algorithms

MDPI

Article

An Algorithm for Solving Zero-Sum Differential Game Related to the Nonlinear $\mathcal{H}_\infty$ Control Problem

Vladimir Milić *, Josip Kasać * and Marin Lukas

Faculty of Mechanical Engineering and Naval Architecture, University of Zagreb, HR-10000 Zagreb, Croatia
* Correspondence: vladimir.milic@fsb.hr (V.M.); josip.kasac@fsb.hr (J.K.); Tel.: +385-(0)1-6168-505

Abstract: This paper presents an approach for the solution of a zero-sum differential game associated with a nonlinear state-feedback $\mathcal{H}_\infty$ control problem. Instead of using the approximation methods for solving the corresponding Hamilton–Jacobi–Isaacs (HJI) partial differential equation, we propose an algorithm that calculates the explicit inputs to the dynamic system by directly performing minimization with simultaneous maximization of the same objective function. In order to achieve numerical robustness and stability, the proposed algorithm uses: quasi-Newton method, conjugate gradient method, line search method with Wolfe conditions, Adams approximation method for time discretization and complex-step calculation of derivatives. The algorithm is evaluated in computer simulations on examples of first- and second-order nonlinear systems with analytical solutions of $\mathcal{H}_\infty$ control problem.

Keywords: $\mathcal{H}_\infty$ control; zero-sum differential game; conjugate gradient method; quasi-Newton method; complex-step method

check for updates

Citation: Milić, V.; Kasać, J.; Lukas, M. An Algorithm for Solving Zero-Sum Differential Game Related to the Nonlinear $\mathcal{H}_\infty$ Control Problem. *Algorithms* **2023**, *16*, 48. https://doi.org/10.3390/a16010048

Academic Editor: Frank Werner

Received: 18 December 2022
Revised: 5 January 2023
Accepted: 9 January 2023
Published: 10 January 2023

1. Introduction

Although the theory of nonlinear $\mathcal{H}_\infty$ control [1,2] is well developed and can be considered standardized, developing algorithms for solving this problem that enable practical implementation is a very active area of research. Furthermore, it is well known that the $\mathcal{H}_\infty$ control problem can be formulated as a two-player zero-sum differential game [3,4] with the objective function including a parameter in such a way that the control vector represents the player that minimizes the objective function while the vector of uncertainty represents the player that maximizes the same objective function. All available scientific research is mainly based on two main approaches: the formulation of the problem in the form of linear matrix inequalities (LMI) [5–8] or on the determination of the approximate solution of the associated Hamilton–Jacobi–Isaacs (HJI) equation [9–12], which is in linear case equivalent to the generalized Riccati equation [13].

Furthermore, methods for solving the nonlinear $\mathcal{H}_\infty$ control problem of singular or descriptor systems have also been developed. The state-feedback scheme, impulse controllability and the well-known implicit function theorem for stability analysis for finite-time $\mathcal{H}_\infty$ control problem of descriptor systems subject to actuator saturation are adopted in [14], while in [15] impulse and hybrid controllers are combined, resulting in less conservative stability conditions than in state-feedback control strategy.

In nonlinear programming-based algorithms that have been proposed in the literature, the system dynamics is treated as an equality constraint and is included in the optimization process using the method of Lagrange multipliers. This results in a HJI equation that is very difficult or almost impossible to solve. For this reason, many approximation methods are developed in which actual computational complexity increases with the number of system states which need to be estimated. In [10,12], the reviews of reinforcement adaptive learning and adaptive dynamic programming techniques to solve multiplayer games related to $\mathcal{H}_\infty$ control are given. In [16], an improved iterative or successive approximation methods for solving the HJI is developed. A game theoretic differential dynamic programming based algorithm for calculating both minimizing and maximizing inputs based on Taylor series expansion of the associated HJI equation around a nominal trajectory is proposed in [11]. Using a critic neural network with time-varying activation functions, the HJI equation is approximately solved in [9]. An event-triggering function for two-player zero-sum games in the presence of control constraints is designed in [17]. Furthermore, in [13], an iterative procedure to find the stabilizing solution of a set of Riccati equations related to discrete-time $\mathcal{H}_\infty$ control problems for periodic systems is addressed. The randomized algorithm based on the Ray-Shooting Method for $\mathcal{H}_\infty$ optimal synthesis of proportional–integral–derivative (PID) controller for linear systems is proposed in [18].

In the case of the methods and algorithms mentioned above, the application of LMI and solving Riccati equation requires the linearization, and in this case the optimality of solution cannot be guaranteed in all operating states of nonlinear system dynamics. On the other hand, solving the HJI equation can be very complex and therefore difficult to apply in real control tasks.

In this paper, the nonlinear state-feedback $\mathcal{H}_\infty$ control problem is formulated as zero-sum differential game, and we approach its solution without the LMI formalism or the need to approximate the solution of the HJI equation. The main idea of the presented approach is in the application of the conjugate gradient method, where the first-order derivatives are calculated by matrix relations backwards in time, which gives a direct numerical calculation of the control and uncertainty variables that explicitly depend on the system states.

In contrast to the approaches proposed in [11,16], which in the case of including a more complex nonlinearity of the dynamic system in the objective function result in a HJI equation so complex that it is practically impossible to solve, in our approach the nonlinear system dynamics is not included in the objective function as an equality constraint, but the state variables, control law and uncertainty are coupled by recursive matrix relations and chain rule for ordered derivatives, which are used to calculate the objective function gradients that appear in conjugate gradient method. Furthermore, in contrast to approaches presented in [10,12], which can be computationally expensive since the tuning of neural network weights is based on a method of weighted residuals that includes calculation of Lebesgue integrals and estimation of state variables, in our approach, the computational complexity does not depend on the dimension of the state space since procedure proposed for gradient calculation has a backward in time structure similar to the back propagation through time algorithm.

Bedsides the conjugate gradient method, which is used for computation of saddle point of zero-sum differential game, the quasi-Newton method for $\mathcal{L}_2$-gain minimization, line search method with Wolfe conditions, Adams approximation method for time discretization and the complex-step method for calculation of derivatives are also systematically integrated into an efficient mathematical tool in order to achieve numerical robustness and stability.

The rest of the paper is organized as follows. In Section 2, the preliminaries of the nonlinear state-feedback $\mathcal{H}_\infty$ control from the concept of dissipativity point of view and also from the zero-sum differential games point of view are presented. Although theories of these concepts are well known, providing basic terms is necessary to understand the contributions contained in the following sections. A complete framework of the derivation of the algorithmic procedure that optimizes the $\mathcal{L}_2$-gain of nonlinear dynamic systems without solving the HJI partial differential equation is given in Section 3. In Section 4, the proposed algorithmic procedure is evaluated on examples of nonlinear systems for which the input variables can be determined exactly by solving the HJI equation. Finally, Section 5 concludes the paper.

Notation

The notation used is fairly standard. Matrices and vectors are represented in bold upper and bold lower case, respectively. Scalars are represented in italic lower case. **I** is an identity matrix and $\mathbf{0}$ is a null matrix. The dimensions of the matrices and vectors can generally be determined trivially by the context. The symbol ∇ stands for the gradient and is considered as a row vector. The symbol $^{\mathrm{T}}$ denotes transposition.

The $\mathrm{vec}(\cdot)$ is a operator that stacks the columns of the matrix one underneath the other. The Kronecker product of the two matrices $\mathbf{A}$ $(m \times n)$ and $\mathbf{B}$ $(p \times q)$, denoted by $\mathbf{A} \otimes \mathbf{B}$ is a $mp \times nq$ matrix defined by $\mathbf{A} \otimes \mathbf{B} = (a_{ij}\mathbf{B})_{ij}$. The definitions of matrix differentials calculus and the algebras related to Kronecker products can be found in [19,20].

The Euclidean norm of the vector is defined as $\|\mathbf{x}\| = \sqrt{\mathbf{x}^{\mathrm{T}}\,\mathbf{x}}$. $\mathcal{L}_2(I, \mathbb{R}^n)$ stands for the standard Lebesgue spaces of vector valued square-integrable and essentially bounded functions mapping an interval $I \subset \mathbb{R}$ to $\mathbb{R}^n$. This space is equipped with an $\mathcal{L}_2$ norm defined by $\|\cdot\|_{\mathcal{L}_2} = \sqrt{\int_{t_0}^{t_f} \|\cdot\|^2 dt}$. We avoid explicitly showing the dependence of the variables from the time when not needed.

2. Preliminaries and Problem Formulation

In this section, we give a review of the basic terms that include nonlinear state-feedback $\mathcal{H}_\infty$ control and related differential game theory, and this is mostly based on classic references from this field [1–4,21,22].

