinear control system that has two state coordinates and can be handled analytically to some extent. The second example exploits the developments of Sections 2 and 3 to design, train, and simulate a stabilizing MPC controller for a nonlinear system with six-dimensional state space.

The numerical simulations were conducted on a desktop machine with Intel CORE i7 vPro processor, and no parallel programming tools were used. The runtimes can be significantly shorter for more powerful machines, especially when parallelization is done.

4.1 | Two-dimensional system

Consider the control system (1) with $x = (x_1, x_2)^\top \in \mathbb{R}^2$, $x(0) = x^0 \in \mathbb{R}^2$, $G = \mathbb{R}^2$, $U \subseteq \mathbb{R}$, and

$$f(x,u) \;=\; \begin{pmatrix} x_1 \,+\, 2x_2 \,+\, u \\ -x_2 \,-\, 2x_1^3 \end{pmatrix} \quad \forall x \in \mathbb{R}^2 \quad \forall u \in U \tag{36}$$

(see [44, Example 1.1]).

First, let $U = \mathbb{R}$. The proper, positive definite, and infinitely differentiable function

$$\check{V}(x) \;=\; \frac{1}{4}\, x_1^4 \,+\, \frac{1}{2}\, x_2^2 \quad \forall x \in \mathbb{R}^2 \tag{37}$$

is a global CLF for this system in the whole state space. Indeed,

$$\left\langle \mathrm{D}\check{V}(x),\, f(x,u) \right\rangle \;=\; x_1^3 (x_1 \,+\, 2x_2 \,+\, u) \,+\, x_2 \left(-x_2 \,-\, 2x_1^3\right) \;=\; x_1^3 (x_1 \,+\, u) \,-\, x_2^2$$
$$\forall x \in \mathbb{R}^2 \quad \forall u \in \mathbb{R},$$

so that, for any constant $b > 0$ and for any bounded continuous function $\chi : \mathbb{R} \to \mathbb{R}$ satisfying

$$\chi(x_1) \geqslant b \quad \forall x_1 \in \mathbb{R}, \tag{38}$$

the feedback control strategy

$$\check{u}(x) \;=\; -(1 \,+\, \chi(x_1))\, x_1 \quad \forall x \in \mathbb{R}^2 \tag{39}$$

is globally stabilizing due to

$$\left\langle \mathrm{D}\check{V}(x),\, f(x, \check{u}(x)) \right\rangle \;=\; -\chi(x_1)\, x_1^4 \,-\, x_2^2 \;\leqslant\; -b x_1^4 \,-\, x_2^2 \quad \forall x \in \mathbb{R}^2.$$

Introduce also the running cost

$$g(x,u) \;=\; 2x_1^4 \,+\, x_2^2 \,+\, \frac{1}{256}\, u^4 \quad \forall x \in \mathbb{R}^2 \quad \forall u \in \mathbb{R}. \tag{40}$$

With the help of the classical verification result [45, Chapter VII, Theorem 2.2], one can show that (37) is the value function in the infinite-horizon optimal control problem

$$\check{V}\left(x^0\right) \;=\; \min_{u(\cdot) \,\in\, L^\infty_{\text{loc}}([0,+\infty),\, \mathbb{R})} \left\{ \int\limits_0^{+\infty} g\left(x\left(t;\, x^0, u(\cdot)\right),\, u(t)\right)\, \mathrm{d}t \right\} \quad \forall x^0 \in \mathbb{R}^2 \tag{41}$$

for the considered system and running cost, and that

$$\breve{u}^*(x) \;=\; -4x_1 \tag{42}$$

is the corresponding optimal feedback control strategy.

Next, take the set of pointwise control constraints as the bounded line segment

$$U \;=\; [-a,a] \tag{43}$$

with a constant $a > 0$. Furthermore, fix a constant $b > 0$ and a bounded continuous function $\chi : \mathbb{R} \to \mathbb{R}$ satisfying (38), and choose a constant $c > 0$ such that the feedback strategy (39) fulfills

$$\breve{u}(x) \;\in\; U \;=\; [-a,a] \quad \forall x \in \Omega_c \tag{44}$$

in the sublevel set

$$\Omega_c \;=\; \left\{x \in \mathbb{R}^2 \,:\, \breve{V}(x) \leqslant c\right\} \tag{45}$$

of $\breve{V}(\cdot)$. Hence, the restrictions of $\breve{V}(\cdot)$ and $\breve{u}(\cdot)$ to Ω_c are a local CLF and a locally stabilizing feedback, respectively. Besides, there exists a constant $c^* > 0$ such that the restriction of $\breve{u}^*(\cdot)$ to Ω_{c^*} is also a locally stabilizing feedback. For convenience, denote

$$V_{\mathrm{loc}}(x) \;=\; \breve{V}(x), \quad u_{\mathrm{loc}}(x) \;=\; \breve{u}(x), \quad u^*_{\mathrm{loc}}(x) \;=\; \breve{u}^*(x) \quad \forall x \in \mathbb{R}^2. \tag{46}$$

Now it is not difficult to check that [1, Theorems 2.21, 2.25, 2.26, 2.28 and 2.29] can be used with the specified local CLF $V_{\mathrm{loc}}(\cdot)$, sublevel set Ω_c, and running cost (40). Recall that a CLF $V(\cdot)$ in the domain of asymptotic null-controllability $\mathcal{D}_0$ is determined by [1, (23), (28)], and its Kruzhkov transform is given by [1, (40)]. Introduce also the Kruzhkov transform of $V_{\mathrm{loc}}(\cdot)$:

$$\upsilon_{\mathrm{loc}}\left(x^0\right) \;\stackrel{\mathrm{def}}{=}\; 1 \,-\, e^{-V_{\mathrm{loc}}(x^0)} \quad \forall x^0 \in \mathbb{R}^2. \tag{47}$$

In order to approximate the CLF and stabilizing feedback, as well as to construct a reasonable inner estimate for the domain of asymptotic null-controllability, we have used the framework that is described in Appendix and explicitly relies on the characteristics based representation of the value function given by [1, Theorem 2.29]. It is not difficult to verify the fulfillment of Assumptions A.2–A.4 from Appendix.

We take

$$\begin{gathered} a \;=\; 1.2, \quad b \;=\; 1.4, \quad c \;=\; 0.015, \quad c_1 \;=\; 0.01, \quad T_{\max} \;=\; 10, \\ \varepsilon \;=\; 10^{-15}, \quad \varepsilon_1 \;=\; 0.005, \quad \delta_1 \;=\; 0.005, \quad \delta_2 \;=\; 0.005 \end{gathered} \tag{48}$$

(the parameters $T_{\max}, \varepsilon, \varepsilon_1$ are introduced in Section A.1, while c_1 and δ_1, δ_2 are described respectively in Sections A.2 and A.3),

$$\chi(x_1) \;=\; \begin{cases} 3, & -4x_1 \;\in\; [-a,a], \\ \max\left\{\dfrac{a}{|x_1|} \,-\, 1,\; b\right\}, & -4x_1 \;\notin\; [-a,a], \end{cases} \tag{49}$$

so that $\chi(\cdot)$ is bounded and continuous, (38) holds,

$$\frac{a}{|x_1|} \;>\; 1 \,+\, b \quad \text{at all points } \; x \in \Omega_c \; \text{ for which } x_1 \neq 0, \tag{50}$$

and (39), (46), (49), (50) imply the following relations:

$$\begin{gathered} u_{\mathrm{loc}}(x) \;=\; \breve{u}(x) \;=\; \begin{cases} u^*_{\mathrm{loc}}(x) \;=\; \breve{u}^*(x) \;=\; -4x_1, & -4x_1 \;\in\; [-a,a], \\ -x_1 \,\max\left\{\dfrac{a}{|x_1|},\; 1+b\right\}, & -4x_1 \;\notin\; [-a,a], \end{cases} \\ u_{\mathrm{loc}}(x) \;=\; -a \,\operatorname{sign} x_1 \quad \text{at all points } \; x \in \Omega_c \; \text{ for which } \; -4x_1 \;\notin\; [-a,a]. \end{gathered} \tag{51}$$

For a stabilizing feedback $u^* : \mathcal{D}_0 \to U$ associated with the CLF $V(\cdot)$, we put

$$u^*(x) \;=\; u_{\mathrm{loc}}(x) \quad \forall x \in \Omega_c. \tag{52}$$

The characteristic Cauchy problems (A.5) and (A.11) were numerically solved via the Dormand–Prince fifth-order Runge–Kutta algorithm from [46, Chapter 17]. When launching the related routine, the initial guess for the stepsize was specified as $2 \cdot 10^{-4}$, and the absolute and relative tolerances were selected as 10^{-6}. The output data was obtained for the uniform time grid on

$[0, T_{\max}]$ with the stepsize $2 \cdot 10^{-4}$. In particular, the shooting costs were approximated from the state trajectory discretizations on this time grid.

The initial states were chosen from the grid on the rectangle $[-2, 2] \times [-2.5, 2.5]$ with the spatial steps 0.0625 and 0.1 along x_1-axis and x_2-axis, respectively. For solving the main and auxiliary finite-dimensional optimization problems formulated in Theorem A.1 and Section A.2, we used the Powell algorithm from [46, §10.7] (which does not require evaluation of derivatives), and the corresponding tolerances were set to 10^{-6} and 10^{-8}, respectively. For each state x^0 on the selected rectangular grid, the Powell iterative process for the auxiliary shooting problem was run from $N_{\text{opt. init. guess}} = 4$ initial guesses that were randomly generated according to the uniform distribution with respect to the angles in the unit sphere parametrization (A.21). The unique point (A.15) and the roots of the function (A.18) were computed via the bisection algorithm from [46, §9.1], and the tolerance was taken as 10^{-13}. An upper bound for the sought-after root λ in (A.15) was chosen as 0.5 (since $\Omega_{c_1} \subset \mathrm{B}_{0.5}(0_2)$). Possible multiple roots of the function (A.18) on $(0, 1)$ were bracketed by using the `zbrak` routine from [46, §9.1]. The bracketing pairs were searched for after dividing the interval $[0, 1]$ into 100 equally spaced segments.

The tolerance for the practical verification of equalities and non-strict inequalities via strict inequalities was set to 10^{-15}.

In the case when we could not find a characteristic reaching the target set Ω_c and generating a cost less than $1 - \varepsilon$ for some initial state x^0 (this might be not only a node on the specified rectangular grid, but also a shooting estimate $\hat{x}^0$ as described in Section A.3), the computation was rerun with the increased parameter values $T_{\max} = T_{\max,\,\text{recomp.}} = 20$ and $N_{\text{opt. init. guess}} = N_{\text{opt. init. guess, recomp.}} = 5$. In such situations, the resulting data was taken from the second attempt ($\upsilon\left(x^0\right)$ might again be estimated as $1 - \varepsilon$, which would indeed be reasonable if $x^0 \notin \mathcal{D}_0$).

The related numerical simulation results are illustrated in Figs. 1–3.

Fig. 1 indicates the Kruzhkov transformed functions $\upsilon(\cdot)$, $\upsilon_{\text{loc}}(\cdot)$ and their difference. Fig. 2 shows the corresponding feedback strategies $u^*(\cdot)$, $u^*_{\text{loc}}(\cdot)$ and their difference (note that the values of $u^*(\cdot)$ at the grid nodes outside Ω_c were computed as described in Section A.3 of Appendix). In Fig. 1, some approximated level sets of $\upsilon(\cdot)$ are depicted as well. In particular, the level $\upsilon(x) = 1 - e^{-c}$ (or, equivalently, $V(x) = c$) describes the boundary $l_c = \partial\Omega_c$, while the level $\upsilon(x) = 1 - \varepsilon_1 = 0.995$ is selected to represent an inner estimate of the domain of asymptotic null-controllability $\mathcal{D}_0$. The illustrations agree with the reasonable expectation that $\upsilon(\cdot)$ and $\upsilon_{\text{loc}}(\cdot)$ should coincide in some region strictly containing Ω_c, and that $u^*(x) = u^*_{\text{loc}}(x)$ for all x lying in this region and satisfying $u^*_{\text{loc}}(x) \in U = [-a, a]$ (due to (51) and (52), one also has $u^*(x) = u_{\text{loc}}(x)$ for all such x and everywhere in Ω_c).

Fig. 3 shows the graphs of the following functions:

- the shooting state error defined as the square root of the numerical estimate of the minimum quadratic shooting cost (see Section A.2 and note that, even when the exact minimum value of the lowest deviation (A.20) is zero, its approximation does not vanish for $x^0 \notin \Omega_c$);
- the shooting time defined as an approximate minimizer in (A.20) for an optimal shooting reverse-time characteristic;
- the shooting value replacement indicator defined as zero if one arrives at a value less than $1 - \varepsilon$ after one or two attempts to compute $\upsilon\left(x^0\right)$, and as the absolute difference $\left|\upsilon_1\left(x^0\right) - \upsilon\left(\hat{x}^0\right)\right|$ in the other case when one uses the first-order estimation technique proposed in Section A.3.

The shooting state error and the shooting value replacement indicator on the considered grid are small enough to conjecture that the whole rectangle is contained in the domain of asymptotic null-controllability $\mathcal{D}_0$. However, rigorous verification of that for the selected bounded control constraint set $U = [-1.2, 1.2]$ remains an open problem. Another open question is whether $\mathcal{D}_0$ is bounded as a whole for a bounded U or not (the obtained inner estimate of $\mathcal{D}_0$ is bounded as seen in Fig. 1).

In order to practically check the obvious fact that, for a sufficiently large a and $U = [-a, a]$, the functions $\upsilon(\cdot)$ and $u^*(\cdot)$ should coincide in the considered bounded rectangle with $\upsilon_{\text{loc}}(\cdot)$ and $u^*_{\text{loc}}(\cdot)$, respectively, we performed numerical simulations for the increased parameter value $a = 20$. We also reduced $T_{\max}$ from 10 to 5. Moreover, the spatial steps along x_1-axis and x_2-axis for the grid on the rectangle $[-2, 2] \times [-2.5, 2.5]$ were increased to 0.1 and 0.15625, respectively. All other parameters kept their values (in particular, $T_{\max,\,\text{recomp.}} = 20$ remained the same). The results are illustrated in Fig. 4.

The average runtime per one initial state was around 5 seconds, which appeared to be about 10 times longer than in case of using GPOPS–II direct collocation software [36, 37]. The choice of the numerical method for this illustrative example was motivated by the clear connection with our characteristics based value function representation. However, GPOPS–II was used in numerical simulations for the next example, because the dynamics of the related high-dimensional characteristic system turned out to be very complicated and sensitive to initial data, which made our characteristics based algorithm hardly applicable. Note that similar algorithms with optimization over solutions of characteristic Cauchy problems nevertheless worked well and fast

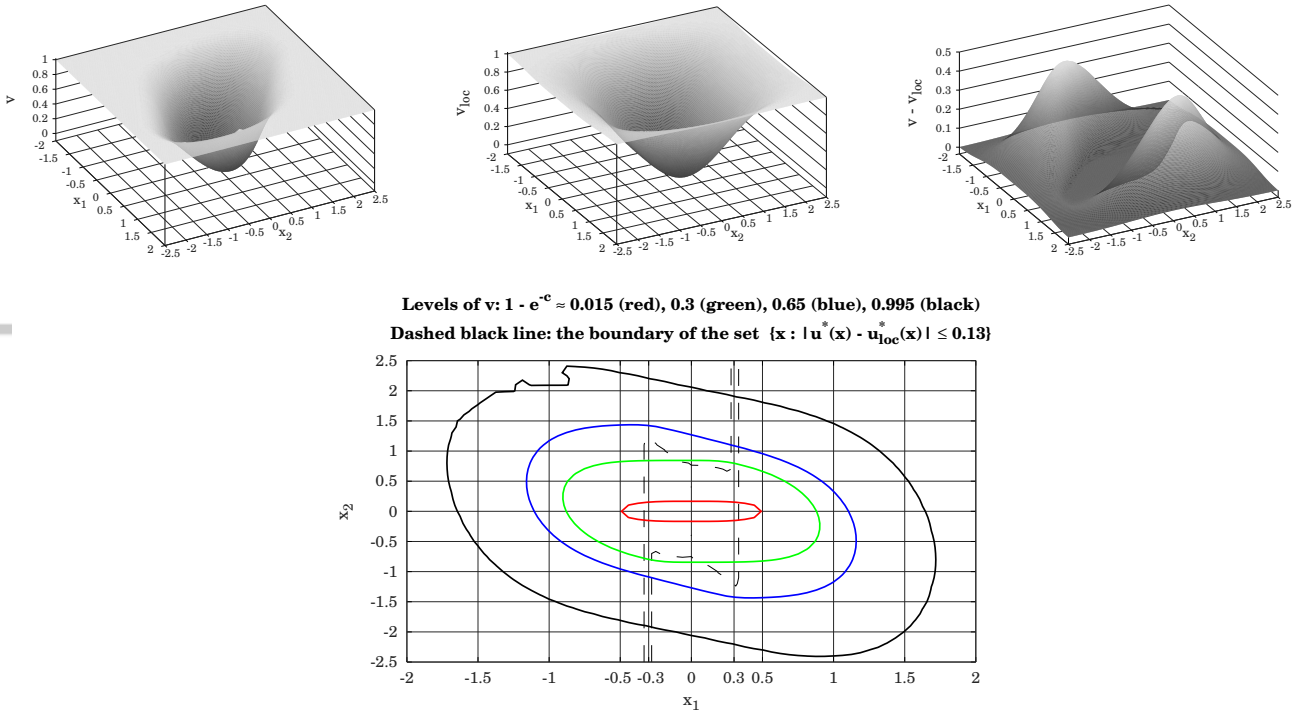


FIGURE 1 The Kruzhkov transformed functions $v(\cdot), v_{\mathrm{loc}}(\cdot)$ and their difference for $U = [-1.2, 1.2]$ in the example of Subsection 4.1. Some approximated level sets of $v(\cdot)$ are shown as well. In order to see the graph of the difference between $v(\cdot)$ and $v_{\mathrm{loc}}(\cdot)$ clearer, the scale of the vertical axis in the third subfigure is modified as compared to that in the first two subfigures.

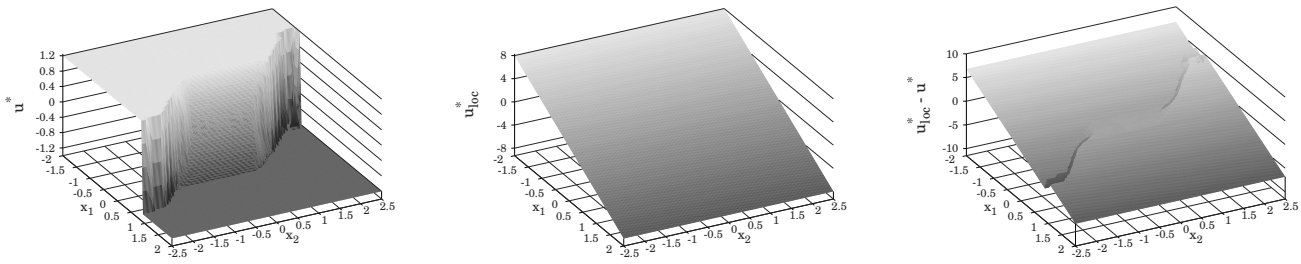


FIGURE 2 The feedback control strategies $u^*(\cdot), u^*_{\mathrm{loc}}(\cdot)$ (corresponding to $v(\cdot), v_{\mathrm{loc}}(\cdot)$, respectively) and their difference for $U = [-1.2, 1.2]$ in the example of Subsection 4.1. In order to see the graphs clearer, we do not fix the same scale for the vertical axes in the subfigures.

for a number of fixed-horizon high-dimensional optimal control problems [47, 48]. An interesting direction of possible future research is the development of more computationally efficient modifications of our characteristics based framework for exit-time problems, e. g., via incorporation of multiple shooting or indirect collocation [49, 50].

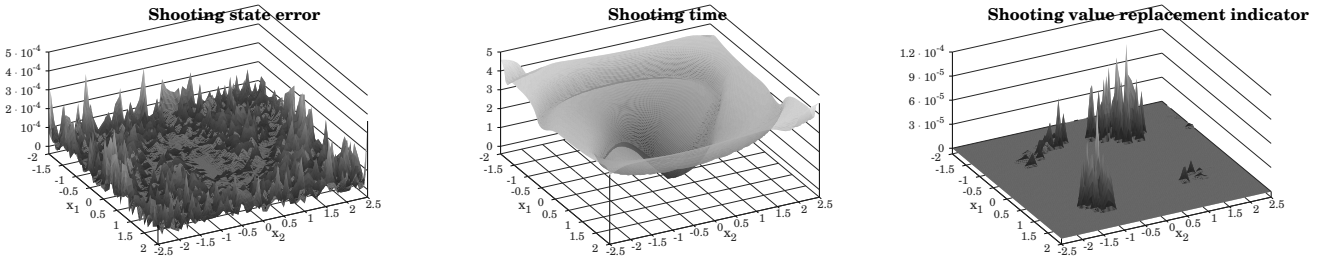


FIGURE 3 The shooting state error, shooting time, and shooting value replacement indicator for $U = [-1.2, 1.2]$ in the example of Subsection 4.1. Different scales are used for the vertical axes in the subfigures.

