A Data-Driven Method for Computing Fixed-Structure Low-Order Controllers With H_∞ Performance

Achille Nicoletti and Alireza Karimi

***Abstract*— Recently, a new data-driven method for robust control with $\mathcal{H}_\infty$ performance has been proposed. This method is based on convex optimization and converges to the optimal performance when the controller order increases. However, for low-order controllers, the performance depends heavily on the choice of some fixed parameters that are used for convexifying the optimization problem. In this paper, several data-driven optimization algorithms are proposed to improve the solution for low-order controllers. A non-convex problem is solved (in a data-driven sense) where the parameters of a fixed-structure low-order controller are optimized; the solution to the problem guarantees the stability of the closed-loop system whilst ensuring robust performance. It is shown that by optimizing all of the controller parameters, the $\mathcal{H}_\infty$ performance for low-order controllers can be significantly improved. The simulation examples illustrate how the proposed method can be used to eliminate the sensitivity associated with the fixed parameters and optimize the system performance.**

I. INTRODUCTION

The data-driven control strategy mitigates the problems with model-based controller designs by avoiding the problem of unmodelled dynamics associated with low-order parametric models. A survey on the differences between the model-based control and data-driven control schemes has been addressed in [1] and [2]. The data-driven controller design methodology can be realized in two manners: using time-domain or frequency-domain data. In this paper, the frequency-domain approach will be utilized for the controller design. It is important to note that the term "data-driven" in this paper signifies that the controller synthesis is independent of the parametric model of the process. In other words, controllers are synthesized by only using the frequency response of the process.

In addition to avoiding the problem of unmodelled dynamics, the use of controllers with pre-defined structures is also important. In the classical robust control design method, the order of the resulting full-order controllers can be quite large; in fact, the order can be as large as the order of the augmented plant [3], [4]. However, in industrial and practical applications, controllers are often implemented in dedicated hardware which are constrained to specific controller structures.

It is well known that fixed-structure controller design in the model-based setting is a non-convex optimization problem. Nonsmooth optimization methods for fixed-structure controllers are used in [5], [6] and [7]; these methods are implemented in the MATLAB Robust Control Toolbox. However, these non-smooth techniques cannot synthesize controllers based on the frequency response of the system (they need a parametric model), and are limited to certain system dynamics (i.e., a pure delay must be approximated by a Padé function).

Solving the $\mathcal{H}_\infty$ problem can be accomplished in a data-driven setting. For fixed-structure controllers, the authors in [8]–[13] use linearly parameterized (LP) controllers in a data-driven setting in order to convexify the $\mathcal{H}_\infty$ problem. However, the results depends on the choice of some fixed parameters in the controller. By convexifying the $\mathcal{H}_\infty$ problem, the global optimal solution to an approximate problem is obtained; however, a question one may ask is why are convexification methods imposed to find a solution to the approximate $\mathcal{H}_\infty$ problem when one can simply use nonlinear solvers to find a local solution of the true $\mathcal{H}_\infty$ problem? This is the question that will be addressed in this paper.

In [14], a new data-driven method for $\mathcal{H}_\infty$ controller design by convex optimization was proposed; a controller was represented as a ratio of two transfer functions (TFs) that were linearly parameterized using a vector of basis functions. It was shown that as the order of these TFs increased, then the $\mathcal{H}_\infty$ performance converged to the global optimal solution (regardless of the basis function that is used). For a low-order controller, however, the results depend on the choice of the basis functions and is not necessarily optimal. This paper presents an extension of the work in [14], and its purpose is to devise a data-driven approach for improving the $\mathcal{H}_\infty$ performance for low-order fixed-structure controllers. Several non-convex optimization problems are proposed to optimize the basis function parameters for fixed-structure low-order controllers (while guaranteeing the closed-loop stability). In particular, a new particle swarm optimization (PSO) algorithm is formulated to optimize the controller parameters and guarantee the stability of the closed-loop system while ensuring robust performance (without any approximation). As with all nonlinear solvers, there are trade-offs that exist between the optimization time and the quality of the optimal solution; these trade-offs will be investigated by comparing the optimal solutions from various methods.

This paper is organized as follows: In Section II, the class of models and controllers are defined; additionally, this section will introduce the notion of $\mathcal{H}_\infty$ control and elicit some of the concepts in [14]. Section III will address the control objectives and the conditions required for satisfying

A. Nicoletti is with the Technology Department, European Organization for Nuclear Research (CERN), Switzerland. A. Karimi is with the Automatic Control Laboratory at Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland. Corresponding author: Alireza Karimi: alireza.karimi@epfl.ch

the $\mathcal{H}_\infty$ criterion. Section IV will demonstrate the effectiveness of the proposed methods by comparing the solutions for two simulation examples. Finally the concluding remarks are given in Section V.

