

Article

μ -Synthesis for Fractional-Order Robust Controllers

Vlad Mihaly ^{*}, Mircea Șușcă ^{*}, Dora Morar [†], Mihai Stănese [†] and Petru Dobra [†]

Department of Automation, Technical University of Cluj-Napoca, Str. G. Barițiu nr. 26-28, 400027 Cluj-Napoca, Romania; Dora.Sabau@aut.utcluj.ro (D.M.); mihai.stanese@aut.utcluj.ro (M.S.); Petru.Dobra@aut.utcluj.ro (P.D.)

* Correspondence: vlad.mihaly@aut.utcluj.ro (V.M.); mircea.susca@aut.utcluj.ro (M.Ș.)

† These authors contributed equally to this work.

Abstract: The current article presents a design procedure for obtaining robust multiple-input and multiple-output (MIMO) fractional-order controllers using a μ -synthesis design procedure with D - K iteration. μ -synthesis uses the generalized Robust Control framework in order to find a controller which meets the stability and performance criteria for a family of plants. Because this control problem is NP-hard, it is usually solved using an approximation, the most common being the D - K iteration algorithm, but, this approximation leads to high-order controllers, which are not practically feasible. If a desired structure is imposed to the controller, the corresponding K step is a non-convex problem. The novelty of the paper consists in an artificial bee colony swarm optimization approach to compute the nearly optimal controller parameters. Further, a mixed-sensitivity μ -synthesis control problem is solved with the proposed approach for a two-axis Computer Numerical Control (CNC) machine benchmark problem. The resulting controller using the described algorithm manages to ensure, with mathematical guarantee, both robust stability and robust performance, while the high-order controller obtained with the classical μ -synthesis approach in MATLAB does not offer this.

Keywords: μ -synthesis; robust control; fractional-order control; swarm optimization; artificial bee colony optimization; CNC machine; mixed sensitivity; D - K iteration; Linear Matrix Inequality



Citation: Mihaly, V.; Șușcă, M.; Morar, D.; Stănese, M.; Dobra, P. μ -Synthesis for Fractional-Order Robust Controllers. *Mathematics* **2021**, *9*, 911. <https://doi.org/10.3390/math9080911>

Academic Editor: António M. Lopes

Received: 28 February 2021

Accepted: 16 April 2021

Published: 20 April 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

One of the active problems with major impact which have been studied for years in Control Theory refers to robustness. Robustness encompasses the sensitivity of a control system with respect to both internal and external disturbances. Several robust methods have been developed in order to achieve robust performance and stability in the presence of uncertainties. Robust control problems use \mathcal{H}_2 and \mathcal{H}_∞ norms defined in frequency domain as a performance measure. To solve $\mathcal{H}_2/\mathcal{H}_\infty$ control problems, there are several approaches. One possible solution is presented in [1] and is based on Algebraic Riccati Equations (AREs). A more numerically stable approach to solve ARE was developed using Popov triplets in [2], approach recently implemented in an open-source manner in [3], with an iterative refinement method presented in [4], but an ARE-based solution presents a limitation due to the impossibility of solving singular problems. An alternative way which manages to solve such problems was introduced in [5], where AREs were replaced by Algebraic Riccati Inequalities (ARIs). ARIs are solved through Linear Matrix Inequalities (LMIs), while regular assumptions are no longer needed due to LMI system versatility. An open-source solver for Robust Control problems using LMIs is presented in [6].

The $\mathcal{H}_2/\mathcal{H}_\infty$ approach designs a suitable controller for the nominal plant, therefore only nominal stability and nominal performance are fulfilled. Additionally, generalized Robust Control framework allows to impose robust stability and robust performance, which cover the previous two aspects for an entire family of physical processes. As such, the μ -synthesis approach extends the \mathcal{H}_∞ optimization in order to obtain a robust controller for the uncertain plant which includes parametric and dynamic uncertainties [7,8]. μ -synthesis

is based on using structured singular values to quantify robustness margins and also on using linear fractional descriptions of the control problem containing the nominal plant model and uncertainty weighting functions. In [9], the authors present the μ -synthesis problem used with the so-called D - K iteration, which, in essence, provides two steps iteratively repeated until the robust performance stops improving: designing a \mathcal{H}_∞ control law and μ analysis on closed-loop system.

μ -synthesis is, however, a non-convex problem and the D - K iteration represents only an approximation, without any convergence guarantees. Another significant concern of D - K iteration is that the method generates high order controllers. In order to solve this issue, various approaches based on fixed structure controller are proposed in different papers [10–12]. The method presented in [10] uses nonsmooth techniques for \mathcal{H}_∞ synthesis. Then, using the same technique, the μ -synthesis was solved using D - K iteration and the result are presented in [13].

The main issue which appears when controller structure constraints are imposed is that the optimization problem is no longer convex and also, μ -synthesis is in general, considered nondeterministic polynomial time hard (NP-hard). A possible solution to that are swarm optimization algorithms. There are different approaches presented in papers [14–16] based on Genetic Algorithms (GA) and Particle Swarm Optimization (PSO). A solution for imposing a fixed structure controller, such as low-order or decentralized, was proposed in paper [14], which splits the problem in two parts: the convex part, solved using the classical ARE approach, and the non-convex part, solved using GA. The same authors proposed in [16] a new technique based on an evolutionary D - KD_0 iteration method, which combines the classical D -step with a KD_0 algorithm based on also a GA. Authors of the paper [15] propose an evolutionary approach to solve the μ -synthesis problem without order reduction by using an improved PSO.

Artificial Bee Colony (ABC) can also be used to solve complex optimization problems with constraints and could possibly outperform the other approaches and return the best solution in shorter execution time. The initial idea was presented in [17] as an extension of another metaheuristic algorithm, namely Honey Bee Swarm (HBS). The efficiency of the algorithm for several state-of-the-art optimization problems, along with an improvement for the stopping criterion, were underlined in [18].

One possible fixed structure controller is fractional-order proportional-integral-derivative (FO-PID). FO-PID is one of the most remarkable fractional order techniques with great interest in research [19]. It is used to generalize the classical PID control by adding extra degrees of freedom [20]. Compared to the integer PID controller, FO-PID brings the advantage of improving the robustness and providing better performance. In [21], the authors propose a FO-PID controller for a fractional-order plant model, presenting an analysis in both frequency and time domains, proving that achieving better control performance is one of the advantages of this approach. The FO-PID was used on a benchmark problem, i.e., the speed control of a DC motor [22], obtaining good results in terms of performance and robustness. Two generalized versions of Kessler's magnitude method with fractional order controllers were developed in [23,24]. A detailed comparison between classical modelling approaches and a fractional integrator approximation as a control baseline model is presented to the servo problem in [25]. A graphical method was developed in [26], while in [27] a fractional order internal model controller with event-based implementation was developed. Other applications comprise in an optimal FO-PID controller for a PMSM speed control, presented in [28], while a robust controller for a steam turbine was developed in [29].

In this paper, we present a new technique to design FO-PID robust controllers using μ -synthesis. The novelty of the current approach consists in implementing an algorithm able to find the nearly optimal values of the controller parameters using an artificial bee colony optimization. The cost function to be minimized is the \mathcal{H}_∞ norm of the closed-loop system, when stable, and, otherwise, a large value affinely dependent on the largest real part of the eigenvalues of the closed-loop state matrix. Therefore, the non-convex part

of the NP-hard μ -synthesis problem is solved using such a swarm optimization, while the D step is solved using the classical LMI technique. More than that, the `realp` object from MATLAB's Robust Control Toolbox embodies a limitation because it cannot be used as an exponent, necessary in approximating the fractional element with an integer-order system. Our approach manages to deal with this limitation in order to obtain the controller parameters, because the only information necessary in our approach is the range of the controller's parameters stored in such variables, which can be replaced with a simpler software object. Additionally, using our approach, a numerical example illustrates that the resulting controller manages to fulfill the robust stability and performance, while the controller obtained using unstructured μ -synthesis does not rigorously guarantee these specifications. Therefore, our method proposes a general framework able to synthesize arbitrary fixed structure fractional order controllers by optimizing their parameters in terms of robustness and performance, surpassing the well-established approach of manually tuning them for a desired problem, harnessing the Robust Control framework's design strongness.

We illustrate our proposed method on a benchmark problem: obtaining a controller for a two axis computer numerical control (CNC) system by solving a mixed-sensitivity μ -synthesis problem. Several control methods using the state space model of the machine are presented in literature. A comparison between the classical pole-placement method and Linear Quadratic Regulator (LQR) method is presented in [30]. For the LQR problem, an energy-based minimization algorithm was proposed, which proves to be suitable in terms of stability and robustness. As presented in [31], a different approach is recommended, in order to obtain a PI controller for each axis, using the state-feedback control algorithm. In this case, the state space model is augmented with an extra state which represents the integral of the position. The PI regulator parameters are obtained from the state-feedback gains.

The paper is organized in four sections. Section 2 introduces several ideas relevant to the proposed method, such as a mathematical foundation of FO-PID, continuing with the fundamental robust control problem based on μ -synthesis, and, finally, the ABC optimization algorithm. After that, the last subsection focuses on solving the non-convex problem of computing fixed structure controllers. Section 3 illustrates an application of the proposed method for position control of a two axis CNC machine, along with numerical results. In Section 4, the previously mentioned results are compared with those obtained using the well-established algorithms from MATLAB. Finally, conclusions are presented in Section 5.

2. Materials and Methods

In this section we present a controller synthesis procedure which manages to find a fixed structure fractional-order controller using the μ -synthesis technique from the Robust Control framework, where the non-convex subproblem involved in the classical D - K iteration is replaced by a swarm optimization algorithm. First, the mathematical background comprised in fractional-order control, robust control and artificial bee colony optimization is underlined, while the fourth subsection presents the proposed design procedure using all the mechanisms briefly described.

