

# Mixed $\mathcal{H}_2/\mathcal{H}_\infty$ -Policy Learning Synthesis

Lekan Molu

Microsoft Research, 300 Lafayette Street, New York City, NY 10012.  
[lekanmolu@microsoft.com](mailto:lekanmolu@microsoft.com).

**Abstract:** A robustly stabilizing optimal control policy in a model-free mixed  $\mathcal{H}_2/\mathcal{H}_\infty$ -control setting is here put forward for counterbalancing the slow convergence and non-robustness of traditional high-variance policy optimization (and by extension policy gradient) algorithms. Leveraging Itô's stochastic differential calculus, we iteratively solve the system's continuous-time (closed-loop) generalized algebraic Riccati equation (GARE) whilst updating its admissible controllers in a two-player, zero-sum differential game setting. Our new results are illustrated by learning-enabled control systems which gather previously disseminated results in this field in one holistic data-driven presentation with greater simplification, improvement, and clarity.

Copyright © 2023 The Authors. This is an open access article under the CC BY-NC-ND license (<https://creativecommons.org/licenses/by-nc-nd/4.0/>)

**Keywords:** Robust control; Data-driven optimal control; Machine learning in modelling, prediction, control and automation.

## 1. INTRODUCTION

We consider system stabilization together with Zames' sensitivity compensation in plants disturbed by additive Wiener process and uncertainties (Zames, 1981) under model-free policy optimization and gradient settings. We pose our solution in a mixed  $\mathcal{H}_2/\mathcal{H}_\infty$  linear quadratic (LQ) optimal control problem (OCP) (Khargonekar et al., 1988) within the family of policy optimization (PO) schemes. Connecting this mixed design synthesis to modern policy optimization algorithms in machine learning, we optimize a performance index that is the upper bound on the  $\mathcal{H}_2$ -norm of the plant transfer function subject to the plant's  $\mathcal{H}_\infty$ -norm constraints: we must find feasible stabilizing policies whilst guaranteeing robustness to a measure of disturbance (Zhang et al., 2019; Cui and Molu, 2023).

PO algorithms, which encapsulate policy gradient (PG) methods (Kakade, 2001; Agarwal et al., 2021), are attractive for modern data-driven problems since they (i) admit continuously differentiable policy parameterization; (ii) are easily extensible to function approximation settings; and (iii) admit structured state and control spaces. As such, PG algorithms are increasingly becoming integral to modern engineering solutions, recommender systems, finance, and critical infrastructure given the growing complexity of the systems that we build and the massive availability of datasets. A major drawback of PG algorithms, however, is that they compute high-variance gradient estimates of the LQR costs from Monte-Carlo trajectory rollouts and bootstrapping. As such, they tend to possess slow convergence guarantees.

To address PG's characteristic non-robustness to uncertainty, and its characteristic slow convergence, recent efforts have proposed mixed  $\mathcal{H}_2/\mathcal{H}_\infty$  control proposal (Zhang et al., 2019; Zhang et al., 2019; Cui and Molu, 2023) as a risk-mitigation design tool: imposing an additional  $\mathcal{H}_\infty$ -norm constraint on the  $\mathcal{H}_2$  cost to be mini-

mized, one guarantees robust stability and performance in the presence of unforeseen uncertainties, noise, worst-case disturbance or incorrectly estimated dynamics – signatures of PG algorithms.

Under stabilizable and observable system parameter conditions, (Zhang et al., 2019) established globally sub-linear and locally super-linear convergence rates in linear quadratic (LQ) zero-sum dynamic game settings. We improved upon these convergence rates in (Cui and Molu, 2023) by solving the PO problem recursively given an initial stabilizing feedback gain that also preserved the  $\mathcal{H}_\infty$  robustness metric. In many modern engineering systems that employ PO, however, stochastic system parameters often have to be identified from nonlinear system trajectory data. For these schemes to work, the control designer may need to linearize nonlinear trajectories about successive equilibrium points (whilst imposing the standard stabilizability and observability constraints on system parameters to be identified). In this paper, we take steps to curb our earlier stabilizability and observability assumptions in (Cui and Molu, 2023).

**Contributions:** We here present a holistic synthesis of our previous dissemination, initiate a search for the initial  $\mathcal{H}_\infty$ -norm constraints-preserving feedback gain,  $K_1$ , and demonstrate the efficacy of our results on a nonlinear numerical experimental setting. The rest of this paper is structured as follows: in §2, we introduce notations and contextualize the problem; in §3 we present our methods; results that back up our claims are set forth in §4. We draw conclusions in §5.

## 2. PRELIMINARIES

### 2.1 Notations

We adopt standard vector-matrix notations throughout. Conventions: Capital and lower-case Roman letters are respectively matrices and vectors; calligraphic letters are

\* The author is with Microsoft Research, NYC.

sets. Exceptions: time variables e.g.,  $t, t_0, t_f, T$  will always be real numbers.

The  $n$ -dimensional Euclidean space is  $\mathbb{R}^n$ . The real and imaginary parts of the complex  $s$ -plane are respectively  $\text{Re}(s)$  and  $\text{Imag}(s)$ . The singular values of  $A \in \mathbb{R}^{n \times n}$  are  $\sigma_i(A), i = 1, \dots, n$ . The standard  $\mathcal{H}_\infty$  norm of a complex matrix-valued function  $G(j\omega)$  is defined over the analytic and bounded functions in the open right-half plane as  $\|G(j\omega)\|_\infty = \sup_{\omega \in \mathbb{R}} \sigma_{\max}(G(j\omega))$  where  $\sigma_{\max}(\cdot)$  denotes the maximum singular value. The  $\mathcal{L}_2$  norm for a signal, function, or the induced matrix norm is denoted  $\|\cdot\|_2$ . We let  $\{\lambda_i(X)\}_{i=1}^n$  denote the  $n$ -eigenvalues of  $X \in \mathbb{R}^{n \times n}$  where  $\lambda_1 < \lambda_2 < \dots < \lambda_n$ . When an optimized variable e.g.,  $u$  is optimal with respect to an index of performance, it shall be denoted  $u^*$ .

All vectors are column-stacked. The Kronecker product of  $A \in \mathbb{R}^{m \times n}$  and  $B \in \mathbb{R}^{p \times q}$  is  $A \otimes B$ . Symmetric  $n$ -dimensional matrices shall belong in  $\mathbb{S}^n$ . A positive definite (resp. negative definite)  $A$  is written  $A \succ 0$  (resp.  $A \prec 0$ ). We denote the index of a given matrix or vector by subscripts. Colon notation denotes the full range of a given index. Indexing ranges over  $1, \dots, n$  for an  $n$ -dimensional vector. The  $i$ th row of a given matrix  $A$  is  $A_{[i,:]}$ , while the  $j$ th column of  $A$  is  $A_{[:,j]}$ . Blockwise indexing follows similar conventions.