II. Preliminaries

In order to avoid the risk of any confusion, the notation that will be used in this paper will first be defined. $\mathbb{R}$ and $\mathbb{R}_+$ represent the set of all real numbers and real numbers greater than zero, respectively. $\Re\{\cdot\}$ denotes the real part of a complex variable. Bold-faced variables will represent column vectors (i.e., the vector $\boldsymbol{v} = [v_1, \dots, v_n]^\top$).

A. Class of Models

The set of all linear time-invariant (LTI) single-input-signal-output (SISO) strictly proper frequency response models belonging to the family of perturbed plants with multimodel and multiplicative uncertainty can be defined as follows:

$$\mathcal{G} = \{G_p(j\omega)[1 + \Delta(j\omega)W_{2_p}(j\omega)]; \quad p = 1, \dots, n_p\} \quad (1)$$

where $G_p(j\omega)$ is the nominal frequency response of the p^{th} process, $W_{2_p}(j\omega)$ is an uncertainty weight with bounded infinity norm, $\Delta(j\omega)$ is an unknown stable TF satisfying $\|\Delta\|_\infty < 1$, and n_p is the number of models. For simplicity, one model from the set $\mathcal{G}$ will be analyzed, and the subscript p will be omitted, but the results are applicable to multi-model uncertainty as well (as will be shown in the simulation examples).

Let the set $\mathbf{RH}_\infty$ represent the family of all stable, proper, real-rational transfer functions with bounded infinity norm. We assume that the plant is represented as $G(s) = N(s)M^{-1}(s)$ such that $\{N(s), M(s)\} \in \mathbf{RH}_\infty$ are called the coprime factors of $G(s)$ over $\mathbf{RH}_\infty$ [15]. The frequency response of such a system is given by:

$$G(j\omega) = N(j\omega)M^{-1}(j\omega), \qquad \omega \in \Omega \quad (2)$$

where $\Omega = \mathbb{R} \cup \{\infty\}$. Note that for stable systems, we assume $M(j\omega) = 1$ and so $N(j\omega)$ is available by spectral analysis from a set of data. For unstable systems, the frequency response of the coprime factors can be identified from a closed-loop experiment with a stabilizing controller [14].

B. Class of Controllers

The controller can be described as $K(s) = X(s)Y^{-1}(s)$, where $\{X(s), Y(s)\} \in \mathbf{RH}_\infty$. The TFs $X(s)$ and $Y(s)$ for this control scheme will be parameterized in the decision vector $\boldsymbol{\rho}$, and can be expressed as

$$X(s, \boldsymbol{\rho}) = \boldsymbol{\rho}_x^\top \boldsymbol{\phi}(s, \xi), \quad Y(s, \boldsymbol{\rho}) = \boldsymbol{\rho}_y^\top \boldsymbol{\phi}(s, \xi) \quad (3)$$

where $\boldsymbol{\rho}_x \in \mathbb{R}^{n_x}$, $\boldsymbol{\rho}_y \in \mathbb{R}^{n_y}$, and $\boldsymbol{\phi}(s, \xi)$ are vectors of stable transfer functions chosen from a set of orthogonal basis functions. An example of such a function is the Laguerre basis function [16]:

$$\phi_1(s) = 1, \quad \phi_q(s, \xi) = \frac{\sqrt{2\xi}\,(s - \xi)^{q-2}}{(s + \xi)^{q-1}} \quad (4)$$

for $q = 2, \dots, \{n_x, n_y\}$, where $\xi \in \mathbb{R}_+$.

A PID controller can also be represented in this form. Suppose that the desired controller structure is given as:

$$K(s, \boldsymbol{\rho}) = k_p + k_i \frac{1}{s} + k_d \frac{s}{T_f s + 1} \quad (5)$$

where $T_f \in \mathbb{R}_+$. Then the controller can be expressed as in (3) with

$$\begin{aligned} \boldsymbol{\rho}_x^\top &= [\rho_1 \quad \rho_2 \quad \rho_3] \\ \boldsymbol{\rho}_y^\top &= [T_f \quad 1 \quad 0] \\ \boldsymbol{\phi}(s, \xi) &= [s^2 \quad s \quad 1]^\top (s + \xi)^{-2} \end{aligned} \quad (6)$$

where the parameters of $\boldsymbol{\rho}_x$ are $\rho_1 = k_p T_f + k_d$, $\rho_2 = k_p + k_i T_f$, $\rho_3 = k_i$.

C. Nominal Performance via Convex Optimization

The methods described in this paper are associated with minimizing the $\mathcal{H}_\infty$ norm of a desired weighted sensitivity function. The sensitivity function $S(s)$ and complementary sensitivity function $T(s)$ associated with this control scheme can be formulated as follows:

$$\begin{aligned} S(s, \boldsymbol{\rho}) &:= e(s)/r(s) = [1 + L(s, \boldsymbol{\rho})]^{-1} \\ &= M(s)Y(s, \boldsymbol{\rho})[N(s)X(s, \boldsymbol{\rho}) + M(s)Y(s, \boldsymbol{\rho})]^{-1} \end{aligned} \quad (7)$$

$$\begin{aligned} T(s, \boldsymbol{\rho}) &:= y(s)/r(s) = L(s, \boldsymbol{\rho})[1 + L(s, \boldsymbol{\rho})]^{-1} \\ &= N(s)X(s, \boldsymbol{\rho})[N(s)X(s, \boldsymbol{\rho}) + M(s)Y(s, \boldsymbol{\rho})]^{-1} \end{aligned} \quad (8)$$

where $L(s, \boldsymbol{\rho}) = G(s)K(s, \boldsymbol{\rho})$; $r(s), y(s)$ and $e(s) = r(s) - y(s)$ are the Laplace transforms of the reference input, the system output, and the tracking error signal, respectively.