Consider the causal input-affine nonlinear system in the state space defined on a some manifold $\mathcal{X} \subseteq \mathbb{R}^n$ in the following form

$$\Sigma : \quad \begin{cases} \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{G}_1(\mathbf{x})\mathbf{u} + \mathbf{G}_2(\mathbf{x})\mathbf{d}, \quad \mathbf{x}(t_0) = \mathbf{x}_0, \\ \mathbf{y} = \mathbf{x}, \\ \mathbf{z} = \mathbf{h}(\mathbf{x}) + \mathbf{L}(\mathbf{x})\mathbf{u}, \end{cases} \tag{1}$$

where $\mathbf{x} \in \mathcal{X}$ is the state vector, $\mathbf{u} \in \mathcal{U} \subseteq \mathbb{R}^m$ is the control vector, $\mathbf{d}$ is the vector representing internal/external uncertainty belonging to the set $\mathcal{D} \subset \mathcal{L}_2\Big(\big[t_0,\ t_f\big],\ \mathbb{R}^s\Big)$. The output vector $\mathbf{y} \in \mathbb{R}^n$ contains all directly measured states of system Σ. The vector $\mathbf{z} \in \mathbb{R}^q$ is the performance variable. Furthermore, the functions $\mathbf{f} : \mathcal{X} \to \mathcal{X}$, $\mathbf{G}_1 : \mathcal{X} \to \mathcal{M}^{n\times m}(\mathcal{X})$, $\mathbf{G}_2 : \mathcal{X} \to \mathcal{M}^{n\times s}(\mathcal{X})$, $\mathbf{h} : \mathcal{X} \to \mathbb{R}^q$, $\mathbf{L} : \mathcal{X} \to \mathcal{M}^{q\times m}(\mathcal{X})$ are real C^1-functions of $\mathbf{x}$.

The following assumptions are employed:

Assumption 1. $\mathbf{x} = \mathbf{0}$ *is a unique equilibrium point, with* $\mathbf{u} = \mathbf{0}$ *and* $\mathbf{d} = \mathbf{0}$*, of system* Σ *and for simplicity* $\mathbf{f}(\mathbf{0}) = \mathbf{0}$, $\mathbf{h}(\mathbf{0}) = \mathbf{0}$.

Assumption 2. *The vector* $\mathbf{h}(\mathbf{x})$ *and matrix* $\mathbf{L}(\mathbf{x})$ *are such that* $\mathbf{h}(\mathbf{x})^{\mathrm{T}}\,\mathbf{L}(\mathbf{x}) = \mathbf{0}$ *i* $\mathbf{L}(\mathbf{x})^{\mathrm{T}}\,\mathbf{L}(\mathbf{x}) = \mathbf{I}$ *form all* $\mathbf{x} \in \mathcal{X}$*, which implies*

$$\mathbf{z} = \begin{bmatrix} \mathbf{h}(\mathbf{x}) \\ \mathbf{u} \end{bmatrix} \Longrightarrow \|\mathbf{z}\|^2 = \|\mathbf{h}(\mathbf{x})\|^2 + \|\mathbf{u}\|^2. \tag{2}$$

By introducing Assumption 2, the so-called singular problem is avoided. More details on solving the singular problem can be found in [23,24].

Assumption 3. *Initial state vector* $\mathbf{x}_0$ *is a priori known.*

Furthermore, the following definition is introduced:

Definition 1 ($\mathcal{L}_2$-gain)**.** $\mathcal{L}_2$*-gain from* $\mathbf{d}$ *to* $\mathbf{z}$ *of system* Σ *is the supremum of* $\gamma > 0$ *satisfies*

$$\|\mathbf{z}\|^2_{\mathcal{L}_2} \leq \gamma^2\|\mathbf{d}\|^2_{\mathcal{L}_2} + \beta(\mathbf{x}_0), \tag{3}$$

for some bounded C^0*-function* $\beta : U \subset \mathcal{X} \to \mathbb{R}$ *such that* $\beta(\mathbf{0}) = 0$.

In general, the problem of nonlinear $\mathcal{H}_\infty$ control of system Σ where all state variables are available (measurable or can be estimated) can be formulated as follows:

Problem 1. *The problem of optimal nonlinear state-feedback $\mathcal{H}_\infty$ control of system Σ is to determine the control law $\mathbf{u}^* = \boldsymbol{\mu}(\mathbf{x}, t)$ and the "worst case" $\mathbf{d}^* = \boldsymbol{\nu}(\mathbf{x}, t)$ such that $\gamma > 0$ is minimized.*

Assumption 4. *The functions $\boldsymbol{\mu}(\mathbf{x}, t)$ and $\boldsymbol{\nu}(\mathbf{x}, t)$ are $\boldsymbol{\mu} \in C^1(\mathcal{X})$, $\boldsymbol{\nu} \in C^1(\mathcal{X})$.*

If for the system Σ there exists some $\gamma \geq \gamma^*$ that satisfies (3), then a zero-sum differential game can be defined. The optimal value of this game is given by the following expression:

$$J^*(\mathbf{x}_0) = \min_{\mathbf{u}} \max_{\mathbf{d}} \int_{t_0}^{t_f} \left(\|\mathbf{h}(\mathbf{x})\|^2 + \|\mathbf{u}\|^2 - \gamma^2 \|\mathbf{d}\|^2 \right) dt, \tag{4}$$

with dynamic equality constraints (1) on a finite time horizon $t_f > t_0$.

The necessary conditions for the saddle point of the zero-sum differential game (4) follow from the minimum and maximum principles, and are of the form

$$\mathbf{u}^*(\mathbf{x}, t) = -\frac{1}{2}\mathbf{G}_1^{\mathrm{T}}(\mathbf{x})\nabla_{\mathbf{x}}^{\mathrm{T}} V(\mathbf{x}, t), \tag{5}$$

$$\mathbf{d}^*(\mathbf{x}, t) = \frac{1}{2\gamma^2}\mathbf{G}_2^{\mathrm{T}}(\mathbf{x})\nabla_{\mathbf{x}}^{\mathrm{T}} V(\mathbf{x}, t), \tag{6}$$

where V is a smooth positive semidefinite solution of the HJI partial differential equation

$$\begin{aligned} &\nabla_t V(\mathbf{x}, t) + \nabla_{\mathbf{x}} V(\mathbf{x}, t)\mathbf{f}(\mathbf{x}) + \frac{1}{4}\nabla_{\mathbf{x}} V(\mathbf{x}, t)\bigg[\frac{1}{\gamma^2}\mathbf{G}_2(\mathbf{x})\mathbf{G}_2^{\mathrm{T}}(\mathbf{x}) \\ &- \mathbf{G}_1(\mathbf{x})\mathbf{G}_1^{\mathrm{T}}(\mathbf{x})\bigg]\nabla_{\mathbf{x}}^{\mathrm{T}} V(\mathbf{x}, t) + \mathbf{h}^{\mathrm{T}}(\mathbf{x})\mathbf{h}(\mathbf{x}) = 0, \quad V(\mathbf{x}(t_f), t_f) = 0. \end{aligned} \tag{7}$$

3. Synthesis of the Algorithm for the Solution of the Zero-Sum Differential Game

In this section, an approach of determining control and uncertainty variables for optimal $\mathcal{H}_\infty$ control (Problem 1) of nonlinear dynamic system Σ is proposed. The proposed approach does not require solving the HJI Equation (7), but the solution is reduced to the direct numerical determination of the saddle point of the related zero-sum differential game.

Control and uncertainty variables are approximated by functions with a linear dependence on a finite number of constant parameters. To calculate these parameters, an approach based on the integration of the quasi-Newton method, the conjugate gradient method, the Adams method and the complex-step derivative approximation into one algorithm is proposed. The goal is to obtain a control variable that explicitly depends on the state variables and in that form is simple for practical implementation. Additionally, the aim is to achieve a numerical solution that uniformly converges towards the optimal solution by increasing the order of complexity of the approximation, i.e., by increasing the number of parameters.