Denote by  $x_{ij}$  the  $(ij)$ 'th entry of  $X \in \mathbb{R}^{m \times n}$  and by  $x_i$  the  $i$ 'th element of  $x \in \mathbb{R}^n$ . The full vectorization of  $X \in \mathbb{R}^{m \times n}$  is the  $mn \times 1$  vector obtained by stacking the columns of  $X$  on top of one another i.e.  $\text{vec}(X) = [x_{11}, x_{21}, \dots, x_{m1}, x_{12}, \dots, x_{m2}, \dots, x_{mn}]^T$ . Let  $P \in \mathbb{S}^n$ , then the half-vectorization of  $P$  is the  $n(n+1)/2$  column vector as a result of a vectorization of upper-triangular part of  $P$  i.e.  $\text{svec}(P) = [p_{11}, p_{12}, \dots, p_{1n}, \dots, p_{nn}]^T$ . The vectorization of the dot product  $\langle x, x^T \rangle$ , where  $x \in \mathbb{R}^n$ , is  $\text{vecv}(x) := [x_1^2, \dots, x_1 x_n, x_2 x_1, x_2^2, x_2 x_3, \dots, x_n^2]^T$ . The inverse of  $\text{vec}(x)$  and  $\text{svec}(y)$  are respectively the full and symmetric matricizations:  $\text{mat}_{m \times n}(x) = (\text{vec}(I_n)^T \otimes I_m)(I_n \otimes x)$ , and  $\text{smat}_m(P)$  so that  $\text{smat}(\text{svec}(p)) = P$ . Here,  $x \in \mathbb{R}^{mn}$  and  $y \in \mathbb{R}^{m(m+1)/2}$  for  $n, m \in \mathbb{R}_{\geq 0}$ . Finally, we denote by  $T_{\text{vec}}(A)$  the vectorization of  $A^T$  i.e.  $\text{vec}(A^T) = T_{\text{vec}}(\text{vec}(A))$ .

## 2.2 System Description

Consider the following nonlinear system

$$\dot{x}(t) = f(x; t) + g(x)u(t) + h(x)w(t), \quad x(0) = x_0 \quad (1a)$$

$$z(t) = \mathcal{G}(x, u; t), \quad z(0) = z_0 \quad (1b)$$

where  $f(\cdot), g(\cdot), h(\cdot)$ , and  $\mathcal{G}(\cdot)$  are nonlinear functions with appropriate dimensions. The state process is  $x \in \mathbb{R}^n$ , the controlled output process is  $z \in \mathbb{R}^m$ , the control input is  $u \in \mathcal{U} \subseteq \mathbb{R}^p$ , and the vector-valued (stochastic) Wiener process is  $w \in \mathcal{W} \subseteq \mathbb{R}^q$ . Let the following finite-dimensional linear time-invariant (FDLTI) system describe the resulting linearized stochastic differential equation

$$dx(t) = Ax(t)dt + B_1u(t)dt + B_2dw(t), \quad x(0) = x_0 \quad (2a)$$

$$z(t) = Cx(t) + Du(t), \quad z(0) = 0, \quad (2b)$$

where  $dw$  is the Gaussian white noise;  $x(0)$  is an arbitrary zero-mean Gaussian random vector independent of  $w(t)$ ; and  $A, B_1, B_2, C, D$  are real matrix-valued functions of

appropriate dimensions. The random signal,  $x(0)$ , and process  $w(t)$  are defined over a complete probability space  $(\Omega, \mathcal{F}, \mathcal{P})$  where  $\Omega$  is  $w$ 's sample space,  $\mathcal{F}$  is the  $\sigma$ -algebra i.e. the filtration generated by  $w$ , and  $\mathcal{P}$  is the probability measure on which  $w(t)$  is drawn for a  $t \in [0, T]$  (where  $T > 0$  is fixed).

**Assumption 1.** We impose the following conditions on the algorithm to be presented. We take  $C^T C \triangleq Q \succ 0$ ,  $D^T(C, D) = (0, R)$  for some  $R \succ 0$ ; and should one wish that the noise process in (2) be statistically independent, then we may take  $B_2 B_2^T = 0$ . Seeing we are seeking a linear feedback controller for (2), we require that the pair  $(A, B_1)$  be *stabilizable*. We expect to compute solutions via an optimization process, therefore we require that *unstable modes of  $A$  must be observable through  $Q$* . Whence  $(\sqrt{Q}, A)$  must be *detectable*.

**Problem 1.** (Problem Statement). *The goal is to keep the controlled process,  $z$ , small in an infinite-horizon LTI constrained optimization setting under a minimizing control  $u$  in spite of unforeseen disturbances  $w$ .*

Let the closed-loop operator (under an arbitrary negative feedback gain  $K \in \mathcal{K}, u(x(t)) = -Kx(t)$ ) mapping  $w$  to  $z$  be  $\|T_{zw}(K)\|_2$ . Then,

$$T_{zw}(K) = (C - DK)(sI - A + B_1 K)^{-1} B_2. \quad (3)$$

Or in (Zhou and Doyle, 1998)'s packed representation,

$$T_{zw}(K) \triangleq \left[ \begin{array}{c|c} A - B_1 K & B_2 \\ \hline C - DK & 0 \end{array} \right]. \quad (4)$$

Design principles in linear control theory exist for solving problem (2) when the covariance of the noise model has a small magnitude. For stochastic  $\mathcal{H}_2$  control problems with large noise intensities (such as PG methods), it suffices to solve a *linear exponential quadratic control problem* under a robustness constraint. To further contextualize the problem, let us formally introduce the design problem.

## 2.3 Risk-Sensitive LEQG as a Mixed Design Problem

In (Zhang et al., 2020), the authors established that the risk-sensitive infinite-horizon linear exponential quadratic Gaussian (LEQG) state-feedback control problem (Jacobson, 1973; Whittle, 1981) is an equivalent mixed- $\mathcal{H}_2/\mathcal{H}_\infty$  control design problem for *linear time-invariant* systems with additive noise of the form (2). We iterate upon this contribution since it introduces a measure of risk-design as an implicit robustness metric when the process noise has a large covariance intensity. And this is typical for policy gradient settings. The state evolves according to (2a) and without loss of generality, the stochastic linear system's performance criterion is

$$\mathcal{J}(K) = \limsup_{t_f \rightarrow \infty} \frac{2\gamma^2}{t_f} \log \mathbb{E} \exp \left[ \frac{1}{2\gamma^2} \int_{t=0}^{t_f} \langle z(t), z(t) \rangle dt \right]. \quad (5)$$

Suppose that the variance term  $\gamma^{-2} \text{var}(z^T z)$  is small, then  $\gamma$  is a measure of *risk-propensity* if  $\gamma > 0$ ; similarly,  $\gamma$  can be considered as a measure of *risk-aversion* if  $\gamma < 0$ ; and  $\gamma$  is a measure of *risk-neutrality* if  $\gamma = 0$  (equivalent to the standard state-feedback LQP). Given LEQG's connection under risk-propensity to the high-variance associated with

PG algorithms, throughout the rest of this paper we take  $\gamma > 0$  in our optimization process.

## 2.4 Mixed $\mathcal{H}_2/\mathcal{H}_\infty$ -Policy Optimization Synthesis

We now define the standard mixed  $\mathcal{H}_2/\mathcal{H}_\infty$  control problem: given system (2) and a real number  $\gamma > 0$ , find an *admissible controller*  $K$  that exponentially<sup>1</sup> stabilizes (3) and renders  $\|T_{zw}\|_\infty < \gamma$ . The set of all *suboptimal* controllers that robustly stabilizes (2) against all (finite gain) stable perturbations  $\Delta$ , interconnected to the system by  $w = \Delta z$ , such that  $\|\Delta\|_\infty \leq 1/\gamma$  can be succinctly denoted as

$$\mathcal{K} = \{K : \lambda_i(A - B_1K) < 0, \|T_{zw}(K)\|_\infty < \gamma\} \quad (6)$$

for  $i = 1, \dots, n$ . We say  $\mathcal{K} \neq \emptyset$  if the pair  $(A, B_1)$  is stabilizable and  $(C, A)$  is detectable c.f. (2).