Given a performance weighting filter $W_1(j\omega)$ with bounded infinity norm, the nominal performance is given by [15]:

$$\|W_1 S\|_\infty < \gamma \quad (9)$$

where $\gamma = 1$. However, the problem of minimizing the upper bound γ will be considered in this paper, where $\gamma \in \mathbb{R}_+$. The condition in (9) can also be expressed as

$$|W_1(j\omega)S(j\omega, \boldsymbol{\rho})| < \gamma, \quad \forall \omega \in \Omega \quad (10)$$

For notation purposes, the dependence in $j\omega$ will be omitted and will only be reiterated when deemed necessary. The dependence on $\boldsymbol{\rho}$ will continue to be highlighted. By substituting the frequency responses of (7) and (8) into (10), the condition for nominal performance can be expressed as

$$\gamma^{-1}|W_1 M Y(\boldsymbol{\rho})| < |\psi(\boldsymbol{\rho})|, \quad \forall \omega \in \Omega \quad (11)$$

where $\psi(\boldsymbol{\rho}) = NX(\boldsymbol{\rho}) + MY(\boldsymbol{\rho})$.

Consider a circle in the complex plane at a specific frequency in Ω which is centered at $\psi(\boldsymbol{\rho})$ and has radius $\gamma^{-1}|W_1 M Y(\boldsymbol{\rho})|$. The constraint in (11) ensures that for any frequency point in Ω, the circle associated with this frequency point will not encircle the origin. In [14], the authors show that there exists a function F that can rotate this circle such that it lies on the right-hand side of the $j\omega$ axis of the complex plane (i.e., all values on and within the circle have positive real parts). This condition is recalled with the following Lemma:

Lemma 1. *Suppose that $H(\boldsymbol{\rho}) = W_1 M Y(\boldsymbol{\rho})\psi^{-1}(\boldsymbol{\rho})$ is the frequency response of a bounded analytic function in the right-half plane. Then, the following constraint is met*

$$\sup_{\omega\in\Omega} |H_1(\boldsymbol{\rho})| < \gamma \tag{12}$$

if and only if there exists a stable transfer function $F(s)$ that satisfies

$$\Re\{\psi(\boldsymbol{\rho})F\} > \gamma^{-1}|W_1 M Y(\boldsymbol{\rho})F|, \quad \forall\omega\in\Omega.$$

Proof : The proof has been omitted to conserve space. However, the proof of a similar condition can be found in [14]. ■

With the above Lemma, a necessary and sufficient condition can be derived for attaining nominal performance whilst ensuring the closed-loop stability. In [14], it was shown that if $X(\boldsymbol{\rho})$ and $Y(\boldsymbol{\rho})$ are linearly parameterized, then a quasi-convex optimization problem can be formulated as follows:

$$\begin{aligned} \underset{\gamma,\boldsymbol{\rho}}{\text{minimize}} \quad & \gamma \\ \text{subject to:} \quad & \gamma^{-1}|W_1 M Y(\boldsymbol{\rho})| < \Re\{\psi(\boldsymbol{\rho})\} \\ & \forall\omega\in\Omega \end{aligned} \tag{13}$$

The optimal solution to the above optimization problem can be obtained by fixing γ and implementing a bisection algorithm, where the problem is iteratively solved until the optimal solution is obtained within a given tolerance [17]. Notice that (13) is a semi-infinite programming (SIP) optimization problem since there are a finite number of optimization variables and an infinite number of constraints. This problem can be transformed into a semi-definite programming (SDP) problem by presetting a frequency grid ω and solving a finite number of constraints. This frequency grid can be predefined in a variety of manners (see [18], [19]).

III. Optimization Problems

In order to preserve the convexity of the $\mathcal{H}_\infty$ problem in (13) (with fixed-structure controllers), it is sufficient to invoke linearly parameterized TFs for $X(\boldsymbol{\rho})$ and $Y(\boldsymbol{\rho})$ where both TFs contain basis functions with fixed values. In [14], it was shown that when the orders of $X(\boldsymbol{\rho})$ and $Y(\boldsymbol{\rho})$ increased, then γ from (13) converged monotonically to the global optimal solution of the $\mathcal{H}_\infty$ problem. However, it is impractical and sometimes impossible to implement the resulting high-order controllers to real systems. For low-order controllers, the optimal solution from the convex problem may be far from the global solution, and is very sensitive to the pre-set values of the basis function parameters [20]. A solution to this problem is to simultaneously optimize the controller parameters $\boldsymbol{\rho}$ and the basis function parameter ξ by a nonlinear optimization algorithm.