2.1. Fractional-Order Controller

The classical integer-order calculus was extended by Riemann and Liouville to fractional-order calculus by introducing the fractional integral operator [32]:

$$\mathcal{I}_a^\alpha \{f(t)\} = \frac{1}{\Gamma(\alpha)} \int_a^t f(\tau)(t-\tau)^{\alpha-1} d\tau, \quad (1)$$

where $\Gamma(\alpha) : \mathbb{C}_+ \rightarrow \mathbb{C}$ is the Euler Gamma function and the order of the integral operator is the complex parameter $\alpha \in \mathbb{C}_+$. This extension develops a new area of research in the

Control System domain. A commonly used definition of this operator was introduced in [33] as:

$$\mathcal{J}_c^\alpha \{f(t)\} = \frac{1}{\Gamma(\alpha)} \int_0^t (t - \tau)^{\alpha-1} f(\tau) d\tau = \mathcal{I}_0^\alpha \{f(t)\}, \tag{2}$$

with $t > 0$ and $\alpha \in \mathbb{R}_+$, having the Laplace transform [34]:

$$\mathcal{L}\{\mathcal{J}_c^\alpha \{f(t)\}\}(s) = s^{-\alpha} F(s), \tag{3}$$

where $F(s) = \mathcal{L}\{f(t)\}(s)$. One of the most common controller structures used in practice is the proportional-integral-derivative (PID) controller, having three degrees of freedom. Using fractional-order calculus, two new degrees of freedom can be added to a PID, having the notation $PI^\lambda D^\mu$. As such, the fractional order PID (FO-PID) has, as extra degrees of freedom, the order $\lambda \in \mathbb{R}_+$ of the integrator and the order $\mu \in \mathbb{R}_+$ of the differentiator, with the resulting transfer function:

$$H_c(s) = K_p + \frac{K_I}{s^\lambda} + K_D s^\mu. \tag{4}$$

The time domain expression of the command signal $c(t)$ can be expressed using the error signal $\varepsilon(t)$ as:

$$c(t) = K_p \cdot \varepsilon(t) + K_I \cdot \mathcal{J}_c^\lambda \{\varepsilon(t)\} + K_D \cdot \mathcal{J}_c^{-\mu} \{\varepsilon(t)\}. \tag{5}$$

One of the major issues of such a fractional element, i.e., \mathcal{J}_c^λ or $\mathcal{J}_c^{-\mu}$, is its implementation. In order to solve this problem, the Oustaloup recursive approximation (ORA) was introduced [35], and allows the approximation of the fractional-order element with an LTI system of pre-specified order N :

$$s^\lambda = \prod_{k=1}^N \frac{1 + s/\omega_{z,k}}{1 + s/\omega_{p,k}}, \tag{6}$$

where the frequency values of the singularities are obtained based on the desired fractional order $\lambda \in (0, 1)$, the integer order of the approximation N , along with the frequency range in which the approximation is valid $[\omega_l, \omega_u]$. Using the following two coefficients:

$$\varepsilon = \left(\frac{\omega_u}{\omega_l}\right)^{\frac{\lambda}{N}} \quad \text{and} \quad \eta = \left(\frac{\omega_u}{\omega_l}\right)^{\frac{1-\lambda}{N}}, \tag{7}$$

the above mentioned frequencies can be computed using:

$$\omega_{z,1} = \omega_l \sqrt{\eta}, \tag{8}$$

$$\omega_{p,n} = \omega_{z,n} \cdot \varepsilon, \quad n = \overline{1, N}, \tag{9}$$

$$\omega_{z,n+1} = \omega_{p,n} \cdot \eta, \quad n = \overline{1, N-1}. \tag{10}$$

For the rest of the possible real values of λ , the approximation can be easily extended as: for $\lambda \in (-1, 0)$ by inverting the relation (6), while for $|\lambda| \geq 1$, the components could be the integer part $[\lambda]$ and the fractional part $\{\lambda\}$, with the fractional part only approximated using (6).

2.2. Robust Control

The Robust Control framework assumes to minimize the $\mathcal{H}_2/\mathcal{H}_\infty$ norm of the lower linear fractional transformation (LLFT) of a plant P and a controller K :

$$P_o = \text{LLFT}(P, K) = P_{11} + P_{12}K(I - P_{22}K)^{-1}P_{21}, \tag{11}$$

where the plant P can be written, in general form, as:

$$P : \begin{pmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{pmatrix} = \left(\begin{array}{c|cc} A & B_w & B_u \\ \hline C_z & D_{zw} & D_{zu} \\ C_y & D_{yw} & D_{yu} \end{array} \right), \tag{12}$$

where the signals involved in the above relation will be further detailed in a more general context. One approach to solving $\mathcal{H}_2/\mathcal{H}_\infty$ problems is using AREs, which presents several limitations that are removed by the LMI approach. However, this framework can be used to ensure nominal stability and nominal performance only, but, the plant P must also contain an augmented model of the real process in which uncertainties are also present. There are two classic uncertainties types: *parametric*, represented by δI , where δ is the maximum bound of the parameter for a physical variable, and *unstructured*, represented by a full block $\Delta \in \mathbb{R}^{m \times m}$. The latter illustrates neglected or unknown dynamics uncertainties. In the mixed-scenario case, the following set is considered:

$$\Delta = \left\{ \Delta = \text{diag} \left(\delta_1 I_{n_1}, \dots, \delta_s I_{n_s}, \Delta_1, \dots, \Delta_f \right) \mid \delta_k \in \mathbb{R}, \Delta_j \in \mathbb{R}^{m_j \times m_j}, k = \overline{1, s}, j = \overline{1, f} \right\}. \tag{13}$$

In the Robust Control field, one of the main tools used for robustness analysis is the structured singular value, defined as follows.

Definition 1. For a square matrix $M \in \mathbb{C}^{N \times N}$ the structured singular value with respect to the set Δ is:

$$\mu_\Delta(M) = \frac{1}{\min_{\Delta \in \Delta} \{ \bar{\sigma}(\Delta) \mid \det(I - M\Delta) = 0 \}}, \tag{14}$$

if there exists $\Delta \in \Delta$ such that the matrix $I - M\Delta$ is rank deficient, otherwise 0.

For an LTI system described by the transfer matrix $M(s)$ and an upper linear fractional transformation (ULFT) connection shown in Figure 1 (left), the structured singular value $\mu_\Delta(M)$ can be defined as:

$$\mu_\Delta(M(s)) = \sup_{\omega \in \mathbb{R}_+} \mu_\Delta(M(j\omega)). \tag{15}$$

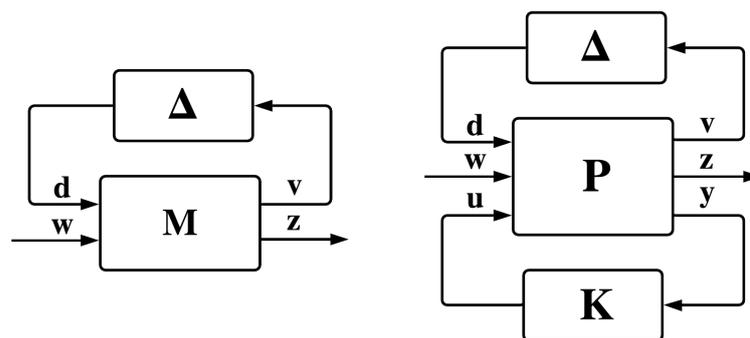


Figure 1. (Left) The generalized M - Δ structure containing the plant and the uncertainty block Δ . (Right) The closed-loop P - Δ - K structure containing the plant, controller and uncertainty block Δ .

Now, considering $M(s) = \text{LLFT}(P, K)(s)$ as being the lower linear fractional transformation (LLFT) between plant P and controller K , the connection illustrated in Figure 1 (right) results. The generalized plant structure is:

$$P_\Delta(s) = \begin{pmatrix} P_{vd}(s) & P_{vw}(s) & P_{vu}(s) \\ P_{zd}(s) & P_{zw}(s) & P_{zu}(s) \\ P_{yd}(s) & P_{yw}(s) & P_{yu}(s) \end{pmatrix} \Leftrightarrow P_\Delta : \begin{pmatrix} \dot{\mathbf{x}}(t) \\ \mathbf{v}(t) \\ \mathbf{z}(t) \\ \mathbf{y}(t) \end{pmatrix} = \left(\begin{array}{c|ccc} A & B_d & B_w & B_u \\ \hline C_v & D_{vd} & D_{vw} & D_{vu} \\ C_z & D_{zd} & D_{zw} & D_{zu} \\ C_y & D_{yd} & D_{yw} & D_{yu} \end{array} \right) \begin{pmatrix} \mathbf{x}(t) \\ \mathbf{d}(t) \\ \mathbf{w}(t) \\ \mathbf{u}(t) \end{pmatrix}, \tag{16}$$

where three types of input signals are present—the command input $\mathbf{u} \in \mathbb{R}^{n_u}$, the performance input $\mathbf{w} \in \mathbb{R}^{n_w}$, and the disturbance input $\mathbf{d} \in \mathbb{R}^{n_d}$ —and three types of outputs—the measurements vector $\mathbf{y} \in \mathbb{R}^{n_y}$, the performances vector $\mathbf{z} \in \mathbb{R}^{n_z}$, and the disturbance output $\mathbf{v} \in \mathbb{R}^{n_v}$.

Besides the well-known $\mathcal{H}_2/\mathcal{H}_\infty$ methods, a controller that meets robust stability and robust performance alike can be computed using the μ -synthesis framework. Robust stability implies that a specific controller manages to stabilize all the processes described by the upper linear fractional transformation (ULFT) presented in Figure 1 (left), while robust performance implies that the controller is able to impose the desired closed-loop performance in the worst case scenario. In order to have a mathematical guarantee that a controller K meets the robust stability and performance, the Main Loop theorem can be used. It implies that closed-loop system meets robust stability and performance if and only if the structural singular value of the LLFT of the plant and controller, with respect to Δ , fulfills:

$$\sup_{\omega \in \mathbb{R}_+} \mu_\Delta(\text{LLFT}(P, K)(j\omega)) < 1. \tag{17}$$

Therefore, the robust control problem can be written as:

$$\inf_{K \text{ stab.}} \sup_{\omega \in \mathbb{R}_+} \mu_\Delta(\text{LLFT}(P, K)(j\omega)), \tag{18}$$

which is not convex. More than that, the structural singular values are hard to be explicitly computed. In practice, there are various bounds which can be used to approximate the structural singular value. One of the most used approximations of the upper bound is in [9]:

$$\mu_\Delta(M) \leq \inf_{D \in \mathcal{D}} \bar{\sigma}(DMD^{-1}), \tag{19}$$

where $\bar{\sigma}$ denotes the largest singular value, and the set \mathcal{D} is defined as:

$$\mathcal{D} = \left\{ \text{diag}\left(D_1, \dots, D_s, d_1 I_{m_1}, \dots, d_f I_{m_f}\right) \mid D_k = D_k^\top \in \mathbb{R}^{n_k \times n_k}, d_j > 0, k = \overline{1, s}, j = \overline{1, f} \right\}. \tag{20}$$

Based on this upper bound, a good approximation of the initial non-convex problem can be employed by solving the following quasi-convex problem:

$$\inf_{K \text{ stab.}} \sup_{\omega \in \mathbb{R}_+} \inf_{D \in \mathcal{D}} \bar{\sigma}\left(D(j\omega) \cdot \text{LLFT}(P, K)(j\omega) \cdot (D(j\omega))^{-1}\right). \tag{21}$$

If the scaling factor, represented by the system D , is fixed, then the problem (21) is nothing but a \mathcal{H}_∞ optimization problem. On the other hand, fixing the controller K , the D scaling step can now be obtained by solving a Parrot problem for a desired frequency set $\Omega = \{\omega_1, \dots, \omega_N\}$ using the following generalized eigenvalue problem:

$$\begin{aligned} & \min \gamma, \\ \text{s.t. } & (\text{LLFT}(P, K)(j\omega_i))^* \cdot X \cdot \text{LLFT}(P, K)(j\omega_i) \leq \gamma^2 X, \end{aligned} \tag{22}$$

where from the solution $X = (D(j\omega_i))^* \cdot D(j\omega_i)$, the matrix $D(j\omega_i)$ can be extracted using a singular value decomposition. After all Parrot problems are solved, a minimum phase system is found in order to approximate the analytical solution $D(s)$. In summary, an iterative algorithm which solves the μ -synthesis problem starts by setting $D = I$, with the following steps applied successively:

1: Fix D and solve the \mathcal{H}_∞ optimal problem to find a controller K :

$$K = \arg \inf_{K \text{ stab.}} \|\text{LLFT}(P, K)\|_\infty. \tag{23}$$

2: Fix the controller K and solve the set of convex problems:

$$D(j\omega) = \arg \inf_{D \in \mathcal{D}} \bar{\sigma} \left(D \cdot \text{LLFT}(P, K)(j\omega) \cdot D^{-1} \right), \tag{24}$$

for a given frequency range Ω and, then, fit a stable minimum phase transfer matrix $D(s)$.

