Frequency-Domain Robust Control Toolbox

Alireza Karimi

***Abstract*— A new frequency-domain robust control toolbox is introduced and compared with some features of the robust control toolbox of Matlab. A summary of the theoretical background for H_∞ controller design using the spectral models is given. The main advantage of this toolbox is that almost all types of model uncertainties like unmodelled dynamics, multimodel uncertainty, spectral uncertainty and parametric uncertainty can be taken into account without conservatism. As a result, the uncertain parametric or frequency-domain identified models of the identification toolbox of Matlab can be used directly for computing robust controllers. The main commands of the new toolbox are briefly explained and the performance of the designed controllers are illustrated via some simulation and experimental results.**

I. Introduction

Robust control problem has been among interesting research problems in control system community for more than three decades. Robust control methods aim to design a controller for an uncertain model belonging to a bounded set. A large amount of research papers and books have been published in this subject and research is still in progress to develop new algorithms with less conservatism. The conservatism can be related to the uncertainty modeling, specially during conversion of one type of uncertainty to one that can be treated by the robust control design algorithm.

The H_∞ control problem was first formulated by Zames in [1]. Then, several elegant solutions to H_∞ loop shaping [2] and optimal H_∞ control using Riccati equations [3] and Linear Matrix Inequalities (LMIs) [4] in state space framework were proposed. In these approaches the model uncertainty is presented by some weighting filters and they usually lead to high order controllers and so a controller order reduction should follow the design.

The most efficient and established algorithms for robust control analysis and design are gathered in Robust Control Toolbox of Matlab together with additional commands for closed-loop and controller structure definition, weighting filter design for performance and uncertainty and controller reduction commands. The main robust control synthesis commands are used for H_∞ loop shaping, optimal H_∞ control, μ-synthesis [5] and H_∞ fixed-structure controller tuning [6]. Some simulation examples are also included that shows how different types of uncertainty can be converted to weighting filters and used to compute optimal robust controllers based on the provided commands. However, these methods are optimal with respect to their prescribed criterion and does not represent the best controller in terms of conventional performance measures and the real model uncertainty.

Recently, a new method for fixed-order robust controller design using the nonparametric frequency-domain models is proposed [7]. This method computes fixed-order linearly parameterized controllers with H_∞ performance by convex optimization. It has been shown that multimodel and frequency-domain uncertainty can be taken into account with no conservatism. Since only the frequency-domain data are used for controller design, input delay and nonparametric identified models can be used directly for controller design.

In this paper, we introduce a new toolbox based on the approach in [7]. Moreover, we show that how parametric uncertainty in a stochastic framework can be taken into account in the proposed approach with almost no conservatism. As a result, the identified models of Identification Toolbox of Matlab, spectral models or parametric models, together with their uncertainty covariance matrix can be used directly in the new toolbox for computing robust controllers. The performance of the proposed approach is compared with that of some examples from Robust Control Toolbox of Matlab. In addition, its full compatibility with the uncertain models of Identification Toolbox is illustrated via some experimental results on a laboratory setup.

It should be mentioned that the Frequency-Domain Robust Control (FDRC) toolbox is available for free in [8]. This toolbox can be used to compute H_∞ decoupling controllers for MIMO systems as well as gain scheduled controllers based on [9] and [10], respectively. It can also be used for PID controller design with constraints on the gain margin, phase margin and the crossover frequency [11]. However, in this paper we do not aim to presents all obtains and abilities of this toolbox and we limit ourself to robust controller design for SISO systems with H_∞ performance.

The paper is organized as follows. Next section recalls the basic framework of the proposed approach. Then, in Section III, we show how different types of model uncertainty can be taken to account in the proposed approach with almost no conservatism. The commands of FDRC toolbox are briefly explained and followed by some simulation examples and experimental results. The results are compared, when is possible, with those of Robust Control Toolbox of Matlab in Section IV. Finally, Section V gives some concluding remarks.

II. H_∞ Controller Design

In this section the class of linearly parameterized controllers are recalled and it is shown that H_∞ performance constraints can be linearized using this type of controllers.

The author is with the Laboratoire d'Automatique of Ecole Polytechnique Fédérale de Lausanne (EPFL), 1015 Lausanne, Switzerland. email: alireza.karimi@epfl.ch

A. *Class of controllers:*

Linearly parameterized controllers are given by :

$$K(\rho) = \phi^T \rho \tag{1}$$

where $\rho^T = [\rho_1, \rho_2, \ldots, \rho_n]$, is the vector of the controller parameters and ϕ is a vector of known orthogonal basis functions. There are some standard orthogonal basis functions that are used generally in the context of function approximation in system identification and controller design. The simplest one which has only one parameter is Laguerre basis function:

$$\phi_1(s) = 1, \quad \phi_i(s) = \frac{\sqrt{2\xi}(s-\xi)^{i-2}}{(s+\xi)^{i-1}}, \quad i = 2, \ldots, n$$

for continuous-systems and:

$$\phi_1(z) = 1, \quad \phi_i(z) = \frac{\sqrt{1-a^2}}{z-a}\left(\frac{1-az}{z-a}\right)^{i-2}, \; i = 2, \ldots, n$$

for discrete-time systems, where $\xi > 0$, and $-1 < a < 1$. The special case with $a = 0$ leads to the FIR (Finite Impulse Response) controller.