An alternative is to formulate an optimization problem based on the results of Lemma 1 in order to improve the performance for low-order controllers.

Theorem 1. *The local optimal solution for obtaining H_∞ performance and closed-loop stability using the fixed-structure controllers $X(\boldsymbol{\rho})$ and $Y(\boldsymbol{\rho})$ is achieved if $F(\boldsymbol{\rho}_f)$ is parameterized with a set of stable orthogonal basis functions and the following optimization problem is realized:*

$$\begin{aligned} \underset{\gamma,\boldsymbol{\rho},\boldsymbol{\rho}_f}{\textit{minimize}} \quad & \gamma \\ \textit{subject to:} \quad & \gamma^{-1}|W_1 M Y(\boldsymbol{\rho})F(\boldsymbol{\rho}_f)| < \Re\{\psi(\boldsymbol{\rho})F(\boldsymbol{\rho}_f)\} \\ & \forall\omega\in\Omega \end{aligned} \tag{14}$$

Proof : According to Lemma 1, it is known that there exists a stable transfer function F such that the constraint to the $\mathcal{H}_\infty$ problem is satisfied. Therefore, $F = F(\boldsymbol{\rho}_f)$ can be chosen such that it incorporates stable basis functions (such as the Laguerre basis functions). Thus the local optimal solution to the $\mathcal{H}_\infty$ problem can be obtained by minimizing γ, fixing the orders of $X(\boldsymbol{\rho})$ and $Y(\boldsymbol{\rho})$, and implementing the optimization problem in (14).

The constraint in this problem implies that $\Re\{F(\boldsymbol{\rho}_f)\psi(\boldsymbol{\rho})\} > 0$. Therefore, the Nyquist plot of $F(\boldsymbol{\rho}_f)\psi(\boldsymbol{\rho})$ will not encircle the origin, and thus the Nyquist plot of $[F(\boldsymbol{\rho}_f)\psi(\boldsymbol{\rho})]^{-1}$ will also not encircle the origin (which implies that $[F(\boldsymbol{\rho}_f)\psi(\boldsymbol{\rho})]^{-1}$ is stable and that the closed-loop system is stable). ■

For continuous-time systems, the function $F(\boldsymbol{\rho}_f)$ can be selected as $F(\boldsymbol{\rho}_f) = \boldsymbol{\rho}_f^\top \boldsymbol{\phi}(\xi_f)$ where $\boldsymbol{\rho}_f \in \mathbb{R}^{n_f}$ and $\boldsymbol{\phi}(\xi_f)$ is the vector of Laguerre basis functions asserted in (4) with the Laguerre parameter defined as $\xi_f \in \mathbb{R}_+$. It is imperative to note that $F(\boldsymbol{\rho}_f)$ is not part of the controller; it is a function which realizes the necessary and sufficient condition in Lemma 1. The type of optimization problem in (14) depends on the parameterization of $X(\boldsymbol{\rho})$, $Y(\boldsymbol{\rho})$, and $F(\boldsymbol{\rho}_f)$.

A. Bilinear Programming

If $X(\boldsymbol{\rho})$, $Y(\boldsymbol{\rho})$, and $F(\boldsymbol{\rho}_f)$ are linearly parameterized (where the Laguerre parameter ξ and ξ_f are fixed for each function), then the optimization problem in (14) becomes a bilinear problem (BP) when an iterative algorithm is used to compute the optimal γ. It is known that if $(a^\star, b^\star)$ is a local solution to a BP given an objective function $f(a,b)$, then

$$\min_a f(a, b^\star) = f(a^\star, b^\star) = \min_b f(a^\star, b) \tag{15}$$

Given this property of BPs, the local solution to the BP can be obtained by solving a finite set of convex optimization problems until convergence is achieved. The basic idea for solving (14) in this manner is to fix the orders of $X(\boldsymbol{\rho})$ and $Y(\boldsymbol{\rho})$ and solve a set of convex problems for increasing orders of $F(\boldsymbol{\rho}_f)$ until convergence is achieved for γ (within a given tolerance). This optimization technique is known as the "Mountain Climbing" method [21].

B. Particle Swarm Optimization

When the basis function parameters ξ and ξ_f in $X(\boldsymbol{\rho})$, $Y(\boldsymbol{\rho})$ and $F(\boldsymbol{\rho}_f)$ are decision variables, then the problem in (14) becomes nonlinear. One of the problems with solving this nonlinear problem is defining the initial values for the decision variables. Since there can be many variables involved in this optimization problem, defining the initial variables to achieve the global optimal solution to the $\mathcal{H}_\infty$ problem may not be trivial.