Since we introduced the Assumption 4, based on Weierstraß's theorem ([25], p. 65) (which refers to polynomial approximation functions) and its generalizations [26–29] (which refer to non-polynomial forms of nonlinear approximation functions) on the uniform approximation of smooth functions, there are constants $p_j^i, r_j^i \in \mathbb{R}$ such that the i-th component of control and uncertainty vector can be written in the following form:

$$\hat{u}_i(\mathbf{x}) = \sum_{j=1}^{n_\theta} p_j^i \theta_j^i(\mathbf{x}), \tag{8}$$

$$\hat{d}_i(\mathbf{x}) = \sum_{j=1}^{n_\psi} r_j^i \psi_j^i(\mathbf{x}), \tag{9}$$

where $\theta_j^i(\mathbf{x}) \in C^1(\mathcal{X}),\ \psi_j^i(\mathbf{x}) \in C^1(\mathcal{X})$ such that $\theta_j^i(\mathbf{0}) = 0,\ \psi_j^i(\mathbf{0}) = 0$. Linear subspaces generated by sets $\{\theta_j^i(\mathbf{x})\}$ i $\{\psi_j^i(\mathbf{x})\}$ are dense in the Sobolev norm $W^{1,\infty}$ [30].

For well chosen functions $\theta_j^i(\mathbf{x})$ and $\psi_j^i(\mathbf{x})$, we have

$$|\hat{u}_i(\mathbf{x}) - u_i(\mathbf{x})| < \varepsilon_{u_i}(\mathbf{x}), \tag{10}$$

$$|\hat{d}_i(\mathbf{x}) - d_i(\mathbf{x})| < \varepsilon_{d_i}(\mathbf{x}), \tag{11}$$

where $\varepsilon_{u_i}(\mathbf{x})$ i $\varepsilon_{d_i}(\mathbf{x})$ are the approximation errors. It follows that $\varepsilon_{u_i}(\mathbf{x}) \to 0$, $\varepsilon_{d_i}(\mathbf{x}) \to 0$ when $n_\theta \to \infty$, $n_\psi \to \infty$, respectively, while for fixed n_θ and n_ψ, approximation errors are bounded by constants on the compact set. It is well known from approximation theory (see for example [31,32]) that it is often possible to determine in advance how many terms of the basis functions expansion should be taken for the required accuracy.

Equations (8) and (9) can be written in the following matrix form:

$$\hat{\mathbf{u}}(\mathbf{x}) = \boldsymbol{\Theta}(\mathbf{x})\,\boldsymbol{\pi}, \tag{12}$$

$$\hat{\mathbf{d}}(\mathbf{x}) = \boldsymbol{\Psi}(\mathbf{x})\,\boldsymbol{\rho}, \tag{13}$$

where

$$\boldsymbol{\Theta}(\mathbf{x}) \equiv \begin{bmatrix} \boldsymbol{\theta}^1(\mathbf{x}) & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\theta}^2(\mathbf{x}) & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \boldsymbol{\theta}^m(\mathbf{x}) \end{bmatrix}, \quad \boldsymbol{\Psi}(\mathbf{x}) \equiv \begin{bmatrix} \boldsymbol{\psi}^1(\mathbf{x}) & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\psi}^2(\mathbf{x}) & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \boldsymbol{\psi}^s(\mathbf{x}) \end{bmatrix},$$

$$\boldsymbol{\theta}^i(\mathbf{x}) \equiv \begin{bmatrix} \theta_1^i(\mathbf{x}) & \theta_2^i(\mathbf{x}) & \cdots & \theta_{n_\theta}^i(\mathbf{x}) \end{bmatrix}, \quad \boldsymbol{\psi}^i(\mathbf{x}) \equiv \begin{bmatrix} \psi_1^i(\mathbf{x}) & \psi_2^i(\mathbf{x}) & \cdots & \psi_{n_\psi}^i(\mathbf{x}) \end{bmatrix}, \tag{14}$$

$$\boldsymbol{\pi} \equiv \begin{bmatrix} \mathbf{p}^1 \\ \mathbf{p}^2 \\ \vdots \\ \mathbf{p}^m \end{bmatrix}, \quad \boldsymbol{\rho} \equiv \begin{bmatrix} \mathbf{r}^1 \\ \mathbf{r}^2 \\ \vdots \\ \mathbf{r}^s \end{bmatrix}, \quad \mathbf{p}^i \equiv \begin{bmatrix} p_1^i \\ p_2^i \\ \vdots \\ p_{n_\theta}^i \end{bmatrix}, \quad \mathbf{r}^i \equiv \begin{bmatrix} r_1^i \\ r_2^i \\ \vdots \\ r_{n_\psi}^i \end{bmatrix}.$$

In this paper, based on the previous considerations, the problem that needs to be solved can be formulated as follows:

Problem 2. *Determine the parameters $\boldsymbol{\pi}$ and $\boldsymbol{\rho}$ such that the $\mathcal{L}_2$-gain of the closed-loop*

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{G}_1(\mathbf{x})\boldsymbol{\Theta}(\mathbf{x})\boldsymbol{\pi} + \mathbf{G}_2(\mathbf{x})\boldsymbol{\Psi}(\mathbf{x})\boldsymbol{\rho}, \tag{15}$$

is minimized. In other words, according to Definition 1, solve the following minimax optimization problem

$$J_{\mu}^{*}(\mathbf{x}_0) = \min_{\boldsymbol{\pi}} \max_{\boldsymbol{\rho}} \left\{ \|\mathbf{h}(\mathbf{x})\|_{\mathcal{L}_2}^2 + \|\boldsymbol{\Theta}(\mathbf{x})\boldsymbol{\pi}\|_{\mathcal{L}_2}^2 - \mu\|\boldsymbol{\Psi}(\mathbf{x})\boldsymbol{\rho}\|_{\mathcal{L}_2}^2 \right\}, \tag{16}$$

which represents a zero-sum differential game, where the minimal $\mathcal{L}_2$-gain is $\gamma^* = \sqrt{\mu}$.

First, to propose an algorithm for minimization of the parameter μ in (16), the first-order derivative of the value of the differential game with respect to μ is needed. Since $J_{\mu}^{*}(\mathbf{x}_0)$ is non-differentiable because it is defined with the min–max operator, the sub-differential is employed. It can easily be shown that sub-differential of $J_{\mu}^{*}(\mathbf{x}_0)$ with respect to μ is

$$\partial J_{\mu}^{*} = -\|\boldsymbol{\Psi}(\mathbf{x})\boldsymbol{\rho}\|_{\mathcal{L}_2}^2. \tag{17}$$

Next, due to the inaccurate calculation of the previously mentioned derivative, the quasi-Newton method is considered, for which the superlinear convergence and numerical robustness have been proven (see for example [33–36]). The quasi-Newton k-th iteration is defined by

$$\begin{aligned} \mu_{k+1} &= \mu_k - \frac{\|\mathbf{h}(\mathbf{x}_k)\|_{\mathcal{L}_2}^2 + \|\boldsymbol{\Theta}(\mathbf{x}_k)\boldsymbol{\pi}_k\|_{\mathcal{L}_2}^2 - \mu_k\|\boldsymbol{\Psi}(\mathbf{x}_k)\boldsymbol{\rho}_k\|_{\mathcal{L}_2}^2}{\partial J_{\mu_k}^{*}} \\ &= \frac{\|\mathbf{h}(\mathbf{x}_k)\|_{\mathcal{L}_2}^2 + \|\boldsymbol{\Theta}(\mathbf{x}_k)\boldsymbol{\pi}_k\|_{\mathcal{L}_2}^2}{\|\boldsymbol{\Psi}(\mathbf{x}_k)\boldsymbol{\rho}_k\|_{\mathcal{L}_2}^2}. \end{aligned} \tag{18}$$

To minimize the $\mathcal{L}_2$-gain of the system (15), we propose the following quasi-Newton algorithm.

It is important to note that in the second step of Algorithm 1, unlike other known methods of nonlinear optimization, the dynamics of the system (15) is not included in the objective function.

Algorithm 1 quasi-Newton method for $\mathcal{L}_2$-gain minimization.

Require: $\mu_0 \in \mathbb{R}_+$, $\varepsilon \in \mathbb{R}_+$.
Ensure: μ^*.

1: Set $k \leftarrow 0$.
2: Solve zero-sum differential game

$$J_{\mu_k}^{*} = \min_{\boldsymbol{\pi}} \max_{\boldsymbol{\rho}} \left\{ \|\mathbf{h}(\mathbf{x}_k)\|_{\mathcal{L}_2}^2 + \|\boldsymbol{\Theta}(\mathbf{x}_k)\boldsymbol{\pi}_k\|_{\mathcal{L}_2}^2 - \mu_k\|\boldsymbol{\Psi}(\mathbf{x}_k)\boldsymbol{\rho}_k\|_{\mathcal{L}_2}^2 \right\}, \tag{19}$$

where $\mathbf{x}_k$, $\boldsymbol{\pi}_k$ and $\boldsymbol{\rho}_k$ are coupled by (15).
3: Calculate

$$\mu_{k+1} = \frac{\|\mathbf{h}(\mathbf{x}_k)\|_{\mathcal{L}_2}^2 + \|\boldsymbol{\Theta}(\mathbf{x}_k)\boldsymbol{\pi}_k\|_{\mathcal{L}_2}^2}{\|\boldsymbol{\Psi}(\mathbf{x}_k)\boldsymbol{\rho}_k\|_{\mathcal{L}_2}^2}. \tag{20}$$

4: If $|\mu_{k+1} - \mu_k| \le \varepsilon$ then terminate. Otherwise, set $k \leftarrow k+1$ and return to step 2.