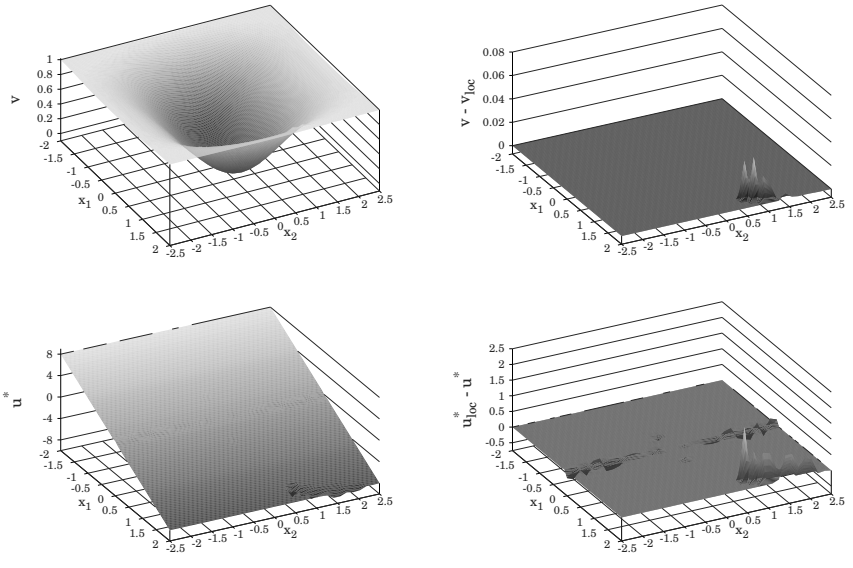


FIGURE 4 The Kruzhkov transformed function $v(\cdot)$, the difference between $v(\cdot)$ and $v_{\mathrm{loc}}(\cdot)$, the feedback control strategy $u^*(\cdot)$, and the difference between $u^*_{\mathrm{loc}}(\cdot)$ and $u^*(\cdot)$ for $U = [-20, 20]$ in the example of Subsection 4.1. In order to see the graphs clearer, we do not fix the same scale for the vertical axes in the subfigures.

4.2 | Six-dimensional system

The dynamics of a planar vertical takeoff and landing (PVTOL) aircraft can be described by the control system [25, 51–54]

$$\begin{cases} \dot{X}_1(t) &= X_2(t), \\ \dot{X}_2(t) &= -(1+u_1(t))\,\sin\left(X_5(t)\right) \;+\; \alpha\, u_2(t)\,\cos\left(X_5(t)\right), \\ \dot{X}_3(t) &= X_4(t), \\ \dot{X}_4(t) &= (1+u_1(t))\,\cos\left(X_5(t)\right) \;+\; \alpha\, u_2(t)\,\sin\left(X_5(t)\right) \;-\; 1, \\ \dot{X}_5(t) &= X_6(t), \\ \dot{X}_6(t) &= u_2(t), \\ t \geqslant 0, \\ X \;=\; (X_1, X_2, X_3, X_4, X_5, X_6)^\top, \quad u \;=\; (u_1, u_2)^\top, \quad n = 6, \quad m = 2, \\ X(0) \;=\; X^0 \;\in\; G \;=\; \mathbb{R}^6, \quad u(\cdot) \;\in\; \mathcal{U} \;=\; L^\infty_{\mathrm{loc}}([0, +\infty), U), \quad U \subseteq \mathbb{R}^2, \end{cases} \tag{53}$$

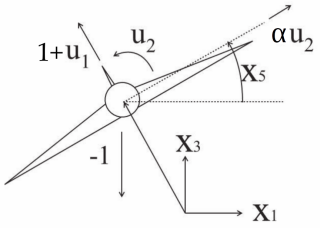


FIGURE 5 The PVTOL aircraft model in the example of Subsection 4.2.

where the following notation is used (see Fig. 5):

- t is a time variable;
- X_1 and X_3 are normalized quantities that correspond to the horizontal and vertical coordinates of the center of mass of the aircraft in a fixed inertial frame;
- X_5 is the roll angle that the aircraft makes with the positive horizontal axis;
- X_2, X_4, and X_6 are the rates of change of X_1, X_3, and X_5, respectively;
- u_1 and u_2 are normalized control inputs such that $1+u_1$ corresponds to the thrust (directed out the bottom of the aircraft), u_2 is related to the angular acceleration (rolling moment), and the origin $X = 0_6$ is a steady state for $u = 0_2$;
- the term -1 in the fourth dynamical equation represents the normalized gravitational acceleration;
- $\alpha > 0$ is a constant coefficient that characterizes the coupling between the rolling moment and the lateral acceleration of the aircraft.

As in [51, 52], introduce the new state variables

$$\begin{aligned} x_1 &= X_1 - \alpha \sin(X_5), \quad x_2 = X_2 - \alpha X_6 \cos(X_5), \\ x_3 &= X_3 + \alpha(\cos(X_5) - 1), \quad x_4 = X_4 - \alpha X_6 \sin(X_5), \\ x_5 &= X_5, \quad x_6 = X_6. \end{aligned} \tag{54}$$

Then the system (53) transforms to

$$\begin{cases} \dot{x}_1(t) = x_2(t), \\ \dot{x}_2(t) = -\left(1 + u_1(t) - \alpha\, x_6^2(t)\right) \sin(x_5(t)), \\ \dot{x}_3(t) = x_4(t), \\ \dot{x}_4(t) = \left(1 + u_1(t) - \alpha\, x_6^2(t)\right) \cos(x_5(t)) - 1, \\ \dot{x}_5(t) = x_6(t), \\ \dot{x}_6(t) = u_2(t), \\ t \geqslant 0, \\ x = (x_1, x_2, x_3, x_4, x_5, x_6)^\top, \quad u = (u_1, u_2)^\top, \\ x(0) = x^0 \in G = \mathbb{R}^6, \quad u(\cdot) \in \mathcal{U} = L^\infty_{\mathrm{loc}}([0, +\infty), U). \end{cases} \tag{55}$$

According to (54), $x = 0_6$ if and only if $X = 0_6$.

Let a_1, a_2 be positive constants and consider the compact convex control constraint set

$$U = [-a_1, a_1] \times [-a_2, a_2]. \tag{56}$$

Note that the forward completeness property (see [1, Item 5 of Assumption 2.1 and Remark 2.2]) is satisfied for (55) despite the terms with x_6^2. This follows from the fact that the forward completeness property holds for the system obtained from (55) by including one more state variable $x_7 = x_6^2$ (with $\dot{x}_7(t) = 2\, x_6(t)\, u_2(t)$) and thereby ensuring sublinear growth estimates for the right-hand sides of the dynamical equations.

Introduce also the running cost

$$g(x,u) \;=\; \frac{\lambda_1}{2}\,\|x\|^2 \;+\; \frac{\lambda_2}{2}\,\|u\|^2 \;+\; \lambda_3 \quad \forall x \in \mathbb{R}^6 \quad \forall u \in \mathbb{R}^2 \tag{57}$$

with constants $\lambda_1 > 0,\ \lambda_2 > 0,\ \lambda_3 \geqslant 0$.

It is not difficult to verify that a quadratic local CLF $V_{\mathrm{loc}}(\cdot)$ can be constructed via linearization as described in Section 2, with Remark 2.2 involved due to exact null-controllability, and that [1, Theorems 2.21, 2.25, 2.26, 2.28 and 2.29] can be used with the running cost (57) and with the mentioned local CLF.

A positive constant term λ_3 in the running cost (57) may be useful if the terminal set Ω_c is relatively small and the first two terms in (57) are rather small on some optimal trajectories near Ω_c.

We took

$$\alpha \;=\; 0.1, \quad a_1 \;=\; a_2 \;=\; 5, \quad \lambda_1 \;=\; 0.2, \quad \lambda_2 \;=\; 0.04, \quad \lambda_3 = 0.05. \tag{58}$$

The algebraic Riccati equation (10) was numerically solved via the `care` routine in the MATLAB environment. The corresponding parameters $\varkappa_1, \varkappa_2$ (see Remark 2.2) were chosen as

$$\varkappa_1 \;=\; \lambda_1 \;=\; 0.2, \quad \varkappa_2 \;=\; 5\lambda_2 \;=\; 0.2. \tag{59}$$

The range of appropriate levels c was approximated with the help of the related recommendations in Remark 2.2: c_{sup} was estimated to lie between 0.096 and 0.097. In order to handle possible multi-extremality, 100 initial guesses were randomly generated for each of the associated nonlinear programming problems, and the sequential quadratic programming algorithm of the `fmincon` routine was used in MATLAB. One might first think that it would be more reasonable to set $(\varkappa_1, \varkappa_2) = (\lambda_1, \lambda_2)$, but the minimal principal axis (12) appears to be markedly smaller in this case as compared to (59) ((12) was estimated as 0.1431 in this case and 0.2113 for (59)). We finally selected

$$c \;=\; 0.092. \tag{60}$$

In order to approximate the transformation $\tilde{V}(\cdot) = \zeta(V(\cdot)) = \ln(1+V(\cdot))$ of the value function as well as the corresponding gradient $\tilde{p}(\cdot) = (1/(1+V(\cdot)))\,\mathrm{D}V(\cdot)$ (the original costate is then given by $p(\cdot) = \mathrm{D}V(\cdot) = \exp\left(\tilde{V}(\cdot)\right)\tilde{p}(\cdot)$) in a certain bounded domain of the state space, we used the approach of Section 3.

The initial interpolation model was built as follows.

First, we constructed three sparse grids with 200, 400, and 600 nodes in $\mathrm{B}_2(0_n)$, $\mathrm{B}_4(0_n)$, and $\mathrm{B}_6(0_n)$, respectively. Halton quasi-random point set objects of MATLAB were involved. Originally, the quasi-random points were generated in the unit hypercube $[0,1]^6$, but they were then transformed to points in the minimal parallelepipeds enclosing the mentioned balls, and we eventually selected the desired numbers of points lying inside these balls and outside the target set Ω_c.

For each remaining grid node, we approximated the values of $\tilde{V}(\cdot)$ and $\tilde{p}(\cdot)$ via the GPOPS–II direct collocation software [36, 37] in MATLAB (which allows for costate estimation in addition to optimal control computation). The following configurations and parameters were in particular chosen in GPOPS–II:

- the tolerance 10^{-6} and at most 2000 iterations for the IPOPT nonlinear programming solver;
- the Patterson–Rao mesh refinement algorithm by default (but the `gpops2` function was rerun with the Liu–Rao–Legendre mesh refinement algorithm in case of any internal error);
- the tolerance 10^{-3}, at least 5 and at most 50 collocation points per mesh interval, and at most 100 iterations for the mesh refinement process;
- the maximal allowed time horizon 50.

For the output of the direct collocation process for each sparse grid node, we checked the fulfillment of the aforementioned tolerances as well as some other requirements:

- the obtained terminal state x_T should satisfy $\left|V_{\mathrm{loc}}(x_T) - c\right| \leqslant 10^{-4}$ (i. e., x_T should be close enough to $l_c = \partial\Omega_c$);
- the obtained initial costate estimate $\tilde{p}_0$ and the related original costate estimate p_0 (which is given by $p_0 = \exp\left(\tilde{V}_0\right)\tilde{p}_0$, where $\tilde{V}_0$ is the obtained approximation of $\tilde{V}(x_0)$ at the initial state x_0) should satisfy the infinitesimal decrease condition

and the Hamiltonian vanishing condition with certain tolerances, that is,

$$\min_{u\in U}\,\langle \tilde{p}_0,\, f(x_0,u)\rangle \;\leqslant\; -10^{-2},$$
$$\left|\min_{u\in U}\,\left\{\langle p_0,\, f(x_0,u)\rangle \,+\, g(x_0,u)\right\}\right| \;\leqslant\; 0.5$$

(the initial state x_0 is a sparse grid node).

When at least one of the imposed tolerances was not satisfied in the direct collocation process for a considered initial state, it was removed from the corresponding sparse grid.

Next, we built three gradient enhanced Kriging models from the obtained data on the three sparse grids via the MATLAB based toolbox [19]. The step tolerance, function tolerance, and maximal number of iterations for the `fminsearch` function used in this toolbox were taken as 10^{-3}, 10^{-5}, and 10^3, respectively. The weights in the composite interpolation model (18) were specified according to (20) and (21). The parameter ε in (21) and the parameter ν for determining the influence radii around the terminal set and grid nodes were chosen as

$$\varepsilon \,=\, 10^{-6}, \quad \nu \,=\, 15. \tag{61}$$

The minimal and maximal allowed radii of influence were set to

$$R_{\min} \,=\, 0.5, \quad R_{\max} \,=\, 5. \tag{62}$$

Orthogonal projections onto the terminal ellipsoid Ω_c were computed via the MATLAB based toolbox [42].

The subsequent model training consisted of 7 iterations, including the zeroth one. On each iteration, we numerically solved the nonlinear programming problem (34) with $E \,=\, \overline{\mathrm{B}}_4(0_6)$, $\Delta \,=\, 0.03$, and $T_{\max} \,=\, 12$. Regarding the number of initial guesses randomly generated in $E \setminus \Omega_c$ (note that the problem is essentially multi-extremal), it was set to 50 for the zeroth and last iterations (initial and final estimation), and 10 for all other iterations. The sequential quadratic programming algorithm of the `fmincon` function in MATLAB was run for each initial guess with the step, function, and constraint tolerances all taken as 10^{-3}, and with at most 10^3 objective function evaluations per single run. The condition of reaching the target set was specified as $V_{\mathrm{loc}}(x) \,-\, c \,+\, 10^{-3} \;\leqslant\; 0$. The dynamical system was numerically integrated by means of the second-order predictor–corrector scheme (also known as the modified Euler method) with time step $5\cdot 10^{-4}$. The control actions u^* were obtained from the states x and transformed costate estimates $\tilde{p}_{\mathrm{appr}}$ as follows: u^* was selected as the unique minimizer of $\left\langle \tilde{p}_{\mathrm{appr}},\, f(x,u)\right\rangle \,+\, 0.01\,\|u\|^2$ over all $u \in U$ when it satisfied $\left\langle \tilde{p}_{\mathrm{appr}},\, f(x,u^*)\right\rangle \,<\, -10^{-6}$ (some regularization was involved to reduce the control sharpness), and we took $u^* \,\in\, \mathrm{Arg}\min_{u\in U}\left\langle \tilde{p}_{\mathrm{appr}},\, f(x,u)\right\rangle$ otherwise (one can easily establish analytical representations for the control components in each of the two cases).

The average optimization time per one initial guess was around 21 minutes, and the total processing time was around 52.48 hours. We did not use parallel programming tools, but they could significantly reduce the total processing time (since different initial guesses could be independently handled at each training iteration).

In the beginning of each iteration except for the zeroth one, our model was modified by incorporating a new Kriging model for a new sparse grid around the state that was obtained by solving the aforementioned nonlinear programming problem on the previous iteration (the model data was computed at this center state and at 50 quasi-random points in the corresponding open ball of radius 1.5).

In the model training process, the level (34) was approximated as 0.3271 on the zeroth iteration and 0.4965 on the last iteration. However, as was already pointed out in the end of Subsection 3.2, the related estimate (35) seems rather conservative, and it has served mainly for the model training purposes.

A number of numerical tests were further performed for the resulting interpolation model (18).

First, we generated 10^4 states in line with the uniform distribution in $\mathrm{B}_5(0_6)\setminus\Omega_c$ and estimated $\tilde{V}_{\mathrm{appr}}(\cdot)$ and $\mathrm{D}\tilde{V}_{\mathrm{appr}}(\cdot)$ at these states via our trained model. We had

$$\tilde{V}_{\mathrm{appr}}(x) \,>\, \zeta\left(V_{l_c}\right) \,=\, \ln(1+c) \tag{63}$$

for all of the selected states x, and

$$\min_{u\in U}\left\langle \mathrm{D}\tilde{V}_{\mathrm{appr}}(x),\, f(x,u)\right\rangle \;<\; -10^{-6} \tag{64}$$

for 92.95% of those states. A similar experiment was conducted with 10^4 states randomly generated in the smaller region $[-1,1]^6\setminus\Omega_c$: (63) was satisfied for all of these states, and (64) held for 97.37% of them. Thus, the infinitesimal decrease condition was fulfilled for the vast majority of the selected states.

Next, we evaluated the interpolation errors of $\tilde{V}_{\mathrm{appr}}(\cdot)$ and $\mathrm{D}\tilde{V}_{\mathrm{appr}}(\cdot)$ with respect to the direct collocation estimates of $\tilde{V}(\cdot)$ and $\tilde{p}(\cdot)$ at 10^3 states that were selected according to the uniform distribution in $\mathrm{B}_5(0_6)\setminus\Omega_c$. Only states with a successful direct

collocation output were taken for this test. The corresponding mean values of $\left|\tilde{V}_{\text{appr}}(x) - \tilde{V}(x)\right|$ and $\left|\mathrm{D}\tilde{V}_{\text{appr}}(x) - \tilde{p}(x)\right|$ were estimated as 0.1758 and 0.3352, respectively, and the standard deviations were computed as 0.1711 and 0.2152, respectively. A similar experiment was conducted with 10^3 states randomly generated in $[-1, 1]^6 \setminus \Omega_c$, which led to the following estimates of the mentioned mean values and standard deviations: 0.0714, 0.3681, 0.058, and 0.1749 (the respective order of these quantities is the same as in the previous sentence). The obtained interpolation errors are certainly not negligible but seem acceptable due to a relatively small number of sparse grid nodes in the six-dimensional state space. There are totally 1383 nodes in the resulting grids, which is substantially smaller than, e. g., 44689 nodes in the six-dimensional Smolyak type sparse grids used in [6, Examples I and II].

The average time to evaluate the CLF and costate at a single state via our interpolation model was around 0.01725 seconds, which appeared to be essentially smaller than 0.7092 seconds that were on average required to do this via the GPOPS–II direct collocation software [36, 37] with the aforementioned configurations and parameters, although direct collocation based estimates were more accurate. Even when reducing the maximal number of mesh refinement iterations in GPOPS–II to 1, the average direct collocation time per single initial state decreased only slightly, from 0.7092 to 0.6629 seconds. Note also that we used the MATLAB toolbox [19] for fast evaluation of the gradients of the Kriging models and differentiated the weighted combination (18) in a semi-analytical fashion. For convenience, the numerical simulations for this subsection were performed in MATLAB. One can expect even smaller processing times for an implementation of our interpolation model (18) (together with the gradient enhanced Kriging tools of [19]) in a compiled programming language such as `C++`.

As was discussed in Subsection 3.1, an updated model (25), (26) may help to expand the practical region of asymptotic null-controllability. A numerical test was also conducted for such a modification with (27)–(29). We took $r_1 = 6$, $r_2 = 5$, and P_{extrapol} was obtained in offline mode as follows. We first generated $N' = 700$ quasi-random states $\left\{x_0^{[i]}\right\}_{i=1}^{N'} \subset \mathrm{B}_7(0_6)$, chose the time horizon $T' = 20$, and constructed 20 positive definite matrices

$$P_{\varkappa_1,\varkappa_2}, \quad \varkappa_1 = 0.2, \quad \varkappa_2 = 0.01,\, 0.02,\, \ldots,\, 0.19,\, 0.2; \tag{65}$$

recall that, for any $\varkappa_1 > 0$ and $\varkappa_2 > 0$, $P_{\varkappa_1,\varkappa_2}$ denotes a unique positive definite solution of the algebraic Riccati equation (10) for the linearized system (11) with diagonal matrices $Q = \varkappa_1 I_{n\times n}$, $R = \varkappa_2 I_{m\times m}$. The nonlinear programming problem

$$\begin{aligned}\gamma' \;=\; \inf_{\substack{P \in \mathbb{R}^{n\times n} \text{ is positive definite,}\\ \|P\| \,=\, 1}} \Bigg\{ & \frac{1}{N'} \sum_{i=1}^{N'} \frac{1}{T'} \int_0^{T'} \left\| x\left(t;\, x_0^{[i]}, P\right)\right\|^2 \mathrm{d}t \;: \\ & x\left(\cdot;\, x_0^{[i]}, P\right) \text{ is a state trajectory with } x(0) = x_0^{[i]} \text{ and feedback control taken from} \\ & \operatorname*{Arg\,min}_{w\in U} \left\langle \mathrm{D}\left(\langle Px, x\rangle\right),\, f(x,w)\right\rangle \;=\; \operatorname*{Arg\,min}_{w\in U} \left\langle 2Px,\, f(x,w)\right\rangle, \quad i = \overline{1, N'} \Bigg\}\end{aligned} \tag{66}$$

(here the matrix norm $\|P\|$ is induced by the Euclidean vector norm) was then solved via the sequential quadratic programming algorithm of the `fmincon` function in MATLAB for each of the 20 initial guesses (65), and P_{extrapol} was selected as the resulting minimizer for the 20 runs. The dynamical system was numerically integrated using the second-order predictor–corrector scheme with time step $2 \cdot 10^{-3}$, and the integral over $[0, T']$ divided by T' was approximated by the mean of the integrand on the time grid. Regarding the sequential quadratic programming algorithm, the step, function, and constraint tolerances were all chosen as 10^{-3}, and the number of objective function evaluations per single run was limited by 10^3. The total processing time for this auxiliary problem was around 21.32 hours.

When performing numerical simulations with the updated model (25)–(29), time sampling for costate recomputation needs to be done only if $x \in D_1$ (for $x \notin D_1$, one has $\tilde{V}_{\text{appr, updated}}(x) = \tilde{V}_{\text{appr, extrapol}}(x)$, and the extrapolation (29) allows for a very fast evaluation together with its gradient). Our first computational test for this model involved trajectory simulations for 10^3 initial states that were randomly generated in $\mathrm{B}_5(0_6) \setminus \Omega_c$. The dynamical system was numerically integrated on the time interval $[0, 60]$ via the second-order predictor–corrector method with time step $5 \cdot 10^{-4}$, and the sampling time $\Delta = 0.03$ was used for $x \in D_1$. Each of the approximated state trajectories eventually reached the target set Ω_c.

A similar numerical experiment was conducted for the other MPC scheme using the trained interpolation model (18) with sampling time $\Delta = 0.03$ in case $x \in \mathrm{B}_5(0_6)$ and direct collocation based open-loop control subarcs with significantly longer sampling time $\Delta_{\text{dir. colloc.}} = 0.6$ in case $x \notin \mathrm{B}_5(0_6)$ (recall the last paragraph in Subsection 3.1). In order to accelerate the online solution of the exit-time optimal open-loop control problems via the GPOPS–II software [36, 37] (though by losing in accuracy),

the maximal number of mesh refinement iterations was reduced to 1, as compared to our previous use of GPOPS–II for building the model data on the sparse grids. In this test, all simulated state trajectories with 10^3 initial states (randomly generated in $B_5(0_6) \setminus \Omega_c$) reached the target set Ω_c within the time interval $[0, 40]$ (the numerical integration algorithm and time step were selected as mentioned in the previous paragraph).

