2012 20th Mediterranean Conference on Control & Automation (MED)
Barcelona, Spain, July 3-6, 2012

Numerical Algorithm for Nonlinear State Feedback $\mathcal{H}_\infty$ Optimal Control Problem

Vladimir Milić, Alberto Bemporad, Josip Kasać and Željko Šitum

Abstract— **In this paper, the numerical algorithm based on conjugate gradient method to solve a finite-horizon min-max optimization problem arising in the $\mathcal{H}_\infty$ control of nonlinear systems is presented. The feedback control and disturbance variables are formulated as a linear combination of basis functions. The proposed algorithm, which has a backward-in-time structure, directly finds very accurate approximations of these feedbacks. Benchmark examples with analytic solutions are provided to demonstrate the effectiveness of the proposed algorithm.**

I. INTRODUCTION

Over the past decades, there has been tremendous progress in the development of nonlinear state feedback $\mathcal{H}_\infty$ controllers for applications in many different engineering fields, see for example [1], [2], [3] to name a few. The formulation of the nonlinear $\mathcal{H}_\infty$ control theory has been well developed [4], [5], [6]. In [7] the $\mathcal{H}_\infty$ control theory has been formulated into nonlinear $\mathcal{L}_2$-gain optimal control problem, which requires the solution of the Hamilton-Jacobi-Isaacs equation (HJIE).

The analytic solution of HJIE is difficult or impossible to find in most cases. An inversion approach which involves solving HJIE like scalar quadratic algebraic equation with the gradient of the smooth scalar as unknown has been reported in [8]. In [9] the HJIE for systems with input constraints has been derived. Authors have introduced a two-player policy iteration scheme that results in a framework that allows the use of neural networks to approximate optimal policies and value functions. In [10] an application of neural networks to find closed-form representation of the feedback strategies and the value function that solves the associated HJIE has been presented. This approach is computationally expensive since the tuning of neural network weights is based on method of weighted residuals which includes calculation of Lebesgue integrals over domain $\mathbb{R}^n$ where n is dimension of the state-space. In other words, the problem is curse of dimensionality when the computational cost increase exponentially with the dimension of the state-space system.

In our approach, the nonlinear state feedback $\mathcal{H}_\infty$ problem is transformed into a nonlinear finite-horizon optimal state feedback control problem with min-max cost. The main aim is to find control variables and disturbance internal/external variables such that control variables minimize an optimization criterion in the presence of worst-case behaviour of disturbance variables. From the standpoint of differential games [11], this means that the control input is the minimizer player and disturbance input is the maximizer player.

In contrast to the approaches founded on neural networks for approximating solutions of HJIE [9], [10], in this paper the tuning of basis functions weights is based on direct minimization of the performance criteria with simultaneous maximization of the same performance criteria. A conjugate gradient approach is used for minimization/maximization of the performance criteria, while the performance criteria gradients are calculated exactly using chain rule for ordered derivatives. Since the control, disturbance and state variables are treated as dependent variables (coupled via plant equations), the final algorithm has a backward-in-time structure similar to the back-propagation-through-time (BPTT) [12] algorithm.

The algorithm presented in this paper is an extension of the recent work in [13], [14] toward finite-horizon $\mathcal{H}_\infty$ optimal state feedback control. In [13], and [14] a conjugate gradient-based BPTT-like algorithm for optimal open-loop control of nonlinear multivariable systems with control and state vectors constraints has been presented. The algorithm performance has been illustrated on a realistic high-dimensional vehicle dynamics control example. The optimization results have demonstrated favourable features of the algorithm in terms of accuracy, robust numerical stability, and relatively fast execution.

Furthermore, with aim to enhance the accuracy of the solution, the higher-order Adams numerical integration schemes [15] are used. One of the reason for using multistep Adams method, among other one-step methods like Runge-Kutta, is that it can be easily transformed to the causal state-space form. The second reason, when compared to the Runge-Kutta method, is that the vector function must be calculated only once in the sampling time. In the other words, the application of the explicit Adams method has no influence on the algorithm complexity except on extension of the overall state-space system dimension.

The rest of paper is organized as follows. In Section II the state feedback nonlinear $\mathcal{H}_\infty$ control problem is transformed into feedback min-max optimal control problem. We formulate the feedbacks as a linear combination of basis functions. In Section III the backward-in-time min-max control algorithm is derived. Section IV illustrates the effectiveness of the proposed algorithm on a nonlinear

The material in this paper is based on work financed by Croatian Science Foundation. Grant number 03.01/170.

Vladimir Milić, Josip Kasać and Željko Šitum are with Faculty of Mechanical Engineering and Naval Architecture, University of Zagreb, Zagreb HR-10000, Croatia. Emails: vladimir.milic@fsb.hr, josip.kasac@fsb.hr, zeljko.situm@fsb.hr
Alberto Bemporad is with IMT (Institutions, Markets, Technologies) Institute for Advanced Studies Lucca, Lucca IT-55100, Italy. Email: alberto.bemporad@imtlucca.it

978-1-4673-2531-8/12/$31.00 ©2012 IEEE

benchmark examples with analytic solution. Finally, Section V concludes the paper.

Notation: The notation used is fairly standard. Matrices are represented as bold upper case. All vectors are intended as column vectors and represented as bold lower case. Scalars are represented as italic lower case. The symbol $^{\mathrm{T}}$ denotes transposition, and $\mathbf{I}$ is identity matrix and $\mathbf{0}$ is null matrix of appropriate dimensions. The derivative of the vector (of order m) with respect to vector (of order n) is the matrix (of order $n \times m$). The operator $\|\cdot\|$ denotes the Euclidean norm. We denote by $\mathbf{u}^*(\mathbf{x})$ and $\mathbf{d}^*(\mathbf{x})$ the exact solutions of nonlinear state feedback $\mathcal{H}_\infty$ control problem and by $\hat{\mathbf{u}}(\mathbf{x})$ and $\hat{\mathbf{d}}(\mathbf{x})$ its numerical solutions. We avoid to explicitly show the dependence of the variables from the time when not needed.