To solve the subproblem in the second step of Algorithm 1, we use the conjugate gradient method as described in the following algorithm.

Note that, in Algorithm 2, the maximization of the objective function is obtained by changing the sign in front of the gradient with respect to $\boldsymbol{\rho}$ in (23). Additionally, it should be noted that the strategy for solving Problem 2 proposed by Algorithms 1 and 2 requires appropriate initialization, i.e., appropriate selection of approximation functions and initial parameters $\boldsymbol{\pi}_0$, $\boldsymbol{\rho}_0$ and μ_0. If they are inadequate, it cannot be guaranteed that the control law is for the "worst case" of uncertainty. In the general, it is very difficult to guarantee whether a solution for the "worst case" of uncertainty is obtained. However, the conditions that would give recommendations for the initialization of the parameters of the proposed algorithms could be derived by applying the concept of inverse min max optimal control. Solving the problem of inverse min–max optimal control goes beyond the scope of the research presented in this paper and will be part of future research.

Algorithm 2 Conjugate gradient method for saddle point computation.

Require: $\mathbf{x}_0 \in \mathbb{R}^n$, $\boldsymbol{\xi}_0 \equiv \begin{bmatrix} \boldsymbol{\pi}_0^{\mathrm{T}} & \boldsymbol{\rho}_0^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^{n_\xi}$, $\mu_k, \beta_0 \in \mathbb{R}_+$, $\eta_0 \in \mathbb{R}_+$, $\epsilon \in \mathbb{R}_+$.
Ensure: $\boldsymbol{\pi}^*, \boldsymbol{\rho}^*$.
1: Set $j \leftarrow 1$.
2: Perform a conjugate gradient descent/ascent algorithm in the following form

$$\boldsymbol{\xi}_j = \boldsymbol{\xi}_{j-1} + \eta_{j-1}\mathbf{s}_{j-1}, \tag{21}$$

$$\mathbf{s}_j = -\mathcal{J}(\boldsymbol{\xi}_j) + \beta_{j-1}\mathbf{s}_{j-1}, \tag{22}$$

where $\mathbf{s}$ is the search direction vector and

$$\mathcal{J}(\boldsymbol{\xi}_j) = \begin{bmatrix} \nabla_{\pi} J - \nabla_{\rho} J \end{bmatrix}^{\mathrm{T}}, \tag{23}$$

$$J = \|\mathbf{h}(\mathbf{x}_j)\|^2_{\mathcal{L}_2} + \|\boldsymbol{\Theta}(\mathbf{x}_j)\boldsymbol{\pi}_j\|^2_{\mathcal{L}_2} - \mu_k\|\boldsymbol{\Psi}(\mathbf{x}_j)\boldsymbol{\rho}_j\|^2_{\mathcal{L}_2}. \tag{24}$$

3: Determine $\eta_j > 0$ by applying the line search strategy that satisfies Wolfe's conditions (see Algorithm 3).
4: Determine $\beta_j > 0$ by applying Dai-Yuan method [37].
5: If $\|\mathcal{J}(\boldsymbol{\xi}_j)\|_\infty \leq \epsilon$ then terminate. Otherwise, set $j \leftarrow j+1$ and return to step 2.

Algorithm 3 Line search method with Wolfe conditions.

Require: $\boldsymbol{\xi}_j$, $\mathbf{s}_j$, $J(\boldsymbol{\xi}_j)$, $\mathcal{J}(\boldsymbol{\xi}_j)$, $0 < c_1 < c_2 < 1$, $\nu \in (0, 1)$.
Ensure: η_j.
1: Set $l \leftarrow 0$.
2: Choose initial η_0.
3: While η_l satisfies Wolfe's conditions

$$J(\boldsymbol{\xi}_j + \eta_l\mathbf{s}_j) \leq J(\boldsymbol{\xi}_j) + c_1\eta_l\mathcal{J}(\boldsymbol{\xi}_j)\mathbf{s}_j, \tag{25}$$

$$\mathcal{J}(\boldsymbol{\xi}_j + \eta_l\mathbf{s}_j)\mathbf{s}_j \geq c_2\mathcal{J}(\boldsymbol{\xi}_j)\mathbf{s}_j, \tag{26}$$

calculate
(i) $\eta_{l+1} = \nu\eta_l$,
(ii) $l \leftarrow l+1$.
4: Set $\eta_j = \eta_l$.

Furthermore, the initialization of parameters β and η is dependent on a specific optimization problem. Through a series of numerical experiments in simulations, we determined that the Dai–Yuan conjugate gradient method reaches a similar level of numerical robustness and accuracy for various initial values of β_0, whereas for the standard gradient algorithm, choosing the initial parameters largely affects the convergence and can cause numerical instabilities.

To determine the convergence step $\eta_j > 0$, we use a line search method that satisfies the Wolfe conditions as described in the following algorithm.

Details related to the line search method and Wolfe conditions can be found in [38].

Gradient Calculation

In this subsection, the relations for calculating the gradients that appear in (22) i.e., (23) is derived. In order to perform chain rule for derivatives backward in time, first the system (15) needs to be rewritten in discrete-time state-space form. For this purpose, we use the multistep Adams method. Compared to the most popular four stage Runge–Kutta method, which requires solving the approximation problem four times at each step, the Adams method requires only one calculation.

The explicit Adams approximation of system (15) can be formulated as follows. Let the time interval $\left[t_0,\ t_f\right]$ consists of $t_i = t_0 + i\tau$ for $i = 0, 1, 2, \ldots, N-1$, such that $\tau = (t_f - t_0)/N$ is the time step length, and let $\hat{\mathbf{x}}$ be the extended $4n$-dimensional state vector (n is dimension of state vector $\mathbf{x}$ and 4 is order of Adams method). Then, the system can be written in the following discrete-time state-space form:

$$\hat{\mathbf{x}}(i+1) = \hat{\mathbf{f}}(\hat{\mathbf{x}}(i)) + \mathbf{a} \otimes \mathbf{B}(\mathbf{x}(i))\boldsymbol{\pi} + \mathbf{a} \otimes \mathbf{D}(\mathbf{x}(i))\boldsymbol{\rho}, \ \ \hat{\mathbf{x}}(0) = \hat{\mathbf{x}}_0, \tag{27}$$

where the vector $\mathbf{a}$ contains a non-zero coefficients of Adams method (see ([39], page 358)), and where, for simplicity, we introduced

$$\mathbf{B}(\mathbf{x}) \equiv \mathbf{G}_1(\mathbf{x})\boldsymbol{\Theta}(\mathbf{x}), \quad \mathbf{D}(\mathbf{x}) \equiv \mathbf{G}_2(\mathbf{x})\boldsymbol{\Psi}(\mathbf{x}). \tag{28}$$

General considerations regarding Adams method are given in [39], while several novel schemes for multistep Adams method are proposed in [40].

The discrete-time form of the objective function (24) is

$$J = \tau \sum_{i=0}^{N-1} \left(\mathbf{h}^{\mathrm{T}}(\mathbf{x}(i))\mathbf{h}(\mathbf{x}(i)) + \boldsymbol{\pi}^{\mathrm{T}}\mathbf{P}(\mathbf{x}(i))\boldsymbol{\pi} - \mu\boldsymbol{\rho}^{\mathrm{T}}\mathbf{R}(\mathbf{x}(i))\boldsymbol{\rho} \right), \tag{29}$$

where for simplicity we introduced

$$\mathbf{P}(\mathbf{x}) \equiv \boldsymbol{\Theta}^{\mathrm{T}}(\mathbf{x})\boldsymbol{\Theta}(\mathbf{x}), \quad \mathbf{R}(\mathbf{x}) \equiv \boldsymbol{\Psi}^{\mathrm{T}}(\mathbf{x})\boldsymbol{\Psi}(\mathbf{x}). \tag{30}$$

The following is a derivation of recursive matrix relations for calculating $\nabla_{\pi} J$ using the basic chain rule arithmetic and matrix differentials calculus as well as certain properties of vectorization of matrices and Kronecker algebra. The relations for calculating $\nabla_{\rho} J$ are obtained in an analogous way.