Although such MPC schemes with direct discretization techniques are expected to show better performance in simulations for deterministic systems, admissible sampling times may often not be enough for online solution of exit-time optimal control problems, and less accurate but faster models, such as that given by (25)–(29), may therefore seem more feasible from the practical point of view.

For testing the stabilization performance and robustness of the feedback control law based on the model (25)–(29), we also introduced a stochastic perturbation of the system (55) with noise in the second, fourth, and sixth dynamical equations (describing the accelerations for the three degrees of freedom):

$$\begin{cases} \dot{x}_1(t) = x_2(t), \\ \mathrm{d}x_2(t) = -\bigl(1 + u_1(t) - \alpha\, x_6^2(t)\bigr)\, \sin(x_5(t))\, \mathrm{d}t + \sigma_2\, \mathrm{d}w_2(t), \\ \dot{x}_3(t) = x_4(t), \\ \mathrm{d}x_4(t) = \bigl(\bigl(1 + u_1(t) - \alpha\, x_6^2(t)\bigr)\, \cos(x_5(t)) - 1\bigr)\, \mathrm{d}t + \sigma_4\, \mathrm{d}w_4(t), \\ \dot{x}_5(t) = x_6(t), \\ \mathrm{d}x_6(t) = u_2(t)\, \mathrm{d}t + \sigma_6\, \mathrm{d}w_6(t), \\ t \geqslant 0, \\ x(0) = x^0 \in \mathbb{R}^6. \end{cases} \tag{67}$$

Here x^0 is a deterministic initial state, σ_2, σ_4, and σ_6 are nonnegative constants (noise intensity parameters), $(w_2(\cdot), w_4(\cdot), w_6(\cdot))$ is a three-dimensional standard Brownian motion (Wiener process) on the time interval $[0, +\infty)$, and the stochastic ordinary differential equations are understood in the Itô sense. An open-loop control strategy can also represent a stochastic process if it is obtained from a closed-loop map.

Two initial conditions were considered:

$$X^0 = \Bigl(2,\, 3,\, 4,\, 1,\, \frac{\pi}{3},\, 1\Bigr)^\top, \tag{68}$$

$$X^0 = \Bigl(2,\, 3,\, 4,\, 1,\, \frac{3\pi}{5},\, 1\Bigr)^\top; \tag{69}$$

x^0 was determined from X^0 in line with (54). Note that (68) is the same initial state as was taken in [52, Section 5], and (69) is obtained from (68) just by increasing the roll angle X_5^0 to $3\pi/5$. We also selected

$$\sigma_2 = \sigma_4 = \sigma_6 = \sigma \tag{70}$$

and studied the two cases

$$\sigma = 0 \ \text{ (deterministic case)}, \quad \sigma = 0.1. \tag{71}$$

The MPC strategy was constructed from the model (25)–(29) with sampling time $\Delta = 0.03$ for $x \in D_1$ (in simulations with this model, no time sampling is needed for $x \notin D_1$ as discussed above); r_1, r_2, and P_{extrapol} were also taken as indicated above.

The Itô stochastic differential equations were numerically integrated on the time interval $[0, 40]$ via the Euler–Maruyama scheme that coincides with the Milstein scheme if the noise intensity matrix is constant and diagonal [55, 56]. The latter condition obviously holds for the system (67). The corresponding time step was chosen as 10^{-4}. Recall that, under certain smoothness and Lipschitz continuity conditions on the drift vector function and noise intensity matrix function, the Milstein scheme has strong convergence order one, while the order of the Euler–Maruyama scheme in general equals $1/2$ if the noise intensity matrix is not constant. In case $\sigma = 0.1$, 10^3 Monte Carlo iterations were performed for each of the two initial conditions (68), (69).

The black solid curves in Figs. 6 and 7 illustrate estimates of the mean values $\mathbb{E}\,\|x(t)\|$ and standard deviations $\sqrt{\mathrm{Var}\,\|x(t)\|}$ on the time interval $[0, 40]$ for our MPC implementation based on the model (25)–(29). Figs. 6 and 7 relate to the initial conditions (68) and (69), respectively.

For comparison, we also integrated the system (67) with the feedback control strategy that was developed and tested on real experiments by Fantoni et al. [52, 53]. This strategy is unconstrained, allows for an explicit analytical representation (which can be handled very fast, so no MPC has to be arranged in simulations), and stabilizes the deterministic PVTOL system if the initial

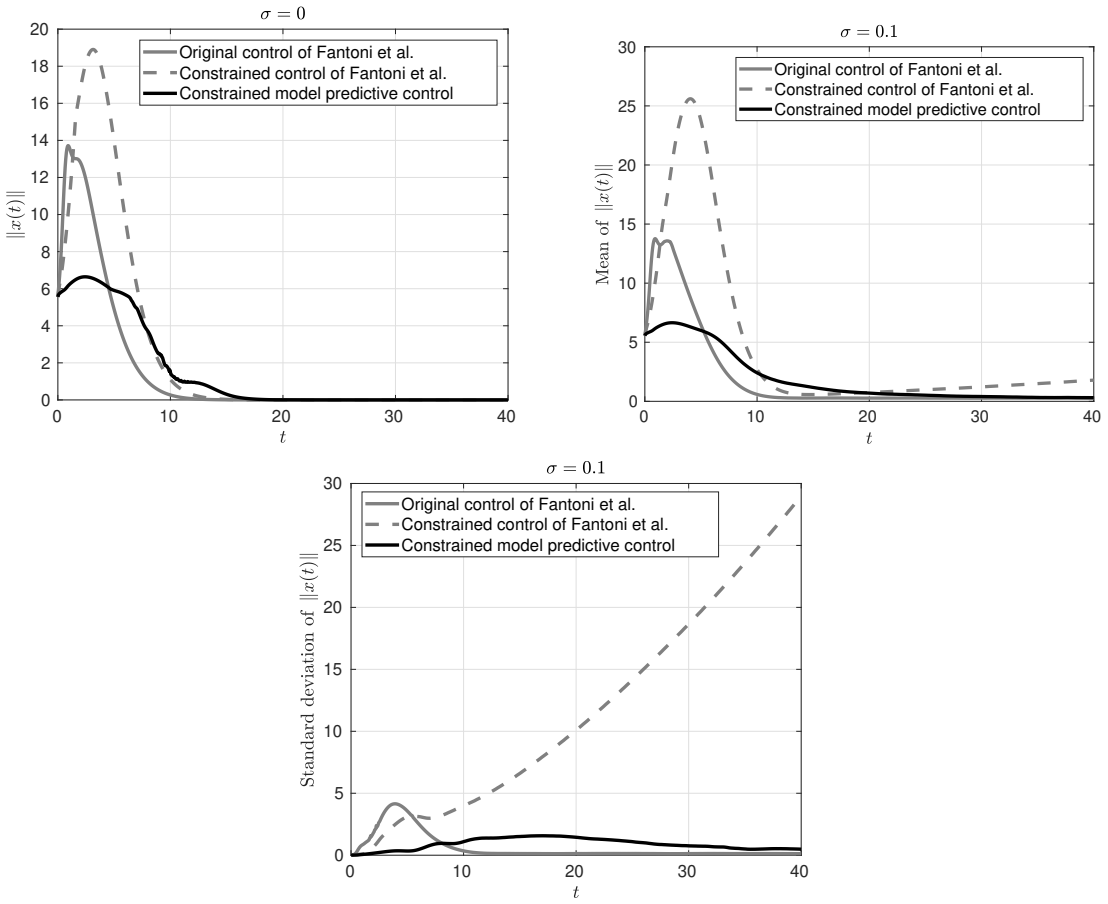


FIGURE 6 Estimates of the mean values $\mathbb{E}\,\|x(t)\|$ and standard deviations $\sqrt{\mathrm{Var}\,\|x(t)\|}$ for the original and constrained versions of the feedback control law of Fantoni et al. [52, 53] as well as for the MPC implementation based on the model (25)–(29) in the example of Subsection 4.2. The initial condition is (68). For the noise intensity parameter $\sigma_2 = \sigma_4 = \sigma_6 = \sigma$, the two cases (71) are considered. In order to see the graphs clearer, we do not fix the same scale for the vertical axes in the subfigures.

roll angle satisfies $|X_5(0)| < \pi/2$ (see [52, Theorem 3.1] or [53, Theorem 1]). It has to be noted that the expressions for the first and second time derivatives of the auxiliary variable r_1 in [53, (16) and (17) on page 414] are incorrectly written ($2\,/\cos(\theta)$ in the formula for $\dot r_1$ should be replaced with $2\tan(\theta)$, and the formula for $\ddot r_1$ should be accordingly modified). However, there is evidence that the correct relations were used in the further theoretical and practical investigation of [53]. In Figs. 6 and 7, the solid gray curves correspond to the original unconstrained strategy of Fantoni et al., while the dashed gray curves indicate its constrained (saturating) version defined as the orthogonal projection to the compact convex control constraint set (56). The latter control law was not considered by Fantoni et al., but we tested it to see how the saturation would reduce the control performance and robustness. The Euler–Maruyama scheme was used with the same time step 10^{-4}, and 10^3 Monte Carlo iterations were again implemented for each of the two initial conditions (68), (69) in case $\sigma = 0.1$.

Figs. 6 and 7 show that the MPC strategy based on the model (25)–(29) successfully mitigates state trajectory deflections from the origin in both cases (71) and for both initial conditions (68), (69), whereas the feedback control law of Fantoni et al. and its constrained modification fail to do so when the initial roll angle $X_5(0)$ exceeds $\pi/2$ ($X_5(0) = \pi/3 < \pi/2$ in (68), and $X_5(0) = 3\pi/5 > \pi/2$ in (69)).

Regarding the graphs of the mean deflections $\mathbb{E}\|x(\cdot)\|$ in Fig. 6 (in the deterministic case $\sigma = 0$, one obviously has $\mathbb{E}\|x(\cdot)\| = \|x(\cdot)\|$), our MPC strategy leads to a somewhat slower stabilization, but also to almost the same first moment $(1/40)\int_0^{40}\mathbb{E}\|x(t)\|\,\mathrm{d}t$ and the second moment $(1/40)\int_0^{40}(\mathbb{E}\|x(t)\|)^2\,\mathrm{d}t$ reduced by nearly half, as compared to the original control law of Fantoni et al. Besides, increasing the control bounds a_1, a_2 in (56) can in general improve the performance of the constrained MPC strategy. However, the original control law of Fantoni et al. is a priori unconstrained (which may limit

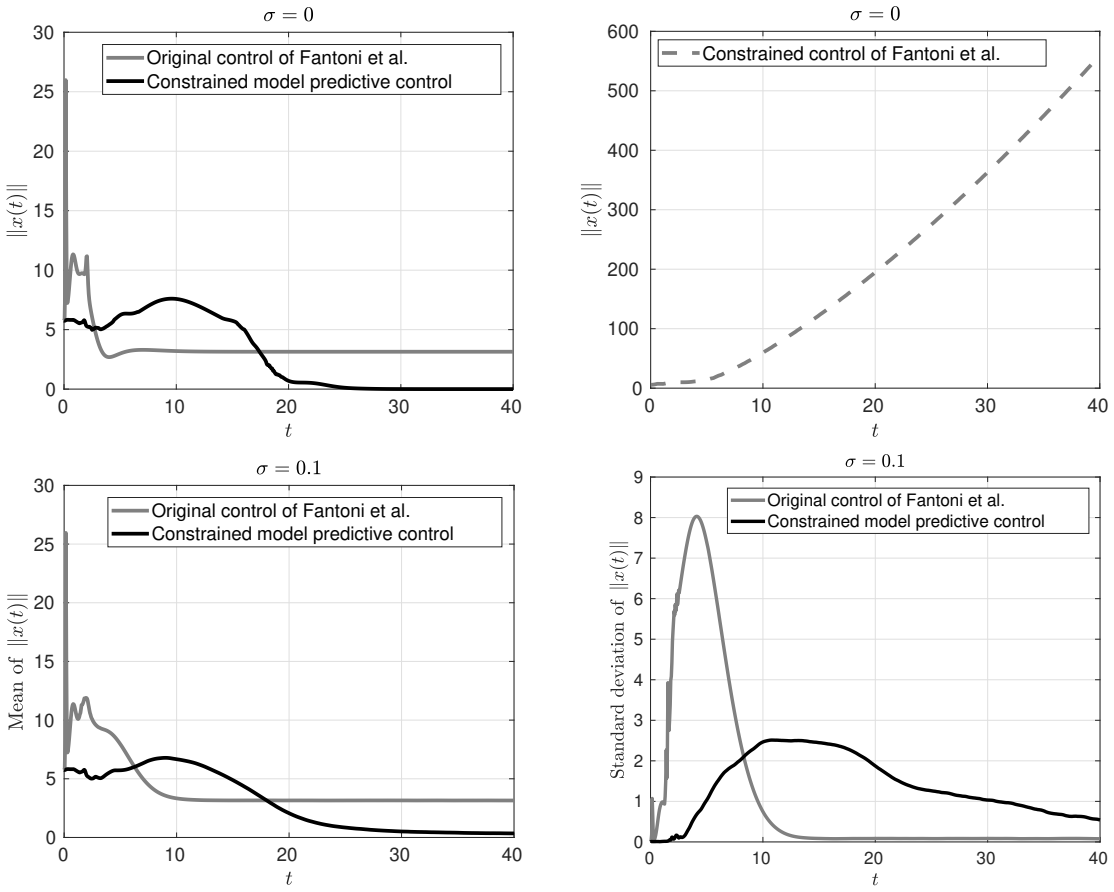


FIGURE 7 Estimates of the mean values $\mathbb{E}\,\|x(t)\|$ and standard deviations $\sqrt{\mathrm{Var}\,\|x(t)\|}$ for the original and constrained versions of the feedback control law of Fantoni et al. [52, 53] as well as for the MPC implementation based on the model (25)–(29) in the example of Subsection 4.2. The initial condition is (69). For the noise intensity parameter $\sigma_2 = \sigma_4 = \sigma_6 = \sigma$, the two cases (71) are considered. In order to see the graphs clearer, we do not fix the same scale for the vertical axes in the subfigures.

its practical use), and its constrained modification may not be robust enough (as follows from the corresponding graphs of the mean and standard deviation in Figs. 6 and 7).

5 | CONCLUSION

This paper has developed a curse-of-dimensionality-free numerical approach to feedback stabilization using the theoretical results of the previous study [1] on CLF characterization via exit-time optimal control. To attenuate also the curse of complexity for global or semi-global solution approximation in a high-dimensional region as well as for fast MPC implementation, we have combined gradient enhanced versions of the Kriging and inverse distance weighting methodologies for scattered grid interpolation. Let us list a number of important features of our approach:

- the interpolation model is composite, i. e., it involves multiple gradient enhanced Kriging models for separate grids, connected by a specific inverse distance weighting model (this is more flexible compared to using a single Kriging model);
- desired values of a CLF and its gradient are obtained not only at a finite number of sparse grid nodes, but also on the whole terminal (target) set in the related exit-time optimal control problem;
- incorporation of the gradient (costate) information often allows smaller sparse grids to be considered;

- new data can be conveniently taken into account within an offline training process (which can be significantly accelerated via parallel computing);
- online control estimation via sparse grid interpolation is typically much faster (though less accurate) than solving direct or characteristics based nonlinear programming problems for each sampling instant;
- in an online MPC implementation, sparse grid interpolation is performed if a current state lies in a certain bounded domain (the control actions are retrieved from the state and costate by means of the Hamiltonian minimum or CLF decrease condition), otherwise one may have to markedly increase the sampling time interval and approximate the corresponding control or costate via direct collocation or a characteristics based optimal control solver (an extrapolation of the form (25), (26) may also be used).

Note that Kriging in our composite model structure may in principle be replaced with other data-driven models, including neural networks [7, 21, 22], polynomial regression [8], and support vector regression [17]. This is a possible subject of future research. It also seems promising to combine our approach with scalable sum-of-squares based methods for stabilization of polynomial dynamical systems [57].

Moreover, the range of practical applicability of characteristics based frameworks to exit-time optimal control problems arising in stabilization problems with relatively high state space dimensions is worth studying further. In particular, an efficient modification of the framework proposed in Appendix may involve multiple shooting or indirect collocation [49, 50].

It is also relevant to design data-driven curse-of-dimensionality-free approaches for more general classes of Hamilton–Jacobi (HJ) equations arising in differential games and stochastic optimal control problems. As was already noted in the introduction, the work [21] has developed a neural network framework for a particular type of HJ equations with possibly nonconvex and nonconcave Hamiltonians depending only on the costate. This may help to treat some feedback differential games with state-independent Hamiltonians. Regarding stochastic optimal control problems [14, 22, 58], curse-of-dimensionality-free techniques to separately estimate the related value function gradients at different states are worth investigating, because their incorporation in scattered grid interpolation schemes or other data-driven models can attenuate also the curse of complexity. A significant challenge is that the complexity of Pontryagin's principle for stochastic optimal control problems [14] makes it more difficult to combine necessary and sufficient optimality conditions (i. e., to connect stochastically evolving costates with value function gradients) compared to the deterministic case.

Finally, one has to note the curse-of-dimensionality-free framework of [59] that uses deep neural networks to approximate compositional Lyapunov functions for systems of nonlinear ordinary differential equations, and another prospective research direction is to extend this framework to CLFs and optimal feedback stabilization problems.

ACKNOWLEDGEMENTS

This work was supported in part by AFOSR/AOARD grant FA2386-16-1-4066. We thank Professor Wei Kang, Naval Postgraduate School, for enlightening discussions and comments.

References

1. Yegorov, I., Dower, P. M., and Grüne, L. Synthesis of control Lyapunov functions and stabilizing feedback strategies using exit-time optimal control. Part I: Theory. Under review in *Optimal Control Applications and Methods*.

2. Michalska, H. and Mayne, D. Q. Robust receding horizon control of constrained nonlinear systems. *IEEE Transactions on Automatic Control* 1993; **38**(11): 1623–1633.

3. Chen, H. and Allgöwer, F. A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability. *Automatica* 1998; **34**(10): 1205–1217.

4. Fontes, F. A. C. C. A general framework to design stabilizing nonlinear model predictive controllers. *Systems & Control Letters* 2001; **42**(2): 127-143.

5. Jadbabaie, A. and Hauser, J. On the stability of receding horizon control with a general terminal cost. *IEEE Transactions on Automatic Control* 2005; **50**(5): 674–678.

6. Kang, W. and Wilcox, L. C. Mitigating the curse of dimensionality: sparse grid characteristics method for optimal feedback control and HJB equations. *Computational Optimization and Applications* 2017; **68**(2): 289–315.

7. Nakamura–Zimmerer, T., Gong, Q., and Kang, W. Adaptive deep learning for high-dimensional Hamilton–Jacobi–Bellman equations. arXiv: 1907.05317.

8. Azmi, B., Kalise, D., and Kunisch, K. Data-driven recovery of optimal feedback laws through optimality conditions and sparse polynomial regression. arXiv: 2007.09753.

9. Grancharova, A. and Johansen, T. A. *Explicit Nonlinear Model Predictive Control*: *Theory and Applications*. Springer-Verlag: Berlin, Heidelberg, 2012.

10. Tonon, D., Aronna, M. S., and Kalise, D. (Eds.). *Optimal Control*: *Novel Directions and Applications*. *Lecture Notes in Mathematics* **2180**. Springer: Cham, 2017.

11. Kalise, D., Kunisch, K., and Rao, Z. *Hamilton–Jacobi–Bellman Equations*: *Numerical Methods and Applications in Optimal Control*. *Radon Series on Computational and Applied Mathematics* **21**. De Gruyter: 2018.

12. Subbotin, A. I. *Generalized Solutions of First-Order PDEs*: *The Dynamical Optimization Perspective*. Birkhauser: Boston, 1995.

13. Melikyan, A. A. *Generalized Characteristics of First Order PDEs*: *Application in Optimal Control and Differential Games*. Birkhauser: Boston, 1998.

14. Yong, J. and Zhou, X. Yu. *Stochastic Controls*: *Hamiltonian Systems and HJB Equations*. Springer-Verlag: New York, 1999.

15. Fleming, W. H. and Soner, H. M. *Controlled Markov Processes and Viscosity Solutions*. Springer-Verlag: New York, 2006.

16. Bardi, M. and Capuzzo-Dolcetta, I. *Optimal Control and Viscosity Solutions of Hamilton–Jacobi–Bellman Equations*. Birkhauser: Boston, 2008.

17. Laurent, L., Le Riche, R., Soulier, B., and Boucard, P. A. An overview of gradient-enhanced metamodels with applications. *Archives of Computational Methods in Engineering* 2019; **26**: 61–106.

18. de Baar, J. H. S., Dwight, R. P., and Bijl, H. Improvements to gradient-enhanced Kriging using a Bayesian interpretation. *International Journal for Uncertainty Quantification* 2014; **4**(3): 205–223.

19. de Baar, J. Gradient-enhanced Kriging. MATLAB Central File Exchange. 2016. URL: https://www.mathworks.com/matlabcentral/fileexchange/60230-gradient-enhanced-kriging

20. Chilan, C. M. and Conway, B. A. Optimal nonlinear control using Hamilton–Jacobi–Bellman viscosity solutions on unstructured grids. *Journal of Guidance, Control, and Dynamics* 2020; **43**(1): 30–38.

21. Darbon, J., Langlois, G. P., and Meng, T. Overcoming the curse of dimensionality for some Hamilton–Jacobi partial differential equations via neural network architectures. *Research in the Mathematical Sciences* 2020; **7**: 20.