Steps 1 and 2 are executed in a loop sequence until the difference between two consecutive \mathcal{H}_∞ norms is less than a prescribed tolerance or the maximum number of iterations is reached.

2.3. Artificial Bee Colony Optimization

The artificial bee colony (ABC) optimization is a nature-inspired algorithm used to minimize a cost function:

$$f : D \rightarrow \mathbb{R}, \text{ where } D = [lb_1, ub_1] \times [lb_2, ub_2] \times \dots \times [lb_d, ub_d] \subset \mathbb{R}^d. \tag{25}$$

The ABC algorithm mimics the behaviour of real honeybees, where each food source represents a possible solution of the optimization problem described above. The location and the amount of nectar correspond to the design variables and the cost function, respectively. The bees are divided in two main groups: employed and unemployed bees, while the unemployed bees could be of two types as well: onlooker and scout bees. The employed bees are the ones that investigate the food source and return to the hive to inform the others by performing the waggle dance; the onlooker bees are the ones that watch the dance and decide whether or not a food source is worthy of being searched or not; the scout bees are former employed bees that have abandoned their previous food source, due to lack of nectar, and which now search for a new one. The Best Solution is represented by the food source, and the quality (or cost) of the solution is represented by the amount of nectar.

The number of employed bees coincides with number of the onlooker bees and represents the dimension of the swarm problem, denoted by N . The employed bees start the foraging process by randomly searching an initial position $\mathbf{x}_i^{(0)}$ in the domain D :

$$\mathbf{x}_i^{(0)} = \begin{pmatrix} lb_1 + \phi_{i,1}^{(0)} \cdot (ub_1 - lb_1) \\ lb_2 + \phi_{i,2}^{(0)} \cdot (ub_2 - lb_2) \\ \vdots \\ lb_d + \phi_{i,d}^{(0)} \cdot (ub_d - lb_d) \end{pmatrix} \in D, \tag{26}$$

where $\phi_{i,1,d}^{(0)} \in [-1, 1]$ are random numbers. After this initialization step, the first Best Solution appears.

Each employed bee searches a new food source based on the location of the current food source $\mathbf{x}_i^{(k)}$ and another food source $\mathbf{x}_j^{(k)}$ randomly selected. The new possible position is:

$$\bar{\mathbf{x}}_i^{(k)} = \text{sat} \left(\mathbf{x}_i^{(k)} + \phi \odot (\mathbf{x}_i^{(k)} - \mathbf{x}_j^{(k)}) \right), \tag{27}$$

where $\phi \in [-1, 1]^d$ is an array of random numbers, \odot is the element-wise multiplication, and sat is the saturation function that does not allow the position to be outside the searching domain D . Now, the position of the i th employed bee for the next iteration will be:

$$\mathbf{x}_i^{(k+1)} = \arg \min \{ f(\mathbf{x}_i^{(k)}), f(\bar{\mathbf{x}}_i^{(k)}) \}, \tag{28}$$

which means that an employed bee will never choose a source with less nectar. If the position for the next iteration will not be changed, the abandonment counter of the i th employed bee increments.

The onlooker bees use the information shared by each employed bee and choose a location around the position of the employed bees. The fitness value of a solution $\mathbf{x}_i^{(k)}$ is given by:

$$\log W(i) = \frac{-f(\mathbf{x}_i^{(k)})}{\frac{1}{N} \sum_j f(\mathbf{x}_j^{(k)})}, \tag{29}$$

and the probability that i th bee’s source will be selected by an onlooker bee is:

$$p_i = \frac{W(i)}{\sum_j W(j)}. \tag{30}$$

Now, using a roulette wheel selection method, the onlooker bee will choose a source i and, using the same searching technique as an employed bee, a new position is computed using (27). If the outlooker bee finds a better solution than the i th employed bee, they change their roles, otherwise the abandonment counter for the i th source increments. After this step, we have N employed bees and N unemployed bees. However, if the abandonment counter for the i th source exceeds a threshold, the i th employed bee becomes a scout and tries to find a new location using the relation (26).

After every loop corresponding to employed, outlooker and scout bees, it is checked if there is a food source with a solution better than the last one. The algorithm is over when the maximum number of cycles is reached or when there is no improvement of the Best Solution after a prescribed number of cycles.

2.4. Proposed Method

The solution of the problem described in (21) using the classical D – K iteration with $\mathcal{H}_2/\mathcal{H}_\infty$ framework leads to a high-order controller. As a solution to this issue, a fixed structure family of controllers \mathcal{K} can be considered, and the problem (21) becomes:

$$\inf_{\substack{\mathcal{K} \in \mathcal{K} \\ \mathcal{K} \text{ stab}}} \sup_{\omega \in \mathbb{R}_+} \inf_{D \in \mathcal{D}} \bar{\sigma} \left(D(j\omega) \cdot \text{LLFT}(P, K)(j\omega) \cdot (D(j\omega))^{-1} \right). \tag{31}$$

The new problem has the disadvantage of being non-convex in terms of the parameters of the proposed controller structure. However, the problem described in (31) will also be solved using a D – K iterative procedure, but the non-convex part of the problem has fewer degrees of freedom and it will be managed with an ABC optimization.

Starting with the model of the process G , a plant P is obtained after the augmentation step:

$$P : \begin{pmatrix} \dot{\mathbf{x}}(t) \\ \mathbf{z}(t) \\ \mathbf{y}(t) \end{pmatrix} = \begin{pmatrix} A & B_w & B_u \\ C_z & D_{zw} & D_{zu} \\ C_y & D_{yw} & D_{yu} \end{pmatrix} \begin{pmatrix} \mathbf{x}(t) \\ \mathbf{w}(t) \\ \mathbf{u}(t) \end{pmatrix}, \tag{32}$$

where the meaning of the inputs and outputs remains the same as in Section 2.2. Considering the advantages of the fractional-order controllers, the proposed structure of the controller is:

$$K^\theta(s) = \begin{pmatrix} C_{FO}^{(1,1)}(s) & C_{FO}^{(1,2)}(s) & \dots & C_{FO}^{(1,n_y)}(s) \\ C_{FO}^{(2,1)}(s) & C_{FO}^{(2,2)}(s) & \dots & C_{FO}^{(2,n_y)}(s) \\ \vdots & \vdots & \ddots & \vdots \\ C_{FO}^{(n_u,1)}(s) & C_{FO}^{(n_u,2)}(s) & \dots & C_{FO}^{(n_u,n_y)}(s) \end{pmatrix}, \tag{33}$$

where $C_{FO}^{(i,j)}$ is a fractional order controller from the i th input to the j th output and has the form:

$$C_{FO}^{(i,j)}(s) = K_P^{(i,j)} + \frac{K_I^{(i,j)}}{s^{\lambda^{(i,j)}}} + K_D^{(i,j)} s^{\mu^{(i,j)}}, \tag{34}$$

with the tunable parameters described by the vector:

$$\theta^{(i,j)} = \left(K_P^{(i,j)} \quad K_i^{(i,j)} \quad \lambda^{(i,j)} \quad K_D^{(i,j)} \quad \mu^{(i,j)} \right)^T \in D_{ABC}^{(i,j)} \subset \mathbb{R}^5. \tag{35}$$

Using these considerations, the desired family of the fixed structure controllers can be described as follows:

$$\mathcal{K} = \left\{ K^\theta(s) \mid \theta \in D_{ABC} \equiv D_{ABC}^{(1,1)} \times D_{ABC}^{(1,2)} \times \dots \times D_{ABC}^{(n_u, n_y)} \right\}, \tag{36}$$

where all parameters are stored in a single vector θ describing all degrees of freedom of the tunable controller. Using the ORA mechanism, each component $K^\theta(s) \in \mathcal{K}$ has a state-space representation:

$$K^\theta : \begin{pmatrix} \dot{\mathbf{x}}_K(t) \\ \mathbf{u}(t) \end{pmatrix} = \begin{pmatrix} A_K(\theta) & B_K(\theta) \\ C_K(\theta) & D_K(\theta) \end{pmatrix} \begin{pmatrix} \mathbf{x}_K(t) \\ \mathbf{y}(t) \end{pmatrix} \tag{37}$$

Next, we denote by P_o^θ the closed-loop system represented by the lower linear fractional transformation between the augmented plant P and controller K^θ , which can be represented as:

$$P_o^\theta = \text{LLFT}(P, K^\theta) : \begin{pmatrix} \dot{\mathbf{x}}_o(t) \\ \mathbf{z}_o(t) \end{pmatrix} = \begin{pmatrix} A_o(\theta) & B_o(\theta) \\ C_o(\theta) & D_o(\theta) \end{pmatrix} \begin{pmatrix} \mathbf{x}_o(t) \\ \mathbf{w}_o(t) \end{pmatrix}, \tag{38}$$

where state vector, input vector and output vector of the closed-loop system are:

$$\mathbf{x}_o = \begin{pmatrix} \mathbf{x} \\ \mathbf{x}_K \end{pmatrix}, \quad \mathbf{w}_o \equiv \mathbf{w} \quad \text{and} \quad \mathbf{z}_o \equiv \mathbf{z}. \tag{39}$$