Aside from the constraint (6), the mixed  $\mathcal{H}_2/\mathcal{H}_\infty$  performance measure can be framed as minimizing an “upper-bound” on the  $\mathcal{H}_2$ -norm of the cost subject to the constraint  $\|T_{zw}\|_\infty < \gamma$  (Bernstein and Haddad, 1989) for a  $\gamma > 0$ . Abusing notation, let  $\mathcal{J}(T_{zw})$  denote the (closed-loop) mixed  $\mathcal{H}_2/\mathcal{H}_\infty$ -control performance measure for the LTI system (2).

**Lemma 1.** The problem (2) with (a slightly abused) quadratic performance measure (5) i.e.

$$\mathcal{J}(T_{zw}) = \mathbb{E} \left\{ \limsup_{t_f \rightarrow \infty} \left[ \exp \left( 2\gamma^{-2} \int_{t=0}^{t_f} \langle z(t), z(t) \rangle dt \right) \right] \right\} \quad (7)$$

admits a unique solution  $x(t)$  to (2) after optimizing  $\min_{K \in \mathcal{K}} \mathcal{J}(\cdot)$  under the *unique and optimal controller*,

$$u^*(x(t)) = -R^{-1}B_1^T P(t)x(t), \quad t \in [0, t_f]. \quad (8)$$

In (8),  $P(t)$  is the unique, symmetric positive solution to the continuous-time (closed-loop) generalized algebraic Riccati equation (GARE)

$$PA + A^T P - P(B_1 R^{-1} B_1^T - \gamma^{-2} B_2 B_2^T) P + Q = 0 \quad (9)$$

if  $Q \succ 0$  for a  $\gamma > 0$ .

**Proof 1.** This Lemma is the infinite-horizon retrofitting of Duncan’s solution to the LEQG control value function based on a standard completion of squares and a Radon-Nikodym derivative (Duncan, 2013, Th II.1).

**Corollary 1.** (Th 9.7 Başar (2008)). The GARE (9) in an infinite-horizon LTI setting admits an equivalent *LQ two-player zero-sum differential game* with the following *upper value*

$$\mathcal{J}(T_{zw}) = \limsup_{t_f \rightarrow \infty} \inf_{u \in \mathcal{U}} \sup_{w \in \mathcal{W}} \int_{t=0}^{t_f} [x^T(t) Q x(t) + u(\cdot)^T R u(\cdot) - \gamma^{-2} w^T(t) w(t)] dt, \quad \forall x \in \mathbb{R}^n \quad (10)$$

subject to assumption 1 (Başar, 2008, §. 9.7). Note that  $\gamma > 0$  can be interpreted as an upper bound on the  $L_2$  gain disturbance attenuation or the  $\mathcal{H}_\infty$ -norm of the system. In addition, let a finite scalar  $\gamma^\infty > 0$  exist, then for all  $\Gamma \triangleq \inf\{\gamma > \gamma^\infty\}$ , (9) has a unique, finite, and positive definite solution if  $(C, A)$  is observable.

<sup>1</sup> Concerning matters relating to linear systems, we take exponential stability to mean internal stability, so that the transfer matrix belongs in the real-rational  $\mathcal{H}_\infty$  space i.e.  $T_{zw} \in \mathcal{RH}_\infty$ .

**Corollary 2.** (Th 4.8, (Başar, 2008)). If  $\Gamma \neq \emptyset$ , and if the LQ zero-sum differential game has a closed-loop perfect-state information structure defined on  $[0, t_f]$ ,  $t_f \rightarrow \infty$ , then (10) admits a unique solution with feedback controls

$$u^*(t) = -R^{-1}B_1^T P_\gamma x(t), \quad w^*(t) = \gamma^{-2} B_2^T P_\gamma x(t) \quad (11)$$

for a  $t \geq 0, \gamma > \gamma^\infty$ . Note that  $P_\gamma$  is the unique solution to (9) in the class of positive definite feedback matrices (where the subscript  $\gamma$  on  $P$  denotes its direct dependence on  $\gamma$ ) which makes the following feedback matrix Hurwitz,

$$A_\gamma = A - (B_1 R^{-1} B_1^T - \gamma^{-2} B_2 B_2^T) P_\gamma. \quad (12)$$

**Remark 1.** Clearly, (10) is a minimax problem whose controller admits the form

$$\min_{u \in \mathcal{U}} \max_{w \in \mathcal{W}} \mathcal{J}(T_{zw}) = \limsup_{t_f \rightarrow \infty} \int_{t=0}^{t_f} [x^T(t) Q x(t) + u^T(\cdot) R u(\cdot) - \gamma^{-2} w^T(t) w(t)] dt, \quad \forall x \in \mathbb{R}^n. \quad (13)$$

Another common form of  $\mathcal{J}(T_{zw})$  easily amenable to policy gradient algorithms is  $J(T_{zw}) = \text{Tr}(P_\gamma B_2 B_2^T)$  (Mustafa, 1989).

**Remark 2.** The cost (13) is nonconvex and not coercive (Zhang et al., 2019). However, our iterative solver (Cui and Molu, 2023) guarantees uniform linear convergence of the iterates during optimization.

**Remark 3.** The objective (13) is differentiable for any  $K \in \mathcal{K}$ , and its policy gradient  $\nabla \mathcal{J}(T_{zw}) := 2(RK - B^T P_\gamma) \Lambda_\gamma$ ; here,  $\Lambda_\gamma$  admits a form amenable to a (continuous time) closed-loop Lyapunov equation (Zhang et al., 2019, Lemma A.4) i.e.,

$$\Lambda_\gamma(A - B_1 K + \gamma^{-2} B_2 B_2^T P_\gamma)^T + (A - B_1 K + \gamma^{-2} B_2 B_2^T P_\gamma) \Lambda_\gamma + B_2 B_2^T = 0. \quad (14)$$

For  $\gamma > 0$  and  $\gamma \neq \sigma_i(B_2), i = 1, \dots, n$ , we define the following closed-loop Hamiltonian matrix for a  $(\gamma, K)$  pair

$$H(\gamma, K) = \begin{bmatrix} A - B_1 K & -\gamma^{-1} B_2 B_2^T \\ -\gamma^{-1} (C^T C + K^T R K) & -(A - B_1 K)^T \end{bmatrix} \quad (15)$$

where we have used  $R = (D^T D - \gamma^2 I)$  and  $S = (D D^T - \gamma^2 I)$  as in (Bruinsma and Steinbuch, 1990, Eq. 2.2).

## 3. METHODS

We now introduce a nonlinear identification procedure, followed by linearization, and closed-loop  $\mathcal{H}_\infty$  parameter search schemes. We close the section with an iterative solver for the GARE (9).

### 3.1 Nonlinear Identification and Linearization

We remark that the user is not limited to the method to be introduced but in our experience, our identification scheme is interpretable and useful for debugging real-world and physical systems. We use the parsimonious **Nonlinear Auto-Regressive Moving Average with exogenous input (NARMAX)** (Chen et al., 1989). which has powerful yet simple parsimonious representation capability on real systems. We first identified a suitable NARMAX structure and model parameters, compute equilibrium points – about which we linearized to a form of

(2), before we estimate the *robustly stabilizing and optimal control policy* for the mixed  $\mathcal{H}_2/\mathcal{H}_\infty$ -control problem.