PSO is a powerful optimization method that can solve both linear and nonlinear problems and can be used to solve the problem in (14) without specifying initial conditions. It is based on the principle that groups of individuals work together to improve both their collective and individual performance [22]. Due to the constraints imposed in (14), an exterior method (i.e., Non-Death-Penalty approach) will be implemented in order to obtain the optimal solution to the problem. With this method, the constrained optimization problem can be transformed to the following unconstrained problem:

$$\underset{\boldsymbol{x}}{\text{minimize}} \quad \Phi(j\omega, \boldsymbol{x}) \tag{16}$$

where $\boldsymbol{x}^\top = [\boldsymbol{\rho}^\top, \boldsymbol{\rho}_f^\top, \boldsymbol{\xi}^\top, \gamma]$ and $\boldsymbol{\xi} = [\xi, \xi_f]^\top$ is the vector of basis function parameters. We have

$$\begin{aligned} \Phi(j\omega, \boldsymbol{x}) &= \gamma + \frac{1}{Q}\sum_{k=1}^{Q} \alpha_k Z_k(j\omega, \boldsymbol{x}) \\ Z_k(j\omega, \boldsymbol{x}) &= [\max(0, z_k(j\omega, \boldsymbol{x}))]^\beta \\ z_k(j\omega, \boldsymbol{x}) &= |W_1(j\omega)M(j\omega)Y(j\omega, \boldsymbol{x})F(j\omega, \boldsymbol{x})| \\ &\quad - \gamma\Re\{\psi(j\omega, \boldsymbol{x})F(j\omega, \boldsymbol{x})\} \end{aligned} \tag{17}$$

The value of β is usually taken to be 1 or 2 and $\alpha_k \in \mathbb{R}_+$ is the penalty factor [22]. In this paper, $\beta = 1$ will be considered. A very large penalty factor will ensure fast convergence to a local solution (even if it is far from the optimal), while a small penalty factor will cause the PSO algorithm to spend much time searching in infeasible regions and may converge to an infeasible solution [23]. For this particular problem, the value of α_k will be a constant, since the weighting factor for each constraint should be the same ($\alpha_k = \alpha$). In other words, the constraint should not be weighted differently for varying frequencies. To obtain more details with regards to the PSO algorithm and how it was applied to the problems in this paper, see [20].

IV. Simulation Examples

Let us now consider two examples in order to determine the validity of the proposed method(s). The `YALMIP` library [24] in conjunction with `MATLAB` was used to solve the convex problem (i.e., the sequential set of convex problems to solve the BP).

For each example, the proposed method will be used where the non-convex problem is solved using three different approaches:

- **Method 1**: Linearly parameterizing $X(\boldsymbol{x})$, $Y(\boldsymbol{x})$, and $F(\boldsymbol{x})$ (where the basis function parameters in $\boldsymbol{\xi}$ are fixed *a-priori*) and use the BP algorithm to solve a sequential set of convex problems until convergence is achieved for increasing n_f.
- **Method 2**: Formulate non-LP functions for $X(\boldsymbol{x})$ and $Y(\boldsymbol{x})$ (with $F(\boldsymbol{x}) = 1$) and use the PSO algorithm to optimize $\boldsymbol{\rho}$ and ξ.
- **Method 3**: Formulate non-LP functions for $X(\boldsymbol{x}), Y(\boldsymbol{x})$ and $F(\boldsymbol{x})$ and use the PSO algorithm to optimize all parameters in $\boldsymbol{x}$.

For comparative purposes, the solutions from all three methods will be compared to the solution obtained from the `hinfstruct` function in `MATLAB`. As a result, examples from the literature with parametric models are chosen.

Remark. *It is emphasized that a direct comparison with* `hinfstruct` *is not the objective of these examples, since the proposed method does not use the parametric models (only the frequency data are used in the optimization problems).*

All optimization problems were solved using a computer with a Intel-i7 core (3.4 GHz) processor and with 8 GB of RAM running on a 64-bit Windows 7 platform. The `MATLAB` version (R2015b) was used for running all algorithms.

A. Example 1: Robust PID Design

Consider the unstable non-minimum phase system analyzed in [25] and [26]:

$$G(s) = \frac{s-1}{s^2 + 0.8s - 0.2} \tag{18}$$

which is subject to multiplicative uncertainty with the weighting filter $W_2(s) = (s + 0.1)(s + 1)^{-1}$. The objective of this case study is to design a robust stabilizing PID controller such that the following robust stability and nominal performance conditions are satisfied:

$$\|W_1 S\|_\infty < \gamma \quad \text{and} \quad \|W_2 T\|_\infty < \gamma \tag{19}$$

where performance weighting filter is chosen as [25]: $W_1(s) = 10(100s + 1)^{-1}$. Since $G(s)$ is unstable, the coprime functions can be selected as $N(s) = (s-1)(s+1)^{-2}$ and $M(s) = (s^2 + 0.8s - 0.2)(s + 1)^{-2}$, where it is evident that $G(s) = N(s)M^{-1}(s)$. For a PID controller, the structure of the functions $X(\boldsymbol{x})$ and $Y(\boldsymbol{x})$ can be selected with the vectors defined in (6). Therefore, the following optimization problem is considered for satisfying (19):

$$\begin{aligned} &\underset{\boldsymbol{x}}{\text{minimize}} && \gamma \\ &\text{subject to:} && \gamma^{-1}|W_1 M Y(\boldsymbol{x})F(\boldsymbol{x})| < \Re\{\psi(\boldsymbol{x})F(\boldsymbol{x})\} \\ &&& \gamma^{-1}|W_2 N X(\boldsymbol{x})F(\boldsymbol{x})| < \Re\{\psi(\boldsymbol{x})F(\boldsymbol{x})\} \\ &&& \omega \in \Omega \end{aligned} \tag{20}$$