II. Problem formulation

Consider the following affine nonlinear dynamical system of the form

$$\begin{aligned} &\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}) + \mathbf{G}_1(\mathbf{x})\mathbf{u}(t) + \mathbf{G}_2(\mathbf{x})\mathbf{d}(t),\ \mathbf{x}(0) = \mathbf{x}_0, \\ &\mathbf{z}(t) = \begin{bmatrix}\mathbf{g}_3^{\mathrm{T}}(\mathbf{x}) & \mathbf{u}^{\mathrm{T}}(t)\end{bmatrix}^{\mathrm{T}},\ \mathbf{f}(0) = 0,\ \mathbf{g}_3(0) = 0, \end{aligned} \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^{n_\bullet}$ is the state vector, $\mathbf{u} \in \mathbb{R}^{n_u}$ is the control input, $\mathbf{d} \in \mathbb{R}^{n_d}$ is the vector representing internal/external disturbance, $\mathbf{z} \in \mathbb{R}^{n_z}$ is the to-be-controlled output or penalty variable. The functions $\mathbf{f}(\cdot)$, $\mathbf{G}_1(\cdot)$, $\mathbf{G}_2(\cdot)$, $\mathbf{g}_3(\cdot)$ are smooth functions of $\mathbf{x}$. It is assumed that $\mathbf{d} \in \mathcal{L}_2[0,\ t_f]$, $t_f \geq 0$, where $\mathcal{L}_2[0,\ t_f]$ denotes the standard Lebesgue space of vector valued square integrable functions over $[0,\ t_f]$.

The objective is to determine a state-feedback controller $\mathbf{u}(\mathbf{x})$, when all the states of the system are available, and determine "worst case" disturbance internal/external variables $\mathbf{d}(\mathbf{x})$, such that finite $\mathcal{L}_2$-gain from $\mathbf{d}$ to $\mathbf{z}$ is less than or equal to some positive number $\gamma > 0$. In other words, for every initial conditions $\mathbf{x}(0) = \mathbf{x}_0$

$$\int_0^{t_f} \|\mathbf{z}\|^2 dt \leq \gamma^2 \int_0^{t_f} \|\mathbf{d}\|^2 dt + J(\mathbf{x}_0). \tag{2}$$

The original idea behind this approach was to formulate the $\mathcal{H}_\infty$ disturbance attenuation problem as a differential game in which $\mathbf{u}$ and $\mathbf{d}$ are two opposing players [16]. It is well known [5] that this problem is equivalent to the solvability of the following min-max optimization problem

$$J^*(\mathbf{x}_0) = \min_{\mathbf{u}} \max_{\mathbf{d}} \left\{ \int_0^{t_f} \left(\|\mathbf{z}\|^2 - \gamma^2 \|\mathbf{d}\|^2 \right) dt \right\}, \tag{3}$$

subject to (1).

The above problem is solved by the feedbacks [7]

$$\mathbf{u}^*(\mathbf{x}) = -\mathbf{G}_1^{\mathrm{T}}(\mathbf{x}) \frac{\partial V}{\partial \mathbf{x}}, \quad \mathbf{d}^*(\mathbf{x}) = \frac{1}{\gamma^2} \mathbf{G}_2^{\mathrm{T}}(\mathbf{x}) \frac{\partial V}{\partial \mathbf{x}}, \tag{4}$$

where $V \geq 0$ is a solution of the HJIE

$$\begin{aligned} \frac{\partial^{\mathrm{T}} V}{\partial \mathbf{x}} \mathbf{f}(\mathbf{x}) + \frac{1}{2} \frac{\partial^{\mathrm{T}} V}{\partial \mathbf{x}} \Bigg[\frac{1}{\gamma^2} \mathbf{G}_2(\mathbf{x}) \mathbf{G}_2^{\mathrm{T}}(\mathbf{x}) - \\ - \mathbf{G}_1(\mathbf{x}) \mathbf{G}_1^{\mathrm{T}}(\mathbf{x}) \Bigg] \frac{\partial V}{\partial \mathbf{x}} + \frac{1}{2} \mathbf{g}_3^{\mathrm{T}}(\mathbf{x}) \mathbf{g}_3(\mathbf{x}) = 0, \end{aligned} \tag{5}$$

with $V(0) = 0$.

Lemma 1: If the nonlinear system (1) is *(i)* asymptotically stable with $\mathbf{d} = 0$ and $\mathbf{u} = \mathbf{u}^*$, *(ii)* has $\mathcal{L}_2$-gain less than γ when $\mathbf{d} \neq 0$, *(iii)* the cost function (3) is smooth then the closed-loop dynamic is asymptotically stable.

Proof: See [7]. ∎

III. Algorithm derivation

In order to numerically solve problem (3) subject to (1), we consider a special form of i-th component of $\mathbf{u}$ and $\mathbf{d}$ from (4) as a linear combination of the basis functions on a compact set $\Omega \subset \mathbb{R}^{n_\bullet}$ as follows

$$\hat{u}_i(\mathbf{x}) = \sum_{k=1}^{n_\sigma} v_k \sigma_k(\mathbf{x}), \quad \hat{d}_i(\mathbf{x}) = \sum_{k=1}^{n_\psi} q_k \psi_k(\mathbf{x}), \tag{6}$$

with $\sigma_k(\mathbf{x}) \in C^1(\Omega)$, $\psi_k(\mathbf{x}) \in C^1(\Omega)$, and $\sigma_k(0) = 0$, $\psi_k(0) = 0$.