The main reason to chose linearly parameterized controllers (with fixed denominator) instead of classical rational transfer functions is that the optimization problem for controller design that will be defined later becomes convex in the controller parameters and can be solved efficiently. In fact, with this parametrization, every point on the Nyquist diagram of the open-loop transfer function $L(j\omega, \rho)$ can be written as a linear function of the controller parameters ρ :

$$L(j\omega, \rho) = G(j\omega)K(j\omega, \rho) = G(j\omega)\phi^T(j\omega)\rho \tag{2}$$

The other reason is that the most used controllers in industry, i.e. the proportional-integral-derivative (PID) controllers, are linearly parameterized. The last reason is that any controller represented by a rational transfer function can be approximated with good accuracy by a high-order linearly parameterized controller if the elements of ϕ are chosen from a set of orthogonal basis functions. In other words, it can be shown that for any stable rational finite order transfer function F and for arbitrary $\varepsilon > 0$ there exists a sufficiently large n such that

$$\|F - \phi^T \rho\|_p < \varepsilon \quad \text{for} \quad 0 < p < \infty$$

The quality of this approximation for a finite n, however, depends on the difference between the poles of F and the poles of ϕ.

B. *H_∞ constraints*

A very standard robust control problem is to design a controller that satisfies $\|W_1 \mathcal{S}\|_\infty < 1$ for a set of models, where $W_1(s)$ is the performance weighting filter and $\mathcal{S} = (1+L)^{-1}$ is the sensitivity function. For a nominal model, this can be written as a set of non-convex constraints in the frequency domain:

$$|W_1(j\omega)| < |1 + L(j\omega, \rho)| \quad \forall \omega \tag{3}$$

Let the left side be multiplied by $|1 + L_d(j\omega)|$, where $L_d(s)$ is the desired open-loop transfer function, and the right side by $|1 + L_d(-j\omega)|$. Then, one gets:

$$|W_1(j\omega)||1 + L_d(j\omega)| < |1 + L_d(-j\omega)||1 + L(j\omega, \rho)| \quad \forall \omega$$

Next, using the fact that

$$R_e\{[1 + L_d(-j\omega)][1 + L(j\omega, \rho)]\} \leq |[1 + L_d(-j\omega)][1 + L(j\omega, \rho)]| \tag{4}$$

a sufficient condition for $\|W_1 \mathcal{S}\|_\infty < 1$ can be derived:

$$|W_1(j\omega)||1 + L_d(j\omega)| - R_e\{[1 + L_d(-j\omega)][1 + L(j\omega, \rho)]\} < 0 \quad \forall \omega \tag{5}$$

However, these constraints do not guarantee the stability of the closed-loop system. A solution to this problem can be found by investigating (4). It is clear that if this constraint is satisfied the second term is strictly positive :

$$R_e\{[1 + L_d(-j\omega)][1 + L(j\omega, \rho)]\} > 0 \qquad \forall \omega \tag{6}$$

Therefore:

$$\text{wno}\{[1 + L_d(-j\omega)][1 + L(j\omega, \rho)]\} = 0 \tag{7}$$

where wno stands for winding number of the Nyquist plot around the origin. Thus we can conclude that:

$$\text{wno}[1 + L_d(j\omega)] = \text{wno}[1 + L(j\omega, \rho)] \tag{8}$$

Thus, if L_d meets the Nyquist stability criterion, and the constraints in (4) are satisfied, the closed-loop system will be stable. Moreover, the behavior of L_d should be the same as L at the vicinity of the poles of L on the imaginary axis such that the number of encirclement can be verified at all ω except those corresponding to the pole on the imaginary axis. For a formal stability proof see [7].

C. *Choice of L_d*

In fact the inequality in (4) and the role of L_d is very crucial in this approximation. It is clear that if $L_d(s) = L(s, \rho)$ there is no conservatism and (5) is equivalent to (3). Therefore, a good choice of $L_d(s)$ should have the following properties:

- $L_d(s)$ should be as close as possible to $L(s)$ to reduce the conservatism. For example it is better that it contains the unstable poles of the plant model as well as the pure time delay, which cannot be canceled by the controller.
- A very good choice for stable systems, in continuous-time as well as in discrete-time, and the controller structures with integral action is $L_d(s) = \omega_c/s$. This choice is coherent with the choice of desired open-loop transfer function in the classical open-loop shaping methods that suggest the magnitude of the open-loop transfer function should be large at low frequencies and small at high frequencies.
- The number of counterclockwise encirclements of $L_d(s)$ around the critical point should be equal to the

number of unstable poles of the plant model to ensure the closed-loop stability.