22. Beck, C., E, W., and Jentzen, A. Machine learning approximation algorithms for high-dimensional fully nonlinear partial differential equations and second-order backward stochastic differential equations. *Journal of Nonlinear Science* 2019; **29**: 1563–1619.

23. Isidori, A. *Nonlinear Control Systems*. Springer-Verlag: London, 1995.

24. Sepulchre, R., Jankovic, M., and Kokotovic, P. V. *Constructive Nonlinear Control*. Springer-Verlag: London, 1997.

25. Fantoni, I. and Lozano, R. *Non-linear Control for Underactuated Mechanical Systems*. Springer-Verlag: London, 2002.

26. Malisoff, M. and Mazenc, F. *Constructions of Strict Lyapunov Functions*. Springer-Verlag: London, 2009.

27. Choukchou-Braham, A., Cherki, B., Djemaï, M., and Busawon, K. *Analysis and Control of Underactuated Mechanical Systems*. Springer International Publishing: Switzerland, 2014.

28. Aguilar-Ibañez, C., Gutiérrez Frias, O., and Suárez Castañón, M. S. Lyapunov-based controller for the inverted pendulum cart system. *Nonlinear Dynamics* 2005; **40**(4): 367–374.

29. Aguilar-Ibañez, C., Martínez-García, J. C., Soria-López, A., and de Jesús Rubio, J. On the stabilization of the inverted-cart pendulum using the saturation function approach. *Mathematical Problems in Engineering* 2011; article ID 856015. DOI: 10.1155/2011/856015

30. Aguilar-Ibañez, C., Suárez Castañón, M. S., and de Jesús Rubio, J. Stabilization of the ball on the beam system by means of the inverse Lyapunov approach. *Mathematical Problems in Engineering* 2012; article ID 810597. DOI: 10.1155/2012/810597

31. Clarke, F. H., Ledyaev, Yu. S., Rifford, L., and Stern, R. J. Feedback stabilization and Lyapunov functions. *SIAM Journal on Control and Optimization* 2000; **39**(1): 25–48.

32. Sontag, E. D. *Mathematical Control Theory*: *Deterministic Finite Dimensional Systems*. Springer: New York, 1998.

33. Zabczyk, J. *Mathematical Control Theory*: *An Introduction*. Birkhäuser: Boston, 1995.

34. Schwarz, H. R., Rutishauser, H., and Stiefel, E. *Numerical Analysis of Symmetric Matrices* (translated by Hertelendy, P.). Prentice-Hall, Englewood Cliffs: New Jersey, 1973.

35. Michel, A. N., Sarabudla, N. R., and Miller, R. K. Stability analysis of complex dynamical systems — Some computational methods. *Circuits, Systems and Signal Processing* 1982; **1**: 171–202.

36. Patterson, M. A. and Rao, A. V. GPOPS–II: A MATLAB software for solving multiple-phase optimal control problems using *hp*-adaptive gaussian quadrature collocation methods and sparse nonlinear programming. *ACM Transactions on Mathematical Software* 2014; **41**(1): 1:1–1:37.

37. Patterson, M. A. and Rao, A. V. GPOPS–II user's guide. 2016. URL: http://www.gpops2.com

38. Nie, Y., Faqir, O., and Kerrigan, E. ICLOCS2 optimal control software. Imperial College London, 2018. URL: http://www.ee.ic.ac.uk/ICLOCS/

39. Becerra, V. M. Solving complex optimal control problems at no cost with PSOPT. *Proceedings of The IEEE Multi-conference on Systems and Control* 2010; 1391–1396.

40. Becerra, V. M. PSOPT optimal control solver user manual. 2011. URL: http://www.psopt.org

41. Bonnans, F., Martinon, P., Giorgi, D., Grélard, V., Maindrault, S., Tissot, O., and Liu, J. BOCOP 2.1.0 — User Guide. October 30, 2017. URL: http://www.bocop.org

42. Luong, B. Euclidean projection on ellipsoid and conic. MATLAB Central File Exchange. 2010. URL: https://www.mathworks.com/matlabcentral/fileexchange/27711-euclidian-projection-on-ellipsoid-and-conic

43. Lange, M., Zühlke, D., Holz, O., and Villmann, T. Applications of l_p-norms and their smooth approximations for gradient based learning vector quantization. *Proceedings of The* 22*nd European Symposium on Artificial Neural Networks*, *Computational Intelligence and Machine Learning*, Bruges, April 23–25, 2014, pp. 271–276. URL: https://www.elen.ucl.ac.be/Proceedings/esann/esannpdf/es2014-153.pdf

44. Shahmansoorian, A. Inverse optimal control and construction of control Lyapunov functions. *Journal of Mathematical Sciences* 2009; **161**(2): 297–307.

45. Afanas'ev, V. N., Kolmanovskii, V. B., and Nosov, V. R. *Mathematical Theory of Control Systems Design*. Kluwer: Dordrecht, 1996.

46. Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. *Numerical Recipes*: *The Art of Scientific Computing*. Cambridge University Press: New York, 2007.

47. Yegorov, I. and Dower, P. Perspectives on characteristics based curse-of-dimensionality-free numerical approaches for solving Hamilton–Jacobi equations. To appear in *Applied Mathematics and Optimization*. DOI: 10.1007/s00245-018-9509-6

48. Chow, Y. T., Darbon, J., Osher, S., and Yin, W. Algorithm for overcoming the curse of dimensionality for state-dependent Hamilton–Jacobi equations. *Journal of Computational Physics* 2019; **387**(15): 376–409.

49. Rao, A. V. A survey of numerical methods for optimal control. *Advances in the Astronautical Sciences* 2010; **135**(1): 497–528.

50. Trélat, E. Optimal control and applications to aerospace: some results and challenges. *Journal of Optimization Theory and Applications* 2012; **154**(3): 713–758.

51. Olfati–Saber, R. Global configuration stabilization for the VTOL aircraft with strong input coupling. *IEEE Transactions on Automatic Control* 2002; **47**(11): 1949–1952.

52. Fantoni, I., Lozano, R., and Castillo, P. A simple stabilization algorithm for the PVTOL aircraft. *IFAC Proceedings Volumes* 2002; **35**(1): 7–11.

53. Fantoni, I., Palomino, A., Castillo, P., Lozano, R., and Pégard, C. Control strategy using vision for the stabilization of an experimental PVTOL aircraft setup. In: Menini, L., Zaccarian, L., Abdallah, C. T. (Eds.), *Current Trends in Nonlinear Systems and Control*, *In Honor of Petar Kokotović and Turi Nicosia*, *Systems and Control*: *Foundations* & *Applications*, Birkhäuser, Boston, 2006, pp. 407–419.

54. Hably, A., Kendoul, F., Marchand, N., and Castillo, P. Further results on global stabilization of the PVTOL aircraft. In: Commault, C. and Marchand, N. (Eds.), *Positive Systems*, *Proceedings of the Second Multidisciplinary International Symposium on Positive Systems*: *Theory and Applications* (*POSTA* 06), *Lecture Notes in Control and Information Sciences* (*LNCIS* 341), Springer-Verlag, Berlin, Heidelberg, 2006, pp. 303–310.

55. Kloeden, P. E. and Platen, E. *Numerical Solution of Stochastic Differential Equations*. Springer-Verlag: Berlin, 1995.

56. Carletti, M. Numerical solution of stochastic differential problems in the biosciences. *Journal of Computational and Applied Mathematics* 2006; **185**(2): 422–440.

57. Kundu, S. and Anghel, M. A sum-of-squares approach to the stability and control of interconnected systems using vector Lyapunov functions. *Proceedings of The* 2015 *American Control Conference*, pp. 5022–5028. DOI: 10.1109/ACC.2015.7172121

58. Kushner, H. J. and Dupuis, P. *Numerical Methods for Stochastic Control Problems in Continuous Time*. Springer: New York, 2001.

59. Grüne, L. Computing Lyapunov functions using deep neural networks. To appear in *Journal of Computational Dynamics*. arXiv: 2005.08965.

□

APPENDIX

SUPPLEMENTARY NUMERICAL METHOD USING A CHARACTERISTICS BASED REPRESENTATION OF THE VALUE FUNCTION

Here we design a supplementary characteristics based method to approximate the value function and costate with the help of [1, Theorem 2.29]. It is used in Subsection 4.1 due to explicit connection with the characteristics based representation of the value function. For the considered exit-time optimal control problems, this method appears to work markedly slower than direct collocation (utilized in Subsection 4.2) and hence requires further development for wider application (which is a possible

subject of future research). Similar algorithms have nonetheless shown sufficiently fast and accurate performance for a number of fixed-horizon high-dimensional optimal control problems [47, 48].

First, recall the following notation from [1, Section 2]:

$$T_{\Omega_c}(x_0, u(\cdot)) \stackrel{\text{def}}{=} \inf\{T \in [0, +\infty) : x(T; x_0, u(\cdot)) \in \Omega_c\} \quad \forall x_0 \in G \quad \forall u(\cdot) \in \mathcal{U}, \tag{A.1}$$

$$\begin{aligned} H(x, u, p, \tilde{p}) &\stackrel{\text{def}}{=} \langle p, f(x, u)\rangle + \tilde{p}\, g(x, u), \\ \mathcal{H}(x, p, \tilde{p}) &\stackrel{\text{def}}{=} \min_{u' \in U} H(x, u', p, \tilde{p}) \\ &\forall\, (x, u, p, \tilde{p}) \in G \times U \times \mathbb{R}^n \times \mathbb{R}, \end{aligned} \tag{A.2}$$

$$U^*(x, p, \tilde{p}) \stackrel{\text{def}}{=} \operatorname{Arg}\min_{u \in U} H(x, u, p, \tilde{p}) \quad \forall\, (x, p, \tilde{p}) \in G \times \mathbb{R}^n \times \mathbb{R}. \tag{A.3}$$

The formulation of [1, Theorem 2.29] is repeated here for convenience.

Theorem A.1. [1, Theorem 2.29] Let Assumptions [1, 2.1, 2.8, 2.16, 2.22 and 2.24] hold. For any initial state $x_0 \in \mathcal{D}_0 \setminus \Omega_c$, the Kruzhkov transformed value $v(x_0)$ defined by [1, (22), (23), (40)] is the minimum of

$$1 - \exp\left\{-\int_0^{T_{\Omega_c}(x_0, u^*(\cdot))} g(x^*(t), u^*(t))\, \mathrm{d}t - c\right\} \tag{A.4}$$

over the solutions of the characteristic Cauchy problems

$$\begin{cases} \dot{x^*}(t) = \mathrm{D}_p H(x^*(t), u^*(t), p^*(t), \tilde{p}^*) = f(x^*(t), u^*(t)), \\ \dot{p^*}(t) = -\mathrm{D}_x H(x^*(t), u^*(t), p^*(t), \tilde{p}^*) \\ \qquad = -(\mathrm{D}_x f(x^*(t), u^*(t)))^\top p^*(t) - \tilde{p}^*\, \mathrm{D}_x g(x^*(t), u^*(t)), \\ u^*(t) \in U^*(x^*(t), p^*(t), \tilde{p}^*), \\ t \in I(x_0, u^*(\cdot)) \stackrel{\text{def}}{=} \begin{cases} [0, T_{\Omega_c}(x_0, u^*(\cdot))], & T_{\Omega_c}(x_0, u^*(\cdot)) < +\infty, \\ [0, +\infty), & T_{\Omega_c}(x_0, u^*(\cdot)) = +\infty, \end{cases} \\ x^*(0) = x_0, \quad p^*(0) = p_0, \end{cases} \tag{A.5}$$

for all extended initial costates

$$(p_0, \tilde{p}^*) \in \{(p, \tilde{p}) : p \in \mathbb{R}^n, \ \tilde{p} \in \{0, 1\}\}. \tag{A.6}$$

Moreover, the same value is obtained when minimizing over the bounded set

$$(p_0, \tilde{p}^*) \in \{(p, \tilde{p}) \in \mathbb{R}^n \times \mathbb{R} : \|(p, \tilde{p})\| = 1, \ \tilde{p} \geqslant 0\}, \tag{A.7}$$

or even over its subset

$$(p_0, \tilde{p}^*) \in \{(p, \tilde{p}) \in \mathbb{R}^n \times \mathbb{R} : \|(p, \tilde{p})\| = 1, \ \tilde{p} \geqslant 0, \ \mathcal{H}(x_0, p, \tilde{p}) = 0\}. \tag{A.8}$$

In order to ensure the uniqueness of the solutions of the characteristic Cauchy problems (A.5) in the normal case $\tilde{p}^* > 0$, the following conditions are imposed.

Assumption A.2. The extremal control map (A.3) is a singleton for $\tilde{p} > 0$, and the corresponding function defined on $G \times \mathbb{R}^n \times (0, +\infty)$ and taking values in U is locally Lipschitz continuous.

This often holds if, for instance, the running cost is regularized by adding a suitable control-dependent term. In the abnormal case $\tilde{p}^* = 0$, there typically exist states and costates for which the extremal control map is nonsingleton, regardless of the running cost. However, one can expect that abnormal characteristics would rarely be optimal, since they do not take the running cost into account. If the extremal control map on a characteristic trajectory becomes nonsingleton at some time during computations, one can select any extremal control action at this time. Note also that, for some particular classes of optimal control problems, an additional analysis via Pontryagin's principle may allow characteristics based methods to be modified so that singular regimes are explicitly handled and the nonuniqueness in the choice of extremal control actions is avoided (see [47, Examples 3.14 and 3.15] related to the case of a fixed finite horizon).

A.1 Practical specification of exit times and the domain of asymptotic null-controllability

Next, recall the representation of the domain of asymptotic null-controllability $\mathcal{D}_0$ in [47, Theorem 2.27]:

$$\mathcal{D}_0 \;=\; \{x_0 \in G \,:\, V(x_0) < +\infty\} \;=\; \{x_0 \in G \,:\, v(x_0) < 1\}. \tag{A.9}$$

Before evaluating the CLF at a particular state $x_0 \in G$, one usually does not know if $x_0 \in \mathcal{D}_0$ or not. Even if the initial state lies in $\mathcal{D}_0$, there may still exist extended initial costates $(p_0, \tilde{p}^*)$ that generate characteristic trajectories with infinite exit time and with the cost (A.4) equal to 1.

Let us provide a practical rule to determine costs and exit (terminal) times during numerical integration of the characteristic Cauchy problems (A.5). A priori, it is reasonable to fix a sufficiently large finite upper bound $T_{\max} > 0$ for exit times, even though this is in general a heuristic choice. Take also a sufficiently small parameter $\varepsilon \in (0, 1)$. It is proposed to stop integrating the characteristic system when at least one of the following conditions starts to hold:

1) the a priori selected upper bound for terminal times is reached, i. e., $t = T_{\max}$;
2) the target set is entered, i. e., $x^*(t) \in \Omega_c$;
3) the accumulated cost becomes very close to the maximum value 1, i. e.,

$$1 \;-\; \exp\left\{-\int\limits_0^t g(x^*(s),\, u^*(s))\, ds \;-\; c\right\} \;\geqslant\; 1 - \varepsilon.$$

These three cases accordingly define practical exit times. In Case 2, the cost is specified as (A.4), while, in Cases 1 and 3, it is set to $1 - \varepsilon$.

Let $\hat{v}:\; G \setminus \Omega_c \;\to\; [0,\, 1-\varepsilon]$ be the approximation of $v(\cdot)$ in $G \setminus \Omega_c$ obtained by incorporating the aforementioned arguments in a numerical method building on Theorem A.1. Select one more parameter $\varepsilon_1 \in (0, 1)$, which is sufficiently small but not less than ε. Then the domain $\mathcal{D}_0$ can be approximated by its inner estimate

$$\hat{\mathcal{D}}_0 \;\stackrel{\text{def}}{=}\; \Omega_c \;\cup\; \{x_0 \in G \setminus \Omega_c \,:\, \hat{v}(x_0) < 1 - \varepsilon_1\}. \tag{A.10}$$

A.2 Shooting problem for reverse-time characteristics

Let us refer to the finite-dimensional optimization problem formulated in Theorem A.1 as the main problem. The cost function in this problem may be essentially multi-extremal for some initial states $x_0 \in G$. There may in particular exist a relatively large subset of (A.7) consisting of the extended initial costates $(p_0, \tilde{p}^*)$ for which the state trajectories of (A.5) do not reach the target set Ω_c and the approximate cost defined above equals $1 - \varepsilon$. It is hence reasonable first to introduce an auxiliary problem, whose solution can then serve as an initial guess for an iterative algorithm applied to the main problem.

Consider the characteristic system rewritten in reverse time τ, that is,

$$\left\{\begin{aligned}
&\frac{\mathrm{d}\hat{x}^*(\tau)}{\mathrm{d}\tau} \;=\; -\mathrm{D}_p H(\hat{x}^*(\tau),\, \hat{u}^*(\tau),\, \hat{p}^*(\tau),\, \tilde{p}^*) \;=\; -f(\hat{x}^*(\tau),\, \hat{u}^*(\tau)),\\
&\frac{\mathrm{d}\hat{p}^*(\tau)}{\mathrm{d}\tau} \;=\; \mathrm{D}_x H(\hat{x}^*(\tau),\, \hat{u}^*(\tau),\, \hat{p}^*(\tau),\, \tilde{p}^*)\\
&\qquad\qquad \;=\; (\mathrm{D}_x f(\hat{x}^*(\tau),\, \hat{u}^*(\tau)))^{\top}\, \hat{p}^*(\tau) \;+\; \tilde{p}^*\, \mathrm{D}_x g(\hat{x}^*(\tau),\, \hat{u}^*(\tau)),\\
&\hat{u}^*(\tau) \;\in\; U^*(\hat{x}^*(\tau),\, \hat{p}^*(\tau),\, \tilde{p}^*),\\
&\tau \;\in\; I(\hat{x}^*(0),\, \hat{p}^*(0),\, \tilde{p}^*),
\end{aligned}\right. \tag{A.11}$$

where $I(\hat{x}^*(0),\, \hat{p}^*(0),\, \tilde{p}^*)$ is the maximum extendability time interval with zero left endpoint during which the related solutions emanating from $(\hat{x}^*(0),\, \hat{p}^*(0))$ satisfy $\hat{x}^*(\tau) \in G$, and take also

$$\hat{x}^*(0) \;\in\; \Omega_{c_1}, \quad \hat{p}^*(0) \;\in\; \mathrm{N}\left(\hat{x}^*(0);\, \Omega_{c_1}\right) \tag{A.12}$$

for some parameter $c_1 \in (0, c]$. The conditions (A.12) come from Pontryagin's principle for the exit-time optimal control problem with the target set

$$\Omega_{c_1} \;\stackrel{\text{def}}{=}\; \left\{x \in \overline{\Omega} \,:\, V_{\mathrm{loc}}(x) \leqslant c_1\right\} \;\subseteq\; \Omega_c \tag{A.13}$$

instead of Ω_c. The aim is to get to a selected state $x_0 \in G$ as close as possible, which leads to a shooting problem. The level c_1 is allowed to be less than c in order to make the shooting more robust, i. e., to increase the possibility that the resulting initial guess for the main optimization problem with $x_0 \in \mathcal{D}_0$ generates a forward-time characteristic state trajectory reaching the original target set Ω_c within the fixed time interval $[0, T_{\max}]$. Note also that a similar reduction of a terminal sublevel set is used in [2] for increasing the robustness of a receding horizon stabilization algorithm.

Additional properties need to be imposed.

Assumption A.3. $c_1 \in (0, c]$ is a constant. [1, Item 4 of Assumption 2.8] holds when c is replaced with c_1 and C_3 remains the same. [1, Item 4 of Assumption 2.16] holds when c is replaced with c_1 and C_6 is replaced with $C_6' \in (0, C_6]$.

Denote also

$$l_{c_1} \stackrel{\text{def}}{=} \left\{ x \in \overline{\Omega} : V_{\text{loc}}(x) = c_1 \right\} = \partial\Omega_{c_1}. \tag{A.14}$$

Assumption A.4. The local CLF $V_{\text{loc}}(\cdot)$ is continuously differentiable in Ω, and, for any direction $\xi \in \mathbb{R}^n$ with $\|\xi\| = 1$, there exists a unique state

$$x^*_{\text{term}}(\xi) \in l_{c_1} \cap \{\lambda\xi : \lambda > 0\}. \tag{A.15}$$

Remark A.5. Let [1, Assumptions 2.1, 2.8] and Assumption A.3 hold. It is not difficult to verify that Assumption A.4 also holds if, for example, the following conditions are fulfilled:

- Ω is convex;
- $V_{\text{loc}}(\cdot)$ is continuously differentiable and convex in Ω (the convexity of $V_{\text{loc}}(\cdot)$ implies the convexity of its sublevel sets, such as Ω_c and Ω_{c_1});
- $V_{\text{loc}}(\cdot)$ satisfies
$$\langle x, \mathrm{D}V_{\text{loc}}(x)\rangle > 0 \quad \forall x \in l_{c_1}$$
(this holds in particular for quadratic local CLFs).