Weierstrass's theorem [17] says that any continuous function on a bounded domain in $\mathbb{R}^{n_\bullet}$ can be approximated by a complete independent basis set. Standard usage of Weierstrass's approximation theorem uses polynomial basis functions. Non-polynomial basis sets have been considered in [18].

The performance criteria (3) implicitly depends on v_k and q_k, so they can be determined according to the performance criteria. The numerical algorithm for tuning v_k weights is based on direct minimization of the performance criteria with simultaneous tuning of q_k weights based on maximization of the same performance criteria.

A. Time discretization

To compute numerical approximation of the nonlinear state feedback $\mathcal{H}_\infty$ problem, we perform discretization of the system dynamics (1) based on explicit Adams method.

Assume that the time interval $[0,\ t_f]$ is divided into $N-1$ sub-intervals of equal length. Then the time grid consists of points $t_i = i\tau$ for $i = 0,\ 1,\ 2,\ \ldots,\ N-1$ where $\tau = t_f/N$ is the time step length.

The discrete-time form of the equation (1) with (6) is

$$\mathbf{x}(i+1) = \boldsymbol{\Phi}_d \left(\mathbf{x}(i),\ \hat{\mathbf{u}}(i),\ \hat{\mathbf{d}}(i) \right), \quad \mathbf{x}(0) = \mathbf{x}_0, \tag{7}$$

where $\hat{\mathbf{u}}(i)$, $\hat{\mathbf{d}}(i)$ denote the control and disturbance sequence over the interval $0,\ 1,\ 2,\ \ldots,\ N-1$, respectively, while

$$\boldsymbol{\Phi}_d \left(\mathbf{x}(i),\ \hat{\mathbf{u}}(i),\ \hat{\mathbf{d}}(i) \right) = \mathbf{x}(i) + \tau \sum_{j=1}^{k} a_j \boldsymbol{\Phi}(i-j+1), \tag{8}$$

is the k-th order Adams approximation of the continuous-time state equation for $i = k-1,\ k,\ k+1,\ \ldots$ and initial conditions $\mathbf{x}(0) = \mathbf{x}_0$, $\mathbf{x}(1) = \mathbf{x}_1$, $\ldots$, $\mathbf{x}(k-1) = \mathbf{x}_{k-1}$, where $\boldsymbol{\Phi}(\cdot) = \mathbf{f}(\mathbf{x}(\cdot)) + \mathbf{G}_1(\mathbf{x}(\cdot))\hat{\mathbf{u}}(\cdot) + \mathbf{G}_2(\mathbf{x}(\cdot))\hat{\mathbf{d}}(\cdot)$. a_j are constant coefficients (for their numerical values see [15]).

The explicit Adams method (8) is a k-th order vector difference equation, which can be conveniently transformed

into the following discrete-time state-space form

$$\begin{aligned} x_j(i+1) &= x_j(i) + \tau a_1 \phi_j(i) + \tau x_{n_\bullet+j}(i), \\ x_{rn_\bullet+j}(i+1) &= a_{r+1}\phi_j(i) + x_{(r+1)n_\bullet+j}(i), \\ x_{(k-1)n_\bullet+j}(i+1) &= a_k \phi_j(i), \end{aligned} \tag{9}$$

for $r = 1, 2, \ldots, k-2$, $j = 1, 2, \ldots, n_0$, $i = k-1, k, k+1, \ldots$, and the initial conditions

$$\begin{aligned} x_j(k-1) &= x_{j(k-1)}, \\ x_{qn_\bullet+j}(k-1) &= \sum_{l=q+1}^{k} a_l \phi_j(k-1+q-l), \end{aligned} \tag{10}$$

for $q = 1, 2, \ldots, k-1$. Using the vector notation, the state-space form of the k-th order Adams method reads

$$\hat{\mathbf{x}}(i+1) = \hat{\boldsymbol{\phi}}\left(\hat{\mathbf{x}}(i),\, \hat{\mathbf{u}}(i),\, \hat{\mathbf{d}}(i)\right), \quad \hat{\mathbf{x}}(0) = \hat{\mathbf{x}}_0, \tag{11}$$

where $\hat{\mathbf{x}}(t)$ is the extended ($n_a = n_0 \cdot k$)-dimensional state vector

$$\hat{\mathbf{x}}(i) = \begin{bmatrix} x_1(i) & x_2(i) & \ldots & x_{n_a-1}(i) & x_{n_a}(i) \end{bmatrix}^{\mathrm{T}}, \tag{12}$$

and

$$\hat{\boldsymbol{\phi}} = \begin{bmatrix} x_1(i) + \tau a_1 \phi_1(i) + \tau x_{n_0+1}(i) & \ldots & a_k \phi_{n_0}(i) \end{bmatrix}^{\mathrm{T}}. \tag{13}$$

The Adams method of the k-th order, as a multistep method, requires knowledge of k initial conditions. In this work, to determine these initial conditions the fourth-order Runge-Kutta method is used. In the case of ordinary differential equations system described by (1), the fourth-order Runge-Kutta method is stated as follows. Start with initial point $(t_0, \mathbf{x}_0)$ and generate the sequence of approximations using $\mathbf{x}(i+1) = \mathbf{x}(i) + \frac{\tau}{6}(\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4)$, where

$$\begin{aligned} \mathbf{k}_1 &= \boldsymbol{\phi}\left(\mathbf{x}(i),\, \hat{\mathbf{u}}(t_i),\, \hat{\mathbf{d}}(t_i)\right), \\ \mathbf{k}_2 &= \boldsymbol{\phi}\left(\mathbf{x}(i) + \frac{\tau}{2}\mathbf{k}_1,\, \hat{\mathbf{u}}(t_i + \frac{\tau}{2}),\, \hat{\mathbf{d}}(t_i + \frac{\tau}{2})\right), \\ \mathbf{k}_3 &= \boldsymbol{\phi}\left(\mathbf{x}(i) + \frac{\tau}{2}\mathbf{k}_2,\, \hat{\mathbf{u}}(t_i + \frac{\tau}{2}),\, \hat{\mathbf{d}}(t_i + \frac{\tau}{2})\right), \\ \mathbf{k}_4 &= \boldsymbol{\phi}\left(\mathbf{x}(i) + \tau\mathbf{k}_3,\, \hat{\mathbf{u}}(t_i + \tau),\, \hat{\mathbf{d}}(t_i + \tau)\right). \end{aligned} \tag{14}$$