In Algorithm 1 the main steps necessary to obtain the parameters of the controller having the structure (33) are presented. The inputs of the algorithm are the closed-loop plant P_o^θ , containing the tunable controller parameters θ , and the parameters α and β which describe the cost function that needs to be minimized using an ABC optimization, but, P_o^θ must also contain the varying parameters and unmodelled dynamics. Therefore, the plant P_Δ obtained after the augmentation step with uncertainties Δ has the form presented in (16). Thus, the first step made in Algorithm 1 consists in transforming the closed-loop system P_o^θ in the generalized closed-loop system $P_{o,\Delta}^\theta$ described as follows:

$$P_{o,\Delta}^\theta = \text{LLFT}(P_\Delta, K^\theta) : \begin{pmatrix} \dot{\mathbf{x}}_o(t) \\ \bar{\mathbf{z}}_o(t) \end{pmatrix} = \begin{pmatrix} \bar{A}_o(\theta) & \bar{B}_o(\theta) \\ \bar{C}_o(\theta) & \bar{D}_o(\theta) \end{pmatrix} \begin{pmatrix} \mathbf{x}_o(t) \\ \bar{\mathbf{w}}_o(t) \end{pmatrix}, \tag{40}$$

where the new extended performance inputs and outputs are:

$$\bar{\mathbf{w}}_o = \begin{pmatrix} \mathbf{d} \\ \mathbf{w}_o \end{pmatrix} \quad \bar{\mathbf{z}}_o = \begin{pmatrix} \mathbf{v} \\ \mathbf{z}_o \end{pmatrix} \tag{41}$$

As mentioned in Section 2.2, the structured singular value can be bounded using two D -scaling factors, one for the left and one for the right scaling, denoted D_L and D_R , respectively. As it can be notice in Figure 2, a new closed-loop plant \bar{P}_o^θ is obtained:

$$\bar{P}_o^{(\theta)}(s) = D_L(s) \cdot P_{o,\Delta}^\theta(s) \cdot D_R^{-1}(s), \tag{42}$$

having the state-space representation:

$$\bar{P}_o^\theta : \begin{pmatrix} \dot{\hat{\mathbf{x}}}_o(t) \\ \hat{\mathbf{z}}_o(t) \end{pmatrix} = \begin{pmatrix} \hat{A}_o(\theta) & \hat{B}_o(\theta) \\ \hat{C}_o(\theta) & \hat{D}_o(\theta) \end{pmatrix} \begin{pmatrix} \hat{\mathbf{x}}_o(t) \\ \hat{\mathbf{w}}_o(t) \end{pmatrix}. \tag{43}$$

All the above mentioned plants are presented in Figure 2. First, the generalized plant P_Δ has a LLFT connection with the controller K^θ , resulting the closed-loop plant $P_{o,\Delta}^\theta$, having the input vector \bar{w}_o and the output vector \bar{z}_o . After the D -scaling step, a new plant \bar{P}_o^θ is obtained, having the input vector \hat{w}_o and the output vector \hat{z}_o .

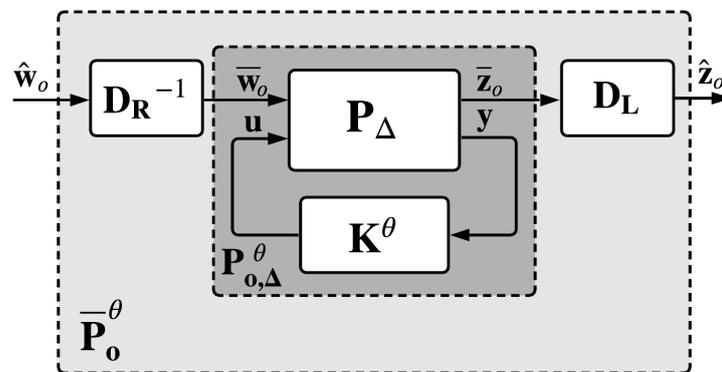


Figure 2. The augmented plant with uncertainties P_Δ in LLFT connection with controller K^θ forms the closed-loop plant $P_{o,\Delta}^\theta$. After each D -scale step, the plant used to find the controller parameters is \bar{P}_o^θ .

Before starting the while loop of Algorithm 1, an initialization of the generalized closed-loop plant with D -scale \bar{P}_o^θ is performed with the initial scale factors $D_L = I_{n_w}$ and $D_R = I_{n_z}$, as seen in line 2. As noticed in line 4, the K step is performed using this generalized plant having as degrees of freedom the tunable parameters θ . In order to compute the controller parameters θ^* , the ABC optimization will be used. The cost function to be minimized is:

$$f : D_{ABC} \rightarrow \mathbb{R}_+, f(\theta) = \begin{cases} \|\bar{P}_o^\theta\|_\infty, & \text{if } \bar{P}_o^\theta \text{ is stable} \\ \alpha \lambda_{max}(\hat{A}_o) + \beta, & \text{if } \bar{P}_o^\theta \text{ is unstable} \end{cases} \quad (44)$$

where the operator λ_{max} is defined by:

$$\lambda_{max} : \mathbb{R}^{M \times M} \rightarrow \mathbb{R}, \lambda_{max}(A) = \max\{\text{Re}(\lambda) \mid \lambda \in \Lambda(A)\}. \quad (45)$$

Algorithm 1: Fixed Structure μ -Synthesis.

Input: $P_o^\theta, \alpha, \beta$

Output: K^{θ^*}

- 1 get uncertain closed-loop plant $P_{o,\Delta}^\theta$ as in (40)
 - 2 set $D_L = I_{n_w}$ and $D_R = I_{n_z}$ and compute $\bar{P}_o^\theta = D_L \cdot P_{o,\Delta}^\theta \cdot D_R^{-1}$
 - 3 **while** $N_{iter} \leq MAX_ITER$ **and** *exists_improvement* **do**
 - 4 $\bar{P}_o^{\theta^*} = \text{computeKstep}(\bar{P}_o^\theta, \alpha, \beta)$
 - 5 update the uncertain plant $P_{o,\Delta}^{\theta^*}$
 - 6 $P_{o,nom}^{\theta^*} = \text{getNominalPlant}(P_{o,\Delta}^{\theta^*})$
 - 7 $[D_L, D_R] = \text{computeDstep}(P_{o,nom}^{\theta^*})$
 - 8 $\bar{P}_o^{\theta^*} = D_L \cdot P_{o,\Delta}^{\theta^*} \cdot D_R^{-1}$
 - 9 check if improvement exists and increase N_{iter}
-

The procedure `computeKstep` used to obtain the controller parameters is briefly presented in Algorithm 2 and will be described below. The inputs of the algorithm are the generalized closed-loop plant with D -scaling step \bar{P}_o^θ , along with the parameters α and β

which describe the cost function (44). The first step of this routine consists in computing the domain D_{ABC} of the cost function based on the constraints of the tunable parameters. The main limitation is represented by the fractional orders of the integral and derivative effects of the controller which must remain in $(0, 1)$, according to ORA.

In the second step of routine Algorithm 2, an initial population is created. Let N be the dimension of the swarm problem. This parameter can be given as input, but as a good practice, this can be chosen 100 times larger than the number of tunable parameters. The initialization step consists in randomly generating the positions $\theta_1^{(0)}, \theta_2^{(0)}, \dots, \theta_N^{(0)}$ of the food sources for each employed bee in the domain D_{ABC} using relation (26). After the initialization step, the first Best Solution (BS) is computed as:

$$\theta^* = \arg \min \{ f(\theta_i^{(k)}) \mid i = \overline{1, N} \} \Rightarrow BS = f(\theta^*). \tag{46}$$

Using this initial population, the main while loop starts. In line 4, the employed bees step is performed. In this step, each employed bee searches a new position around using relation (27), resulting N new possible positions for the next step, denoted by $\bar{\theta}_1^{(k)}, \bar{\theta}_2^{(k)}, \dots, \bar{\theta}_N^{(k)}$, and the proposed positions of the employed bees at step $k + 1$ will be:

$$\hat{\theta}_i^{(k+1)} = \arg \min \{ f(\theta_i^{(k)}), f(\bar{\theta}_i^{(k)}) \}, i = \overline{1, N}. \tag{47}$$

If, for a specific food source i , the proposed position coincides with the previous position, the abandonment counter increments.

Using the proposed solutions of the above step, each onlooker bee selects a new possible solution based on the roulette wheel selection mechanism and relation (27), resulting a new set of proposed solutions: $\tilde{\theta}_1^{(k)}, \tilde{\theta}_2^{(k)}, \dots, \tilde{\theta}_N^{(k)}$. If the i th onlooker bee has a better solution than i th employed bee, they exchange their roles, otherwise the abandonment counter for the i th food source increments. After this step, the new set of proposed positions for the employed bees are:

$$\hat{\theta}_i^{(k+1)} = \arg \min \{ f(\tilde{\theta}_i^{(k)}), f(\hat{\theta}_i^{(k+1)}) \}, i = \overline{1, N}. \tag{48}$$

Algorithm 2: Compute K Step.

Input: $\bar{P}_o^\theta, \alpha, \beta$

Output: $\bar{P}_o^{\theta^*}$

- 1 compute the domain D of the optimization problem
 - 2 create an initial population of employed bees using (26) and select the first Best Solution
 - 3 **while** $N_{cycles} < MNC$ **and** $exists_improvement$ **do**
 - 4 perform the employed bees step
 - 5 perform the outlooker bees step
 - 6 perform the scout bees step
 - 7 find the new Best Solution, check if improvements exist and increase N_{cycles}
 - 8 **end**
-

After performing the outlooker bees step from line 5, the abandonment counter for each active food source will be interrogated. If the abandonment counter of the i th position exceeds a prescribed threshold, denoted by LIMIT, the employed bee becomes a scout bee and its new position is obtained using (26). As a good practice, this paper proposed an improved mechanism of converting employed bees in scout bees. If the value of the cost function at the proposed position $f(\hat{\theta}_i^{(k+1)})$ is over β , this means that, in the case of a good calibration of parameters α and β , the solution corresponds to an unstable closed-loop

system and can be dropped. The positions of the employed bees for the next iteration become $\theta_1^{(k+1)}, \theta_2^{(k+1)}, \dots, \theta_N^{(k+1)}$.

The last step of the main while loop, presented in line 7, consists in computing the Best Solution after this new iteration:

$$\theta^* = \arg \min \{ f(\theta_{old}^*), f(\theta_1^{(k)}), f(\theta_2^{(k)}), \dots, f(\theta_N^{(k)}) \} \Rightarrow BS = f(\theta^*), \tag{49}$$

and, then, checking if there exist any improvements after this step. In order to have a good trade-off between execution time and solution accuracy, it can be useful to establish a threshold for the number of steps when no improvement appears in Best Solution and mark if there exists such an improvement. Being a metaheuristic optimization algorithm, the runtime can be made deterministic and theoretically finite by imposing the variable *MNC*, without convergence guarantees. In practice, such methods have been successfully employed in various global optimization problems and, being that it is an offline optimization, it is not problematic for our approach.