- $L_d(s)$ should contain all poles of $L(s)$ on the imaginary axis.
- For systems with known unstable poles, a good choice for $L_d(s)$ is:

$$L_d(s) = \frac{\omega_c}{s} \prod_i \frac{s - \alpha_i}{s + \alpha_i} \tag{9}$$

  where α_i are unstable poles of the plant model. However, it should be verified that L_d satisfies the Nyquist criterion.
- If the first choice of $L_d(j\omega)$ leads to a non feasible set, the iterative windsurfing approach can be used to compute an appropriate $L_d(s)$. In this approach we start with modest specifications by reducing the gain of W_1 so that a feasible solution ρ_1 is obtained. Then $L_d(j\omega) = L(j\omega, \rho_1)$ is chosen and the specifications will be tightened by increasing the gain of W_1. A feasible solution ρ_2 for the second feasibility problem will be used to compute a new $L_d(j\omega) = L(j\omega, \rho_2)$. Although the convergence of this iterative approach to the optimal solution cannot be proved, good results in practice can be obtained. This approach is used in FDRC Toolbox when a bisection algorithm is used to minimize $\|W_1\mathcal{S}\|_\infty$.
- If a stabilizing controller $K_0(j\omega)$ is available, $L_d(j\omega) = G(j\omega)K_0(j\omega)$ is a good choice to compute a robust low-order linearly parameterized controller.

III. Model Uncertainty

A. Unmodeled Dynamics

If the set of models is represented by multiplicative uncertainty, i.e.

$$\tilde{G}(s) = G(s)[1 + W_2(s)\Delta(s)] \text{ with } \|\Delta\|_\infty < 1,$$

the necessary and sufficient condition for robust performance is given by [12]:

$$\| |W_1\mathcal{S}| + |W_2\mathcal{T}| \|_\infty < 1 \tag{10}$$

where $\mathcal{T} = L(1 + L)^{-1}$ is the complementary sensitivity function.

The basic idea is to represent the robust performance constraints in (10) as a set of linear or convex constraints. Let us write (10) in the frequency domain :

$$|W_1(j\omega)\mathcal{S}(j\omega)| + |W_2(j\omega)\mathcal{T}(j\omega)| < 1 \quad \forall\omega \tag{11}$$

Multiplying the robust performance condition in (11) by $|1+L(j\omega, \rho)|$ gives:

$$|W_1(j\omega)| + |W_2(j\omega)L(j\omega, \rho)| < |1 + L(j\omega, \rho)| \quad \forall\omega \tag{12}$$

Note that $|1 + L(j\omega, \rho)|$ is the distance between the critical point and $L(j\omega, \rho)$. Hence, this constraint is satisfied if and only if there is no intersection in the Nyquist diagram between a circle centered at the critical point with a radius of $|W_1(j\omega)|$ and a circle centered at $L(j\omega, \rho)$ with a radius of $|W_2(j\omega)L(j\omega, \rho)|$ for all ω.

There exist two alternatives in order that this condition to be satisfied for all models in the uncertainty set represented by a circle centered at $L(j\omega, \rho)$. The first alternative is to approximate the uncertainty circle by a polygon of $q > 2$ vertices. Then, the robust performance condition in (11) is satisfied if all vertices are located in the right side of $d(\omega)$. This can be represented by the following linear constraints :

$$\begin{aligned} &|W_1(j\omega)[1 + L_d(j\omega)]| \\ &\quad - R_e\{[1 + L_d(-j\omega)][1 + L_i(j\omega, \rho)]\} < 0 \quad \forall\omega \end{aligned} \tag{13}$$

and $i = 1, \dots, q$ where $L_i(j\omega, \rho) = G_i(j\omega)K(j\omega, \rho)$ and

$$G_i(j\omega) = G(j\omega)\left[1 + \frac{|W_2(j\omega)|}{\cos(\pi/q)} e^{j2\pi i/q}\right] \tag{14}$$

It can be observed that the number of linear constraints are multiplied by q when the uncertainty circle is approximated by a polygon of q vertices.

The second alternative is to increase the radius of the performance circle by $|W_2(j\omega)L(j\omega, \rho)|$ which leads to the following convex constraints:

$$\begin{aligned} &|W_1(j\omega)[1 + L_d(j\omega)]| + |W_2(j\omega)L(j\omega, \rho)[1 + L_d(j\omega)]| - \\ &\quad R_e\{[1 + L_d(-j\omega)][1 + L(j\omega, \rho)]\} < 0 \quad \forall\omega \end{aligned} \tag{15}$$

This alternative has less constraints and no conservatism but leads to a slightly more complex convex optimization problem (convex constraints instead of linear constraints).