□

In line with Assumptions A.3, A.4 and the Hamiltonian vanishing condition in Pontryagin's principle, the initial data (A.12) for the reverse-time characteristic system (A.11) is taken as

$$\begin{aligned} \hat{x}^*(0) &= x^*_{\text{term}}(\xi) \in l_{c_1}, \quad \xi \in \mathbb{R}^n, \quad \|\xi\| = 1, \\ \hat{p}^*(0) &= \varkappa\, \mathrm{D}V_{\text{loc}}(\hat{x}^*(0)), \quad \varkappa \geqslant 0, \\ \mathcal{H}(\hat{x}^*(0), \hat{p}^*(0), \tilde{p}^*) &= 0, \quad \tilde{p}^* \geqslant 0. \end{aligned} \tag{A.16}$$

By adopting the normalization condition $\|(\hat{p}^*(0), \tilde{p}^*)\| = 1$, one obtains

$$\tilde{p}^* \in [0, 1], \quad \varkappa = \frac{\sqrt{1 - (\tilde{p}^*)^2}}{\|\mathrm{D}V_{\text{loc}}(\hat{x}^*(0))\|}, \tag{A.17}$$

and $\mathrm{D}V_{\text{loc}}(\hat{x}^*(0)) \neq 0_n$ due to the infinitesimal decrease condition on the local CLF and $\inf\{g(x,u) : x \in l_{c_1},\ u \in U\} > 0$ (recall [1, Assumptions 2.8, 2.16] and Assumption A.3). The latter properties also yield that, for any $\xi \in \mathbb{R}^n$ with $\|\xi\| = 1$, the function

$$[0, 1] \ni \tilde{p} \mapsto \mathcal{H}\left(x^*_{\text{term}}(\xi), \ \frac{\sqrt{1 - \tilde{p}^2}}{\|\mathrm{D}V_{\text{loc}}(x^*_{\text{term}}(\xi))\|}\, \mathrm{D}V_{\text{loc}}(x^*_{\text{term}}(\xi)), \ \tilde{p} \right) \tag{A.18}$$

is negative at $\tilde{p} = 0$ and positive at $\tilde{p} = 1$, i. e., at least one root $\tilde{p}^*$ exists and satisfies

$$0 < \tilde{p}^* < 1 \tag{A.19}$$

(which in particular excludes the abnormal case $\tilde{p}^* = 0$). Possible multiple roots on $(0, 1)$ can be numerically bracketed, e. g., by using the `zbrak` routine from [46, §9.1]. Each of them is further handled, and a root with the best shooting performance is finally selected.

The shooting goal is to minimize the lowest deviation

$$\min_{\tau \in [0, T_{\max}] \cap I(\hat{x}^*(0),\, \hat{p}^*(0),\, \tilde{p}^*)} \left\| \hat{x}^*(\tau) - x_0 \right\|^2 \tag{A.20}$$

from a state $x_0 \in G$ over the solutions of the reverse-time characteristic Cauchy problems (A.11), (A.16), (A.17). One therefore arrives at optimizing over the vectors ξ on the unit sphere in $\mathbb{R}^n$, which can be parametrized as follows:

$$\begin{cases} \xi_1 = \prod_{i=1}^{n-1} \sin\theta_i, \\ \xi_j = \cos\theta_{j-1} \prod_{i=j}^{n-1} \sin\theta_i, \quad j = \overline{2, n-1}, \\ \xi_n = \cos\theta_{n-1}, \\ 0 \leqslant \theta_1 < 2\pi, \quad 0 \leqslant \theta_j \leqslant \pi, \quad j = \overline{2, n-1}. \end{cases} \tag{A.21}$$

The periodicity in the angles θ_i, $i = \overline{1,n}$, allows for performing unconstrained optimization over them. Possible multi-extremality in the auxiliary problem can be treated by applying an iterative optimization method to a fixed number of random initial guesses generated according to the uniform angles distribution.

A.3 Main problem for forward-time characteristics and first-order approximation of the value function

If $(\hat{x}^*(\cdot), \hat{p}^*(\cdot), \tilde{p}^*)$ is an optimal shooting characteristic and τ' is a minimizer in (A.20), then the extended costate

$$\left(\frac{\hat{p}^*(\tau')}{\tilde{p}^*}, 1 \right) \tag{A.22}$$

(normalized so as to make the last coordinate equal to 1) can specify the initial guess for an iterative optimization method applied to the main problem for forward-time characteristics. The normalization in (A.22) allows for optimizing with respect to $p_0 \in \mathbb{R}^n$ in the main problem, as well as for representing the gradient $\mathrm{D}V(\cdot)$ of the original value function along optimal characteristics in case $x_0 \in \mathcal{D}_0$ ($V(x_0) < +\infty$) directly via the costate.

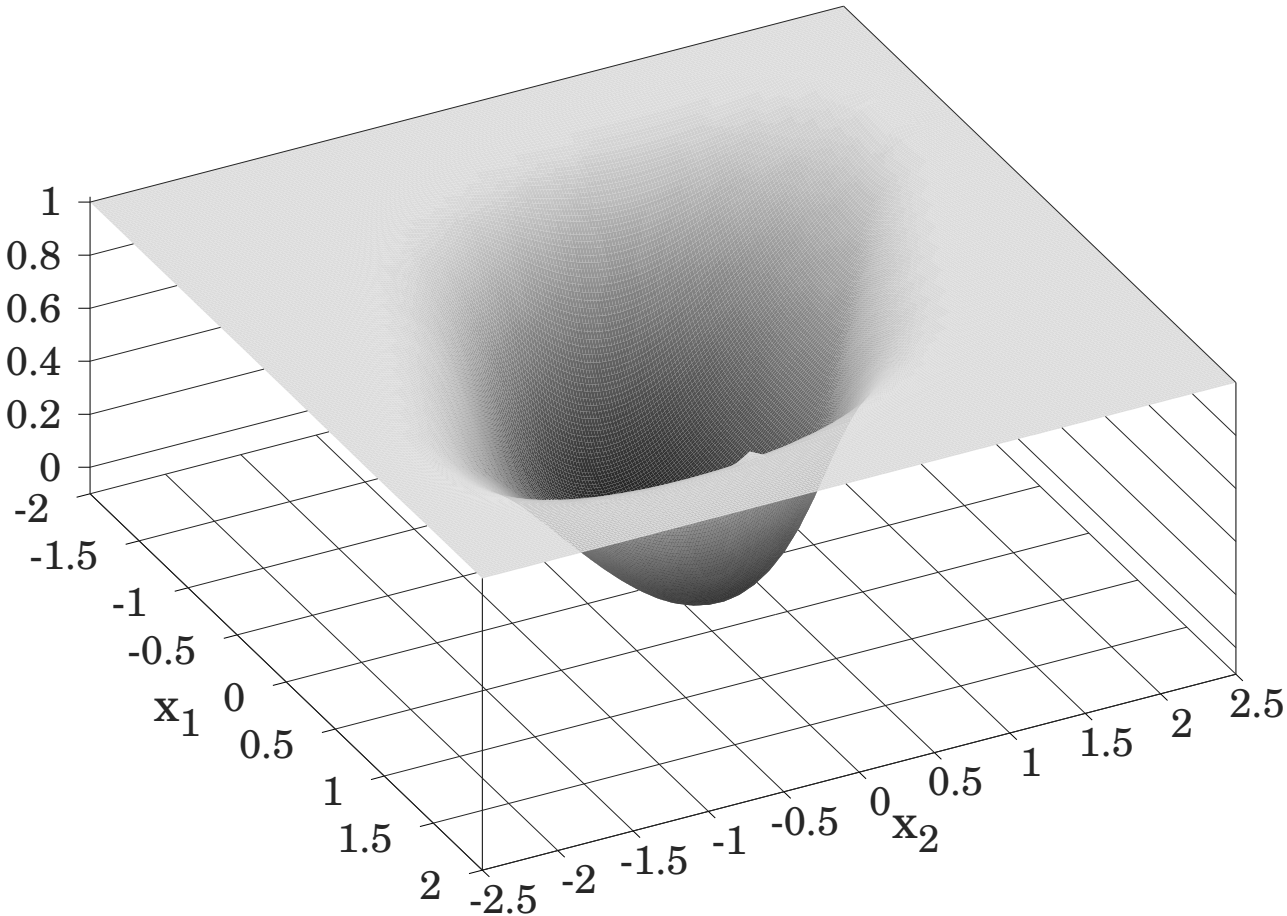
If p_0 is an optimal initial costate in the main problem (with $\tilde{p}^* = 1$) for an initial state x_0, the optimal control action at this state is chosen from $U^*(x_0, p_0, 1)$.

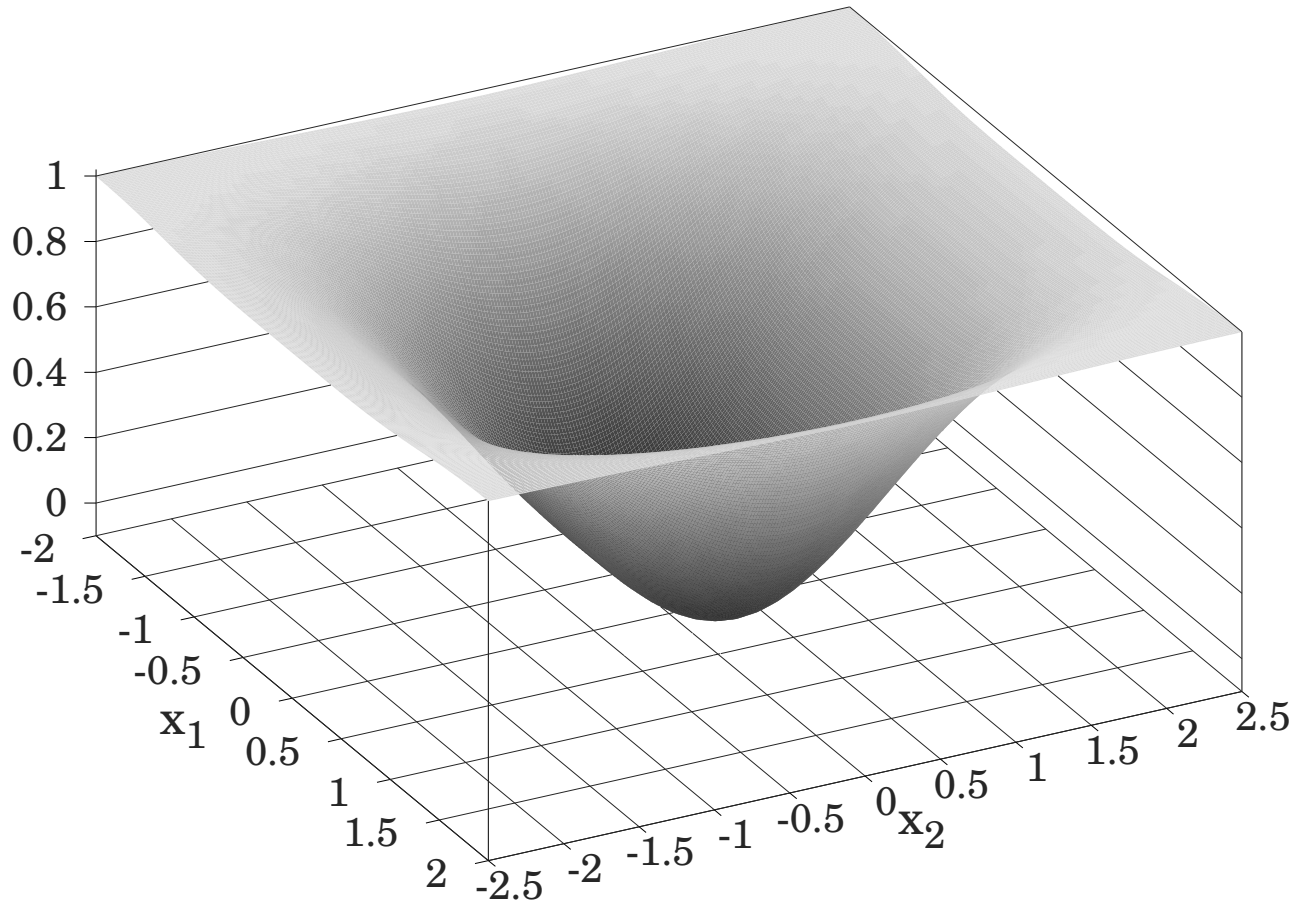
Even if $x_0 \in \mathcal{D}_0$ and the optimal cost (A.20) in the auxiliary shooting problem is rather small, the related initial guess (A.22) for the main problem and also some neighboring costates might still lead to a forward-time characteristic state trajectory that emanates from x_0 but does not reach the target set Ω_c within the time interval $[0, T_{\max}]$. In this situation, the shooting is not accurate enough, but the following technique for evaluating $v(x_0)$ may be helpful. Select two sufficiently small positive parameters δ_1, δ_2. If the square root of the optimal shooting cost (A.20) is less than δ_1 and $\hat{x}_0 = \hat{x}^*(\tau')$ is the closest state to x_0 on the optimal shooting characteristic, then one can solve the main optimization problem (with $\tilde{p}^* = 1$) for the initial state $\hat{x}_0$ and use the corresponding value $v(\hat{x}_0)$ and optimal initial costate $\hat{p}_0$ in order to approximate $v(x_0)$. More precisely, if $\|\hat{x}_0 - x_0\| < \delta_1$ and $v(\hat{x}_0) = 1 - e^{-V(\hat{x}_0)} < 1 - \varepsilon$, consider the first-order estimate

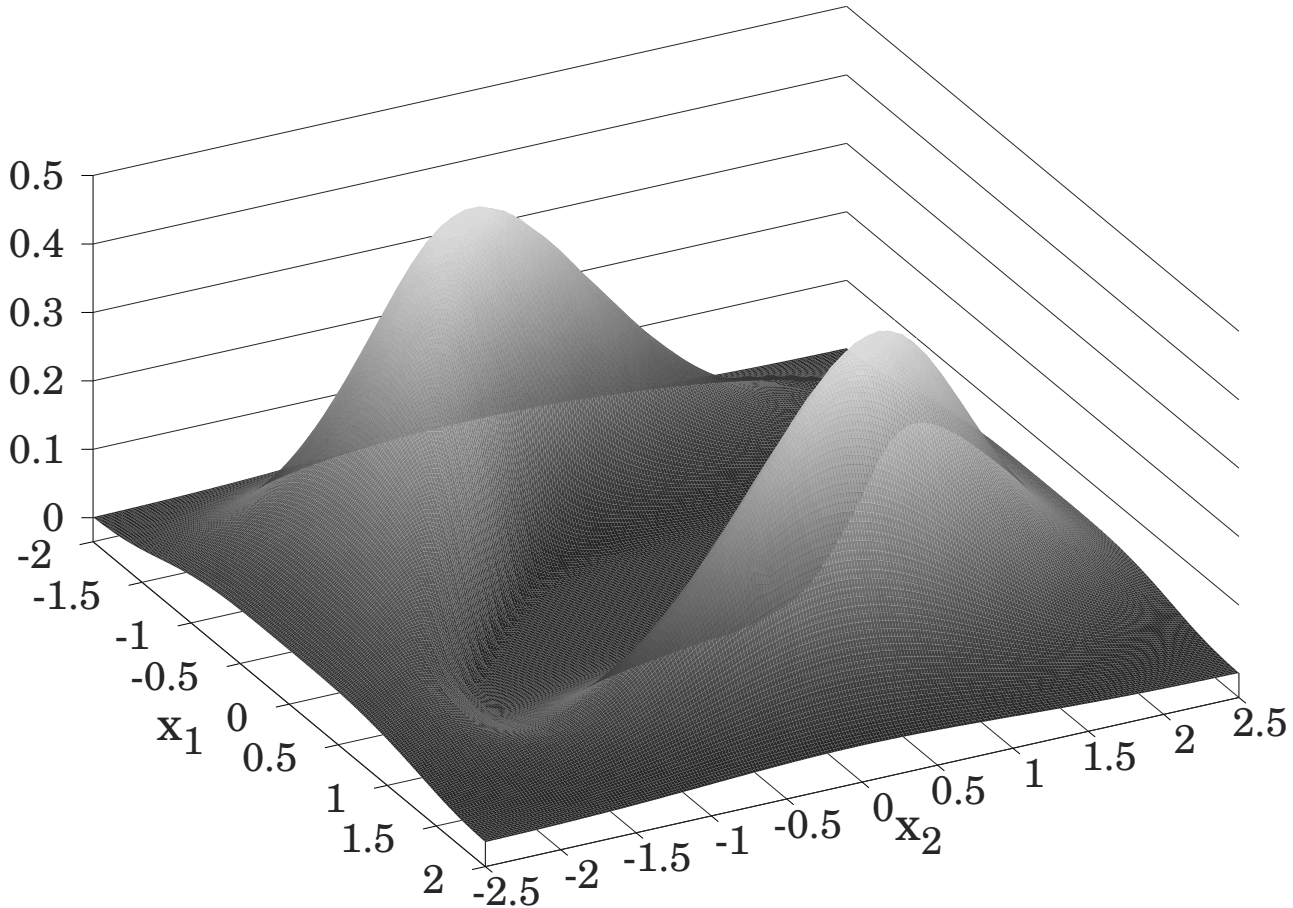
$$\begin{gathered} V_1(x_0) \stackrel{\text{def}}{=} V(\hat{x}_0) + \langle \hat{p}_0, x_0 - \hat{x}_0 \rangle, \quad V(\hat{x}_0) = -\ln(1 - v(\hat{x}_0)), \\ v_1(x_0) \stackrel{\text{def}}{=} 1 - e^{-V_1(x_0)}, \end{gathered} \tag{A.23}$$

and, if also $0 \leqslant v_1(x_0) < 1 - \varepsilon$ and $|v_1(x_0) - v(\hat{x}_0)| < \delta_2$, take $v(x_0) \approx v_1(x_0)$. If at least one of the tested conditions does not hold, put $v(x_0) \approx 1 - \varepsilon$. Moreover, the sought-after control action at the state x_0 is selected from $U^*(x_0, \hat{p}_0, 1)$.

According to [1, Theorem 2.21], $v(\cdot)$ is differentiable almost everywhere in $\mathcal{D}_0$. If it is differentiable at $\hat{x}_0$ and some convex open neighborhood of $\hat{x}_0$ containing x_0 lies in $\mathcal{D}_0$, then (A.23) indeed gives a first-order approximation of $v(x_0)$, because the costate $\hat{p}_0$ (for which $\tilde{p}^* = 1$) represents $\mathrm{D}V(\hat{x}_0)$.