From expression (14) it can be seen that fourth-order Runge-Kutta method requires the calculation in mid-points $\hat{\mathbf{u}}(i+1/2) \equiv \hat{\mathbf{u}}(t_i + \tau/2)$ and $\hat{\mathbf{d}}(i+1/2) \equiv \hat{\mathbf{d}}(t_i + \tau/2)$. A rough approximation $\hat{\mathbf{u}}(t_i + \tau/2) \approx \hat{\mathbf{u}}(t_i)$ and $\hat{\mathbf{d}}(t_i + \tau/2) \approx \hat{\mathbf{d}}(t_i)$ would significantly deteriorate the algorithm accuracy. The value of control and disturbance vector in mid-points can be approximated by $\hat{\mathbf{u}}(t_i + \tau/2) \approx (\hat{\mathbf{u}}(t_i) + \hat{\mathbf{u}}(t_{i+1}))/2$, $\hat{\mathbf{d}}(t_i + \tau/2) \approx \left(\hat{\mathbf{d}}(t_i) + \hat{\mathbf{d}}(t_{i+1})\right)/2$. This approximation is verified by simulations and provides satisfactory accuracy in the range of fourth order approximation. More details on Adams and Runge-Kutta methods can be found in [15].

The discrete-time form of the $\mathcal{H}_\infty$ performance criteria reads

$$J(\mathbf{x}_0) = \tau \sum_{i=0}^{N-1} \left(\|\mathbf{z}(i)\|^2 - \gamma^2 \|\hat{\mathbf{d}}(i)\|^2 \right). \tag{15}$$

B. Conjugate gradient algorithm

In this work the optimization approach is based on conjugate gradient descent/ascent algorithm in the following form

$$w_k^{(l+1)} = w_k^{(l)} + \eta^{(l)} s_k^{(l)}, \tag{16}$$

$$s_k^{(l+1)} = \lambda \frac{\partial J}{\partial w_k^{(l+1)}} + \beta^{(l)} s_k^{(l)}, \tag{17}$$

where w_k is a k-th component of the vector which contains the basis function weights from equations (6). $\lambda = -1$ if w_k corresponds to weights of basis function for control variables, and $\lambda = 1$ if w_k corresponds to weights of basis function for disturbance variables.

Furthermore, s_k is k-th component of the search direction vector, and $l = 1, 2, \ldots, M$ is the number of gradient algorithm iterations.

Note that maximization of the performance criteria is provided by simple change of sign in front of gradient of cost function with respect to w_k in (17).

The standard method for computing $\eta^{(l)}$ is line search algorithm which requires one-dimensional minimization of the performance criteria. This is a computationally expensive method which may require many evaluations of the performance criteria during one iteration of the gradient algorithm. Also, if the performance criteria is not appropriately scaled, the line search algorithm may exhibit poor convergence properties [14]. In order to avoid these issues, in this work we use the SuperSAB approach [19] which requires only the information on gradient directions in two consecutive iterations of the gradient algorithm. The algorithm is modified in terms of using a scalar convergence rate $\eta^{(l)}$ (as oppose to a matrix formulation), in order to avoid discontinuities in disturbance vector $\mathbf{d}$ and optimized control vector $\mathbf{u}$. The modified SuperSAB algorithm is given by

$$\eta^{(l)} = \begin{cases} d^+ \eta^{(l-1)} & \text{if} \quad \sum_{k=1}^{n_w} \frac{\partial J}{\partial w_k^{(l)}} \frac{\partial J}{\partial w_k^{(l-1)}} > 0, \\ d^- \eta^{(l-1)} & \text{if} \quad \sum_{k=1}^{n_w} \frac{\partial J}{\partial w_k^{(l)}} \frac{\partial J}{\partial w_k^{(l-1)}} < 0, \\ \eta^{(l-1)} & \text{if} \quad \sum_{k=1}^{n_w} \frac{\partial J}{\partial w_k^{(l)}} \frac{\partial J}{\partial w_k^{(l-1)}} = 0, \end{cases} \tag{18}$$

where $0 < d^- < 1 < d^+$ are dilatation coefficients (decreasing/increasing factors).

Scalar $\beta^{(l)}$ is determined by

$$\beta^{(l)} = \frac{\mu \mathbf{g}^{(l+1)} \mathbf{g}^{(l+1)\mathrm{T}} + [1-\mu]\mathbf{g}^{(l+1)} \left[\mathbf{g}^{(l+1)} - \mathbf{g}^{(l)}\right]^{\mathrm{T}}}{\nu \mathbf{g}^{(l)} \mathbf{g}^{(l)\mathrm{T}} + [1-\nu]\mathbf{s}^{(l)} \left[\mathbf{g}^{(l+1)} - \mathbf{g}^{(l)}\right]^{\mathrm{T}}}, \tag{19}$$

where $\mathbf{g} = \begin{bmatrix} \frac{\partial J}{\partial w_1} & \frac{\partial J}{\partial w_2} & \ldots & \frac{\partial J}{\partial w_{n_w}} \end{bmatrix}$, $\mu \in [0,\, 1]$, $\nu \in [0,\, 1]$. If the scalars μ and ν take only their extreme values, 0 or 1, then four possible combinations are obtained: Fletcher-Reeves method [20] for $\mu = 1$ and $\nu = 1$, Polak-Ribere method [21] for $\mu = 0$ and $\nu = 1$, Hestenes-Stiefel method [22] for $\mu = 0$ and $\nu = 0$, Dai-Yuan method [23] for $\mu = 1$ and $\nu = 0$. The numerical comparison of these methods are reported in [24].