B. Multimodel Uncertainty

The results can be extended to the case of multimodel uncertainty as well. In this case the constraints in (15) should be repeated for all models $G_i(j\omega)$ from $i = 1, \dots, m$. If the desired open-loop transfer function and the performance and uncertainty filters are different for each model, i.e. we have L_{d_i}, W_{1i} and W_{2i}, that can be used in (15).

Since the constraints in (15) are convex with respect to $G_i(j\omega)$ for the case of multimodel uncertainties, they will also guarantee the robust performance for the convex combinations of the models defined by:

$$G(\lambda, j\omega) = \sum_{i=1}^{m} \lambda_i G_i(j\omega)$$

C. Spectral Uncertainty

Consider the input signal $u(t)$ and the output signal $y(t)$ of a discrete-time system $G(q^{-1})$ are available for a finite number of $t = 1, \dots, N$, where q^{-1} is the backward shift operator. Assume that the data are noise-free and the initial and final conditions for u and y are zero, i.e $u(t) = y(t) = 0$ for $t \leq 0$ and $t > N$. Then

$$\hat{G}(e^{-j\omega}) = \frac{Y(\omega)}{U(\omega)} \tag{16}$$

where $U(\omega)$ and $Y(\omega)$ are the periodograms of $u(t)$ and $y(t)$ defined by :

$$U(\omega) = \frac{1}{\sqrt{N}} \sum_{t=1}^{N} u(t)e^{-j\omega t}$$

$$Y(\omega)=\frac{1}{\sqrt{N}}\sum_{t=1}^{N}y(t)e^{-j\omega t}$$

For noisy data (16) gives a spectral model which is asymptotically unbiased. The estimates $R_e\{\hat{G}(e^{j\omega})\}$ and $I_m\{\hat{G}(e^{j\omega})\}$ are asymptotically uncorrelated and normally distributed with a variance of $\Phi_v(\omega)/2|U(\omega)|^2$, where $\Phi_v(\omega)$ is the spectrum of the disturbance $v(t)$ at the output of the plant [13]. Since $v(t)$ is not measurable, it can be estimated using the unbiased estimate of the plant model, i.e., $\hat{v}(t)=y(t)-\hat{G}(q^{-1})u(t)$. So its spectrum is given by:

$$\Phi_{\hat{v}}(\omega)=\Phi_y(\omega)-\frac{|\Phi_{uy}(\omega)|^2}{\Phi_u(\omega)} \qquad (17)$$

In order to see the shape of this uncertainty in the Nyquist diagram we can define a random variable vector $\hat{G}_v(\omega)$ as

$$\hat{G}_v(\omega)=[R_e\{\hat{G}(e^{-j\omega})\} \quad I_m\{\hat{G}(e^{-j\omega})\}] \qquad (18)$$

This vector has a joint normal distribution with a diagonal covariance matrix (because of the uncorrelation of real and imaginary parts of the estimates):

$$C_G(\omega)=\operatorname{cov}\left(\hat{G}_v(\omega)\right)=\frac{\Phi_{\hat{v}}(\omega)}{2|U(\omega)|^2}\left[\begin{array}{cc}1 & 0\\ 0 & 1\end{array}\right] \qquad (19)$$

Therefore, the true plant model $G(e^{-j\omega})$ belongs to the following disk in the Nyquist diagram with a probability of $1-\alpha$:

$$\left[\begin{array}{c}x-R_e\{\hat{G}\}\\ y-I_m\{\hat{G}\}\end{array}\right]^T C_G^{-1}(\omega)\left[\begin{array}{c}x-R_e\{\hat{G}\}\\ y-I_m\{\hat{G}\}\end{array}\right]\leq\mathcal{X}_2^2(\alpha) \qquad (20)$$

where $\mathcal{X}_2^2$ is the chi-square distribution with two degrees of freedom. For a confidence interval of 0.95 ($\alpha=0.05$), we have $\mathcal{X}_2^2(0.05)=5.99$. This uncertainty set can be represented by multiplicative model uncertainty given by:

$$\tilde{G}(e^{-j\omega})=\hat{G}(e^{-j\omega})[1+W_2(j\omega)\Delta]\,;\quad \omega\in[0\quad\frac{\pi}{T_s}]$$

where T_s is the sampling period and

$$W_2(j\omega)=\frac{1}{\hat{G}(e^{-j\omega})}\sqrt{\frac{5.99\Phi_{\hat{v}}(\omega)}{2|U(\omega)|^2}}$$

is the uncertainty frequency function for 0.95 probability.