The gradient of the objective function (29) with respect to the vector $\boldsymbol{\pi}$ is

$$\nabla_{\pi} J = \tau \sum_{i=0}^{N-1} \frac{\partial F(i)}{\partial \boldsymbol{\pi}}, \tag{31}$$

where

$$F(i) = \mathbf{h}^{\mathrm{T}}(\mathbf{x}(i))\mathbf{h}(\mathbf{x}(i)) + \boldsymbol{\pi}^{\mathrm{T}}\mathbf{P}(\mathbf{x}(i))\boldsymbol{\pi} - \mu\boldsymbol{\rho}^{\mathrm{T}}\mathbf{R}(\mathbf{x}(i))\boldsymbol{\rho}. \tag{32}$$

Then, an operation of partial derivative with respect to vector $\boldsymbol{\pi}$ is performed over the expression (32), which gives:

$$\begin{aligned} \nabla_{\pi} F(i) &= 2\mathbf{h}^{\mathrm{T}}(\mathbf{x}(i)) \cdot \nabla_{\hat{\mathbf{x}}}\mathbf{h}(\mathbf{x}(i)) \cdot \nabla_{\pi}\hat{\mathbf{x}}(i) - \mu[\boldsymbol{\rho} \otimes \boldsymbol{\rho}]^{\mathrm{T}} \cdot \nabla_{\hat{\mathbf{x}}}\mathrm{vec}(\mathbf{R}(\mathbf{x}(i))) \cdot \nabla_{\pi}\hat{\mathbf{x}}(i) \\ &+ [\mathrm{vec}(\mathbf{P}(\mathbf{x}(i)))]^{\mathrm{T}} \cdot (\boldsymbol{\pi} \oplus \boldsymbol{\pi}) + (\boldsymbol{\pi} \otimes \boldsymbol{\pi})^{\mathrm{T}} \cdot \nabla_{\hat{\mathbf{x}}}\mathrm{vec}(\mathbf{P}(\mathbf{x}(i))) \cdot \nabla_{\pi}\hat{\mathbf{x}}(i). \end{aligned} \tag{33}$$

In (33) it is necessary to calculate $\nabla_{\pi}\hat{\mathbf{x}}(i)$, which is obtained based on the expression (27) as follows:

$$\begin{aligned} \nabla_{\pi}\hat{\mathbf{x}}(i) &= \mathbf{a} \otimes \mathbf{B}(\mathbf{x}(i-1)) + \nabla_{\hat{\mathbf{x}}}[\mathbf{a} \otimes \mathbf{B}(\mathbf{x}(i-1))\boldsymbol{\pi}] \cdot \nabla_{\pi}\hat{\mathbf{x}}(i-1) \\ &+ \nabla_{\hat{\mathbf{x}}}\hat{\mathbf{f}}(\hat{\mathbf{x}}(i-1)) \cdot \nabla_{\pi}\hat{\mathbf{x}}(i-1) + \nabla_{\hat{\mathbf{x}}}[\mathbf{a} \otimes \mathbf{D}(\mathbf{x}(i-1))\boldsymbol{\rho}] \cdot \nabla_{\pi}\hat{\mathbf{x}}(i-1), \end{aligned} \tag{34}$$

where

$$\nabla_{\hat{\mathbf{x}}}[\mathbf{a} \otimes \mathbf{B}(\mathbf{x}(i-1))\boldsymbol{\pi}] = \mathbf{a} \otimes \left(\boldsymbol{\pi}^{\mathrm{T}} \otimes \mathbf{I}_n\right) \cdot \nabla_{\hat{\mathbf{x}}}\mathrm{vec}(\mathbf{B}(\mathbf{x}(i-1))), \tag{35}$$

$$\nabla_{\hat{\mathbf{x}}}[\mathbf{a} \otimes \mathbf{D}(\mathbf{x}(i-1))\boldsymbol{\rho}] = \mathbf{a} \otimes \left(\boldsymbol{\rho}^{\mathrm{T}} \otimes \mathbf{I}_n\right) \cdot \nabla_{\hat{\mathbf{x}}}\mathrm{vec}(\mathbf{D}(\mathbf{x}(i-1))). \tag{36}$$

The expressions (34)–(36) represent recursive matrix relations for $i = 1, 2, \ldots, N-1$ with initial conditions

$$\nabla_{\pi}\hat{\mathbf{x}}(0) = \mathbf{0}. \tag{37}$$

Note that the functions, $\hat{\mathbf{f}}(\hat{\mathbf{x}})$, $\mathbf{h}(\mathbf{x})$, $\mathbf{P}(\mathbf{x})$, $\mathbf{R}(\mathbf{x})$, $\mathbf{B}(\mathbf{x})$ and $\mathbf{D}(\mathbf{x})$, which appear in previous expressions, are known from system dynamics and predetermined approximation functions basis and their derivatives can be easily calculated to machine precision by applying a complex-step method.

The first-order derivative of $\hat{\mathbf{f}}(\hat{\mathbf{x}})$ with respect to $\hat{\mathbf{x}}$ using complex-step approximation is accomplished by approximating its component (let us say the k-th component) with a complex variable using a Taylor's series expansion [41,42]

$$\hat{f}_k(\hat{\mathbf{x}} + i\,h\,\mathbf{e}_j) = \hat{f}_k(\hat{\mathbf{x}}) + i\,h\,\nabla_{\hat{\mathbf{x}}}\hat{f}_k(\hat{\mathbf{x}}) \cdot \mathbf{e}_j + \frac{(i\,h)^2}{2!}\mathbf{e}_j^{\mathrm{T}} \cdot \nabla^2_{\hat{\mathbf{x}}}\hat{f}_k(\hat{\mathbf{x}}) \cdot \mathbf{e}_j + \ldots. \tag{38}$$

where $\mathbf{e}$ is unit vector. Taking only the imaginary parts gives

$$\nabla_{\hat{\mathbf{x}}}\hat{f}_k(\hat{\mathbf{x}}) \cdot \mathbf{e}_j = \frac{\mathrm{Im}\left\{\hat{f}_k(\hat{\mathbf{x}} + i\,h\,\mathbf{e}_j)\right\}}{h} + \mathcal{O}(h^2). \tag{39}$$

In the above expressions, i represents an imaginary unit $i^2 = -1$, while previously with i we denoted the i-th time point. For this reason, the $\hat{\mathbf{x}}(i)$ is omitted in these expressions, but this is implied. It can be seen that subtraction does not appear in the numerator of the expression (39), i.e., the subtractive cancellation problem has been removed. This means that the step h can be extremely small, so the derivative can be calculated to machine precision.

The implementation of a complex-step method can be very simply achieved by using a high-level programming language that does not require a prior definition of the variable types, but it is possible to define complex variables automatically by applying built-in functions.

The complex-step method for calculating $\nabla_{\hat{\mathbf{x}}}\hat{\mathbf{f}}(\hat{\mathbf{x}})$ is described by the pseudocode given in Algorithm 4. The calculation of derivatives of other known functions can be obtained in an analogous manner.

Algorithm 4 Calculation of derivatives at the i-th time point using the complex-step method.

Require: $n, h, \hat{\mathbf{x}}, \hat{\mathbf{f}}(\hat{\mathbf{x}})$
Ensure: $\nabla_{\hat{\mathbf{x}}}\hat{\mathbf{f}}(\hat{\mathbf{x}})$
- $\mathbf{x1} \leftarrow \hat{\mathbf{x}}$
- **for** $j = 1$ to n **do**
  - $x1_j \leftarrow complex(\hat{x}_j, h)$
  - $\mathbf{f1} \leftarrow \hat{\mathbf{f}}(\mathbf{x1})$
  - **for** $k = 1$ to n **do**
    - $\dfrac{\partial \hat{f}_k(\hat{x}_j)}{\partial x_j} \leftarrow \dfrac{imag(f1_k)}{h}$
  - **end for**
  - $\mathbf{x1} \leftarrow \hat{\mathbf{x}}$
- **end for**

4. Benchmark Examples with Analytic Solution

In this section, the algorithmic procedure described in the previous sections is tested on examples of nonlinear systems where the HJI equation can be solved analytically, thereby exactly determining the control and uncertainty vectors. In this way, it can be directly assessed whether the proposed algorithmic procedure calculates the required solutions with sufficient numerical efficiency in terms of convergence and accuracy. The algorithm is written and implemented in the MATLAB software package. When solving a particular problem, it is necessary to initialize the algorithm in a suitable way.