It is important to say that, in order to ensure numerical stability of the algorithm the parameter $\beta^{(l)}$ is limited to β_{max}. If the parameter $\beta^{(l)}$ has a constant value, $0 < \beta^{(l)} < 1$, then the conjugate gradient algorithm becomes equivalent to a standard gradient algorithm with momentum.

C. Gradient calculation

The gradient of the performance criteria (3) according to the w_k in the l-th iteration of the gradient algorithm is given by

$$\frac{\partial J}{\partial w_k} = \tau \sum_{i=0}^{N-1} \left(\sum_{r=1}^{n_\bullet} \frac{\partial F(i)}{\partial \hat{x}_r(i)} \frac{\partial \hat{x}_r(i)}{\partial w_k} + \right. \\ \left. + \sum_{r=1}^{n_u} \frac{\partial F(i)}{\partial \hat{u}_r(i)} \frac{\partial \hat{u}_r(i)}{\partial w_k} + \sum_{r=1}^{n_d} \frac{\partial F(i)}{\partial \hat{d}_r(i)} \frac{\partial \hat{d}_r(i)}{\partial w_k} \right), \tag{20}$$

for $k = 1, 2, \ldots, n_\sigma + n_\psi$. Note that if w_k corresponds to weights of basis function for control variables then $\partial \hat{d}_r(i)/\partial w_k = 0$, and if w_k corresponds to weights of basis function for disturbance variables then $\partial \hat{u}_r(i)/\partial w_k = 0$.

From (11) it follows that

$$\frac{\partial \hat{x}_r(i)}{\partial w_k} = \sum_{j=1}^{n_\bullet} \frac{\partial \hat{\phi}_r(i-1)}{\partial \hat{x}_j(i-1)} \frac{\partial \hat{x}_j(i-1)}{\partial w_k} + \\ + \sum_{m=1}^{n_u} \frac{\partial \hat{\phi}_r(i-1)}{\partial \hat{u}_m(i-1)} \frac{\partial \hat{u}_m(i-1)}{\partial w_k} + \\ + \sum_{n=1}^{n_d} \frac{\partial \hat{\phi}_r(i-1)}{\partial \hat{d}_n(i-1)} \frac{\partial \hat{d}_n(i-1)}{\partial w_k}. \tag{21}$$

Further, from (6) it follows that

$$\frac{\partial \hat{u}_m(i)}{\partial w_k} = \sum_{j=1}^{n_\sigma} \left(\frac{\partial w_j}{\partial w_k} \sigma_j + w_j \sum_{p=1}^{n_0} \frac{\partial \sigma_j}{\partial \hat{x}_p(i)} \frac{\partial \hat{x}_p(i)}{\partial w_k} \right), \tag{22}$$

for $k = 1,\ 2,\ \ldots,\ n_\sigma$, and

$$\frac{\partial \hat{d}_n(i)}{\partial w_k} = \sum_{j=1}^{n_\psi} \left(\frac{\partial w_{n_\sigma + j}}{\partial w_k} \psi_j + w_{n_\sigma + j} \sum_{p=1}^{n_0} \frac{\partial \psi_j}{\partial \hat{x}_p(i)} \frac{\partial \hat{x}_p(i)}{\partial w_k} \right), \tag{23}$$

for $k = n_\sigma + 1,\ n_\sigma + 2,\ \ldots,\ n_\sigma + n_\psi$.

D. Jacobians calculation

The extended Jacobian matrix for Adams method can be expressed based on (9) as functions of the basic Jacobian matrix as follows

$$\frac{\partial \hat{\boldsymbol{\Phi}}(i)}{\partial \hat{\mathbf{x}}(i)} = \begin{bmatrix} \mathbf{I} + \tau a_1 \frac{\partial \boldsymbol{\Phi}(i)}{\partial \mathbf{x}(i)} & \tau \mathbf{I} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ a_2 \frac{\partial \boldsymbol{\Phi}(i)}{\partial \mathbf{x}(i)} & \mathbf{0} & \mathbf{I} & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{k-1} \frac{\partial \boldsymbol{\Phi}(i)}{\partial \mathbf{x}(i)} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{I} \\ a_k \frac{\partial \boldsymbol{\Phi}(i)}{\partial \mathbf{x}(i)} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \end{bmatrix}, \tag{24}$$

$$\frac{\partial \hat{\boldsymbol{\Phi}}(i)}{\partial \hat{\mathbf{u}}(i)} = \begin{bmatrix} \tau a_1 \frac{\partial \boldsymbol{\Phi}(i)}{\partial \mathbf{u}(i)} \\ a_2 \frac{\partial \boldsymbol{\Phi}(i)}{\partial \mathbf{u}(i)} \\ \vdots \\ a_k \frac{\partial \boldsymbol{\Phi}(i)}{\partial \mathbf{u}(i)} \end{bmatrix}, \quad \frac{\partial \hat{\boldsymbol{\Phi}}(i)}{\partial \hat{\mathbf{d}}(i)} = \begin{bmatrix} \tau a_1 \frac{\partial \boldsymbol{\Phi}(i)}{\partial \mathbf{d}(i)} \\ a_2 \frac{\partial \boldsymbol{\Phi}(i)}{\partial \mathbf{d}(i)} \\ \vdots \\ a_k \frac{\partial \boldsymbol{\Phi}(i)}{\partial \mathbf{d}(i)} \end{bmatrix}.$$

Similarly, the gradient of function in the sum from (15) with respect to extended state vector is related to the basic gradient as follows

$$\frac{\partial F(i)}{\partial \hat{\mathbf{x}}(i)} = \left[\left(\frac{\partial F(i)}{\partial \mathbf{x}(i)} \right)^{\mathrm{T}} \quad 0 \quad \ldots \quad 0 \right]^{\mathrm{T}}. \tag{25}$$

These basic Jacobians and gradient can be calculated using automatic differentiation. Automatic (or algorithmic) differentiation (AD) is now a widely used tool within scientific computing. The standard reference is the book [25]. Over the past decades, extensive research activities led to a through understanding and analysis of two basic modes: the forward and reverse modes. For the previously derived algorithm the forward mode has been chosen.