D. Parametric Uncertainty

In many practical applications a model of the system is available, but its parameters are not exactly known. In most cases the parameters can be considered as random variables with known mean, variance and distribution. This is the case for parametric models obtained by system identification from a set of noisy data. In physical modeling of the systems, the parameters are measured by an instrument that has some accuracy. Therefore, the measured value can be considered as a random variable whose mean and variance can be estimated by repeating measurements. Even in the deterministic case when each parameter belongs to an interval, a stochastic approach can be used to represent the uncertainty.

In a stochastic framework the covariance of $\hat{G}_v(\omega)$ in (18) can be estimated from the covariance of its parameters using a linear approximation. It can be shown that if the parametric uncertainty comes from the noise effect in system identification, this approximation is accurate for large data length [14]. In this case we have:

$$C_G(\omega)=\left(\frac{\partial\hat{G}_v(\omega)}{\partial\theta}\right)\operatorname{cov}(\theta)\left(\frac{\partial\hat{G}_v(\omega)}{\partial\theta}\right)^T \qquad (21)$$

This covariance is computed in Identification Toolbox of Matlab and can be used to compute an uncertainty model set for robust controller design.

Note that the covariance matrix in this case is not diagonal, so in the Nyquist diagram, this uncertainty is represented by an ellipse. In this case, the spectral model cannot be represented directly by a multiplicative uncertainty model. However, an n_q-side polygon of minimum area can be computed that circumscribes each ellipse. Therefore, the uncertainty set, e.g. for a probability of 0.95, can be approximated by the convex combination of the vertices of the polygon.

$$G(\lambda)=\sum_{k=1}^{n_q}\lambda_i\tilde{G}_k(e^{-j\omega})\,: \qquad (22)$$

where

$$\tilde{G}_k(e^{-j\omega})=\hat{G}(e^{-j\omega})+[1\;\;j]\sqrt{5.99C_G(\omega)}\left[\begin{array}{c}\frac{\cos(2\pi k/n_q)}{\cos(\pi/n_q)}\\ \frac{\sin(2\pi k/n_q)}{\cos(\pi/n_q)}\end{array}\right]$$

The last vector in the above equation gives the coordinates of a vertex of a polygon circumscribing the unit circle and $\sqrt{5.99C_G}$ is a 2×2 matrix that defines the size and direction of the uncertainty (for 0.95 probability). In fact $\sqrt{5.99C_G}$ projects the unit circle to an ellipse with the size of uncertainty and consequently it projects the polygon circumscribed the unit circle to a polygon circumscribed about the ellipse.

Therefore, the uncertainty ellipse in the frequency domain related to stochastic parametric uncertainty can be completely covered with multimodel uncertainty including n_q models.

IV. Simulation Examples and Comparison

A. FDRC Toolbox

The Frequency-Domain Robust Controller Design Toolbox is a tool for designing robust linearly parameterized controllers in the Nyquist diagram. It can be used to design linearly parameterized controllers of any order for parametric models or nonparametric models obtained for example by the identification toolbox of MATLAB. The robust controllers are designed in terms of H_∞ performance or classical robustness margins such as the gain and phase margin, for single/multi-model, SISO/MIMO systems. The toolbox also supports designing gain-scheduled controllers.

In all of design cases, linear or convex optimization problems are solved. For linear and quadratic optimization the well-known `linprog` or `quadprog` (depending on the problem) commands of the Optimization toolbox of MATLAB are used. While convex optimization problems are formulated with YALMIP [15] and can be solved with all available solvers. Many commands of the Control toolbox of MATLAB are used as well.

The procedure of design comprises three steps. First the type (or structure) of the controller is determined. Then the desired performance characteristics are specified, and finally a controller with the desired type and required performance is designed. In the following comes a brief description of these three steps with corresponding commands.

1) Determining controller structure: The first step of design is determining the desired controller type. By defining the controller type in fact the vector of basis transfer functions ϕ is specified. In the following command the controller type and subsequently the vector ϕ are specified by the user.

```
phi = conphi (ConType , ConPar , CorD , F)
```

where `ConType` can be `'Laguerre'` or other basis functions which are explained in the user's manual of the toolbox [8]. The parameters of the basis functions are defined in `ConPar`, e.g. for Laguerre basis function, $[100 \quad 4]$ defines a fourth order controller with $\zeta = 100$. The choice of continuous-time or discrete time basis functions is defined by `CorD` that can be `'s'` or `'z'`, respectively. Finally, `F` is a fixed term in the controller, e.g. $1/s$ to have an integrator.

2) Determining control performance: The desired control performance are determined by the following command:

```
per = conper (PerType , par , Ld)
```

where `PerType` can be `Hinf` or 'LS' for H_∞ or loop shaping performance. For H_∞ performance `par` defines the weighting filters and for loop shaping the modulus margin. The desired open-loop transfer function is given by `Ld`.