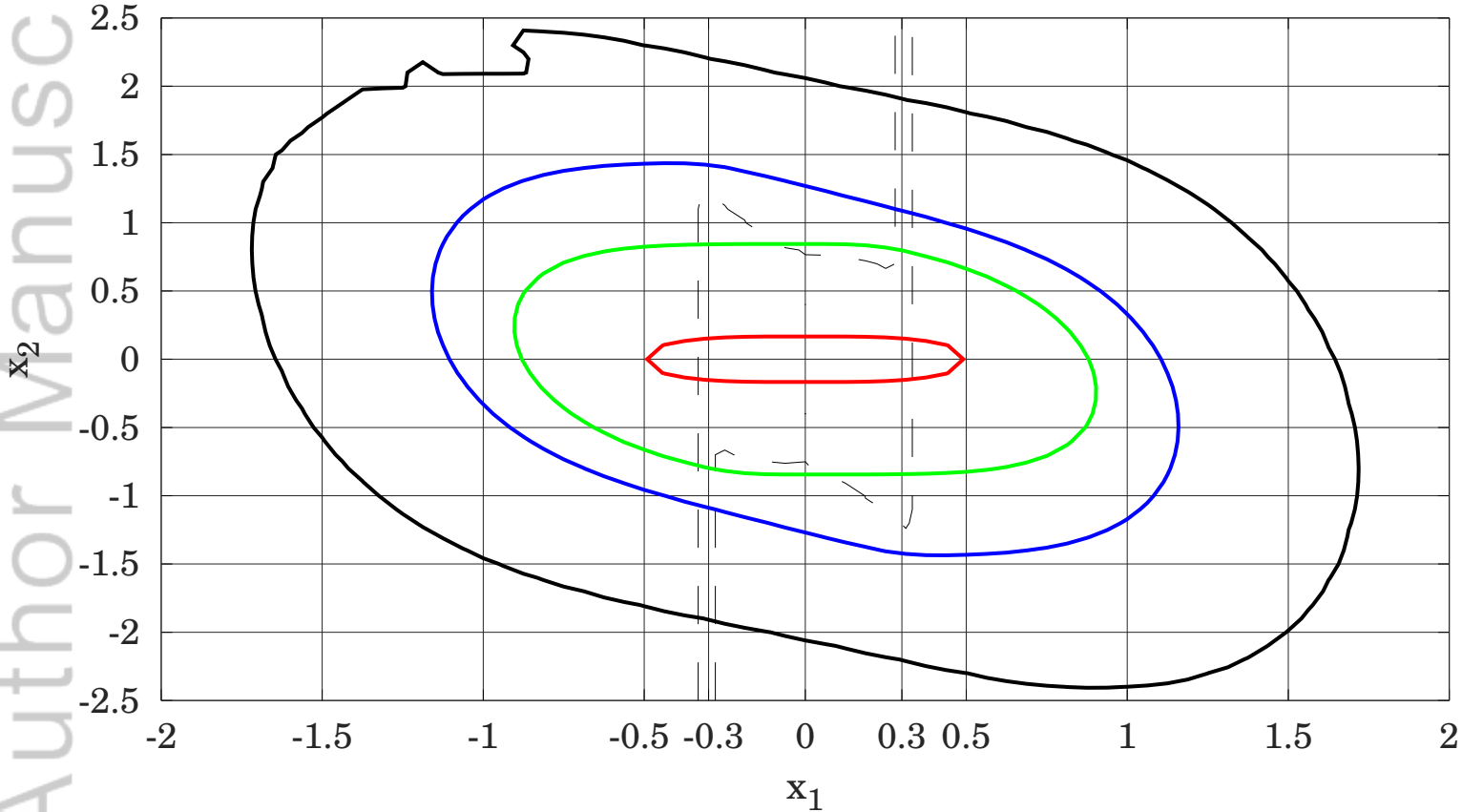


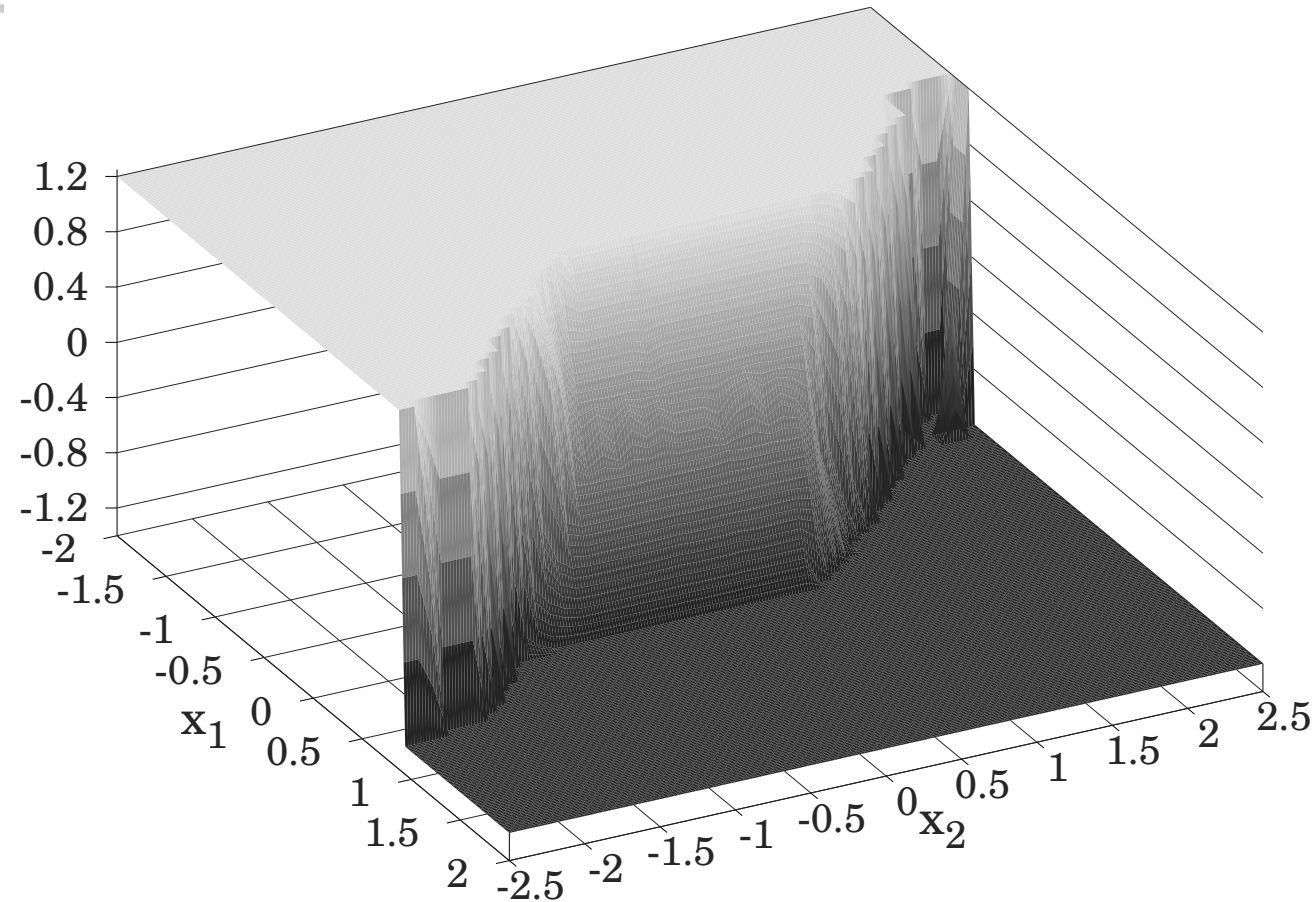


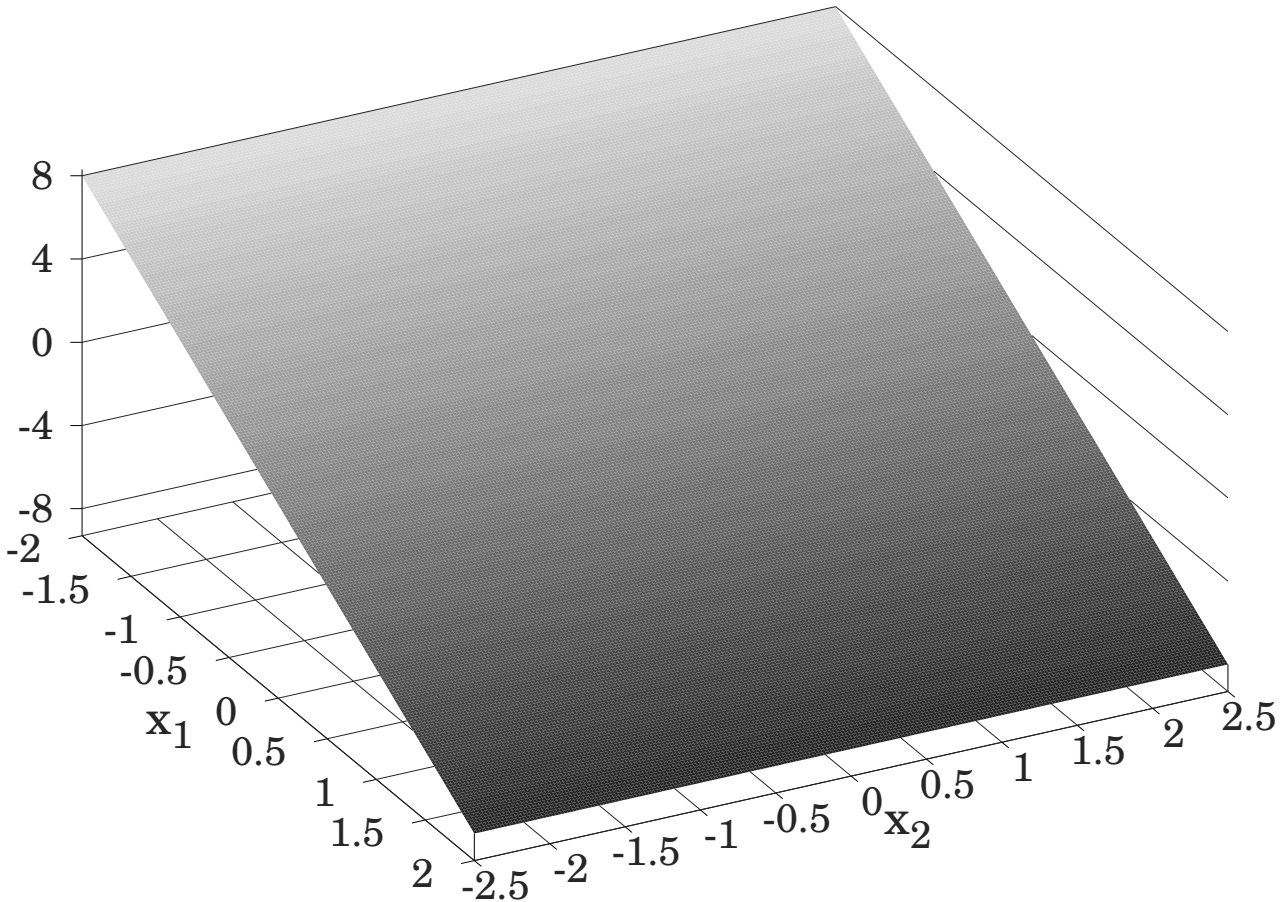


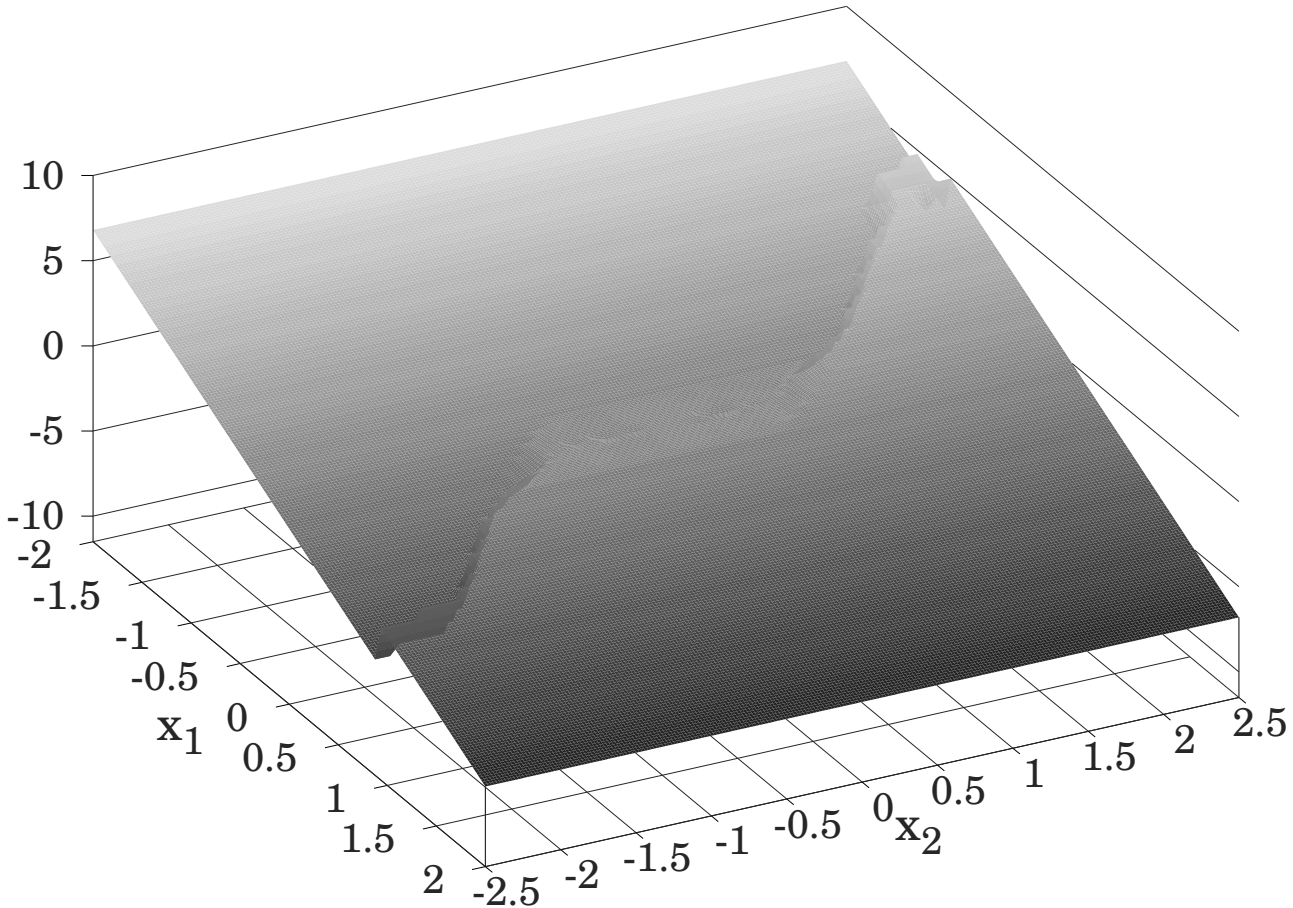
Levels of v: $1 - e^{-c} \approx 0.015$ (red), 0.3 (green), 0.65 (blue), 0.995 (black)

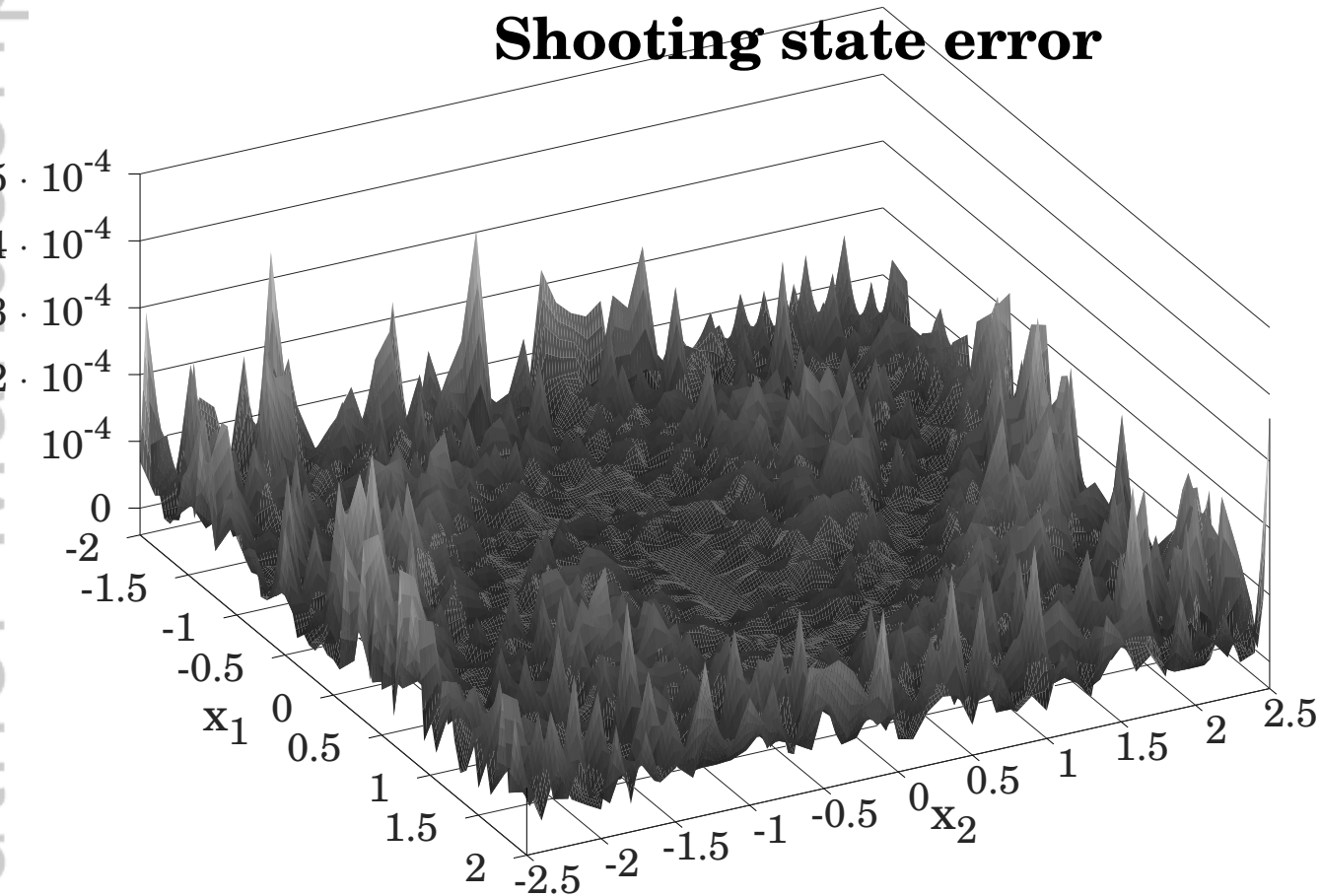
Dashed black line: the boundary of the set $\{x : |u^*(x) - u^*_{loc}(x)| \le 0.13\}$