Suppose that an input-output data from a real system defined by (1) has been collected. Denote this as  $D^N = \{z_1, \dots, z_m, u_1, \dots, u_p, w_1, \dots, w_q\}$ . Let the maximum lags in the input, disturbance, and output data be denoted by  $n_u$ ,  $n_w$ , and  $n_y$  respectively. We fit a polynomial NARMAX model to  $D^N$  with the power-form  $\ell$ -degree polynomial,

$$z(t) = \theta_0 + \sum_{i_1=1}^n \theta_{i_1} x_{i_1}(t) + \sum_{i_1=1}^n \sum_{i_2=i_1}^n \theta_{i_1} \theta_{i_2} x_{i_1}(t) u_{i_2}(t) \cdots + \sum_{i_1=1}^n \cdots \sum_{i_\ell=i_{\ell-1}}^n \theta_{i_1} \cdots \theta_{i_\ell} x_{i_1}(t) \cdots x_{i_\ell}(t) + e(t) \quad (16)$$

whose parameters  $\theta_{i_1 \dots i_m}, m \in [1, \ell]$  are to be identified. The model structure has order  $n = n_z + n_u + n_w + n_e$ , where  $n_e$  is the maximum order for a *pseudo-random binary sequence*  $e(k)$  that aids identification robustness. The state variables are explicitly,

$$x_m(k) = \begin{cases} z(k-m), & 1 \leq m \leq n_z \\ u(k-(m-n_z)), & n_z+1 \leq m \leq n_z+n_u \\ w(k-(m-n_z-n_u)), & n_z+n_u+1 \leq m \leq n_z+n_u+n_w \\ e(k-(m-n+n_e)), & n-n_e+1 \leq m \leq n. \end{cases} \quad (17)$$

Equation (16) admits a linear regression model of the form  $z(t) = \sum_{i=1}^M \phi_i(t) \theta_i + e(t)$  for  $t = 1, \dots, N$  and a process noise  $e(t)$ . Or in matrix form:  $Z = \Phi \Theta + \Xi$ , where  $\Phi = [\phi_1 \cdots \phi_M]$  denotes the regression matrix, and  $\Theta = [\theta_1, \dots, \theta_M]$  are parameters to be learned, typically in a regression process. The solution to the least squares cost  $\min_{\Theta} \|z - \Phi \Theta\|_2$  yields the parameter estimates  $\Theta$  for the nonlinear model.

We adopt the computationally efficient Householder transformation in transforming the information matrix,  $\Phi^T \Phi$  into a well-conditioned QZ-matrix partition. Afterwards, we recover NARMAX parameters by solving the resulting triangular system of linear equations in a least squares sense.

Given that the nonlinear structure is unknown ahead of time, we start with large values of  $n_z, n_u$ , and  $n_w$  in  $P$  – adding regression variables that capture natural properties such as damping and friction e.t.c. in order to capture as many nonlinear variation that exist in the data as possible. We then iteratively pruned the parameters using the *error reduction ratio* algorithm (Billings, 2013) within the forward orthogonal regression least square algorithm (Chen et al., 1989).

### 3.2 NARMAX Linearization

The conditions stipulated in Assumption 1 must be realized before we can implement a learning-based procedure. The identified NARMAX model is then linearized about a suitable equilibrium point to obtain a form of (2) in state space form. In our experience, we have always found Assumption 1 to be satisfied after linearization. Suppose that the pair  $(A, B)$  is still not controllable after linearization (we have not seen this in practice), a perturbation can be made of the multi-input LTI system as follows: Suppose

that there exists a diagonalizable matrix  $M$  such that  $\bar{A} = M^T A M$ , and  $\bar{B} = M^T B$ . Then  $(A, B)$  can be reduced to  $(\bar{A}, \bar{B})$  as follows:

$$\bar{A} = \begin{bmatrix} A_{cnt} & * \\ 0 & \bar{A}_{cnt} \end{bmatrix}, \quad \bar{B} = [B_{cnt} \cdots 0]^T \quad (18)$$

for<sup>2</sup>

$$A_{cnt} = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1,p-1} & A_{1p} \\ A_{21} & A_{22} & \cdots & A_{2,p-1} & A_{2p} \\ 0 & A_{32} & \cdots & A_{3,p-1} & A_{3p} \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & \cdots & & A_{pp} \end{bmatrix}, \quad B_{cnt} = \begin{bmatrix} B_1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (19)$$

where  $p$  is a pair's controllability index, blocks  $B_1, A_{21}, \dots, A_{p,p-1}$  possess full row ranks, and  $\dim(\bar{A}_{cnt}) = \dim[(\bar{A}, \bar{B})_{cnt}]$ .

### 3.3 LEQG/LQ Differential Game

In (Cui and Molu, 2023, Algorithm II), we introduced an iterative solver for the closed-loop controls  $u$  and  $w$  respectively. A key drawback is the need for the first  $K_1 \in \mathcal{K}$  to be known. This limits the practicality of the algorithm to data-driven PO schemes. In addition, our examples did not illustrate a means of respecting the *stabilizability* and *detectability* assumptions needed to guarantee a solution to the minimax problem (13). We now provide an all-encompassing learning scheme for obtaining the solution to the mixed  $\mathcal{H}_2/\mathcal{H}_\infty$ -control problem in a purely data-driven setting compatible with modern model-free policy optimization schemes.

Let  $p \in \mathcal{P}$  and  $q \in \mathcal{Q}$  denote the iteration indices at which the controllers  $u_p(t) = K_p x(t)$  and  $w_q(t) = L_p^q x(t)$  are updated in (11), where  $K_p$  and  $L_p^q$  are feedback gains

$$K_p = -R^{-1} B_1^T P_p^q, \quad L_p^q = \gamma^{-2} B_2^T P_p^q. \quad (20)$$

Furthermore, let the closed-loop transition matrix under the gains of (20), and quadratic matrix term in (13) be (see (Cui and Molu, 2023, Equation 12))

$$A_\gamma = A - B_1 K_p + B_2 L_p^q, \quad Q_\gamma = C^T C + K_p^T R K_p. \quad (21)$$

**Observe:**  $P_p^q$  is finite if and only if the closed-loop system matrix  $A_\gamma$  possesses eigenvalues with negative real parts. In this case,  $P_p^q$ , for  $p, q = 0, 1, 2, \dots$  is the unique positive definite solution to the closed-loop GARE

$$A_\gamma^T P_p^q + P_p^q A_\gamma + Q_\gamma - \gamma^{-2} L_p^{qT} L_p^q = 0 \quad (22)$$

where recursively, (20) holds for  $p, q = 1, 2, \dots$ . Note that  $K_1$  must be chosen such that  $A_\gamma^1 = A - B_1 K_1 + B_2^T L_1^1$  is Hurwitz and its closed-loop  $\mathcal{H}_\infty$  norm transfer function is bounded from above by a user-defined  $\gamma > 0$  (Kleinman, 1968). Then (i)  $K_1 \leq P_{p+1}^q \leq P_p^q \leq \dots$ ,  $p, q = 1, 2$ , (ii)  $\lim_{(p,q) \rightarrow \infty} P_p^q = P$ . See proof in (Cui and Molu, 2023).

### 3.4 Mixed Sensitivity Initialization

In (Cui and Molu, 2023), we had established that in order for our model-free algorithm to work in a purely data-driven setting,  $K_1$  must be in the constraint set  $\mathcal{K}$ . The means for finding a  $K_1$  that satisfies the constraints equation (6) is itemized in algorithm 1.

<sup>2</sup>  $\bar{X}_{cnt}$  signifies the non-controllable part of  $X$ .