Returning to Algorithm 1, after the *K* step is performed, a new parameter vector θ^* results, which leads to a new generalized closed-loop plant $\bar{P}_o^{\theta^*}$ and to a new uncertain plant $P_{o,\Delta}^{\theta^*}$. The closed-loop plant $P_{o,\Delta}^{\theta^*}$ is used to compute the next *D*-scale factors. From $P_{o,\Delta}^{\theta^*}$ we extract a nominal plant $P_{o,nom}^{\theta^*}$ by fixing the tunable parameters of the controller with the values determined in step 4.

The `computeDstep` routine from line 7 receives as input this nominal plant and returns the left and the right *D*-scale factors. Based on the poles and the transmission zeros of the nominal plant $P_{o,nom}^{\theta^*}$, a set $\Omega = \{\omega_1, \omega_2, \dots, \omega_F\}$ of frequencies is generated. Then, we need to get the frequency response data for each scaling factor by solving the following generalized eigenvalue problem:

$$\begin{aligned} & \min \gamma, \\ \text{s.t. } & \bar{\sigma} \left(D_L(j\omega_i) \cdot P_{o,nom}^{\theta^*}(j\omega_i) \cdot D_R^{-1}(j\omega_i) \right) < \gamma, \end{aligned} \tag{50}$$

for each $i = \overline{1, F}$, which is nothing but a Parrot problem which can be solved point by point using LMI techniques, as mentioned in Section 2.2. Once the frequency response data points are obtained for each value in Ω , we need to fit two minimum phase systems, one for each scaling factor, then perform the *D*-scaling step, giving a new generalized closed-loop system:

$$\bar{P}_o^{\theta^*}(s) = D_L(s) \cdot P_{o,\Delta}^{\theta^*}(s) \cdot D_R^{-1}(s). \tag{51}$$

In a similar manner with the `computeKstep` routine, there are possible stopping criteria which can be used. First, a threshold for the maximum number of *D*-*K* iterations can be imposed. Another important stopping condition appears if the upper bound of the structural singular value is less than 1, because this fact already guarantees that the controller ensures robust stability and robust performance. In accordance with the allowed maximum number of steps, a stopping criterion could be to check if there are any improvements after a certain number of steps.

3. Numerical Results

In this section we illustrate how the proposed method can be used on a benchmark problem. The process is represented by a Computer Numerical Control (CNC) machine with two orthogonal axis which are operated by two servo DC motors. A Trio Motion Coordinator family controller was used for the CNC motors. The programming language was Trio Basic, which provides various functions such as linear, circular and helical interpolation, variable speed and acceleration profile functions and control functions to ensure smooth and synchronized motions.

A mathematical model for each axis was determined on the basis of measured data: angular speeds ω_x and ω_y , and angular positions θ_x and θ_y . The state space mathematical model of the machine is described as follows:

$$G : \begin{pmatrix} \dot{\omega}_x \\ \dot{\theta}_x \\ \dot{\omega}_y \\ \dot{\theta}_y \\ \theta_x \\ \theta_y \end{pmatrix} = \begin{pmatrix} -\frac{1}{T_{mx}} & 0 & 0 & 0 & \frac{K_{mx}}{T_{mx}} & K_{xy} \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{1}{T_{my}} & 0 & K_{yx} & \frac{K_{my}}{T_{my}} \\ 0 & 0 & 1 & 0 & 0 & 0 \\ \hline 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} \omega_x \\ \theta_x \\ \omega_y \\ \theta_y \\ u_x \\ u_y \end{pmatrix}, \tag{52}$$

where the state vector is $\mathbf{x} = (\omega_x \ \theta_x \ \omega_y \ \theta_y)^\top$, the input vector is $\mathbf{u} = (u_x \ u_y)^\top$ and the output vector is $\mathbf{y} = (\theta_x \ \theta_y)^\top$. The model parameters, along with their nominal values and uncertainty range are detailed in Table 1.

Table 1. Nominal values and uncertainty ranges of the CNC model parameters.

Parameter	Nominal Value	Uncertainty Range	Parameter	Nominal Value	Uncertainty Range
T_{mx}	0.02448	$\pm 10\%$	T_{my}	0.01139	$\pm 10\%$
K_{mx}	25.8017	$\pm 10\%$	K_{my}	25.1494	$\pm 10\%$
K_{xy}	26.65	$\pm 10\%$	K_{yx}	24.46	$\pm 10\%$

For this control problem a mixed-sensitivity loop shaping technique is used, which provides a good trade-off between performance and robustness. In order to use this technique, a new plant model will be obtained after the augmentation process, as in Figure 3. The performance inputs for the resulting augmented plant P are the references for both axis $\mathbf{w} \equiv \mathbf{r} = (r_x \ r_y)^\top$. For the augmentation procedure, the closed-loop transfer functions need to be weighted from the reference signals \mathbf{r} to their corresponding error signals \mathbf{e} , output signals \mathbf{y} and command signals \mathbf{u} , which are named: sensitivity function $S = (I + GK)^{-1}$, complementary sensitivity function $T = I - S$ and control effort KS , respectively. The performance output vector is composed from the weighted outputs of these three vector-valued functions:

$$\mathbf{z} = \begin{pmatrix} \mathbf{z}_S \\ \mathbf{z}_T \\ \mathbf{z}_{KS} \end{pmatrix}, \text{ where } \mathbf{z}_S = \begin{pmatrix} z_{S,x} \\ z_{S,y} \end{pmatrix}, \mathbf{z}_T = \begin{pmatrix} z_{T,x} \\ z_{T,y} \end{pmatrix} \text{ and } \mathbf{z}_{KS} = \begin{pmatrix} z_{KS,x} \\ z_{KS,y} \end{pmatrix}. \tag{53}$$

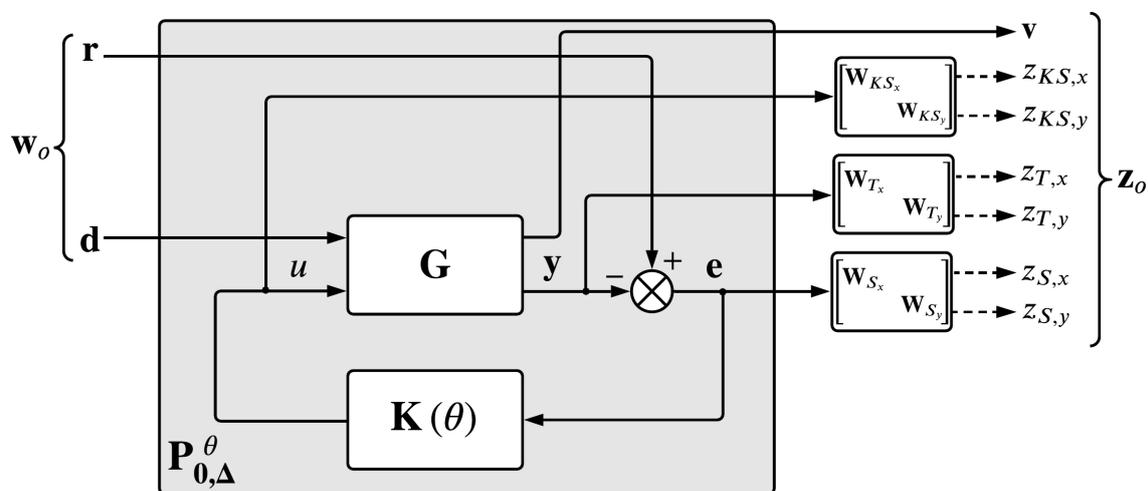


Figure 3. Closed-loop augmented plant.

In order to ensure good disturbance rejection, good reference tracking and stabilization of an unstable plant, the sensitivity function must have small magnitude, which means that the magnitude of the open loop must be large. On the other hand, in order to ensure mitigation of measurement noise and robust stability, the complementary sensitivity function must have small magnitudes, which implies that the magnitude of the open loop system must be small. More than that, the command signal must have small magnitude, which implies that the control effort transfer function must also have small magnitude. However, although these requirements seem to be conflicting, the frequency range where the magnitude of the open loop must be small and high are mostly disjunctive: the magnitude should be high for low frequencies and should be low for high frequencies. In order to ensure the desired shape of these three functions, three weighting functions must be designed, respectively.

The sensitivity function is a very good indicator of the closed-loop system tracking performance and has the advantage of being sufficient to consider only the magnitude. Typical specifications for sensitivity weighting functions are: minimum bandwidth frequency ω_B^* , maximum steady-state error A and maximum peak magnitude M , imposed by the following model [36]:

$$W_S(s) = \frac{1}{M} \frac{s + \omega_B^*}{s + \omega_B^* A}. \tag{54}$$

In a similar manner, the weighting function for the complementary sensitivity must be designed using the following specifications: the maximum peak amplitude M_T , the maximum value for high frequencies A_T , the minimum bandwidth ω_{BT}^* and the roll-off n , formulated as:

$$W_T(s) = \frac{(s + \omega_{BT}^*)^n}{(A_T^{1/n} s + \omega_{BT}^* M_T^{1/n})^n}. \tag{55}$$

For the control effort weighting function, the main performance specifications are the maximum value of the magnitude at low and high frequencies, denoted by *DC* and *HF*, respectively, and an intermediate point of interest. Sometimes, the main goal is simply to maintain the command signal under a prescribed value due to physical limitations of the system or other causes.

The major advantage of this approach consists in sculpting the relevant loop functions to impose performances implying good tracking and dynamic behaviour. Of great use are the rise time limitation through ω_B , steady-state error through A , while the roll-off slope of the closed-loop system imposed using n is directly coupled with sensor noise characteristics. These performances are specified for different frequency ranges, using the adequately selected weighting functions presented above.