4.1. Example—First-Order System

Consider a scalar nonlinear system described by the following equation [21]:

$$\dot{x} = u + \mathrm{arctg}(x) \cdot d, \quad \mathbf{z} = \begin{bmatrix} x \\ u \end{bmatrix}. \tag{40}$$

If $\gamma > \frac{\pi}{2}$, then by solving the corresponding HJI equation (for details see [21]), the analytical feedbacks are

$$\begin{aligned} u^* &= -x\left(1 - \frac{1}{\gamma^2}\mathrm{arctg}^2(x)\right)^{-\frac{1}{2}}, \\ d^* &= \frac{1}{\gamma^2} x\, \mathrm{arctg}(x)\left(1 - \frac{1}{\gamma^2}\mathrm{arctg}^2(x)\right)^{-\frac{1}{2}}. \end{aligned} \tag{41}$$

Approximation functions of control and uncertainty variables are chosen in the following form

$$\begin{aligned} \hat{u}(x) &= p_1 x + p_2 x^3 + p_3 x^5 + p_4 x^7 + p_5 x^9, \\ \hat{d}(x) &= r_1 x + r_2 x^3 + r_3 x^5 + r_4 x^7 + r_5 x^9, \end{aligned} \tag{42}$$

i.e., written in the form (12) and (13) we have

$$\mathbf{\Theta}(x) = \mathbf{\Psi}(x) = \begin{bmatrix} x & x^3 & x^5 & x^7 & x^9 \end{bmatrix}, \ \boldsymbol{\pi} = \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \end{bmatrix}, \ \boldsymbol{\rho} = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \\ r_5 \end{bmatrix}. \tag{43}$$

In this example, we set the following values of the algorithm parameters: we divided the time interval from $t_0 = 0$, to $t_f = 10$ [s] into N =100,000 equal subintervals, i.e., the discretization step is $\tau = 10^{-4}$ [s] and we applied the Adams method of the 4-th order; vector of initial conditions $\mathbf{x}_0 = [1]$; initial value of the parameter $\mu_0 = \gamma_0^2 = 5$; vectors of initial parameters of approximation functions $\boldsymbol{\pi}_0 = \boldsymbol{\rho}_0 = \mathbf{0}$; stopping criterion of the quasi-Newton method $\varepsilon = 10^{-3}$; stopping criterion of conjugate-gradient method $\epsilon = 10^{-6}$, Dai–Yuan method is used by default, the initial numerical values of the conjugate gradient algorithm parameters are chosen as $\eta_0 = 0.1$ and $\beta_0 = 0.5$, coefficients $c_1 = 10^{-3}$, $c_2 = 0.9$ and $\nu = 0.8$.

We obtain the following parameter μ and parameters of approximation functions:

$$\mu^* = 2.4667, \quad \boldsymbol{\pi}^* = \begin{bmatrix} -0.9999 \\ -0.2045 \\ 0.0781 \\ -0.0391 \\ 0.0112 \end{bmatrix}, \quad \boldsymbol{\rho}^* = \begin{bmatrix} 0.0414 \\ 0.9638 \\ -1.7517 \\ 1.8462 \\ -0.7361 \end{bmatrix}, \tag{44}$$

i.e., minimum $\mathcal{L}_2$-gain is $\gamma^* = \sqrt{\mu^*} = 1.5706$.

Figure 1 shows the time dependence of the state variable, i.e., the response of the system (40) from the initial condition $x_0 = 1$, where the control variable and uncertainty variable are of the form (42) with parameters (44). The control and uncertainty variables from (42) obtained by the proposed algorithm with the parameters from (44) in comparison with the analytical solutions from the expression (41) depending on the state variable are shown in Figure 2. Figure 3 shows the solutions obtained by the derived algorithm in comparison with the analytical solutions (41) as a function of time. Based on everything shown, it can be concluded that the numerically obtained solutions approximate the analytical ones well with negligible error.

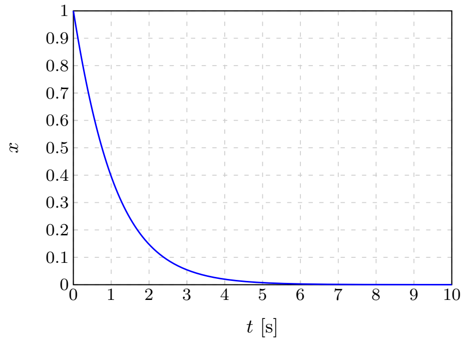


Figure 1. Time dependence of the state variable (Example 4.1).

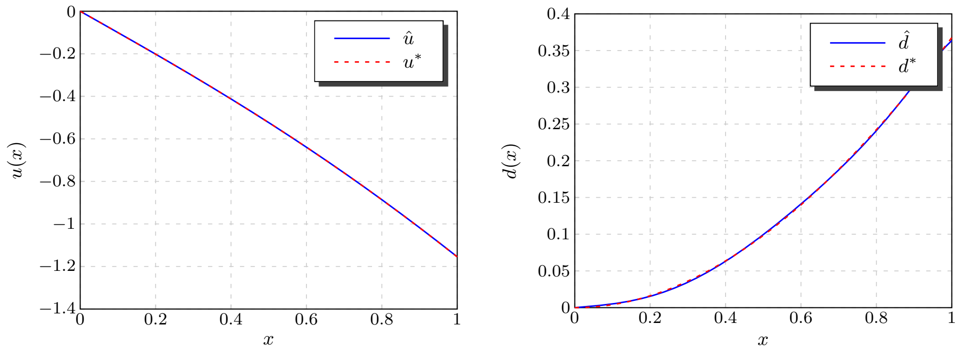


Figure 2. Control and uncertainty variables in dependence on the state variable (Example 4.1).

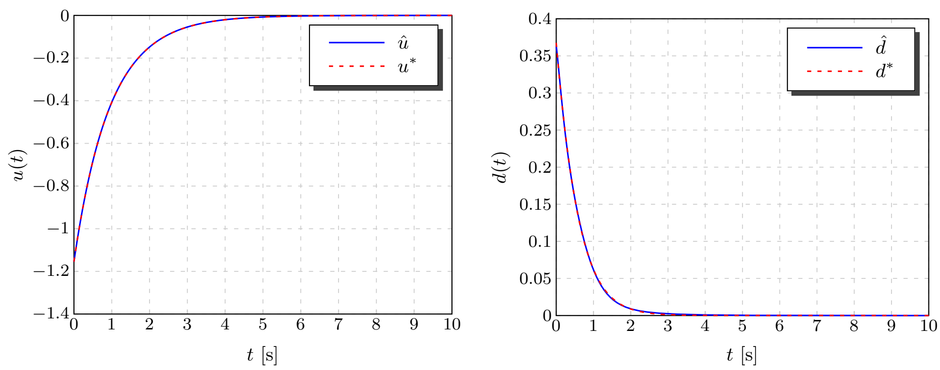


Figure 3. Time dependence of the control and uncertainty variables (Example 4.1).

4.2. Example—Second-Order System

Consider the second order nonlinear system [43]

$$\begin{aligned}
\dot{x}_1 &= -\frac{1}{8}\left(29x_1 + 87x_1x_2^2\right) - \frac{1}{4}\left(2x_2 + 3x_2x_1^2\right) + u_1 + \frac{1}{2}d, \\
\dot{x}_2 &= -\frac{1}{4}\left(x_1 + 3x_1x_2^2\right) + 3u_2 + d, \\
\mathbf{z} &= \begin{bmatrix} \sqrt{2}(2x_1 + 6x_1x_2^2) \\ \sqrt{2}(4x_2 + 6x_1^2x_2) \\ u_1 \\ u_2 \end{bmatrix}.
\end{aligned} \tag{45}$$

If $\gamma^* = 1$ then by solving the corresponding HJI equation (for details see [43]), the analytical feedbacks are

$$\begin{aligned}
u_1^*(\mathbf{x}) &= -x_1 - 3x_1x_2^2, \\
u_2^*(\mathbf{x}) &= -6x_2 - 9x_1^2x_2, \\
d^*(\mathbf{x}) &= \frac{1}{2}x_1 + 2x_2 + 3x_1^2x_2 + \frac{3}{2}x_1x_2^2.
\end{aligned} \tag{46}$$