**Algorithm 1** Search for the closed-loop  $\mathcal{H}_\infty$ -norm

---

```

1: Given a user-defined step size  $\eta > 0$ 
2: Set the initial upper bound on  $\gamma$  as  $\gamma_{ub} = \infty$ .
3: Initialize a buffer for possible  $\mathcal{H}_\infty$  norms for each  $K_1$ 
   to be found,  $\Gamma_{buf} = \{\}$ .
4: Initialize ordered poles  $\mathcal{P} = \{p_i \in \text{Re}(s) < 0 \mid i =$ 
    $1, 2, \dots\} \triangleright p_1 < p_2 < \dots$ 
5: for  $p_i \in \mathcal{P}$  do
6:   Place  $p_i$  on (2);  $\triangleright$  (Tits and Yang, 1996)
7:   Compute stabilizing  $K_1^{p_i}$ 
8:   Find lower bound  $\gamma_{lb}$  for  $H(\gamma, K_1^{p_i})$ ;  $\triangleright$  using (23)
9:    $\Gamma_{buf}(i) = \text{get\_hinf\_norm}(T_{zw}, \gamma_{lb}, K_1^{p_i})$ .
10: end for
11: function  $\text{get\_hinf\_norm}(T_{zw}, \gamma_{lb}, K_1^{p_i})$ 
12:   while  $\gamma_{ub} = \infty$  do
13:      $\gamma := (1 + 2\eta) \gamma_{lb}$ ;
14:     Get  $\lambda_i(H(\gamma, K_1^{p_i}))$   $\triangleright$  c.f. (15)
15:     if  $\text{Re}(\Lambda) \neq \emptyset$  for  $\Lambda = \{\lambda_1, \dots, \lambda_n\}$  then
16:       Set  $\gamma_{ub} = \gamma$ ; exit
17:     else
18:       Set buffer  $\Gamma_{lb} = \{\}$ 
19:       for  $\lambda_k \in \{\text{Imag}(\Lambda)_{:p-1}\}$  do  $\triangleright k = 1$  to  $K$ 
20:         Set  $m_k = \frac{1}{2}(\omega_k + \omega_{k+1})$ 
21:         Set  $\Gamma_{lb}(k) = \max\{\sigma[T_{zw}(jm_k)]\}$ ;
22:       end for
23:        $\gamma_{lb} = \max(\Gamma_{lb})$ 
24:     end if
25:     Set  $\gamma_{ub} = \frac{1}{2}(\gamma_{lb} + \gamma_{ub})$ .
26:   end while
27:   return  $\gamma_{ub}$ 
28: end function

```

---

Following Corollary 2, we must first find the upper bound of  $\gamma$  i.e.  $\gamma^\infty := \gamma_{ub}$  whereupon the unique, finite, and positive definite solution to the GARE is satisfied. Let us now introduce the following proposition.

**Proposition 1.** (Bruinsma and Steinbuch (1990)). For all  $\omega_p \in \mathbb{R}$ , we have that  $j\omega_p$  is an eigenvalue of the Hamiltonian  $H(\gamma_1)$  if and only if  $\gamma_1$  is a singular value of  $T_{zw}(j\omega_p)$ .

**Remark 4.** Singular value computation is easily obtainable given a frequency of a system. Proposition 1 allows us to obtain all frequencies that correspond to a single eigenvalue.

**Procedure:** We search for stabilizing gains  $K_1^{p_i}$  over the space of negative reals (or range matrix inequalities for multiple input systems) such that each  $K_1^{p_i}$  on line 7 of Algorithm 1 is stabilizing. A starting lower bound for each  $\gamma_i$  corresponding to gain  $K_1^{p_i}$  can be set

$$\gamma_{lb} := \max\{\sigma_{\max}(G(0)), \sigma_{\max}(G(j\omega_p)), \sigma_{\max}(B_2)\} \quad (23)$$

where  $\omega_p$  is chosen as specified in (Bruinsma and Steinbuch, 1990, Eq. (4.4)). The  $\mathcal{H}_\infty$  computation scheme is fast and has a guaranteed quadratic convergence (since  $\gamma_{lb}(i+1) > \gamma_{lb}(i)$ ) if the poles are chosen to make  $(A - B_1K)$  Hurwitz. Given a user-defined step size,  $\eta$ , the rest of the algorithm consists in iteratively increasing the value of  $\gamma_{lb}$  until all eigenvalues of the closed-loop system Hamiltonian c.f. (15) have no imaginary part.

The  $\mathcal{H}_\infty$  norm computation scheme is based on the singular values of (3) and the eigenvalues of (15). For poles far to the left of the origin,  $\gamma$  will be small. However, as

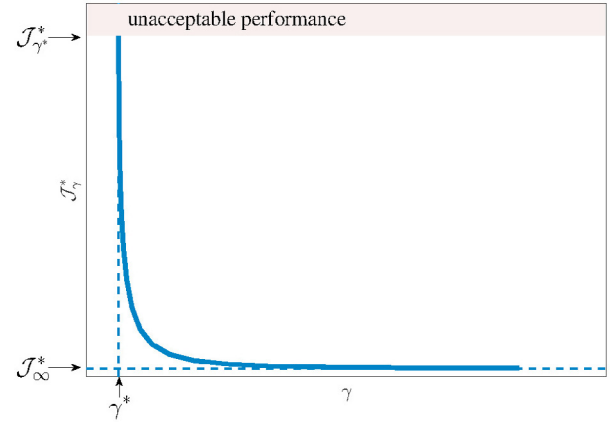


Fig. 1. Performance Robustness Trade-off Curve.

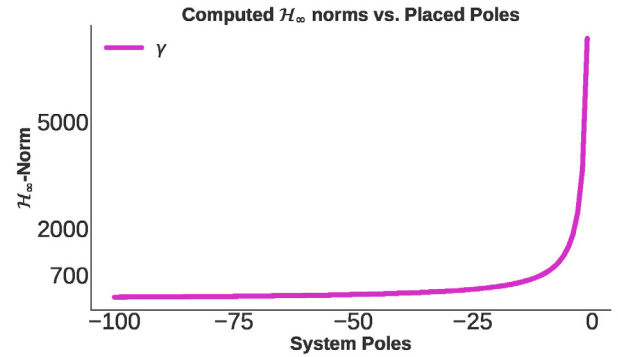


Fig. 2. Computed  $\mathcal{H}_\infty$ -norm for searched poles. We see that the closer to the origin, the greater the value of  $\|T_{zw}\|_{\mathcal{H}_\infty}$

$\lambda \rightarrow 0, \gamma \rightarrow \infty$ . Thus, a critical value of  $\gamma^*$  can be obtained (see Fig. 2) above which the system becomes unstable (the cost becomes infinite; c.f. (Ogunmolu et al., 2018, Fig. 1)). We employ this heuristic to choose a  $K_1$  that satisfies constraint (6). This algorithm 1 finds the system's  $\mathcal{H}_\infty$  norm; part of it is an adaptation of (Bruinsma and Steinbuch, 1990)'s fast  $\mathcal{H}_\infty$  computation algorithm which enjoys quadratic convergence as opposed to the popular bisection algorithm (Zhou and Doyle, 1998). On line 6, we used (Tits and Yang, 1996)'s globally fast and convergent pole placement algorithm and leveraged the implementation provided in the `scipy` library.