A variety of tools exist for AD of the standard programming languages. In this work, TOMLAB/MAD [26] mathematical software is used.

Application of AD comparing with numerical differentiation [14] provide significant reduction of the algorithm computational time.

E. Algorithm summary

The following steps summarize the previously derived algorithm:

Step 1: Initialization: starting with $l = 0$ set initial state vector $\mathbf{x}_0$; choose a complete set of C^1 basis function and initial weights defined in (6); set initial time t_0, final time t_f, number of time intervals N, time step length τ; set number of gradient algorithm iterations M, initial convergence rate $\eta^{(0)}$, dilatation parameters d^- and d^+, and β_{max}.

Step 2: Calculation of state vector (11) using Adams method.

Step 3: Calculation of extended Jacobians (24) and gradient (25) using automatic differentiation.

Step 4: Backward-in-time calculation of (21), (22) and (23).

Step 5: Calculation of gradient (20).

Step 6: Calculation of new basis functions weights using conjugate gradient methods (16) and (17).

Step 7: Shift the index $l = l + 1$ and go back to step 2.

IV. Simulation results

In this section we test the performance of the proposed numerical approach on two benchmark examples with analytic solution.

A. Benchmark examples with analytic solution

Example 1. Consider the scalar nonlinear system [7]

$$\dot{x} = u + x\,d, \quad \mathbf{z} = \begin{bmatrix} x & u \end{bmatrix}^{\mathrm{T}}. \tag{26}$$

For this example, the HJIE only has solution for $|x| < \gamma$. If $\gamma = 1$ then analytic solution of HJIE leads to feedbacks

$$u^*(x) = -\frac{x}{\sqrt{1 - x^2}}, \quad d^*(x) = \frac{x^2}{\sqrt{1 - x^2}}, \tag{27}$$

for $-1 < x < 1$. One selects the basis functions as

$$\{\sigma_k\} = \{\psi_k\} = \left\{ x,\ x^3,\ x^5,\ x^7,\ x^9 \right\}. \tag{28}$$

To numerically compute weights of these basis functions, we carry out proposed algorithm from Section III in MATLAB program which is executed by utilizing a portable (notebook) PC computer with Intel Core Duo CPU (2.00 GHz). The terminal time is $t_f = 20$ sec and the number of optimization time intervals is $N = 20000$ so that the sampling interval is $\tau = 0.001$ sec. The conjugate gradient Hestenes-Stiefel method is used, and the number of iterations of the gradient algorithm is $M = 400$. The average algorithm execution time is about 0.14 sec per iteration. The numerical values of the algorithm parameters in (18) and (19) are chosen as: $d^+ = 1.05$, $d^- = 0.95$, $\eta^{(0)} = 1.0$, $\beta_{max} = 1.0$.

The state variable initial condition is $x_0 = 0.8$ and the initial basis functions weights are chosen as $v_{k_\bullet} = q_{k_\bullet} = -5$. After 400 iterations of the algorithm the final numerical values of weights for control variable $\hat{u}$ are: $v_1 = -1.0011$, $v_2 = -0.4532$, $v_3 = -0.8197$, $v_4 = 1.1990$, $v_5 = -2.1017$; and for disturbance variable $\hat{d}$ are: $q_1 = 0.0825$, $q_2 = 3.0317$, $q_3 = -6.6278$, $q_4 = 11.2674$, $q_5 = -5.5801$.

Fig. 1 shows the maximum errors between the numerical solutions $\hat{u}$, $\hat{d}$ and the exact analytical solutions given by (27) for the Adams method of 1st and 4th order. It can been seen that the error progressively decreases as the Adams method of 4th order is applied.

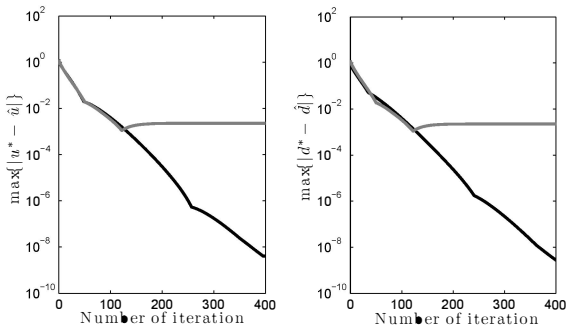

Fig. 1. Maximum errors of control and disturbance variables depending on the number of iterations for first (grey line) and fourth (black line) order Adams method.

Example 2. Consider the second order nonlinear system [27]

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

If $\gamma = 1$ then by solving the corresponding HJIE analytical feedbacks are

$$\begin{aligned}
&u_1^*(\mathbf{x}) = -x_1 - 3x_1x_2^2, \quad u_2^*(\mathbf{x}) = -6x_1 - 9x_1^2x_2,\\
&d^*(\mathbf{x}) = \frac{1}{2}x_1 + 2x_2 + 3x_1^2x_3 + \frac{3}{2}x_1x_2^2.
\end{aligned} \tag{30}$$

The following basis functions are chosen:

$$\begin{aligned}
&\{\sigma_k\} = \{\psi_k\} =\\
&= \left\{x_1,\ x_2,\ x_1x_2,\ x_1^2x_2,\ x_1x_2^2,\ x_1^2x_2^2,\ x_1^2,\ x_2^2\right\}.
\end{aligned} \tag{31}$$

In this example the terminal time is $t_f = 3$ sec and the number of optimization time intervals is $N = 3000$ so that the sampling interval is $\tau = 0.001$ sec. The conjugate gradient Dai-Yuan method is used, and the number of iterations of the gradient algorithm is $M = 500$. The average algorithm execution time is about 1.3 sec per iteration. The numerical values of the algorithm parameters in (18) and (19) are chosen as: $d^+ = 1.05$, $d^- = 0.95$, $\eta^{(0)} = 1.0$, $\beta_{max} = 1.0$.