Being a MIMO system with two inputs and two outputs, all weighting functions must be described by 2×2 transfer matrices, but, following a standard decoupling procedure, the weighting functions will be 2×2 diagonal transfer matrices. For the sensitivity, we consider two nearly similar weighting transfer functions, one for each axis, having the maximum bandwidth $\omega_{B,x}^* = 3$ [rad/s] and $\omega_{B,y}^* = 5$ [rad/s], the maximum steady-state error $A_x = A_y = 10^{-2}$ and the desired maximum sensitivity peak $M_x = M_y = 1.5$, resulting in:

$$W_S(s) = \begin{pmatrix} W_{S,x}(s) & 0 \\ 0 & W_{S,y}(s) \end{pmatrix}, \text{ where } W_{S,x}(s) = \frac{0.6667s + 3}{s + 0.03} \text{ and } W_{S,y}(s) = \frac{0.6667s + 5}{s + 0.05}. \tag{56}$$

For the complementary sensitivity weighting function, the same 2×2 diagonal structure approach will be used, imposing the same parameters for both axis: maximum complementary bandwidth $\omega_{BT,x}^* = 50$ [rad/s], maximum peak magnitude $M_{T,x} = M_{T,y} = 1.5$,

maximum magnitude at high frequencies $A_{T,x} = A_{T,y} = 10^{-2}$ and roll-off $n_x = n_y = 1$, resulting in:

$$W_T(s) = \begin{pmatrix} W_{T,x}(s) & 0 \\ 0 & W_{T,y}(s) \end{pmatrix}, \text{ where } W_{T,x}(s) = W_{T,y}(s) = \frac{s + 50}{0.01s + 75}, \quad (57)$$

while the control effort weighting function being designed to encompass only the physical limitation of the command signal (between -1 and 1), resulting the transfer matrix:

$$W_{KS}(s) = \begin{pmatrix} W_{KS,x}(s) & 0 \\ 0 & W_{KS,y}(s) \end{pmatrix}, \text{ where } W_{KS,x}(s) = W_{KS,y}(s) = 1. \quad (58)$$

The proposed structure of the controller is a decentralized one with two $PI^\lambda D^\mu$ controllers:

$$\mathcal{K} = \left\{ K^\theta(s) = \begin{pmatrix} K_{P,x} + K_{I,x}s^{-\lambda_x} + K_{D,x}s^{\mu_x} & 0 \\ 0 & K_{P,y} + K_{I,y}s^{-\lambda_y} + K_{D,y}s^{\mu_y} \end{pmatrix} \mid \theta \in D \subset \mathbb{R}^{10} \right\}, \quad (59)$$

where the tunable parameters are:

$$\theta = (K_{P,x} \ K_{I,x} \ \lambda_x \ K_{D,x} \ \mu_x \ K_{P,y} \ K_{I,y} \ \lambda_y \ K_{D,y} \ \mu_y)^\top. \quad (60)$$

The problem to be solved with the proposed method is the mixed-sensitivity fixed structure μ -synthesis one, described as:

$$\begin{aligned} \min_{\substack{K \in \mathcal{K} \\ K \text{ stab}}} \sup_{\omega \in \mathbb{R}_+} \inf_{D \in \mathcal{D}} \bar{\sigma} \left(D(j\omega) \cdot \text{LLFT}(P, K)(j\omega) \cdot (D(j\omega))^{-1} \right), \\ \text{s.t. } \left\| \begin{matrix} W_S S \\ W_T T \\ W_{KS} K S \end{matrix} \right\|_\infty < 1. \end{aligned} \quad (61)$$

The settings for `computeKstep` used in the experiments are: the swarm dimension $N = 1000$, the maximum number of cycles $MNC = 50$, the maximum number of cycles with no improvements $NOIMP = 10$, the limit for the abandonment counter $LIMIT = 10$. The parameters necessary to describe the cost function (44) are $\alpha = 1$ and $\beta = 10^5$. The maximum number of D - K iterations is $MAX_ITER = 10$ and the maximum window length for assessing lack of progress is 4. Using this setup, the mixed-sensitivity fixed structure μ -synthesis problem (61) is solved using four D - K iterations, as noticed in Table 2. The fractional order controller has been approximated using ORA with the following parameters: the frequency range is $[\omega_l, \omega_u] = [1 \times 10^{-4}, 1 \times 10^3]$ [rad/s] and the order of the approximation is $N = 3$. Given that, the resulting controller is:

$$K^{\theta^*}(s) = \begin{pmatrix} 0.01 + 7.1116 \cdot s^{-0.7648} + 0.1133 \cdot s^{0.0909} & 0 \\ 0 & 0.1464 + 10 \cdot s^{-0.9926} + 0.1344 \cdot s^{0.0549} \end{pmatrix}. \quad (62)$$

Table 2. The evolution of the structural singular value in the D - K iteration procedure used to solve the mixed-sensitivity fixed structure μ -synthesis problem for the case study—FO-PID structure.

D-K Iteration Number	1	2	3	4
Number of ABC Iterations	50	39	50	45
Peak Value of μ	1.8956	1.0185	1.0021	0.9822

The imposed shape of the sensitivity, complementary sensitivity and control effort functions for both axis are depicted in Figure 4, along with the obtained shapes of those functions with the resulting controller for 100 Monte Carlo simulations. It can be noticed that all resulting Bode diagrams are under the prescribed shapes, which is guaranteed by the fact that the upper bound of the structured singular value $\mu_\Delta(P_{o,\Delta}^{\theta^*})$ is less than 1. The

time-domain performances of the systems $r_x \rightarrow \theta_x$ and $r_y \rightarrow \theta_y$ are depicted in Figure 5, which are correlated with the frequency-domain performances. As such, the minimum values of the bandwidths for both x and y axis are over $\omega_{B,x}^* = 3$ [rad/s] and $\omega_{B,y}^* = 5$ [rad/s], which means that the rise time is less than 0.33 [s] and 0.2 [s], respectively. For the system $r_x \rightarrow \theta_x$, the rise time of the nominal system is 0.248 [s] and varies from 0.227 [s] to 0.281 [s], while the settling time is 0.558 [s] and varies from 0.496 [s] to 0.664 [s]. On the other hand, for the system $r_y \rightarrow \theta_y$, the rise time of the nominal system is 0.211 [s] and varies from 0.19 [s] to 0.235 [s], having the settling time for the nominal system 0.405 [s] and varying from 0.363 [s] to 0.454 [s]. According to the shape of the actual obtained sensitivity functions, there is no overshoot and no steady-state error for neither of the experiments presented using Monte Carlo simulations. Moreover, the reciprocal axis influence is small, as resulted from numerical simulations, where the peak amplitude from r_y to θ_x varies from 0.01 to 0.0143, and the peak amplitude from r_x to θ_y varies from 3.55×10^{-3} to 4.97×10^{-3} , respectively.

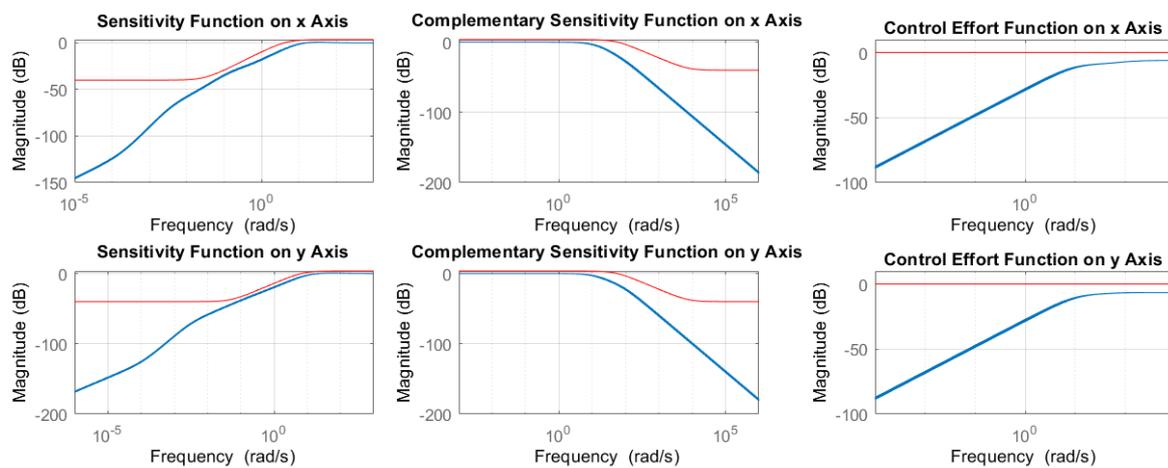


Figure 4. Imposed shapes of the sensitivity, complementary sensitivity and control effort frequency responses for both axis, along with 100 Monte Carlo simulations with FO-PID closed-loop systems.

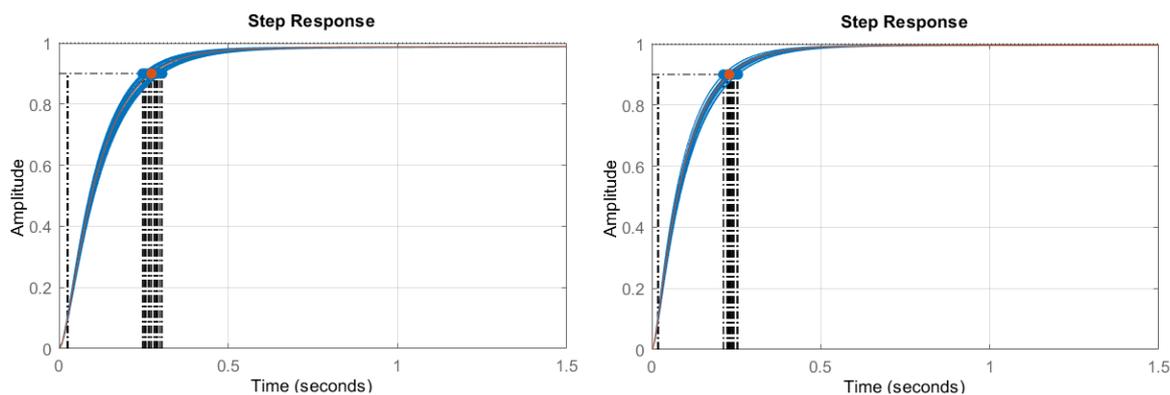


Figure 5. Time-domain closed-loop FO-PID performances for $r_x \rightarrow \theta_x$ and $r_y \rightarrow \theta_y$ systems.

4. Discussion

As noticed, the resulting controller contains a small fractional D term, which may be negligible. As such, the controller may be resynthesized with an imposed diagonal PI^λ structure:

$$\mathcal{K} = \left\{ K^\theta(s) = \begin{pmatrix} K_{P,x} + K_{I,x}s^{-\lambda_x} & 0 \\ 0 & K_{P,y} + K_{I,y}s^{-\lambda_y} \end{pmatrix} \middle| \theta \in D \subset \mathbb{R}^6 \right\}, \quad (63)$$

where the tunable parameters are:

$$\theta = (K_{P,x} \quad K_{I,x} \quad \lambda_x \quad K_{P,y} \quad K_{I,y} \quad \lambda_y)^\top. \tag{64}$$

The control problem remains the same as in (61), maintaining the constraints as in (56)–(58). The settings for this experiment were kept the same as for the previous case: $N = 1000$, $MNC = 50$, $NOIMP = 10$, $LIMIT = 10$, $\alpha = 1$, $\beta = 10^5$, $MAX_ITER = 10$, $[\omega_l, \omega_u] = [1e-4, 1e3]$ [rad/s] and $N = 3$. Using this setup, the mixed-sensitivity fixed structure μ -synthesis problem (61) is solved using four D - K iterations, as noticed in Table 3. The resulting controller is:

$$K_{FO-PI}^{\theta^*}(s) = \begin{pmatrix} 0.2229 + 5.9392 \cdot s^{-0.7792} & 0 \\ 0 & 0.3949 + 10 \cdot s^{-0.9733} \end{pmatrix}. \tag{65}$$

Table 3. The evolution of the structural singular value in the D - K iteration procedure used to solve the mixed-sensitivity fixed structure μ -synthesis problem for the case study—FO-PI structure.