### 3.5 Iterative Two-Player LQ Zero-Sum Game

We now analyze the dynamic game. Putting (21) into (22), we have

$$(A^T P_p^q + P_p^q A) + (Q_p - \gamma^{-2} L_p^{qT} L_p^q) + L_p^{qT} B_2^T P_p^q - K_p^T B_1^T P_p^q - P_p^q B_1 K_p + P_p^q B_2 L_p^q = 0, \quad (24)$$

which in vector form can be written as

$$\begin{aligned} & \text{svec}(A^T P_p^q + P_p^q A) + \text{svec}(Q_p - \gamma^{-2} L_p^{qT} L_p^q) \\ & - \text{smat}[(I_n \otimes K_p^T) + (K_p^T \otimes I_n) T_{\text{vec}}] \text{vec}(B_1^T P_p^q) \\ & + \text{smat}[I_n \otimes L_p^{qT} B_2^T + (L_p^q B_2^T \otimes I_n)] \text{mat}(\text{svec}(P_p^q)) \end{aligned} \quad (25)$$

where  $p, q$  are iteration indices for the controller  $u$  and disturbance  $w$  respectively (introduced formally in Algorithm 2) and  $n, m$  are as defined in §2.2. We know that the differential game (13) admits equal upper and lower

optimal values owing to the GARE (10) having a positive definite solution i.e.  $\mathcal{J}^* = x^T P x$  (Başar, 2008, Th 4.8 (iii)). Control laws must therefore be computed along the trajectories of (2), using the derivative of  $\mathcal{J}^* = x^T P x$ . At the iteration pair  $(p, q)$ ,  $d(\mathcal{J}^*(x; t))$  admits the solution (by Itô's differential rule)

$$d(x^T P_p^q x) = x^T (A^T P_p^q + P_p^q A_p^q) x dt + 2x^T P_p^q B_1 u_p dt + 2x^T P_p^q B_2 dw + \text{Tr}(B_2^T P_p^q B_2) dt \quad (26)$$

where  $\text{Tr}(M)$  denotes the trace of  $M$ .

Letting  $\phi(t) = [\text{vec}^T(x), 2(x^T \otimes u^T), 1]^T$ , and integrating the above on the interval  $[0, t_f]$ , we find that

$$\underbrace{\frac{1}{t_f} \int_0^{t_f} \phi d(\text{vec}^T(x)) \text{svec}(P_p^q)}_{\hat{\Psi}(t_f)} = \begin{bmatrix} \text{svec}(A^T P_p^q + P_p^q A) \\ \text{vec}(B_1^T P_p^q) \\ \text{Tr}(B_2^T P_p^q B_2) \end{bmatrix} \times \underbrace{\frac{1}{t_f} \int_0^{t_f} \phi \phi^T dt}_{\hat{\Phi}(t_f)} + \frac{1}{t_f} \int_0^{t_f} 2\phi x^T P_p^q B_2 dw \quad (27)$$

where the last term tends to zero as  $t_f \rightarrow \infty$ . We now recall Lemmas A.7 and A.8 in (Cui and Molu, 2023), so that the following holds almost surely:  $\lim_{t_f \rightarrow \infty} \hat{\Phi}(t_f) = \Phi \equiv \mathbb{E}(\phi \phi^T)$ , and  $\Psi = \lim_{t_f \rightarrow \infty} \hat{\Psi}(t_f)$ . Hence,

$$\begin{bmatrix} \text{svec}(A^T P_p^q + P_p^q A) \\ \text{vec}(B_1^T P_p^q) \\ \text{Tr}(B_2^T P_p^q B_2) \end{bmatrix} = \Phi^{-1}(t_f) \Psi(t_f) \text{svec}(P_p^q). \quad (28)$$

Furthermore, let  $n_1 := n(n+1)/2$  and  $n_2 := n_1 + mn$ . Then, we may write

$$\text{svec}(A^T P_p^q + P_p^q A) = [\Phi^{-1}]_{[1:n_1]} \Psi \text{svec}(P_p^q) \quad (29a)$$

$$\text{vec}(B_1^T P_p^q) = [\Phi^{-1}]_{[1+n_1:n_2]} \Psi \text{svec}(P_p^q). \quad (29b)$$

Define  $\hat{\Phi}_1 = [\Phi^{-1}]_{[1:n_1]}$ ,  $\hat{\Phi}_2 = [\Phi^{-1}]_{[1+n_1:n_2]}$ , so that (25) in light of (29) becomes

$$\begin{aligned} & \hat{\Phi}_1 \Psi \text{svec}(P_p^q) + \text{svec}(Q_p - \gamma^{-2} L_p^{qT} L_p^q) \\ & - \text{smat} [(I_n \otimes K_p^T) + (K_p^T \otimes I_n) T_{\text{vec}}] \hat{\Phi}_2 \Psi \text{svec}(P_p^q) \\ & + \text{smat} [I_n \otimes L_p^{qT} B_2^T + (L_p^q B_2^T \otimes I_n)] \text{mat}(\text{svec}(P_p^q)) = 0 \end{aligned} \quad (30)$$

Rearranging the above and letting

$$\begin{aligned} \Upsilon_p^q &= \hat{\Phi}_1 \Psi - \text{smat} [(I_n \otimes K_p^T) + (K_p^T \otimes I_n) T_{\text{vec}}] \hat{\Phi}_2 \Psi \\ &+ \text{smat} [I_n \otimes L_p^{qT} B_2^T + (L_p^q B_2^T \otimes I_n)] \text{mat} \end{aligned} \quad (31)$$

it can be verified that the cost matrix  $P_p^q$  (c.f. (11)) admits the solution

$$\text{svec}(P_p^q) = (\Upsilon_p^q)^{-1} \text{svec}(Q_p - \gamma^{-2} L_p^{qT} L_p^q). \quad (32)$$

The entire procedure for updating the control laws in an iterative manner is described in Algorithm 2.

We note in passing that inversion of matrix terms are efficiently computed using standard Cholesky factorizations. We refer interested readers to (Cui and Molu, 2023) for the convergence and robustness analyses of our results.

## Algorithm 2 Mixed $\mathcal{H}_2/\mathcal{H}_\infty$ -Control Synthesis

- 1: Collect I/O data, and identify the nonlinear model (1) following §3.1.
- 2: Obtain  $Z^L = \{A, B_1, B_2, C, D\}$  by linearizing the NARMAX model in step 1 ▷ See §3.1
- 3: Form matrices  $R := D^T D \succ 0$ ,  $Q := C^T C$
- 4: Using  $Z^L$ , find  $\hat{K}_1 \in \mathcal{K}$  ▷ alg. 1; Pick a suitable  $\gamma$
- 5: Run (2)'s time response with  $K_1 \in \mathcal{K}$  on  $Z^L$ .
- 6: Form state-control data  $(\hat{\Phi}_1, \hat{\Phi}_2, \hat{\Psi})$  ▷ Eq. (31);
- 7:  $\hat{K}_p^q, \hat{L}_p^q, \hat{P}_p^q = \text{robust\_gains}(\hat{\Phi}_1, \hat{\Phi}_2, \hat{\Psi}, x(t), \bar{p}, \bar{q}, \gamma)$
- 8: **for**  $t = 1, \dots, t_f$  **do**
- 9:   Apply  $u(x) = \hat{K}_p^q x(t)$ ,  $w(x) = \hat{L}_p^q x(t)$  to eq. (2)
- 10: **end for**
- 11: **function** `robust_gains`( $\hat{\Phi}_1, \hat{\Phi}_2, \hat{\Psi}, \bar{p}, \bar{q}, \gamma$ )
- 12:   Initialize  $q = 1$
- 13:   Initialize  $L_1^1$  ▷ Set to zeros or randomly initialize
- 14:   **for**  $p \in 1$  to  $\bar{p}$  **do**
- 15:     **while**  $q \leq \bar{q}$  **do**
- 16:       Find in order:  $\hat{\Upsilon}_p^q, \hat{P}_p^q$  ▷ Eq. (31) & (32);
- 17:       Compute  $\hat{L}_p^q \leftarrow \gamma^{-2} B_2^T \hat{P}_p^q$  ▷ Eq. (11)
- 18:       Update  $q \leftarrow q + 1$
- 19:     **end while**
- 20:     Form `smat`(`vec`( $\widehat{B_1^T P_p^q}$ )) ▷ Eq. (29b)
- 21:     Compute  $\hat{K}_{p+1}^q \leftarrow R^{-1} \widehat{B_1^T P_p^q}$  ▷ Feedback gain;
- 22:   **end for**
- 23:   **return**  $\hat{K}_p^q, \hat{L}_p^q, \hat{P}_p^q$
- 24: **end function**

## 4. RESULTS

We now present numerical results of the algorithm described in the foregoing.