3) Controller Design: The controller is designed by the following command:

```
K = condes (G , phi , per , options)
```

where `G` can be a cell containing multiple models with any acceptable format of LTI models in Matlab. The options and their default values are given in the user's manual of the toolbox [8]. The specific options used in this paper will be explained in the following examples.

B. Example 1: Unmodeled Dynamics

Consider the family of plants described by the following multiplicative uncertainty model [16]:

$$\tilde{G}(s) = \frac{(s+1)(s+10)}{(s+2)(s+4)(s-1)}[1 + W_2(s)\Delta(s)]$$

where

$$W_2(s) = 0.8\frac{1.1337s^2 + 6.8857s + 9}{(s+1)(s+10)}$$

The nominal performance is defined by $\|W_1\mathcal{S}\|_\infty < 1$ with :

$$W_1(s) = \frac{2}{(20s+1)^2}$$

The Robust Control Toolbox of Matlab leads to an unstable 7-th order controller for this problem using the following codes:

```
s=tf('s');
G=(s+1)*(s+10)/((s+2)*(s+4)*(s-1));
W{1}=2/(20*s+1)^2;
W{2}=0.8*(1.1337*s^2+6.8857*s+9)/((s+1)*(s+10));

P=augw(G,W{1},[],W{2});
C=hinfsyn(P);
```

This optimal controller gives:

$$\|[W_1\mathcal{S} \quad W_2\mathcal{T}]\|_\infty = 0.6793 \tag{23}$$

$$\||W_1\mathcal{S}| + |W_2\mathcal{T}|\|_\infty = 0.8363 \tag{24}$$

The same problem is solved to design a third-order controller using the FDRC toolbox with the following codes:

```
phi=conphi('Laguerre',[100 2],'s',1/s);
Ld=10*(s+1)/s/(s-1);
hinfper=conper('Hinf',W,Ld);
opt=condesopt('nq',[],'gamma',[0.01 2 0.001],...
'lambda',[1 1 0 0]);
K=condes(G,phi,hinfper,opt)
```

The resulting stable controller

$$K(s) = \frac{2430.3(s+25.19)(s+1.08)}{s(s+100)^2} \tag{25}$$

gives:

$$\|[W_1\mathcal{S} \quad W_2\mathcal{T}]\|_\infty = 0.7201 \tag{26}$$

$$\||W_1\mathcal{S}| + |W_2\mathcal{T}|\|_\infty = 0.7203 \tag{27}$$

Remarks:

- Since `hinfsyn` minimizes $\|[W_1\mathcal{S} \quad W_2\mathcal{T}]\|_\infty$, a smaller value is obtained with respect to the result of FDRC toolbox. However, better results in terms of the *true* robust performance criterion in (10) is obtained by the new toolbox.
- The variable `opt` shaws the options taken for the controller design. `nq=[]` indicates that no approximation of the uncertainty circle by a polygon is considered. `gamma=[0.01 2 0.001]` initializes respectively the minimum, maximum and tolerance of γ in a bisection algorithm. Finally, `lambda=[1 1 0 0]` indicates that $\||W_1\mathcal{S}| + |W_2\mathcal{T}|\|_\infty$ should be minimized.
- A Laguerre orthogonal basis function is used for the linearly parameterized controller. The pole of this basis function is chosen as $\zeta = 100$ and one integrator is fixed in the controller. We will use the same choice in the next example.
- The choice of L_d is based on the remarks of Section II-C. A desired crossover frequency of $\omega_c = 10$ rad/s and the existence of an unstable pole in the plant model leads to

$$L_d(s) = \frac{10}{s}\frac{s+1}{s-1}$$

Since, in this example an iterative method is used to improve the performance, the final results is not sensitive to the choice of $L_d(s)$. By changing ω_c from 2 to 100, very similar results are obtained.

C. Example 2: Multimodel Uncertainty

This example is taken from Robust Control Toolbox of Matlab. A first order unstable model is considered :

$$P_0(s) = \frac{2}{s-2} \tag{28}$$

This model is perturbed by different type of uncertainty like extra lag, high frequency resonance, time delay and pole/gain migration to obtain the following models:

$$\begin{aligned} P_1(s) &= P_0(s)\frac{1}{0.06s+1}, & P_4(s) &= P_0(s)\frac{-0.02s+1}{0.02s+1} \\ P_2(s) &= P_0(s)\frac{50^2}{s^2+10s+50^2}, & P_5(s) &= \frac{2.4}{s-2.2} \\ P_3(s) &= P_0(s)\frac{70^2}{s^2+28s+70^2}, & P_6(s) &= \frac{1.6}{s-1.8} \end{aligned}$$

The performance weighting filter is chosen as a first order low-pass filter with a static gain of 500 and a bandwidth of $\omega_c = 4.5$ rad/s :

$$W_{\text{perf}}(s) = \frac{0.33s + 4.248}{s + 0.008496} \tag{29}$$

A noise filter is also defined as:

$$W_{\text{noise}}(s) = \frac{0.1975s^2 + 0.6284s + 1}{7.901e^{-5}s^2 + 0.2514s + 400} \tag{30}$$

The control objective is to obtain

$$\|W_{\text{perf}}\mathcal{S}\|_\infty < \gamma \quad \text{and} \quad \|W_{\text{noise}}\mathcal{T}\|_\infty < \gamma$$

with $\gamma < 1$ for all 7 models.