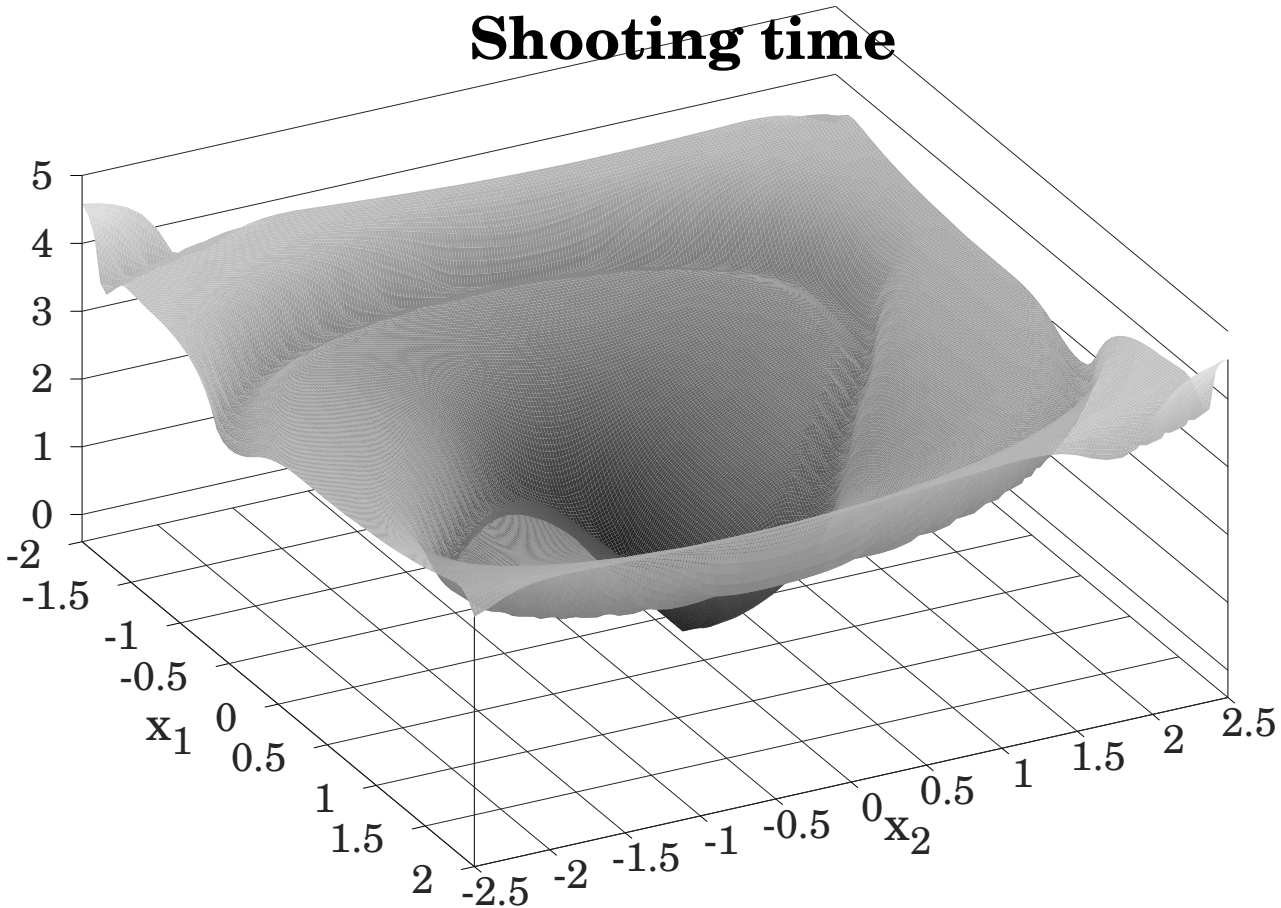


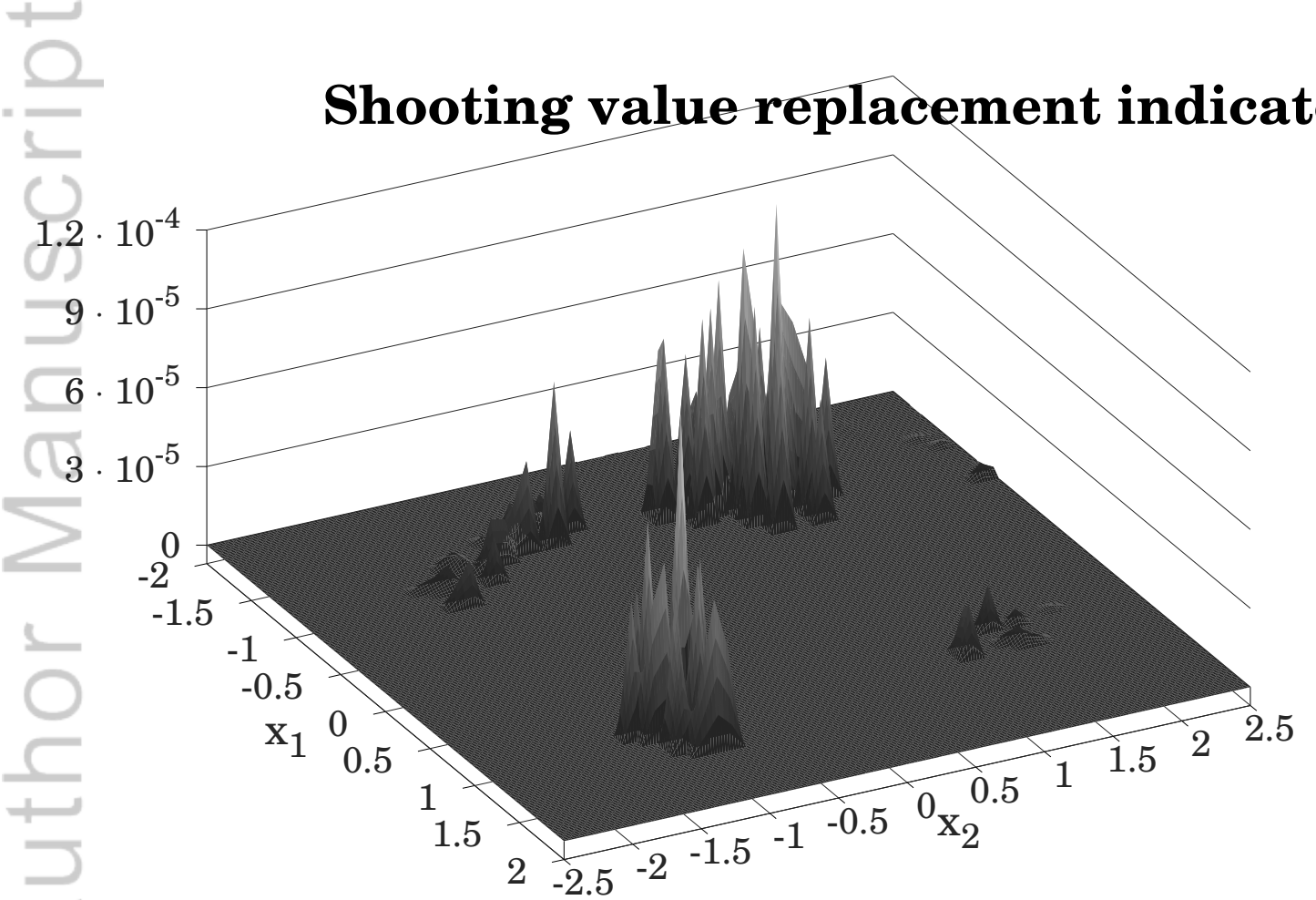


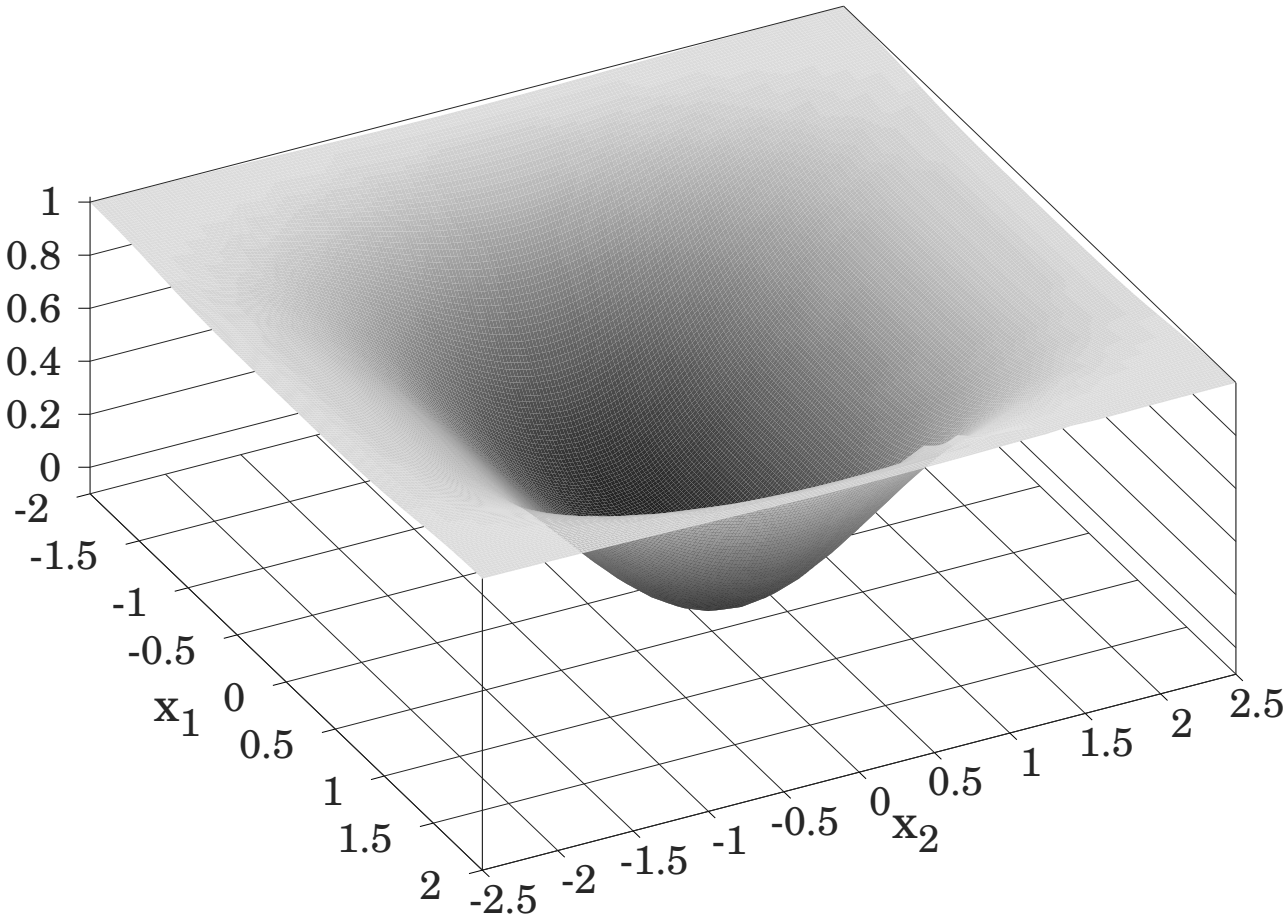


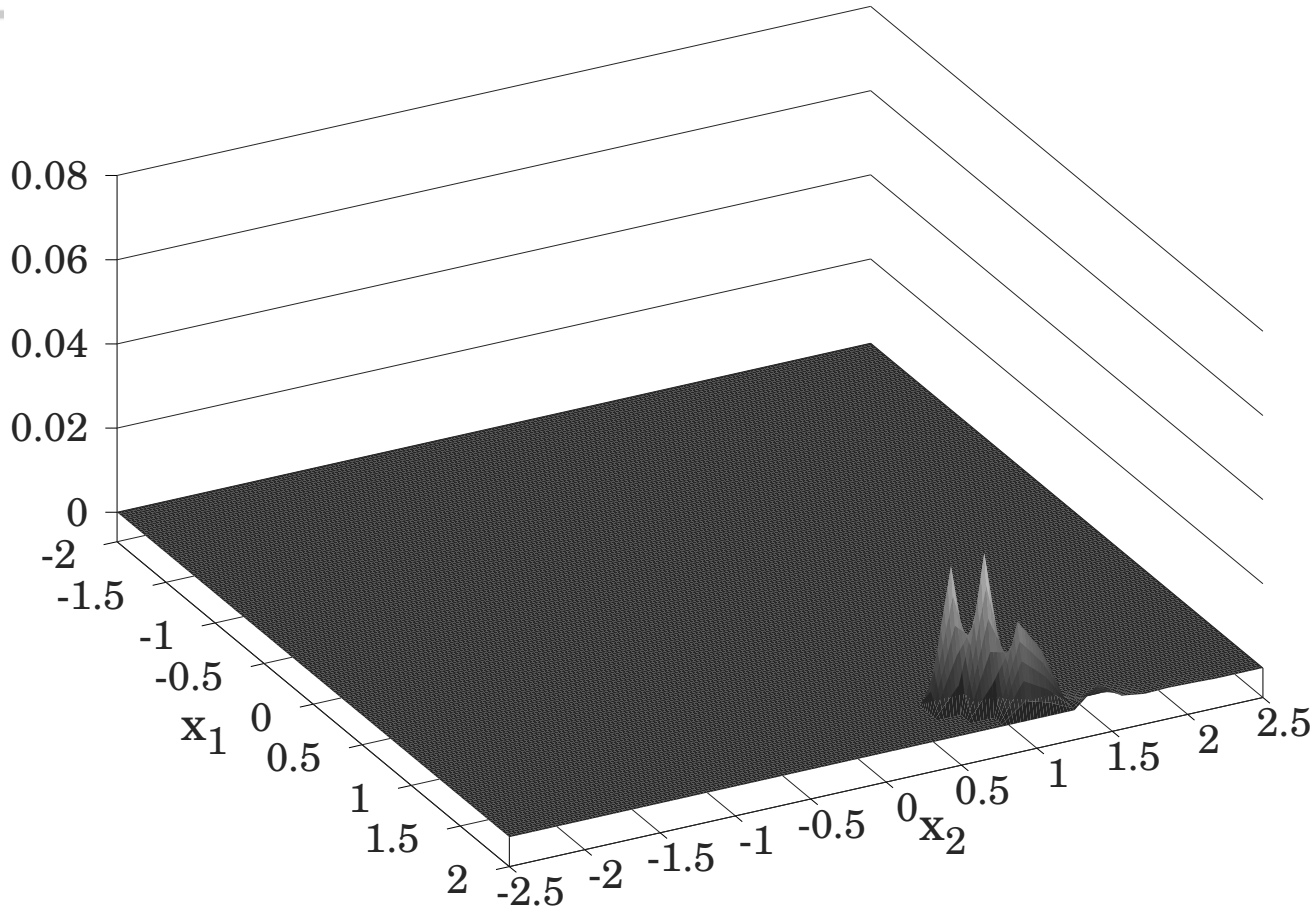


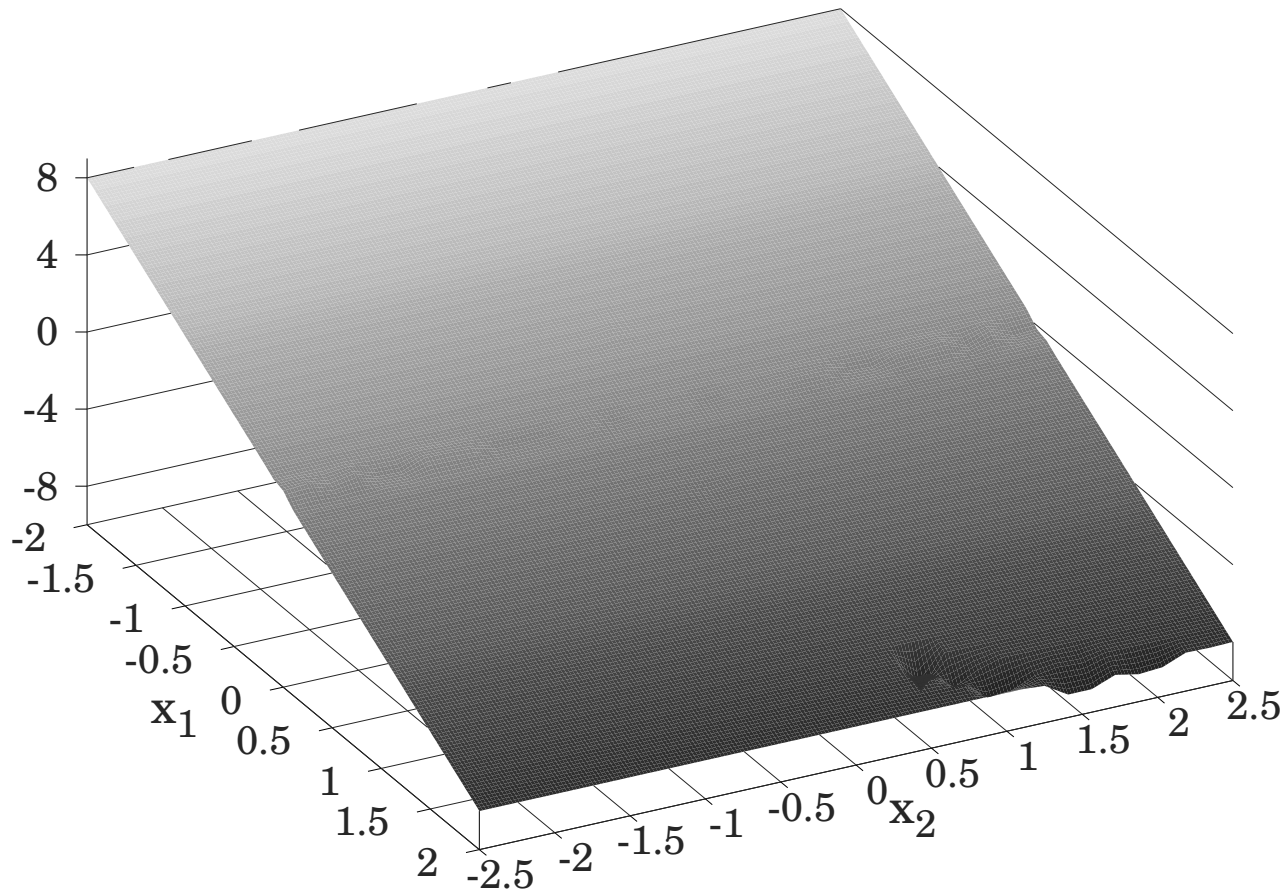


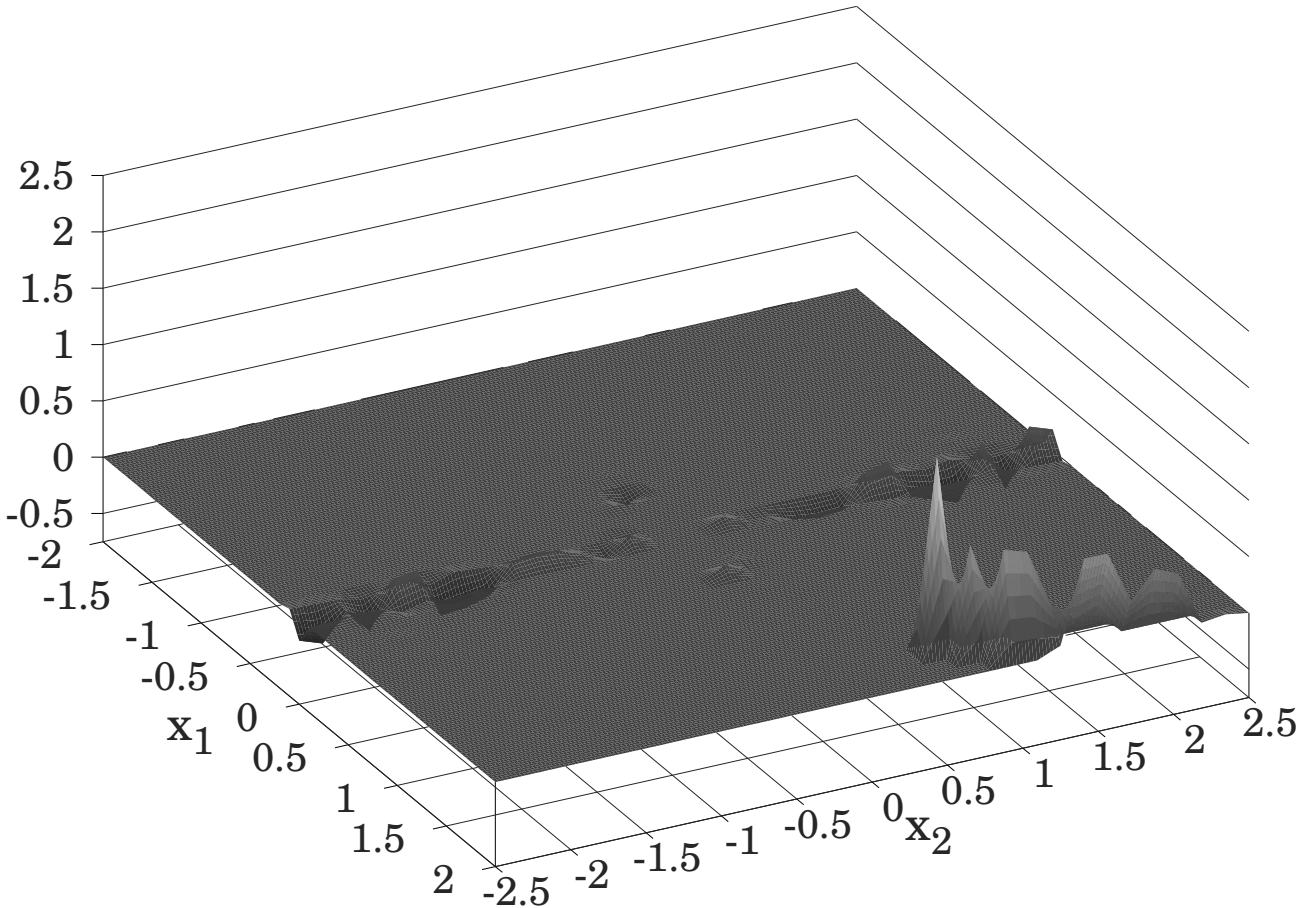


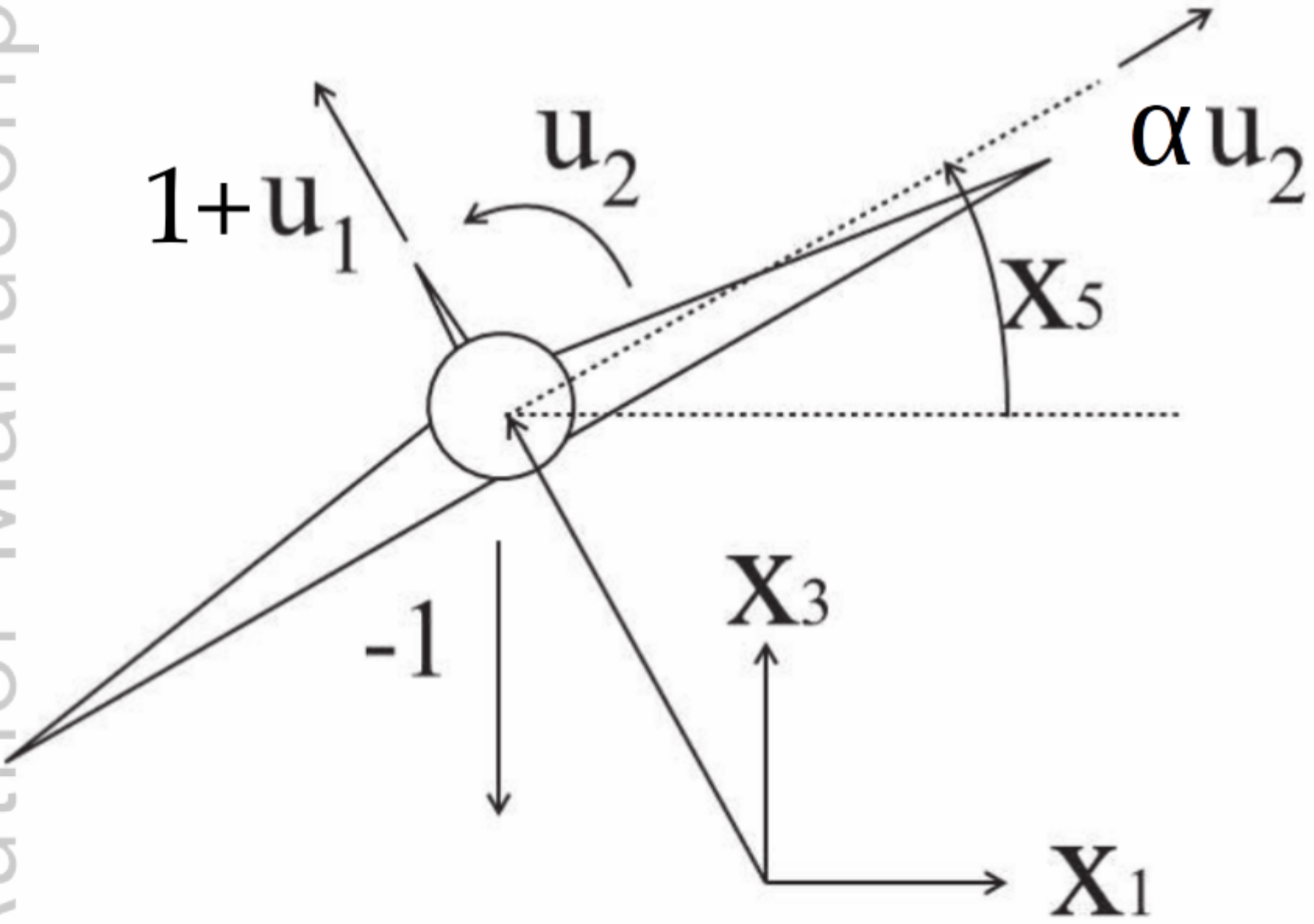


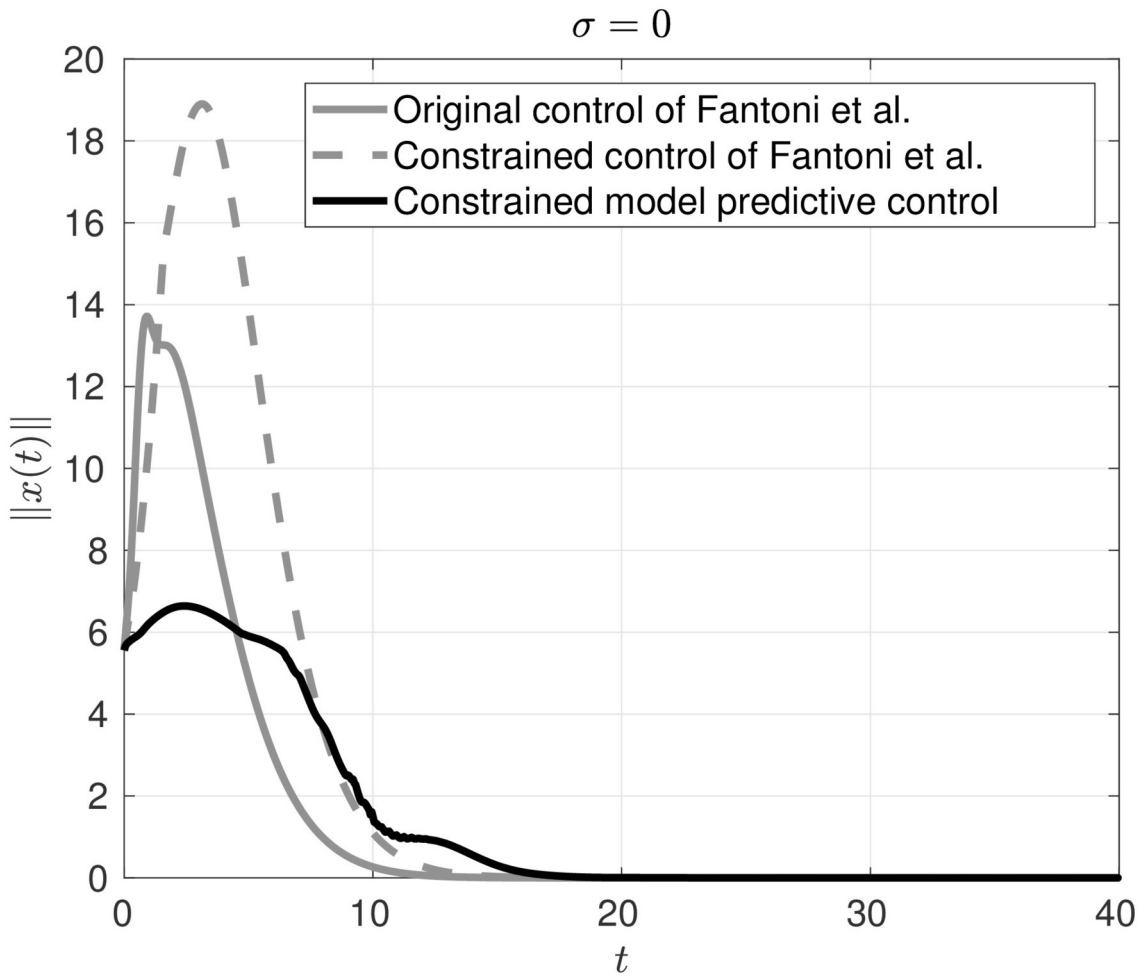




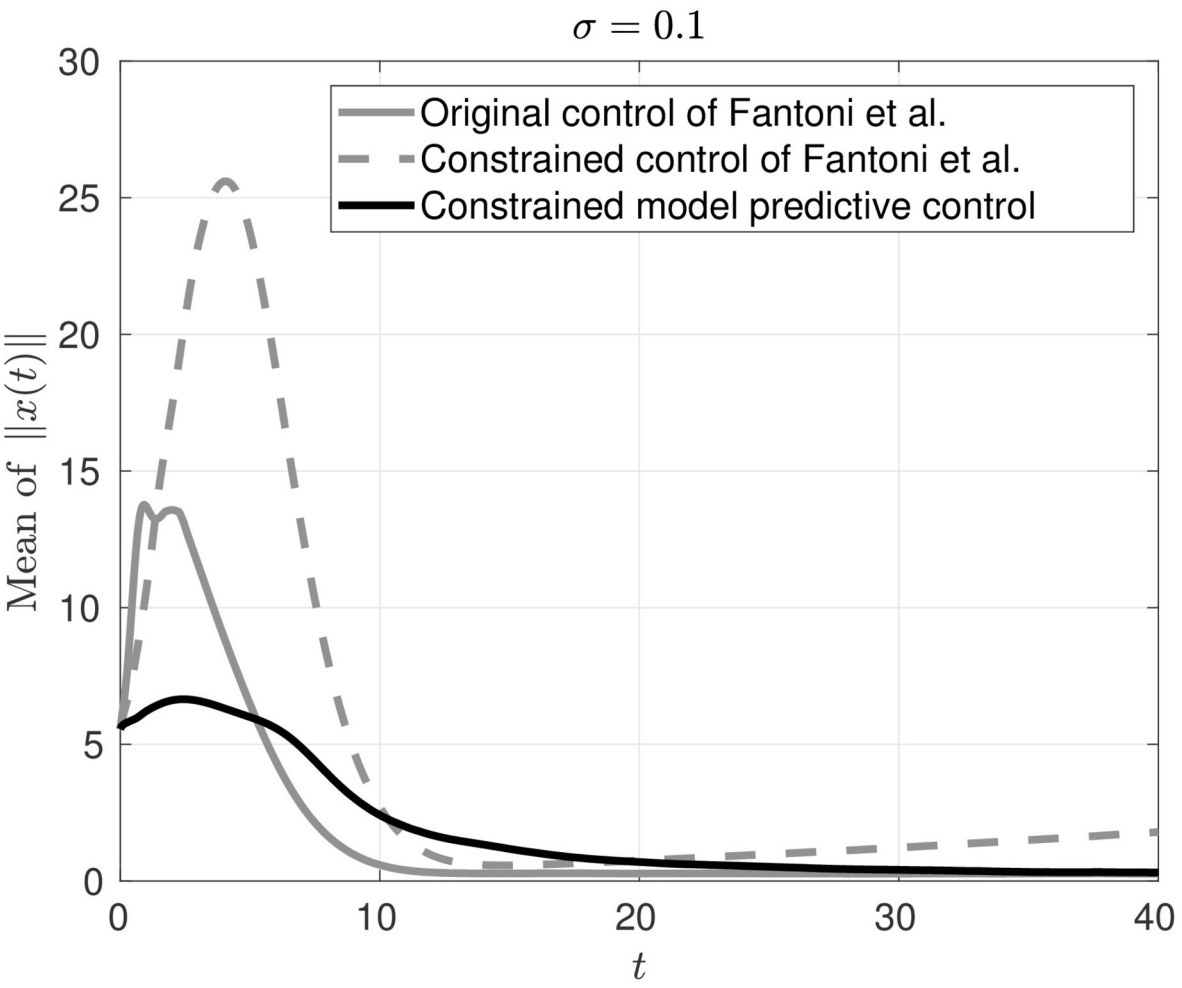