### 4.1 Car Cruise Control System

We consider a car *cruise control* system (Åström and Murray, 2021, §3.1) whereupon a controller  $u(x(t)) = [u_1(t), u_2(t)]$  must maintain a constant velocity  $v$  (the state), whilst automatically adjusting the car's throttle,  $u_1(t), t \in [0, T]$  despite disturbances characterized by road slope changes ( $u_3 = \theta$ ), rolling friction ( $F_r$ ), and aerodynamic drag forces ( $F_d$ ).

This control design problem is well-suited to our robust control formulation because (i) the disturbances and state variables are separable and can be lumped into the form of the stochastic differential equations (1) and (2); (ii) it is a multiple-input (throttle, gear, vehicle speed) single-output (vehicle acceleration) system that introduces modeling challenges; (iii) the entire operating range of the system is nonlinear though there is a reasonable linear bandwidth that characterize the input/output (I/O) system as we will see shortly. The model is

$$\begin{aligned} m \frac{dv}{dt} &= \alpha_n u \tau (\alpha_n v) - mg C_r \text{sgn}(u) \\ &- \frac{1}{2} \rho C_d A |v| v - mg \sin \theta \end{aligned} \quad (33)$$

where  $v$  is the velocity profile of the vehicle (taken as the system's state),  $m$  is vehicle's mass,  $\alpha_n$  is the inverse of the vehicle's effective wheel radius,  $\tau$  is the vehicle's torque – it is controlled by the throttle  $u := u_1$ . The rolling friction



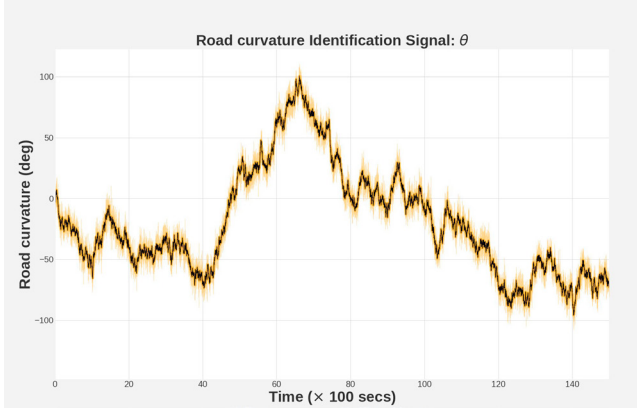


Fig. 3. Car inclination input identification signal.

coefficient is  $C_r$  and  $C_d$  is the aerodynamic drag constant for a vehicle of area  $A$ . The road curvature,  $\theta$ , is modeled as a Wiener process c.f. (2) with  $w_i(t) \sim \mathcal{N}(0, 1)$  where for  $i \in [1, \dots, ]$ ,  $dw_i = \sum_{j=1}^i w_j$ . If we let  $x := v$ ,  $u_1 := u$ ,  $u_2 := \alpha_n$ , and  $u_3 := \theta$  and set  $C_r = 0.01$ ,  $C_d = 0.32$ ,  $\rho = 1.3 \text{ kg/m}^3$ ,  $A = 2.4 \text{ m}^2$  (following (Åström and Murray, 2021)), then the torque  $\tau$  is  $\tau = \tau_m - \tau_m \beta (\omega/\omega_m - 1)^2$ , where  $\beta = 0.4$ ,  $\omega_m = 420$  and  $\tau_m = 190$ . Simplified, we write

$$\tau = 190 - 76 \left( \frac{39x}{420} - 1 \right)^2.$$

#### 4.2 Nonlinear Identification and Linearization

In our NARMAX structure selection and model estimation scheme, we first start with a large number of parameters and regressors that consists of the polynomial expansion in (16), **sinuoidal**, and **signum** functions following (33). We choose a polynomial degree of 3 and the inputs  $u$  and state lags  $x$  were chosen as  $[1, 1, 1]$  and  $[1]$  respectively. We employed the forward regression orthogonal least squares algorithm (Chen et al., 1989) in estimating the parametric terms of the model. We then employed the **error reduction ratio** (Chen et al., 1989; Billings, 2013) algorithm in pruning away extraneous terms. This whittled down the eventual model to the following parsimonious representation

$$\begin{aligned} \dot{x}(t) = & 0.062518u_2(t)x(t) - 0.12051u_1(t)u_2^2(t) \\ & + 0.00081339u_3^3(t) + 0.9767 \sin(u_3(t)). \end{aligned} \quad (34)$$

The NARMAX structure selection and model estimation step produced a **root relative test error** of 0.0661 (see Fig. 4). Using (33) with  $u_3 \sim \mathcal{N}(0, 0.05)$ , a constant gear ratio of 40 and a car mass of  $1600 \text{ kg}$ , we collect I/O data with 40,000 samples in continuous time as shown in Fig. 3. With the input-output data, a NARMAX model was identified whose prediction error with respect to held-out validation data is shown in Fig. 4.

To amend the nonlinear control problem (1) to the setup (2), we compute the values of the states, inputs, and outputs for system (16)'s equilibrium points:  $x_{eq} := v_{ref}$ ,  $u_{eq} := [u_1, u_2]$ , and  $z_{eq} \triangleq v_{ref}$  given initial values  $x(0) = 20$ ,  $u(0) = [0, 40]$ , and  $z(0) = 20$ . The resulting linearized system (2) is

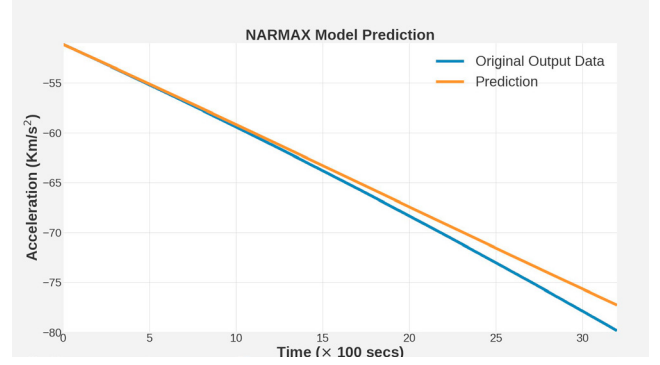


Fig. 4. Model prediction vs. actual car acceleration.

$$A = [100.0288], \quad B_1 = [-193.072, 137.3123],$$

$$B_2 = [-17014.7221, -10557.48189] \quad C = [1, 0] \quad (35)$$

and  $D$  is  $[1, 1]$ . As seen, the pairs  $(A, B_1)$  is stabilizable and  $(C, A)$  is observable – notable features of linearizing the NARMAX model in that it faithfully captures a system's parsimonious model.

#### 4.3 Efficacy of the Learning Algorithm

After running Algorithm 1, we found a  $\gamma$  of value 500 to be a suitable value for robustly compensating for a change in road slope with angle  $\theta = 40$ . The goal is to regulate the speed of the car so that despite the change in slope, a constant speed of  $40 \text{ m/s}$  is maintained. We then run Alg. 2 for  $R = I$ , solve for  $\Upsilon$  and equations (31) and (32) based on collected data on the linearized system (35). We run Alg. 2 on the collected data (see line 6 of Alg. 2). We then test the efficacy of the computed solutions to the final gains  $K_p^q$  and cost matrix  $P_p^q$  using our iterative solver (c.f. Alg. 2) against (known) computed optimal values for the cost matrix  $P^*$  and gain  $K^*$  using Duncan's Riccati equation (9) and control law in (8). We set iteration max indices to 20 and 30. The relative errors between our solver and these solutions are shown in Fig. 5. We see that both parameters converge to their optimal values in within the first two iterations; this is despite the disturbance and unknown model aforesaid.