D - K Iteration Number	1	2	3	4
Number of ABC Iterations	30	30	24	30
Peak Value of μ	3.0676	1.1961	1.0026	0.9959

The imposed shape of the sensitivity, complementary sensitivity and control effort functions for both axis are depicted in Figure 6, along with the obtained shapes of those functions with the resulting controller for 100 Monte Carlo simulations. It can be noticed that all resulting Bode diagrams are under the prescribed shapes, which is guaranteed by the fact that the upper bound of the structured singular value $\mu_\Delta(P_{o,\Delta}^{\theta^*})$ is less than 1.

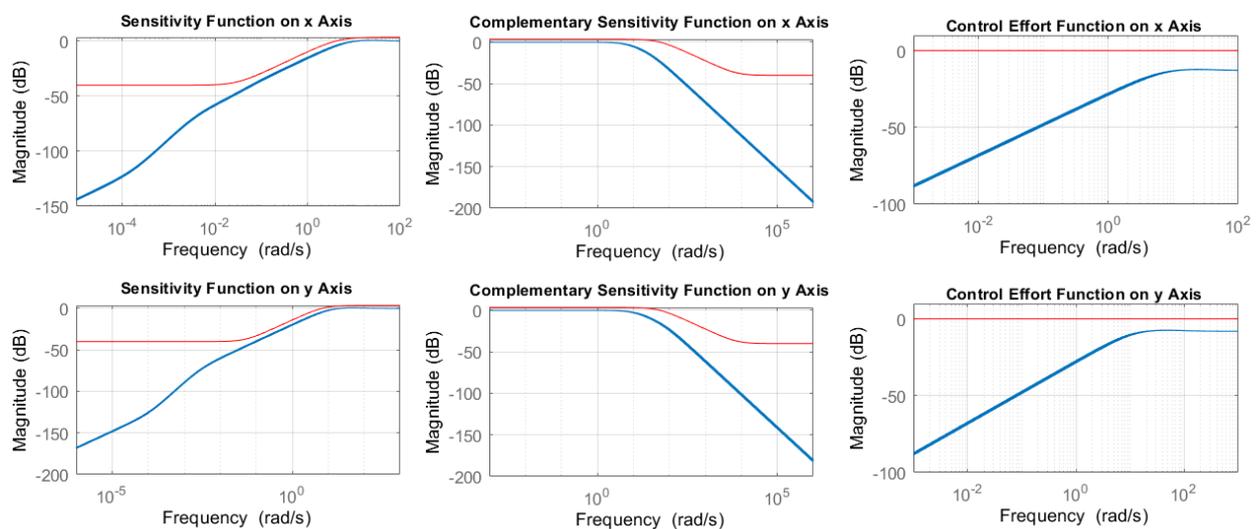


Figure 6. Imposed shapes of the sensitivity, complementary sensitivity and control effort frequency responses for both axis, along with 100 Monte Carlo simulations with actual FO-PI closed-loop systems.

The proposed method results are compared with those obtained using the *musyn* routine from MATLAB [37]. The *musyn* routine manages to solve both fixed structure and free-structure μ -synthesis control problems. The first solution of the problem (61) is obtained using the proposed method with results already presented in Section 3. Starting from the same control problem, the structured μ -synthesis version of the *musyn* routine with the same stopping criteria was used. The resulting controller after 2 iterations achieved its best performance $\mu_\Delta(P_{o,\Delta}^{\theta^*}) = 0.9842$, which means that there is a mathematical guarantee for robust stability and robust performance. The transfer matrix of this controller is:

$$K^{\theta^*}(s) = \begin{pmatrix} 0.0101 + 7.4689 \cdot s^{-0.7860} + 0.1729 \cdot s^{0.0542} & 0 \\ 0 & 0.0881 + 5.4840 \cdot s^{-0.9749} + 0.2031 \cdot s^{0.0348} \end{pmatrix}. \quad (66)$$

In opposition with the above approaches, the controller obtained with μ -synthesis procedure from MATLAB without imposing a fixed structure is of order 74 and, after 10 iterations, the best achieved performance is $\mu_{\Delta}(P_{o,\Delta}^{\theta^*}) = 1.003$. The frequency-domain data for structured singular values corresponding to these three numerical simulations are presented in Figure 7. As mentioned, the peak value of the μ values for the unstructured problem is over the critical value 1, while the FO-PID controller manages to fulfill all requirements. A comparison between the frequency responses of the controllers is shown in Figure 8.

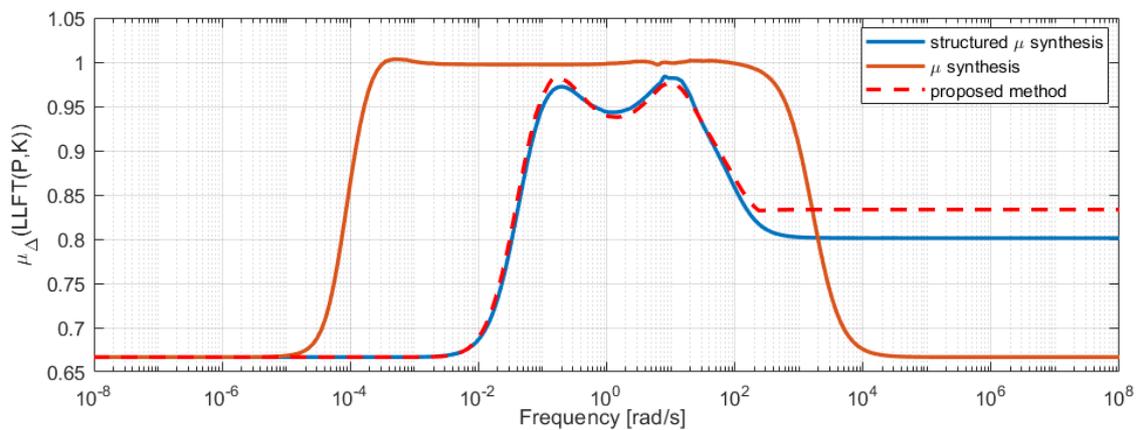


Figure 7. Comparison between structured singular values’ frequency response obtained with the structured μ -synthesis from MATLAB (blue), with the unstructured μ -synthesis from MATLAB (orange) and with the proposed method (red), respectively.

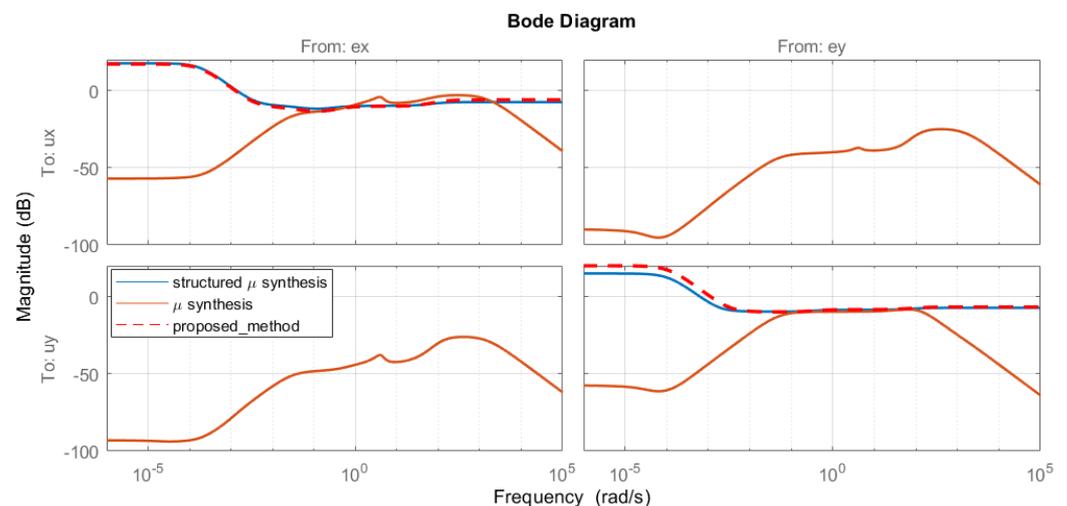


Figure 8. Obtained controllers’ frequency responses with the structured μ -synthesis from MATLAB (blue), unstructured (orange) μ -synthesis from MATLAB, and with the proposed method (red).

The integer-order approximation of a fractional element using ORA contains two exponential terms, as seen in (7), which presently cannot be treated using the `realp` objects in MATLAB. The actual solution used for this paper is to approximate the exponential function using Taylor series truncation. As stated in the Introduction, our approach requires only the range of the controller parameters and, thus, another object could be used instead of the entire structure of `realp` necessary in nonsmooth optimization-based algorithms used in MATLAB `hinfstruct` and `systeme` routines. Based on this limitation, the toolbox presented in [38] will be extended in order to address the previously mentioned

mathematical issues, due to the fact that only the fixed-structure \mathcal{H}_∞ synthesis algorithm uses the `realp` object, which can be replaced by our approach, where the NP-hard non-convex problem is solved using an ABC swarm optimization.

Compared to other approaches previously described in the introduction, where LQR methods were considered for CNC machines, the advantages of the proposed method consist in the generality of the method and the flexibility of the robustness, with the possibility to compensate measurement noise, unmodelled dynamics and input-output disturbances. Therefore, while LQR requires the complete state vector to be measured at runtime, the proposed method requires only the provided measurements, modelled by the actual outputs of the system. Although the LQR method can be augmented with a state estimator in order to obtain output feedback, the main limitation of this approach is that the model of the plant must be accurate, while in the Robust Control framework, utilized in our method, an entire family of uncertain plants can be taken into consideration at the design phase. Moreover, it is difficult to impose exact limitations on the maximum allowed command signals, using the energy-based approach, which generally is an intrinsic limitation of the execution element.