1) Simulation Results: The problem in (20) was solved using the proposed method with a frequency grid from 10^{-2} to 10^2 [rad/s] (using 200 logarithmically spaced points). Let γ_{n_f} denote the optimal solution to the problem for a given order n_f of $F(\boldsymbol{x})$. The BP was solved with the iterative convex method (i.e., the Mountain Climbing method) for different basis function values (with $\xi = \xi_f$); Fig. 1 displays the optimal solution as a function of n_f. It can be observed that regardless of the basis function parameter, the solution

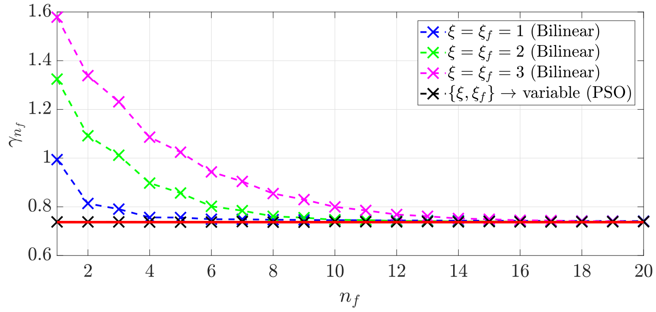

Fig. 1. Optimal solution to (20) using the proposed bilinear and PSO algorithms. The optimal solution produced by `hinfstruct` (solid-red line).

TABLE I
COMPARISON OF OPTIMAL SOLUTIONS WITH OPTIMIZATION TIME

	$\gamma^\star$	Optimization Time [min]
Method 1	0.737	[4, 21]
Method 2	0.799	0.48
Method 3	0.737	0.50
Convex Method [14]	1.113	0.84
`hinfstruct`	0.737	0.18

converges to the same value (which in this case, is $\gamma^\star = 0.737$). The `hinfstruct` function from `MATLAB` produces the same value.

Now consider the parameterizations of Method 2 and Method 3; with Method 3, $F(\boldsymbol{x})$ was selected with $n_f = 2$. The author in [20] has provided the details with regards to the parameters used in the PSO algorithm and how the optimal solution was obtained. Table I compares the optimal solutions obtained with the optimization time for each method. The convex method refers to the algorithm in [14] with $\xi = 1$. Note that the optimization time of Method 1 varies based on which basis function parameter is used. The variation time shown in this table is based on the values used in Fig. 1. Method 3 achieves a low optimal value with little time; therefore, *in a data-driven sense*, optimizing $X(\boldsymbol{x})$, $Y(\boldsymbol{x})$ and $F(\boldsymbol{x})$ using the PSO algorithm proves to be the more efficient solution for this problem.

Figure 1 also displays the solution for varying n_f using Method 3. It can be observed that when these basis function parameters are optimized, a 1$^{\text{st}}$ order function for $F(\boldsymbol{x})$ produces a solution approximately equal to the solution from `hinfstruct`. Thus optimizing the basis function parameters using the PSO algorithm proves to be more efficient, since convergence to a solution is obtained without implementing high orders of $F(\boldsymbol{x})$.

B. Example 2: Multimodel Uncertainty

For this example, a robust controller will be designed for a family of unstable systems. This example is taken from the Robust Control Toolbox of `MATLAB`. The nominal plant model for this family of systems is given as $G_0(s) = 2(s-2)^{-1}$; this model is perturbed through various types of uncertainties such as time delay, high frequency resonance, pole/gain migration, and extra lag; the family of perturbed models are given as follows:

$$\begin{aligned} G_1(s) &= G_0(s)\frac{1}{0.06s+1} & G_4(s) &= G_0(s)e^{-0.04s} \\ G_2(s) &= G_0(s)\frac{50^2}{s^2+10s+50^2} & G_5(s) &= \frac{2.4}{s-2.2} \\ G_3(s) &= G_0(s)\frac{70^2}{s^2+28s+70^2} & G_6(s) &= \frac{1.6}{s-1.8} \end{aligned} \tag{21}$$

The performance filter $W_1(s)$ and noise filter $W_2(s)$ are chosen to be equal to the filters asserted in [16], i.e.,

$$\begin{aligned} W_1(s) &= \frac{0.33s+4.248}{s+0.008496} \\ W_2(s) &= \frac{0.1975s^2+0.6284s+1}{7.901\cdot 10^{-5}s^2+0.2514s+400} \end{aligned} \tag{22}$$

For this problem, the control objective is to minimize γ and satisfy the following criteria for all seven models:

$$\|W_1 S_\ell\|_\infty < \gamma \quad \text{and} \quad \|W_2 T_\ell\|_\infty < \gamma \tag{23}$$

for $\ell = 0, \ldots, 6$. The `hinfstruct` function in `MATLAB`'s Robust Control Toolbox uses this criteria to design a controller, and achieves an optimal solution of $\gamma^\star = 0.886$ (with 200 random initiations) when a 6$^{\text{th}}$ order controller is used (with integral action).