The state variables initial conditions are $\mathbf{x}_0 = [1\ \ 1]^{\mathrm{T}}$ and the initial basis functions weights are chosen as $v_{k_\bullet} = q_{k_\bullet} = -0.1$. After 500 iterations of the algorithm the final numerical values of weights for control variable $\hat{u}_1$ are: $v_1 = -0.9997$, $v_2 = -0.0243$, $v_3 = 0.0792$, $v_4 = -0.0683$, $v_5 = -3.0584$, $v_6 = 0.0428$, $v_7 = -0.0003$, $v_8 = 0.0291$; for control variable $\hat{u}_2$ are: $v_1 = 0.0004$, $v_2 = -6.0325$, $v_3 = 0.1246$, $v_4 = -9.1230$, $v_5 = -0.1557$, $v_6 = 0.1033$, $v_7 = -0.0009$, $v_8 = 0.0838$; and finally for disturbance variable $\hat{d}$ are: $q_1 = 0.4995$, $q_2 = 2.0478$, $q_3 = -0.1576$, $q_4 = 3.1370$, $q_5 = 1.6151$, $q_6 = -0.0847$, $q_7 = 0.0008$, $q_8 = -0.0579$.

As can be seen from Fig. 1 a relatively small error of the numerical solution compared to the analytical solution is achieved.

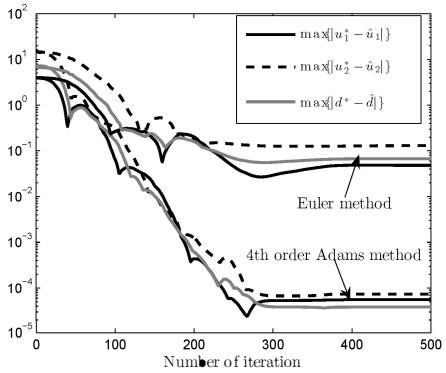

Fig. 2. Maximum errors of control and disturbance variables depending on the number of iterations for first (Euler method) and fourth order Adams method.

B. Discussion on numerical robustness and computing efficiency

The algorithm contains several free parameters such as the time step τ, and the parameters $\eta^{(0)}$, d^-, d^+, and β_{max} of the modified conjugate gradient methods. The numerical stability is not affected by decreasing the time step τ. But, naturally, there is a minimal value of the time step τ which guaranties numerical stability of the Adams integration methods.

With respect to parameters d^-, d^+, and β_{max}, the tuning region is known in advance, while the initial learning rate $\eta^{(0)}$ is dependent on specific optimization problem. The

algorithm convergence is more sensitive to the choice of parameter β_{max} than to choice of other parameters. The larger the parameter β_{max}, the faster the convergence. The algorithm convergence is largely insensitive to the choice of dilatation parameters d^+ and d^-, if they lie in the intervals $0.85 \leq d^- \leq 0.95$ and $1.05 \leq d^+ \leq 1.15$. Similar values of dilatation coefficients are reported in the neural networks literature [19]. A recommended relationship between dilatations parameters is $d^+ \approx 1/d^-$.

According to the literature dealing with the conjugate gradient algorithm there is no a general rule how to chose the appropriate method for calculating the quantity $\beta^{(l)}$. The Hestenes-Stiefel and the Dai-Yuan methods have shown the best convergence properties and related accuracy on benchmark examples (26) and (29). The limit value of $\beta^{(l)}$ which guaranties numerical stability is $\beta_{max} = 1$, but depending on particular optimization problem this limit can be increased (e.g. $\beta_{max} = 1.2$), in order to provide a faster convergence of the algorithm.

It is illustrated in [13] that the conjugate gradient methods are less sensitive to the choice of initial learning rate $\eta^{(0)}$ than the standard gradient algorithm. The conjugate gradient method reaches a similar level of solution accuracy for various initial learning rate values $\eta^{(0)}$, while for the standard gradient algorithm the choice of $\eta = \eta^{(0)}$ largely affects the algorithm convergence and can cause numerical instabilities.

V. Conclusion

In this paper a numerical algorithm for solving the nonlinear state feedback $\mathcal{H}_\infty$ control problem has been presented. A backward-in-time conjugate gradient optimal control algorithm with min-max cost has been proposed and tested on first and second-order affine nonlinear benchmark examples with analytic solutions. The results have illustrated favourable features of the algorithm in terms of accuracy and consistent numerical stability. The algorithm can easily be applied to higher-order system with increasing complexity.

While the individual methods such as backward-in-time technique, conjugate gradient optimization algorithms, Adams method for solving ODEs, and AD are known from the literature, in our approach they are integrated together to provide an effective, novel algorithm for numerical solution of the state feedback $\mathcal{H}_\infty$ control problems.

Comparison of the algorithm with other existing methods is a subject of ongoing work and future publications. Also, the future research will be oriented towards extension of the proposed algorithm to the case of noisy state measurements and the proposed algorithm for $\mathcal{H}_\infty$ feedback control will be extended with dynamic observer providing optimal output feedback control.

References

[1] M. Hardt, J. W. Helton, and K. Kreutz-Delgado, "Numerical solution of nonlinear $\mathcal{H}_2$ and $\mathcal{H}_\infty$ control problems with application to jet engine compressors," *IEEE Transactions on Control Systems Technology*, vol. 18, no. 1, pp. 98–111, 2000.