5. Conclusions and Future Work

The current paper presents a new design method for fixed structure fractional-order controllers using the Robust Control framework. The proposed method manages to return nearly optimal parameters of a MIMO FO-PID as a solution for a mixed-sensitivity fixed structure μ -synthesis control problem. Although the μ -synthesis control problem is NP-hard, the D - K iteration algorithm represents a good approximation which allows to convert it into a P-hard problem. However, the returned controller is of high order, which means that an order reduction must be performed in order to implement the control law. Therefore, the imposed structure is an increasingly explored approach, although such a problem presents a non-convex component for the K step. Our approach consists of an artificial bee colony swarm optimization as a solution to this non-convex fixed structure \mathcal{H}_∞ subproblem. This solution requires only the range of the controller parameters, as opposed to the nonsmooth optimization-based approach from MATLAB's Robust Control Toolbox, where the parameters can be used only in polynomial structured expressions, which is an inherent limitation when fractional-order controllers are desired. Further, the case study of mixed-sensitivity μ -synthesis position control problem for both axis of a CNC machine manages to underline the strong points of the method and of the imposed structure of the controller: it provides a mathematical guarantee for robust stability and robust performance, while the unstructured version of μ -synthesis from MATLAB does not manage to offer it.

As future work, we want to include our method in a toolbox, as stated in the Discussion section, which starts from an initial process model and a set of desired performances, and manages to automatically obtain the augmented plant, the controller decoupling transfer matrix, the optimal values of the controller parameters, followed by closed-loop simulations and validations. Additionally, we propose to extend this technique for certain types of nonlinear systems.

Author Contributions: Conceptualization, V.M. and M.Ş.; methodology, V.M.; software, V.M.; validation, M.Ş. and M.S.; formal analysis, V.M. and P.D.; investigation, V.M. and D.M.; resources, V.M.; data curation, D.M. and M.S.; writing—original draft preparation, M.V., D.M. and M.Ş.; writing—review and editing, M.Ş., D.M., M.S. and P.D.; visualization, M.Ş.; supervision, P.D.; project administration, V.M.; funding acquisition, D.M. All authors have read and agreed to the published version of the manuscript.

Funding: This paper was supported by the Project “Entrepreneurial competences and excellence research in doctoral and postdoctoral programs—ANTREDOC”, project co-funded by the European Social Fund.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

Abbreviations

The following abbreviations are used in this manuscript:

ABC	Artificial Bee Colony
CNC	Computer Numerical Control
FO-PID	Fractional-Order PID
LMI	Linear Matrix Inequality
LLFT	Lower Linear Fractional Transform
LTI	Linear Time-Invariant
MIMO	Multiple-Input Multiple-Output
NP	Nondeterministic Polynomial Time
PID	Proportional-Integral-Derivative
ULFT	Upper Linear Fractional Transform

References

- Doyle, J.; Glover, K.; Khargonekar, P.; Francis, B.A. State-space solutions to standard \mathcal{H}_2 and \mathcal{H}_∞ control problems. *IEEE Trans. Autom. Control* **1989**, *34*, 831–847. [\[CrossRef\]](#)
- Ionescu, V.; Oară, C.; Weiss, M. *Generalized Riccati Theory and Robust Control—A Popov Function Approach*; John Wiley & Sons: Chichester, UK, 1999.
- Șuşcă, M. Solving Algebraic Riccati Equations Using Proper Deflating Subspaces for $\mathcal{H}_2/\mathcal{H}_\infty$ Synthesis. Master's Thesis, Technical University of Cluj-Napoca, Cluj-Napoca, Romania, 2019.
- Șuşcă, M.; Mihaly, V.; Stănescu, M.; Dobra, P. Iterative Refinement Procedure for Solutions to Algebraic Riccati Equations. In Proceedings of the 2020 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR), Cluj-Napoca, Romania, 21–23 May 2020.
- Gahinet, P.; Apkarian, P. A linear matrix inequality approach to \mathcal{H}_∞ control. *Int. J. Robust Nonlinear Control* **1994**, *4*, 421–448. [\[CrossRef\]](#)
- Mihaly, V. General Purpose Linear Matrix Inequality Solver with Applications in Robust and Nonlinear Control. Master's Thesis, Technical University of Cluj-Napoca, Cluj-Napoca, Romania, 2020.
- Liu, K.Z.; Yao, Y. *Robust Control—Theory and Applications*; John Wiley & Sons: Singapore, 2016.
- Gu, D.-W.; Petkov, P.H.; Konstantinov, M.M. *Robust Control Design with MATLAB*; Springer London Limited: London, UK, 2005.
- Packard, A.; Doyle, J.; Balas, G. Linear Multivariable Robust Control with a μ Perspective. *J. Dyn. Syst. Meas. Control* **1993**, *115*, 426–438. [\[CrossRef\]](#)
- Apkarian, P.; Noll, D. Nonsmooth \mathcal{H}_∞ Synthesis. *IEEE Trans. Autom. Control* **2006**, *41*, 71–86. [\[CrossRef\]](#)
- Bompart, V.; Noll, D.; Apkarian, P. Second-order nonsmooth optimization for \mathcal{H}_∞ synthesis. *Numer. Math.* **2007**, *107*, 433–454. [\[CrossRef\]](#)
- Apkarian, P.; Noll, D. The \mathcal{H}_∞ Control Problem is Solved. *Aerosp. Lab* **2017**, 1–11. [\[CrossRef\]](#)
- Apkarian, P. Nonsmooth μ synthesis. *Int. J. Robust Nonlinear Control* **2011**, *21*, 1493–1508. [\[CrossRef\]](#)
- Farag, A.; Werner, H. A Riccati-Genetic Algorithms Approach To Fixed-Structure Controller Synthesis. In Proceedings of the 2004 American Control Conference, Boston, MA, USA, 30 June–2 July 2004; pp. 2799–2804.
- Lari, A.; Khosravi, A. An Evolutionary Approach to Design Practical μ Synthesis Controllers. *Int. J. Control Autom. Syst.* **2013**, *11*, 167–174. [\[CrossRef\]](#)
- Farag, A.; Werner, H. Fixed-Structure μ -Synthesis—An Evolutionary Approach. In Proceedings of the 2006 American Control Conference, Minneapolis, MN, USA, 14–16 June 2006; pp. 4332–4337.
- Dervis, K. *An Idea Based on Honey Bee Swarm For Numerical Optimization*; Technical Report—TR06; Erciyes University: Kayseri, Egypt, 2005.
- Mihaly, V.; Covaci, R.; Andrei, S. Artificial Bee Colony Optimization. In Proceedings of the 21th International Conference on Automation, Quality and Testing, Robotics (AQTR), THETA, Student Forum, Cluj-Napoca, Romania, 24–26 May 2018.
- Petráš, I. *Fractional Order Systems*; Printed Edition of the Special Issue Published in *Mathematics*; MDPI: Basel, Switzerland, 2019.
- Monje, C.A.; Chen, Y.; Vinagre, B.M.; Xue, D.; Feliu, V. *Fractional-Order Systems and Controls: Fundamentals and Applications*; Springer Science and Business Media: London, UK, 2010.
- Zhao, C.; Xue, D.; Chen, Y.Q. A fractional order PID tuning algorithm for a class of fractional order plants. In Proceedings of the IEEE International Conference Mechatronics and Automation, Niagara Falls, ON, Canada, 29 July–1 August 2005; pp. 216–221.
- Mihaly, V.; Dulf, E. Novel fractional order controller design for first order systems with time delay. In Proceedings of the 2020 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR), Cluj-Napoca, Romania, 21–23 May 2020; pp. 1–4.

23. Dulf, E.H.; Şuşcă, M.; Kovacs, L. Novel Optimum Magnitude Based Fractional Order Controller Design Method. *IFAC-PapersOnLine* **2018**, *51*, 912–917. [[CrossRef](#)]
24. Dulf, E.H. Simplified Fractional Order Controller Design Algorithm. *Mathematics* **2019**, *7*, 1166. [[CrossRef](#)]
25. Vinagre, B.M.; Monje, C.A.; Calderón, A.J.; Chen, Y.Q.; Feliu, V. The fractional integrator as a reference function. In Proceedings of the 1st IFAC Workshop on Fractional Differentiation and Its Application, Bordeaux, France, 19–21 July 2004.
26. Garrido, S.; Monje, C.A.; Martin, F.; Moreno, L. Design of Fractional Order Controllers Using the PM Diagram. *Mathematics* **2020**, *8*, 2022. [[CrossRef](#)]
27. Muresan, C.I.; Birs, I.R.; Dulf, E.H. Event-Based Implementation of Fractional Order IMC Controllers for Simple FOPDT Processes. *Mathematics* **2020**, *8*, 1378. [[CrossRef](#)]
28. Zheng, W.; Luo, Y.; Chen, Y.Q.; Wang, X. A Simplified Fractional Order PID Controller's Optimal Tuning: A Case Study on a PMSM Speed Servo. *Entropy* **2021**, *23*, 130. [[CrossRef](#)] [[PubMed](#)]
29. Iannino, V.; Colla, V.; Innocenti, M.; Signorini, A. Design of a \mathcal{H}_∞ Robust Controller with μ -Analysis for Steam Turbine Power Generation Applications. *Energies* **2017**, *10*, 1026. [[CrossRef](#)]
30. Sabău, D.; Dobra, P. State-feedback control algorithms for a CNC machine. In Proceedings of the 2018 22nd International Conference on System Theory, Control and Computing (ICSTCC), Sinaia, Romania, 10–12 October 2018; pp. 615–620.
31. Sabău, D.; Dobra, P. A PI controller based on state-feedback algorithm for an XY positioning system. In Proceedings of the 2019 22nd International Conference on Control Systems and Computer Science (CSCS), Bucharest, Romania, 28–30 May 2019; pp. 56–60.
32. Oldham, K.B.; Spanier, J. *The Fractional Calculus*; Academic Press: New York, NY, USA; London, UK, 1974.
33. Caputo, M. Linear model of dissipation whose Q is almost frequency independent-II. *Geophys. J. Int.* **1967**, *13*, 529–539. [[CrossRef](#)]
34. Podlubny, I. *Fractional Differential Equations*; Mathematics in Science and Engineering; Academic Press: San Diego, CA, USA, 1999; Volume 198.
35. Xue, D.; Chen, Y.Q.; Atherton, D.P. *Linear Feedback Control—Analysis and Design with MATLAB*; SIAM: Philadelphia, PA, USA, 2007.
36. Skogestad, S.; Postlethwaite, I. *Multivariable Feedback Control—Analysis and Design*, 2nd ed.; John Wiley & Sons: New York, NY, USA, 2005.
37. Balas, G.; Chiang, R.; Packard, A.; Safonov, M. *Robust Control Toolbox—User's Guide*; The MathWorks: Natick, MA, USA, 2020.
38. Şuşcă, M.; Mihaly, V.; Stănescu, M.; Morar, D.; Dobra, P. Unified CACSD Toolbox for Hybrid Simulation and Robust Controller Synthesis with Applications in DC-to-DC Power Converter Control. *Mathematics* **2021**, *9*, 731. [[CrossRef](#)]