In Robust Control Toolbox of Matlab, the multimodel uncertainty is converted to multiplicative uncertainty and a shaping filter that covers the relative behavior of the plant is designed. Then the μ-synthesis approach is used to compute an 18-th order controller that meets the performance for all models with $\gamma = 1.024$. The order of this model is then reduced to 6 by an order reduction algorithm that ensures almost the same performance. Figure 1a shows the step disturbance response of this controller.

The same problem is solved with the new toolbox. An $(n+1)$-th order controller with integral action is designed using a Laguerre basis function of order n with a high frequency pole at $\zeta = 100$. The desired open-loop transfer function is chosen according to the rules in Section II-C. Then the following code is used to compute the controller:

```
w=logspace(-2,3,1000);
opt=condesopt('w',w,'gamma',[0.1,4,0.001]);
Ld=4.5*(s+2)/s/(s-2);
W{1}=Wperf;W{2}=Wnoise;
per=conper('Hinf',W,Ld);
phi=conphi('Laguerre',[100 n],'s',1/s);
K=condes(p,phi,per,opt);
```

For $n = 8$ (9-th order controller), the resulting controller achieves better performance than 18-th order optimal controller with $\gamma = 0.8341$. Figure 1b shows the step disturbance response of the 7 models. Comparing with the optimal solution of the μ-synthesis approach in Fig. 1a, it has less undershoot and shorter settling time. For $n = 4$, a 5th-order controller gives $\gamma = 1.076$ and achieves similar results as the μ-synthesis controller.

Remarks:

- The main reason to obtain better performance than the μ-synthesis method is the conservatism in the multimodel uncertainty approximation with a weighting filter in the the μ-synthesis approach. The multimodel uncertainty is taken into account without any conservatism in FDRC toolbox.
- In this example pure time delay in the model $P_4(s)$ is introduced by a first-order Pade approximation. It should be mentioned that time delay can be considered in FDRC toolbox without any approximation.

D. Example 3: Spectral Uncertainty

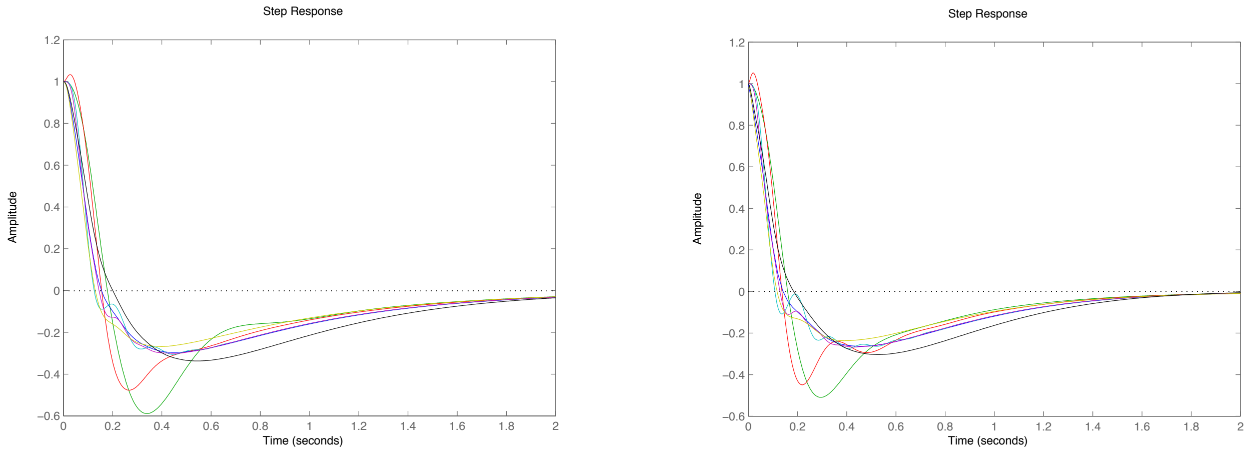
In this example a laboratory setup is used to show how the spectral uncertainty can be taken into account in robust controller design by the proposed toolbox. The laboratory setup is an electro-mechanical flexible transmission system. The plant consists of a drive motor (servo actuator) which is coupled via a timing belt to a drive disk with variable inertia. Another timing belt connects the drive disk to the speed reduction assembly while a third elastic belt completes the drive train to the load disk. A proportional feedback controller is used to control the position of the load disk. The input of the system is the position reference of the load disk which is measured by a high resolution incremental encoder. The input of the system is excited with a PRBS signal with a sampling period of $T_s = 40$ms and a data length of 765. A picture of the plant and the input/output data are given in Fig. 2.