The same problem is now solved using the proposed approach. First, the coprime factors $N_\ell(s)$ and $M_\ell(s)$ for $\ell = 0, \ldots, 6$ must be established. Since each model is unstable, then each coprime factor must be selected such that $\{N_\ell(s), M_\ell(s)\} \in \mathbf{RH}_\infty$ for all ℓ. A simple choice is to divide both the numerator and denominator of each model by a factor $(s+100)^{\lambda_\ell}$ (as defined in [14]), where λ_ℓ is the largest degree of the denominator of the ℓ-th respective plant model.

1) Simulation Results: The problem in (23) is solved using the proposed approach by considering a logarithmically spaced frequency grid with 300 points from 10^{-1} [rad/s] to 10^4 [rad/s]. First, consider the parameterization process asserted in Method 1; a 6$^{\text{th}}$ order controller is designed (5$^{\text{th}}$ order controller with one integrator) using the Laguerre basis functions defined in (4) with $\xi = 20$ (as defined in [14]) and with $\xi_f = 20$. The BP is then solved using the "Mountain Climbing" method until convergence is achieved (within 10^{-5}) for $n_f = 10$.

Now consider the parameterization method asserted in Method 2 and Method 3 (where Method 3 will use a function $F(\boldsymbol{x})$ with $n_f = 10$). The author in [20] has provided the details with regards to the parameters used in the PSO algorithm and how the optimal solution was obtained. A comparison of the optimal solutions with the optimization time for each method satisfying the criteria in (23) are tabulated in Table II (where each method implements a 6$^{\text{th}}$ order controller). From all of these algorithms, it can be observed that the proposed method using Method 3 yields the best solution for this problem. Note that reducing the number of random initializations with `hinfstruct` produces a

TABLE II

COMPARISON BETWEEN OPTIMAL SOLUTIONS AND OPTIMIZATION TIME FOR MULTIMODEL PROBLEM

	$\gamma^\star$	Optimization Time [min]
Method 1	0.830	606
Method 2	0.880	207
Method 3	0.817	401
Convex Method [14]	0.881	15
hinfstruct	0.886	212

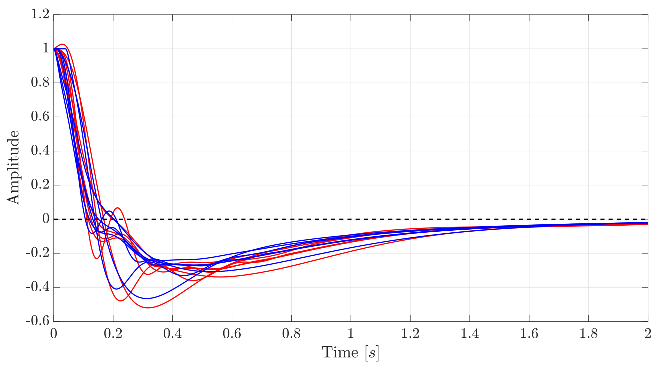

Fig. 2. Step responses of $S_\ell(s)$ $\forall \ell$ using the proposed PSO algorithm (blue) and with the controller obtained from `hinfstruct` (red).

solution close to the solution in Table II (a difference of approximately 1%).

Figure 2 shows a comparison of the step responses of $S_\ell(s)$ using the proposed PSO method (with function parameterization as asserted in Method 3) and `hinfstruct`. It can be observed that the results are comparable; however, the proposed method produces reduced overshoot (at the expense of a larger optimization time).

V. CONCLUSION

A data-driven approach has been implemented in order to design robust controllers that achieve $\mathcal{H}_\infty$ performance. A new PSO algorithm was formulated in a data-driven setting to solve a non-convex optimization problem and optimize all parameters of a fixed-structure controller while guaranteeing the stability of the closed-loop system. The simulation examples show that for very low-order controllers (such as the PID controller), the solutions to the non-convex optimization problems yield better results in a short amount of time. For higher order controllers, the convex method produces a reasonable value (with respect to the optimal values of the non-convex problems) in a relatively short time. For future work, it will be desired to compare the solutions and optimization times using other nonlinear solvers (such as genetic algorithms, evolutionary programming, and differential evolution).

VI. ACKNOWLEDGEMENT

The authors would like to thank Dr. Dan Simon from Cleveland State University for his contribution to this work.