[2] C.-C. Kung, "Nonlinear $\mathcal{H}_\infty$ robust control applied to F-16 aircraft with mass uncertainty using control surface inverse algorithm," *Journal of the Franklin Institute*, vol. 345, no. 8, pp. 851–876, 2008.

[3] A. G. A. Siqueira and M. A. Terra, "Neural network-based $\mathcal{H}_\infty$ control for fully actuated and underactuated cooperative manipulators," *Control Engineering Practice*, vol. 17, no. 3, pp. 418–425, 2009.

[4] M. Abu-Khalaf, J. Huang, and F. L. Lewis, *Nonlinear $\mathcal{H}_2/\mathcal{H}_\infty$ Costrained Feedback Control: A Practical Design Approach Using Neural Networks*. London: Springer-Verlag, 2006.

[5] T. Basar and P. Bernard, *$\mathcal{H}_\infty$ Optimal Control and Related Minimax Design Problems, Second Edition*. Boston, MA: Birkhuser, 1995.

[6] J. W. Helton and M. R. James, *Extending $\mathcal{H}_\infty$ Control to Nonlinear Systems*. Philadelphia: SIAM, 1999.

[7] A. Van Der Schaft, "$\mathcal{L}_2$-gain analysis of nonlinear systems and nonlinear state feedback $\mathcal{H}_\infty$, control," *IEEE Transactions on Automatic Control*, vol. 37, no. 6, pp. 770–784, 1992.

[8] M. D. S. Aliyu, "An approach for solving the Hamilton-Jacobi-Isaacs equation (HJIE) in nonlinear $\mathcal{H}_\infty$ control," *Automatica*, vol. 39, no. 5, pp. 877–884, 2003.

[9] M. Abu-Khalaf, F. L. Lewis, and J. Huang, "Policy iterations on the Hamilton-Jacobi-Isaacs equation for $\mathcal{H}_\infty$ state feedback control with input saturation," *IEEE Transactions on Automatic Control*, vol. 51, no. 12, pp. 1989–1995, 2006.

[10] ——, "Neurodynamic programming and zero-sum games for constrained control systems," *IEEE Transactions on Neural Networks*, vol. 19, no. 7, pp. 1243–1252, 2008.

[11] R. Isaacs, *Differential Games. A Mathematical Theory with Application to Warfare and Pursuit, Control and Optimization*. New York: John Wiley and Sons, Inc., 1965.

[12] P. J. Werbos, "Backpropagation through time: What it does and how to do it," in *Proceedings of IEEE*, vol. 78, October 1990, pp. 1550–1560.

[13] J. Kasać, J. Deur, B. Novaković, and I. Kolmanovsky, "A Conjugate Gradient-based BPTT-like Optimal Control Algorithm," in *IEEE International Conference on Control*, Saint Petersburg, Russia, July 8-10 2009.

[14] J. Kasać, J. Deur, B. Novaković, I. Kolmanovsky, and F. Assadian, "A conjugate gradient-based BPTT-like optimal control algorithm with vehicle dynamics control application," *IEEE Transactions on Control Systems Technology*, vol. 19, no. 6, pp. 1587–1595, 2011.

[15] E. Hairer, S. P. Nørsett, and G. Wanner, *Solving Ordinary Differential Equations I - Nonstiff Problems, Second Revised Edition*. Berlin: Springer-Verlag, 2008.

[16] P. Tsiotras, M. Corless, and M. Rotea, "An $\mathcal{L}_2$ disturbance attenuations solution to the nonlinear benchmark problem," *Int. J. Robust Nonlinear Control*, vol. 8, pp. 311–330, 1998.

[17] R. Adams and J. Fournier, *Sobolev Spaces*. New York: Academic Press, 2003.

[18] I. W. Sandberg, "Notes on uniform approximation of time-varying systems on finite time intervals," *IEEE Transactions on Circuits and Systems 1: Fundamental Theory and Applications*, vol. 45, no. 8, pp. 863–865, 1998.

[19] T. Tollenaere, "SuperSAB: Fast adaptive backpropagation with good scaling properties," *Neural Networks*, vol. 3, no. 5, pp. 561–573, 1990.

[20] R. Fletcher and C. M. Reeves, "Function minimization by conjugate gradients," *The Computer Journal*, vol. 7, no. 2, pp. 149–154, 1964.

[21] E. Polak and G. Ribiere, "Note sur la convergence de mthodes de directions conjugues," *Revue franaise d'informatique et de recherche oprationnelle, srie rouge*, vol. 3, no. 1, pp. 35–43, 1969.

[22] M. R. Hestenes and E. Stiefel, "Methods of Conjugate Gradients for Solving Linear Systems," *Journal of Research of the National Bureau of Standards*, vol. 49, no. 6, pp. 409–436, 1952.

[23] Y. H. Dai and Y. Yuan, "A Nonlinear Conjugate Gradient Method with a Strong Global Convergence Property," *SIAM J. on Optimization*, vol. 10, no. 1, pp. 177–182, 1999.

[24] Y. H. Dai and Q. Ni, "Testing different conjugate gradient methods for large scale unconstrained optimization," *Journal of Computational Mathematics*, vol. 21, no. 3, pp. 311–320, 2003.

[25] A. Griewank and A. Walther, *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation, Second Edition*. Philadelphia, USA: SIAM, 2008.

[26] S. A. Forth, "An efficient overloaded implementation of forward mode automatic differentiation in MATLAB," *ACM Transactions on Mathematical Software*, vol. 32, no. 2, pp. 195–222, 2006.

[27] T. Dierks and S. Jagannathan, "Optimal control of affine nonlinear continuous-time systems using an online hamilton-jacobi-isaacs formulation," in *IEEE Conference on Decision and Control*, Atlanta, GA, USA, 15-17 Dec. 2010, pp. 3048–3053.