Approximation functions of control and uncertainty variables are chosen in the following form:

$$\begin{aligned}
\hat{u}_1(\mathbf{x}) &= p_1^1x_1 + p_2^1x_2 + p_3^1x_1x_2 + p_4^1x_1^2x_2 + p_5^1x_1x_2^2 + p_6^1x_1^2x_2^2 + p_7^1x_1^2 + p_8^1x_2^2, \\
\hat{u}_2(\mathbf{x}) &= p_1^2x_1 + p_2^2x_2 + p_3^2x_1x_2 + p_4^2x_1^2x_2 + p_5^2x_1x_2^2 + p_6^2x_1^2x_2^2 + p_7^2x_1^2 + p_8^2x_2^2, \\
\hat{d}(\mathbf{x}) &= r_1x_1 + r_2x_2 + r_3x_1x_2 + r_4x_1^2x_2 + r_5x_1x_2^2 + r_6x_1^2x_2^2 + r_7x_1^2 + r_8x_2^2,
\end{aligned} \tag{47}$$

i.e., written in the form (12) and (13), we have

$$\begin{aligned}
&\boldsymbol{\theta}^1(\mathbf{x}) = \boldsymbol{\theta}^2(\mathbf{x}) = \boldsymbol{\psi}(\mathbf{x}) = \begin{bmatrix} x_1 & x_2 & x_1x_2 & x_1^2x_2 & x_1x_2^2 & x_1^2x_2^2 & x_1^2 & x_2^2 \end{bmatrix}, \\
&\boldsymbol{\Theta}(\mathbf{x}) = \begin{bmatrix} \boldsymbol{\theta}^1(\mathbf{x}) & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\theta}^2(\mathbf{x}) \end{bmatrix}, \quad \boldsymbol{\Psi}(\mathbf{x}) = \boldsymbol{\psi}(\mathbf{x}), \\
&\mathbf{p}^1 = \begin{bmatrix} p_1^1 \\ p_2^1 \\ \vdots \\ p_8^1 \end{bmatrix}, \quad \mathbf{p}^2 = \begin{bmatrix} p_1^2 \\ p_2^2 \\ \vdots \\ p_8^2 \end{bmatrix}, \quad \boldsymbol{\pi} = \begin{bmatrix} \mathbf{p}^1 \\ \mathbf{p}^2 \end{bmatrix}, \quad \boldsymbol{\rho} = \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_8 \end{bmatrix}.
\end{aligned} \tag{48}$$

In this example, we set the following values of the algorithm parameters: we divided the time interval from $t_0 = 0$, to $t_f = 3$ [s] into $N = 5000$ equal subintervals, i.e., the discretization step is $\tau = 10^{-4}$ [s] and we applied the Adams method of the 4-th order by default; vector of initial conditions $\mathbf{x}_0 = [1 \;\; 1]^{\mathrm{T}}$; initial value of the parameter $\mu_0 = \gamma_0^2 = 5$; vectors of initial parameters of approximation functions $\boldsymbol{\pi}_0 = \boldsymbol{\rho}_0 = \mathbf{1}$; stopping criterion of the quasi-Newton method $\varepsilon = 10^{-3}$; stopping criterion of conjugate-gradient method $\epsilon = 10^{-3}$, Dai–Yuan method is used by default, the initial numerical values of the conjugate gradient algorithm parameters are chosen as $\eta_0 = 0.1$ and $\beta_0 = 0.5$, coefficients $c_1 = 10^{-3}$, $c_2 = 0.9$ and $\nu = 0.8$.

We obtain the following parameter μ and parameters of approximation functions:

$$\mu^* = 1.0021, \quad \mathbf{p}^{1*} = \begin{bmatrix} -0.9997 \\ -0.0241 \\ 0.0785 \\ -0.0676 \\ -3.0576 \\ 0.0422 \\ -0.0003 \\ 0.0287 \end{bmatrix}, \ \mathbf{p}^{2*} = \begin{bmatrix} 0.0004 \\ -6.0319 \\ 0.1224 \\ -9.1208 \\ -0.1533 \\ 0.1016 \\ -0.0009 \\ 0.0825 \end{bmatrix}, \ \boldsymbol{\rho}^* = \begin{bmatrix} 0.4995 \\ 2.0478 \\ -0.1576 \\ 3.1370 \\ 1.6151 \\ -0.0847 \\ 0.0007 \\ -0.0579 \end{bmatrix}, \tag{49}$$

i.e., the minimum $\mathcal{L}_2$-gain is $\gamma^* = \sqrt{\mu^*} = 1.0010$.

Figure 4 shows the time dependence of the state variables, i.e., the response of the system (45) from the initial conditions $\mathbf{x}_0 = [1 \ \ 1]^{\mathrm{T}}$, where the control variables and uncertainty variable are of the form (47) with parameters (49). Figure 5 shows the solutions obtained by the derived algorithm in comparison with the analytical solutions (46). In this example, as in the previous, it is evident that the numerical solution approximates the analytical solution well.

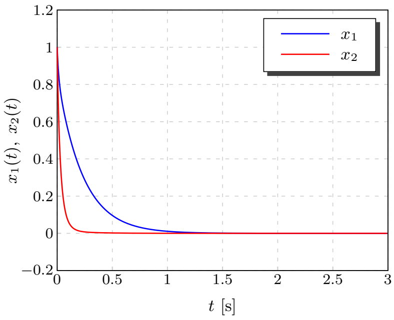
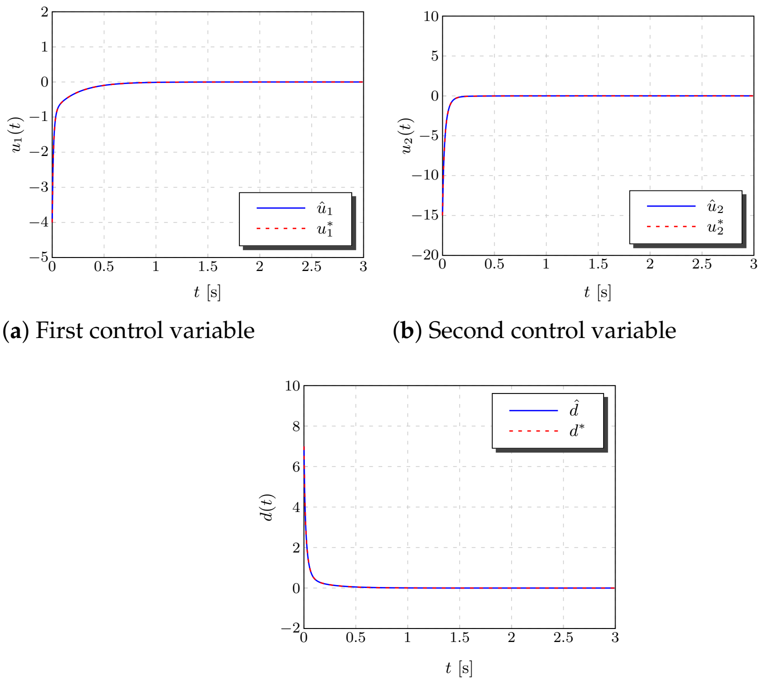


Figure 4. Time dependence of the state variables (Example 4.2).



(**a**) First control variable (**b**) Second control variable

(**c**) Uncertainty variable

Figure 5. Time dependence of the control and uncertainty variables (Example 4.2).

5. Conclusions

In this paper, an approach for the solution of a nonlinear $\mathcal{H}_\infty$ control problem is presented. Instead of using the approximation methods for solving the corresponding HJI equation, the solution is obtained by direct numerical calculation of the control and uncertainty variables that explicitly depend on the system states. In order to achieve numerical efficiency, the proposed algorithmic procedure uses quasi-Newton method, conjugate gradient method, line search method with Wolfe conditions, Adams approximation method for time discretization, and complex-step calculation of derivatives. In spite of the fact that the methods used in this paper are known from the references cited, in our approach, they are integrated together to provide a suitable mathematical tool for numerical solution of the zero-sum differential game related to the nonlinear $\mathcal{H}_\infty$ control problem.

The extension of this approach can be continued in two research directions: *(i)* consider output measurement-feedback $\mathcal{H}_\infty$ optimal control problem, and *(ii)* the case where the initial state vector is unknown and treated as an uncertainty, i.e., the maximizing "player". It can be assumed that the proposed strategy can be extended to these two cases without a significant increase in its complexity.