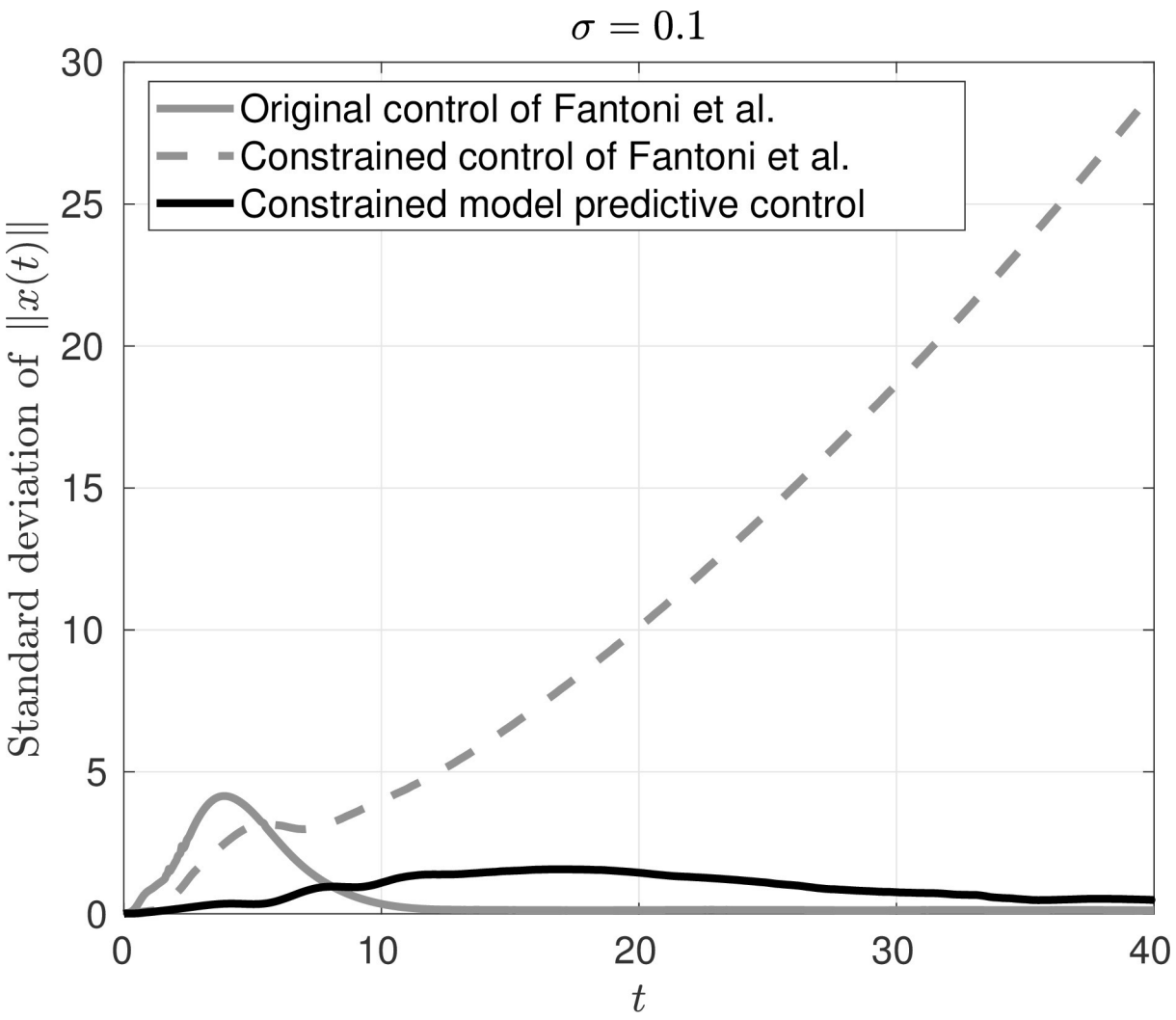




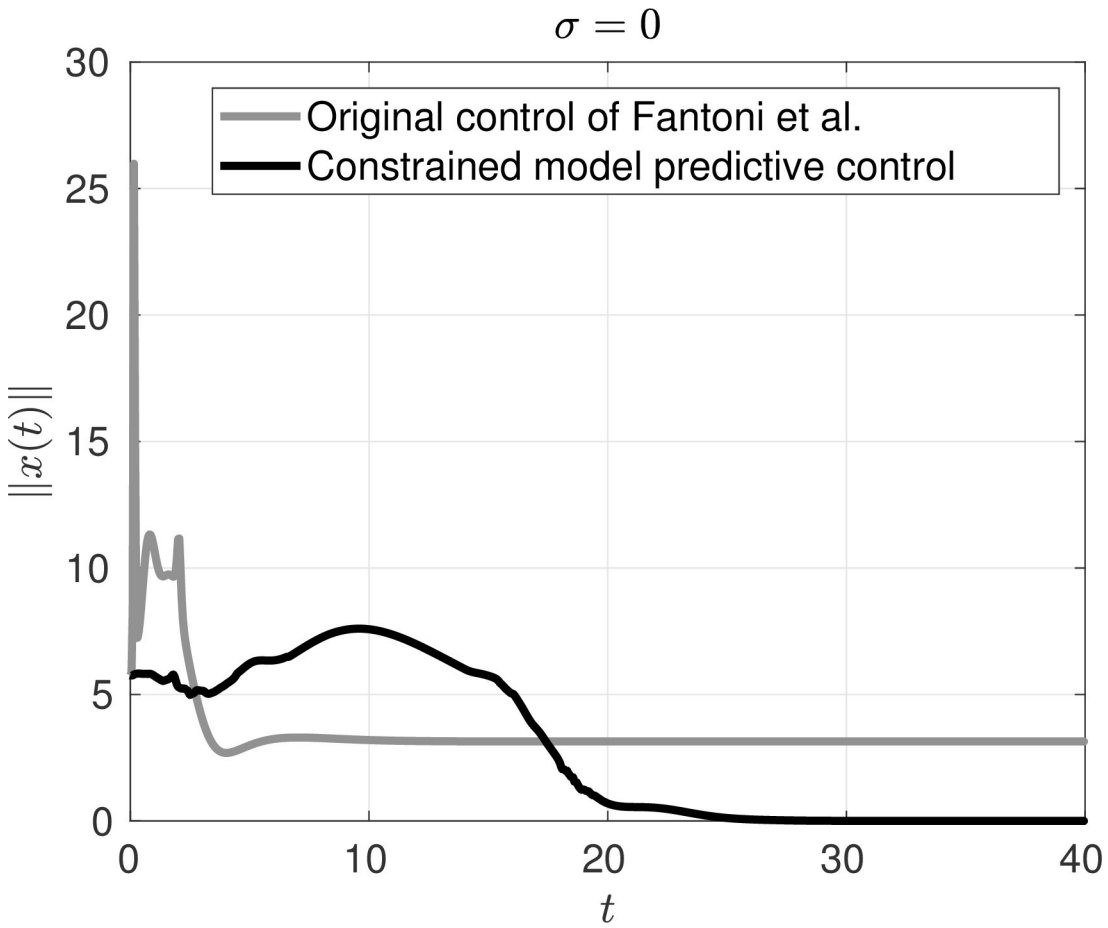
oca_2733_fig_8a.eps



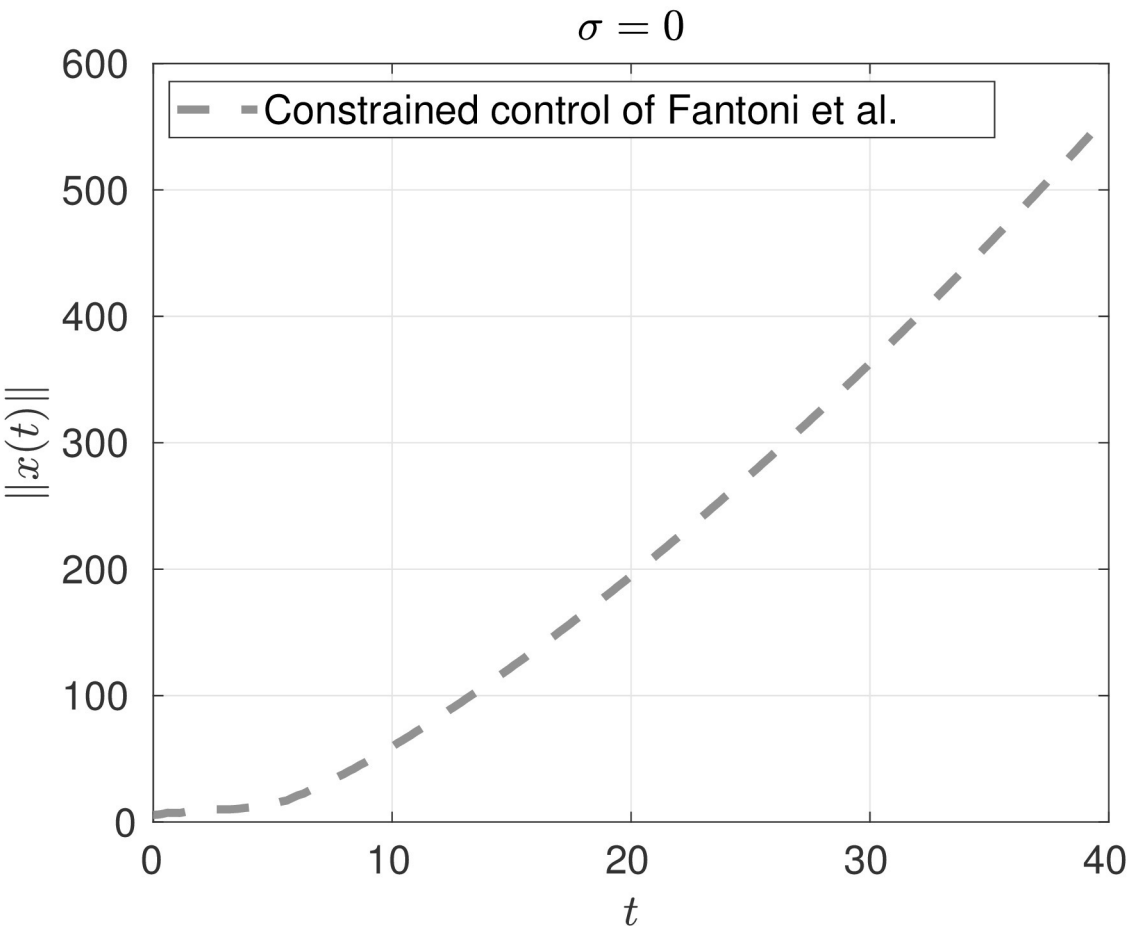
oca_2733_fig_8b.eps



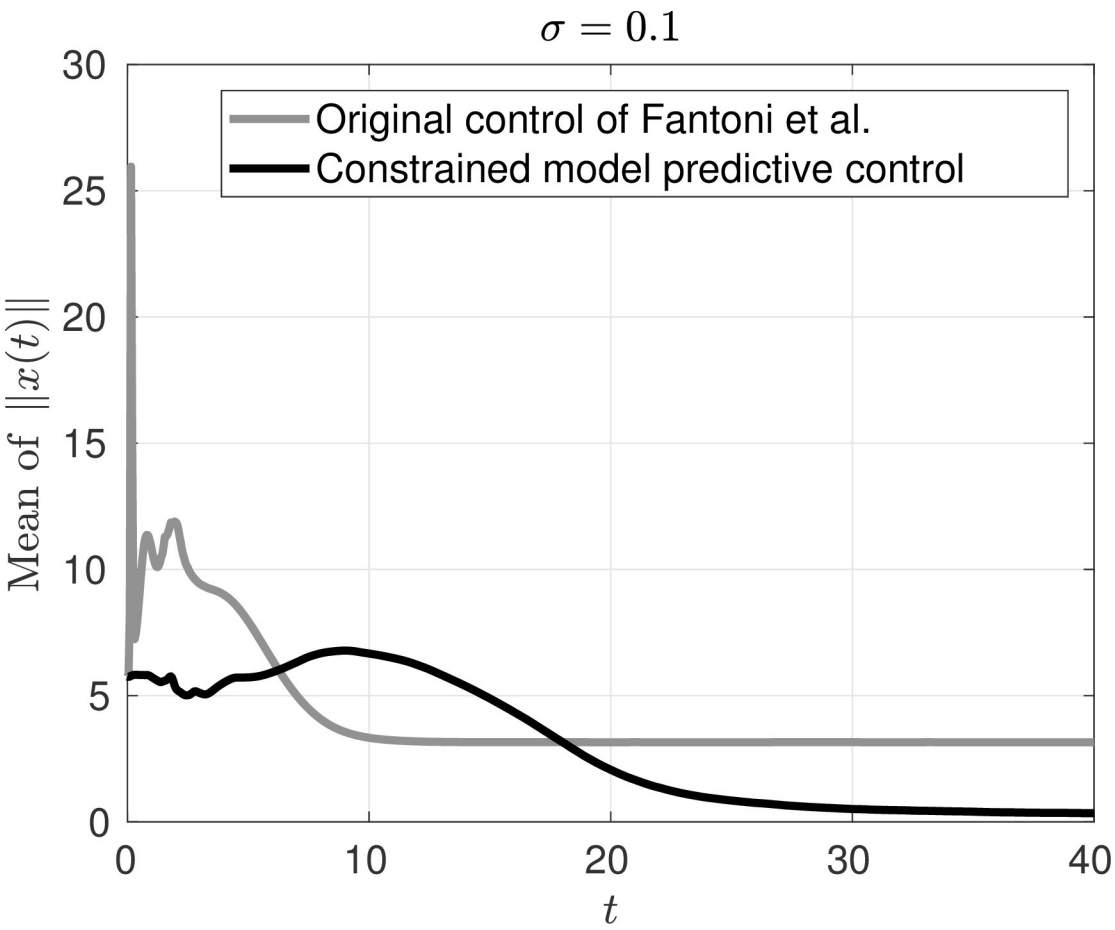
oca_2733_fig_8c.eps



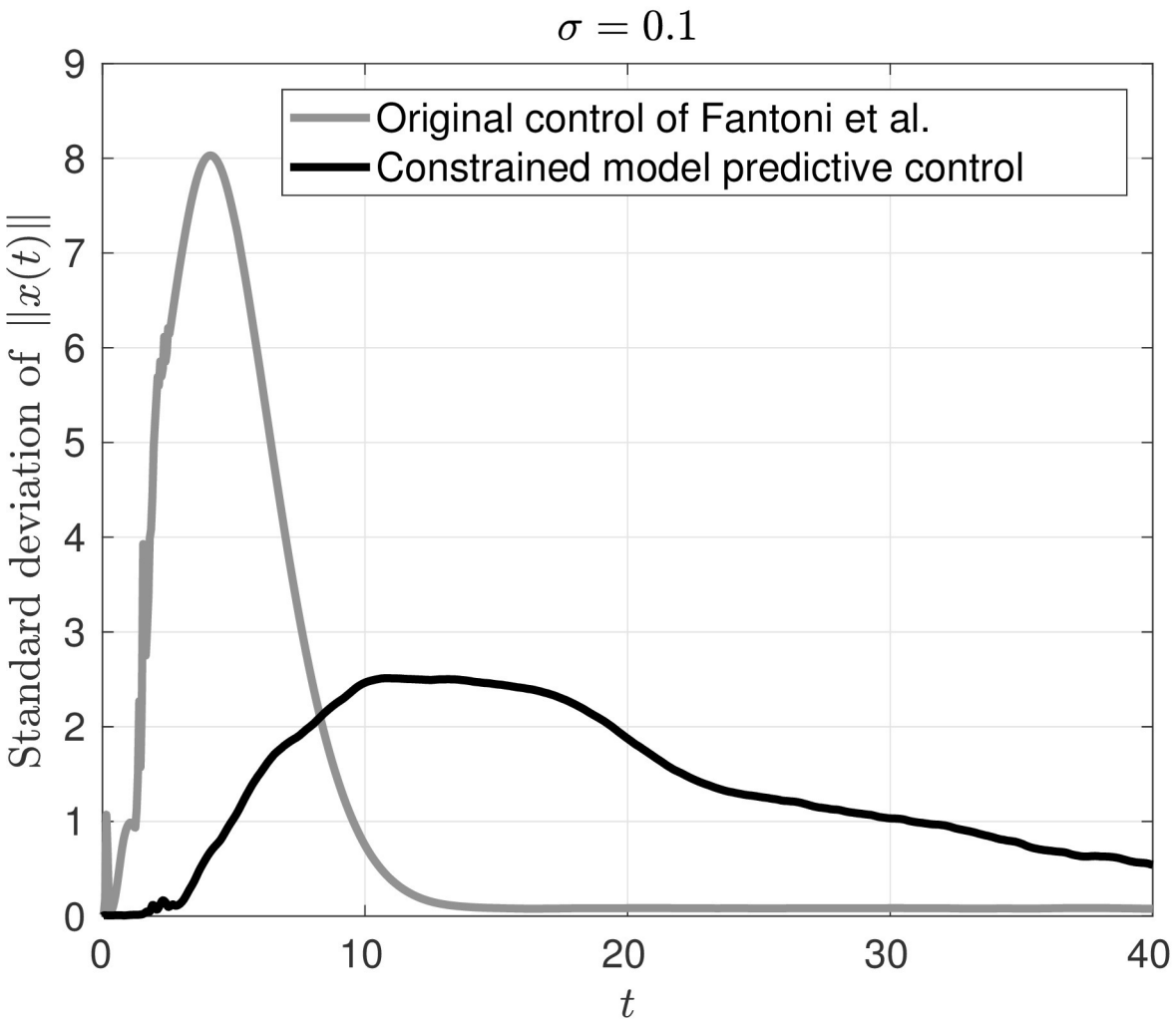
oca_2733_fig_9a.eps



oca_2733_fig_9b.eps



oca_2733_fig_9c.eps



oca_2733_fig_9d.eps