REFERENCES

[1] Z.-S. Hou and Z. Wang, "From model-based control to data-driven control: Survey, classification and perspective," *Information Sciences*, vol. 235, pp. 3–35, 2013.

[2] S. Formentin, K. Van Heusden, and A. Karimi, "A comparison of model-based and data-driven controller tuning," *Int. Journal of Adaptive Control and Signal Processing*, 2013.

[3] P. Gahinet and P. Apkarian, "An LMI-based parametrization of all H_∞ controllers with applications," in *Proceedings of the 32nd IEEE Conference on Decision and Control*. IEEE, 1993, pp. 656–661.

[4] J. C. Doyle, K. Glover, P. P. Khargonekar, and B. A. Francis, "State-space solutions to standard H_2 and H_∞ control problems," *IEEE Transactions on Automatic Control*, vol. 34, no. 8, pp. 831–847, Aug 1989.

[5] P. Apkarian and D. Noll, "Nonsmooth H_∞ synthesis," *IEEE Transactions on Automatic Control*, vol. 51, no. 1, pp. 71–86, 2006.

[6] P. Gahinet and P. Apkarian, "Decentralized and fixed-structure H_∞ control in MATLAB," in *50th IEEE Conference on Decision and Control (CDC)*, Orlando, FL, USA, 2011.

[7] ——, "Structured H_∞ synthesis in MATLAB," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 1435–1440, 2011.

[8] A. Karimi and G. Galdos, "Fixed-order H_∞ controller design for nonparametric models by convex optimization," *Automatica*, vol. 46, no. 8, pp. 1388–1394, 2010.

[9] A. Karimi, H. Khatibi, and R. Longchamp, "Robust control of polytopic systems by convex optimization," *Automatica*, vol. 43, no. 6, pp. 1395–1402, 2007.

[10] E. Solingen, J. Wingerden, and T. Oomen, "Frequency-domain optimization of fixed-structure controllers," *International Journal of Robust and Nonlinear Control*, 2016.

[11] M. Saeki, M. Ogawa, and N. Wada, "Low-order H_∞ controller design on the frequency domain by partial optimization," *International Journal of Robust and Nonlinear Control*, vol. 20, no. 3, pp. 323–333, 2010.

[12] M. Hast, K. J. Aström, B. Bernhardsson, and S. Boyd, "PID design by convex-concave optimization," in *European Control Conference*, Zurich, Switzerland, 2013, pp. 4460–4465.

[13] S. Boyd, M. Hast, and K. J. Åström, "MIMO PID tuning via iterated LMI restriction," *International Journal of Robust and Nonlinear Control*, vol. 26, no. 8, pp. 1718–1731, 2016.

[14] A. Karimi, A. Nicoletti, and Y. Zhu, "Robust H_∞ controller design using frequency-domain data via convex optimization," *International Journal of Robust and Nonlinear Control*, 2016. [Online]. Available: http://dx.doi.org/10.1002/rnc.3594

[15] C. J. Doyle, B. A. Francis, and A. R. Tannenbaum, *Feedback Control Theory*. New York: Mc Millan, 1992.

[16] A. Karimi, "Frequency-domain robust control toolbox," in *52nd IEEE Conference in Decision and Control*, 2013, pp. 3744 – 3749.

[17] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge,UK: Cambridge University Press, 2004.

[18] P. Seiler, B. Vanek, J. Bokor, and G. J. Balas, "Robust H_∞ filter design using frequency gridding," in *American Control Conference (ACC), 2011*. IEEE, 2011, pp. 1801–1806.

[19] G. Galdos, A. Karimi, and R. Longchamp, "H_∞ controller design for spectral MIMO models by convex optimization," *Journal of Process Control*, vol. 20, no. 10, pp. 1175 – 1182, 2010.

[20] A. Nicoletti, "A data-driven frequency-domain approach for robust controller design via convex optimization," Ph.D. dissertation, Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland, 2018.

[21] H. Konno, "A cutting plane algorithm for solving bilinear programs," *Mathematical Programming*, vol. 11, no. 1, pp. 14–27, 1976.

[22] D. Simon, *Evolutionary optimization algorithms*. John Wiley & Sons, 2013.

[23] Ö. Yeniay, "Penalty function methods for constrained optimization with genetic algorithms," *Mathematical and Computational Applications*, vol. 10, no. 1, pp. 45–56, 2005.

[24] J. Löfberg, "YALMIP: A toolbox for modeling and optimization in MATLAB," in *CACSD Conference*, http://control.ee.ethz.ch/joloef/yalmip.php, 2004.

[25] Z. Zhen-ping, "Determination of the admissible parameter set in designing PID controllers with guaranteed robust performance," in *Electrical and Control Engineering (ICECE), 2010 International Conference on*. IEEE, 2010, pp. 3625–3628.

[26] M.-T. Ho, "Synthesis of H_∞ PID controllers: A parametric approach," *Automatica*, vol. 39, no. 6, pp. 1069–1075, 2003.