## 5. CONCLUSION

Following up on our recent contribution (Cui and Molu, 2023), we have presented a framework for discarding the restricting assumption of *stabilizability* and *observability* in linear plants under a mixed sensitivity design framework. We first identified the nonlinear system, linearized it, found appropriate  $\mathcal{H}_\infty$  bounds for the system and then deployed our learning algorithm. We introduced a fast means for finding the linearized system's  $\mathcal{H}_\infty$ -norm; and then simplified the two-loop mixed sensitivity algorithm earlier disseminated in Cui and Molu (2023). Further numerical results on a car's cruise controller is here presented to fortify the credibility of our previous results. Some open problems, which we intend to treat in the near future include

- (1) systems with finite-escape time in the solution to differential equations (1) and (2) i.e. there exists discontinuity that incapacitates the Lipschitz continuity

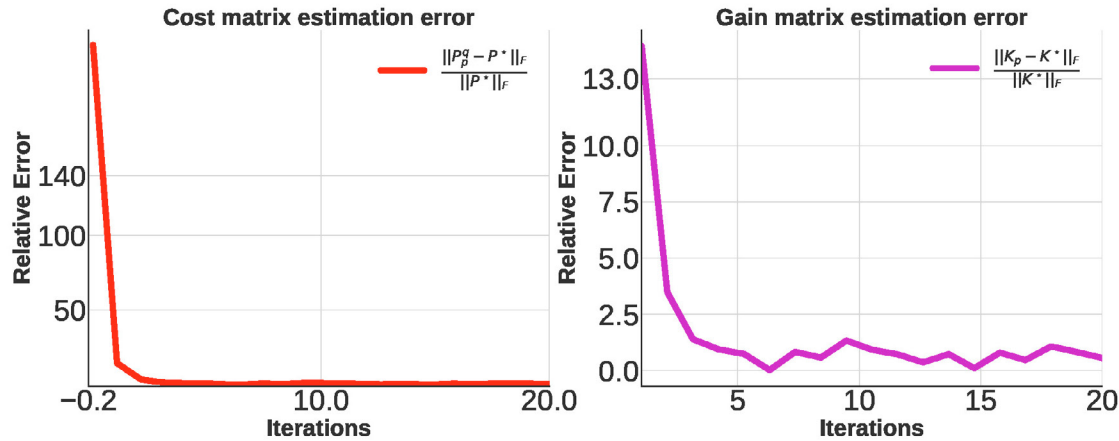


Fig. 5. Relative estimation error for the cost and gain matrices.

assumption of  $f(x, u)$  in  $x$ . Of what relevance is finite-time stability in such mixed sensitivity analyses?

- (2) multiplicative noise in the system dynamics; and
- (3) a large scale study of robustness analyses to distributed computing systems.

#### REFERENCES

- Agarwal, A., Kakade, S.M., Lee, J.D., and Mahajan, G. (2021). On the Theory of Policy Gradient Methods: Optimality, Approximation, and Distribution Shift. *J. Mach. Learn. Res.*, 22(98), 1–76.
- Åström, K.J. and Murray, R.M. (2021). *Feedback Systems: An Introduction for Scientists and Engineers*. Princeton University Press.
- Başar, T. (2008).  *$H_\infty$ -Optimal Control and Related Minimax Design Problems: A Dynamic Game Approach*. Springer.
- Bernstein, D. and Haddad, W. (1989). LQG control with an  $\mathcal{H}_\infty$  performance bound: a Riccati equation approach. *IEEE Transactions on Automatic Control*, 34(3), 293–305. doi:10.1109/9.16419.
- Billings, S. (2013). *Nonlinear System Identification: NARMAX Methods in the Time, Frequency, and Spatio-Temporal Domains*, volume 39. John Wiley & Sons, Ltd.
- Bruinsma, N. and Steinbuch, M. (1990). A Fast Algorithm to Compute the  $H_\infty$ -norm of a Transfer Function Matrix. *Systems & Control Letters*, 14, 287–293.
- Chen, S., Billings, S.A., and Luo, W. (1989). Orthogonal Least Squares Methods and Their Application to Nonlinear System Identification. *International Journal of Control*, 50(5), 1873–1896.
- Cui, L. and Molu, L. (2023). Mixed  $H_2/H_\infty$  Control for Robust Policy Optimization Under Unknown Dynamics. URL <https://arxiv.org/abs/2209.04477>.
- Duncan, T.E. (2013). Linear-Exponential-Quadratic Gaussian control. *IEEE Transactions on Automatic Control*, 58(11), 2910–2911. doi: 10.1109/TAC.2013.2257610.
- Jacobson, D. (1973). Optimal stochastic linear systems with exponential performance criteria and their relation to deterministic differential games. *IEEE Transactions on Automatic Control*, 18(2), 124–131. doi: 10.1109/TAC.1973.1100265.
- Kakade, S.M. (2001). A Natural Policy Gradient. *Advances in Neural Information Processing Systems*, 14.
- Khargonekar, P., Petersen, I., and Rotea, M. (1988).  $\mathcal{H}_\infty$  optimal control with state-feedback. *IEEE Transactions on Automatic Control*, 33(8), 786–788. doi: 10.1109/9.1301.
- Kleinman, D.Z. (1968). On an iterative technique for riccati equation computations. *IEEE Transactions on Automatic Control*, 13, 114–115.
- Mustafa, D. (1989). Relations between maximum-entropy/ $\mathcal{H}_\infty$  control and combined  $\mathcal{H}_\infty$ /LQG control. *Systems and Control Letters*, 12(3), 193–203.
- Ogunmolu, O., Gans, N., and Summers, T. (2018). Minimax iterative dynamic game: Application to nonlinear robot control tasks. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 6919–6925. IEEE.
- Tits, A. and Yang, Y. (1996). Globally convergent algorithms for robust pole assignment by state feedback. *IEEE Transactions on Automatic Control*, 41, 1432–1452.
- Whittle, P. (1981). Risk-sensitive linear/quadratic/gaussian control. *Advances in Applied Probability*, 13(4), 764–777.
- Zames, G. (1981). Feedback and optimal sensitivity: Model reference transformations, multiplicative seminorms, and approximate inverses. *IEEE Transactions on Automatic Control*, 26(2), 301–320. doi: 10.1109/TAC.1981.1102603.
- Zhang, K., Hu, B., and Başar, T. (2019). Policy Optimization for  $\mathcal{H}_2$  Linear Control with  $\mathcal{H}_\infty$  Robustness Guarantee: Implicit Regularization and Global Convergence. *arXiv e-prints*, arXiv:1910.09496.
- Zhang, K., Hu, B., and Basar, T. (2020). Policy optimization for  $\mathcal{H}_2$  linear control with  $\mathcal{H}_\infty$  robustness guarantee: Implicit regularization and global convergence. In *Learning for Dynamics and Control*, 179–190. PMLR.
- Zhang, K., Yang, Z., and Basar, T. (2019). Policy optimization provably converges to nash equilibria in zero-sum linear quadratic games. In *Advances in Neural Information Processing Systems*, volume 32.
- Zhou, K. and Doyle, J.C. (1998). *Essentials of robust control*, volume 104. Prentice Hall, Upper Saddle River, NJ.