Author Contributions: Conceptualization, V.M. and J.K.; methodology, V.M. and J.K.; software, V.M.; validation, V.M., J.K. and M.L.; formal analysis, V.M.; investigation, V.M., J.K. and M.L.; resources, V.M.; data curation, V.M., J.K. and M.L.; writing—original draft preparation, V.M.; writing—review and editing, V.M., J.K. and M.L.; visualization, V.M., J.K. and M.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work has been supported by the European Regional Development Fund, Operational Program Competitiveness and Cohesion 2014–2020, grant number KK.01.1.1.04.0092.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

HJI	Hamilton–Jacobi–Isaacs
LMI	Linear Matrix Inequalities
PID	Proportional–Integral–Derivative

References

1. Helton, J.W.; James, M.R. *Extending $\mathcal{H}_\infty$ Control to Nonlinear Systems*; SIAM: Philadelphia, PA, USA, 1999.
2. Van Der Schaft, A. $\mathcal{L}_2$-gain analysis of nonlinear systems and nonlinear state feedback $\mathcal{H}_\infty$ control. *IEEE Trans. Autom. Control* **1992**, *37*, 770–784. [CrossRef]
3. Basar, T.; Olsder, G.J. *Dynamic Noncooperative Game Theory*; SIAM: Philadelphia, PA, USA, 1999.
4. Basar, T.; Bernard, P. *$\mathcal{H}_\infty$ Optimal Control and Related Minimax Design Problems, Second Edition*; Birkhuser: Boston, MA, USA, 1995.
5. Khanbaghi, M.; Zečević, A. An LMI-based control strategy for large-scale systems with applications to interconnected microgrid clusters. *IEEE Access* **2022**, *10*, 111554–111563. [CrossRef]
6. Chen, B.S.; Ma, Y.S.; Lee, M.Y. Stochastic robust $\mathcal{H}_\infty$ decentralized network formation tracking control of large-scale team satellites via event-triggered mechanism. *IEEE Access* **2022**, *10*, 62011–62036. [CrossRef]
7. Chatavi, M.; Vu, M.T.; Mobayen, S.; Fekih, A. $\mathcal{H}_\infty$ robust LMI-based nonlinear state feedback controller of uncertain nonlinear systems with external disturbances. *Mathematics* **2022**, *10*, 3518. [CrossRef]
8. Gritli, H.; Belghith, S. Robust feedback control of the underactuated inertia wheel inverted pendulum under parametric uncertainties and subject to external disturbances: LMI formulation. *J. Frankl. Inst.* **2018**, *355*, 9150–9191. [CrossRef]
9. Xi, A.; Cai, Y. A nonlinear finite-time robust differential game guidance law. *Sensors* **2022**, *22*, 6650. [CrossRef]
10. Liu, D.; Xue, S.; Zhao, B.; Luo, B.; Wei, Q. Adaptive dynamic programming for control: A survey and recent advances. *IEEE Trans. Syst. Man Cybern. Syst.* **2021**, *51*, 142–160. [CrossRef]

11. Sun, W.; Pan, Y.; Lim, J.; Theodorou, E.A.; Tsiotras, P. Min-max differential dynamic programming: Continuous and discrete time formulations. *J. Guid. Control. Dyn.* **2018**, *41*, 2568–2580. [CrossRef]
12. Vamvoudakis, K.G.; Modares, H.; Kiumarsi, B.; Lewis, F.L. Game theory-based control system algorithms with real-time reinforcement learning: How to solve multiplayer games online. *IEEE Control Syst. Mag.* **2017**, *37*, 33–52.
13. Ivanov, I.G.; Bogdanova, B.C. The iterative solution to discrete-time $\mathcal{H}_\infty$ control problems for periodic systems. *Algorithms* **2016**, *9*, 20. [CrossRef]
14. Lu, X.; Li, H. A hybrid control approach to $\mathcal{H}_\infty$ problem of nonlinear descriptor systems with actuator saturation. *IEEE Trans. Autom. Control* **2021**, *66*, 4960–4966. [CrossRef]
15. Lu, X.; Li, H. Prescribed finite-time $\mathcal{H}_\infty$ control for nonlinear descriptor systems. *IEEE Trans. Circuits Syst. II Express Briefs* **2021**, *68*, 2917–2921. [CrossRef]
16. Aliyu, M.D.S. An improved iterative computational approach to the solution of the Hamilton-Jacobi equation in optimal control problems of affine nonlinear systems with application. *Int. J. Syst. Sci.* **2020**, *51*, 2625–2634. [CrossRef]
17. Mu, C.; Wang, K. Approximate-optimal control algorithm for constrained zero-sum differential games through event-triggering mechanism. *Nonlinear Dyn.* **2019**, *95*, 2639–2657. [CrossRef]
18. Peretz, Y. A Randomized Algorithm for Optimal PID Controllers. *Algorithms* **2018**, *11*, 81. [CrossRef]
19. Graham, A. *Kronecker Products and Matrix Calculus: With Applications*; Ellis Horwood Limited: West Sussex, UK, 1981.
20. Brewer, J.W. Kronecker products and matrix calculus in system theory. *IEEE Trans. Circuits Syst.* **1978**, *25*, 772–781. [CrossRef]
21. Van Der Schaft, A. *$\mathcal{L}_2$-Gain and Passivity Techniques in Nonlinear Control*; Springer: London, UK, 1996.
22. Isaacs, R. *Differential Games. A Mathematical Theory with Application to Warfare and Pursuit, Control and Optimization*; John Wiley and Sons, Inc.: New York, NY, USA, 1965.
23. Astolfi, A. Singular $\mathcal{H}_\infty$ control for nonlinear systems. *Int. J. Robust Nonlinear Control* **1997**, *7*, 727–740. [CrossRef]
24. Maas, W.C.A.; Van der Schaft, A.J. Singular nonlinear $\mathcal{H}_\infty$ optimal control by state feedback. In Proceedings of the The 33rd IEEE Conference on Decision and Control, Lake Buena Vista, FL, USA, 14–16 December 1994; Volume 2, pp. 1415–1420.
25. Courant, R.; Hilbert, D. *Methods of Mathematical Physics: Volume 1*; Interscience Publishers, Inc.: New York, NY, USA, 1966.
26. Cybenko, G. Approximation by superpositions of a sigmoidal function. *Math. Control. Signals Syst.* **1989**, *2*, 303–314. [CrossRef]
27. Barron, A.R. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Trans. Inf. Theory* **1993**, *39*, 930–945. [CrossRef]
28. Sandberg, I.W. Notes on uniform approximation of time-varying systems on finite time intervals. *IEEE Trans. Circuits Syst. I Fundam. Theory Appl.* **1998**, *45*, 863–865. [CrossRef]
29. Sandberg, I.W. Uniform approximation of periodically-varying systems. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2004**, *51*, 1631–1638. [CrossRef]
30. Adams, R.A.; Fournier, J.J.F. *Sobolev Spaces*; Pure and Applied Mathematics; Elsevier Science: Amsterdam, The Netherlands, 2003.
31. Davis, P.J. *Interpolation and Approximation*; Dover Publications Inc.: New York, NY, USA, 1975.
32. Meinardus, G. *Approximation of Functions: Theory and Numerical Methods*, Larry, L., Transed; Schumaker Springer: Berlin, Germany, 1967.
33. Pu, D.; Zhang, J. Inexact generalized Newton methods for second order C-differentiable optimization. *J. Comput. Appl. Math.* **1998**, *93*, 107–122. [CrossRef]
34. Qi, L. On superlinear convergence of quasi-Newton methods for nonsmooth equations. *Oper. Res. Lett.* **1997**, *20*, 223–228. [CrossRef]
35. Pang, J.S.; Qi, L. Nonsmooth equations: Motivation and algorithms. *SIAM J. Optim.* **1993**, *3*, 443–465. [CrossRef]
36. Qi, L.; Sun, J. A nonsmooth version of Newton's method. *Math. Program.* **1993**, *58*, 353–367. [CrossRef]
37. Dai, Y.H.; Yuan, Y. A nonlinear conjugate gradient method with a strong global convergence property. *SIAM J. Optim.* **1999**, *10*, 177–182. [CrossRef]
38. Nocedal, J.; Wright, S.J. *Numerical Optimization*; Springer Science + Business Media, LLC: New York, NY, USA, 2006.
39. Hairer, E.; Nørsett, S.P.; Wanner, G. *Solving Ordinary Differential Equations I—Nonstiff Problems, Second Revised Edition*; Springer: Berlin, Germany, 2008.
40. Pesterev, D.; Druzhina, O.; Pchelintsev, A.; Nepomuceno, E.; Butusov, D. Numerical integration schemes based on composition of adjoint multistep methods. *Algorithms* **2022**, *15*, 463. [CrossRef]
41. Squire, W.; Trapp, G. Using complex variables to estimate derivatives of real functions. *SIAM Rev.* **1998**, *40*, 110–112. [CrossRef]
42. Fornberg, B. Numerical differentiation of analytic functions. *ACM Trans. Math. Softw.* **1981**, *7*, 512–526. [CrossRef]
43. Dierks, T.; Jagannathan, S. Optimal control of affine nonlinear continuous-time systems using an online Hamilton-Jacobi-Isaacs formulation. In Proceedings of the 49th IEEE Conference on Decision and Control, Atlanta, GA, USA, 15–17 December 2010; pp. 3048–3053.