A spectral model is identified using `spafdr` command of Identification toolbox of Matlab. The Nyquist diagram of this spectral model together with the uncertainty circles of 0.95 probability are given in Fig. 3a. This spectral model can be used directly in the FDRC toolbox to compute a digital robust controller by using the following codes:

```
data=iddata(y,u,Ts);
G=spafdr(data);

phi=conphi('Laguerre',[Ts , 0 , 4],'z',z/(z-1));
Ld=15/s;
per=conper('LS',0.5,Ld);
K=condes(G,phi,per);
```

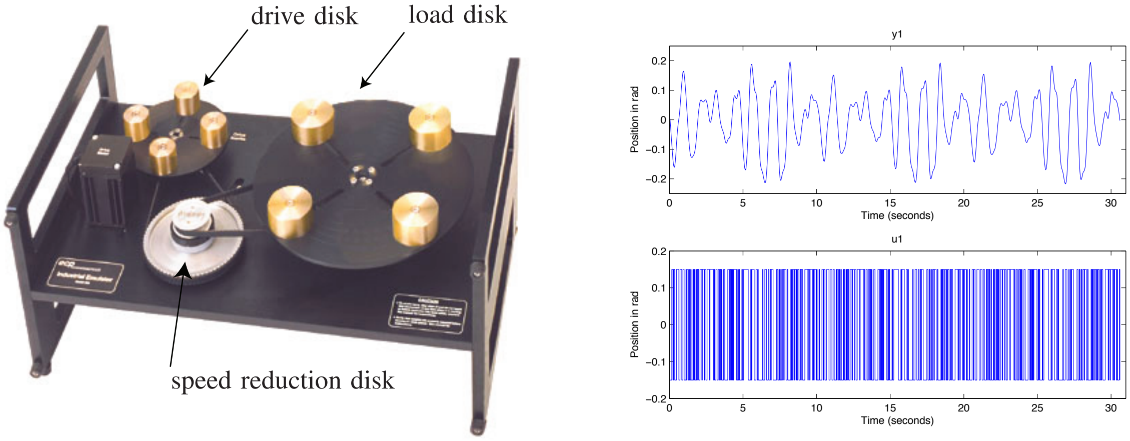
The discrete-time Laguerre basis functions of order 4 with $a = 0$ (FIR controller) and an integrator are used for the controller structure. The controller is designed by loopshaping method with guaranteeing a modulus margin of 0.5. In this method the weighted two-norm of the difference between L and L_d is minimized under the constraint on the modulus



(a)

(b)

Fig. 1. Step disturbance response of Example 2 with the controller of (a) Robust Control toolbox of Matlab, (b) FDRC toolbox.



(a)

(b)

Fig. 2. Flexible transmission system (a) Picture of the plant (b) Identification data.

margin (the open-loop Nyquist plot will not enter a circle of radius 0.5 centered at the critical point). Figure 3b shows the open-loop Nyquist diagram of the designed controller. It can be observed that the Nyquist plot and its uncertainty disks meet the performance specifications.

E. *Example 4: Parametric Uncertainty*

Based on the acquired data in Example 3, a fourth order parametric model can be identified using the identification toolbox of Matlab. The prediction error method with output error structure is used to identify the model parameters and their covariance matrix. Figure 3c shows the Nyquist diagram of the identified parametric model together with the uncertainty ellipses. The identified model is directly used by the FDRC toolbox to compute a robust controller with the same structure and performance specification as in Example3. The following codes are used:

```
G=oe(data,[4 4 1])
phi=conphi('Laguerre',[Ts , 0 , 4],'z',z/(z-1));
Ld=15/s;
per=conper('LS',0.5,Ld);
K=condes(G,phi,per);
```

The Nyquist plot of the open-loop transfer function is given in Fig. 3d. It can be seen that the specifications are satisfied in all frequencies. The results are a bit better than those of Example 3, because the uncertainties are smaller.

V. Conclusions

All robust controller design methods suffer from conservatism. The conservatism can be in different parts of the design procedure from the optimization criterion, modeling uncertainty and convexification of the optimization constraints. In this paper a new toolbox for robust controller design based on the frequency domain models is introduced. The main advantage of this toolbox is that almost all types of model uncertainty can be taken into account with almost no conservatism. In fact, the conservatism is moved to the convexification of the H_∞ constraints via the choice of the

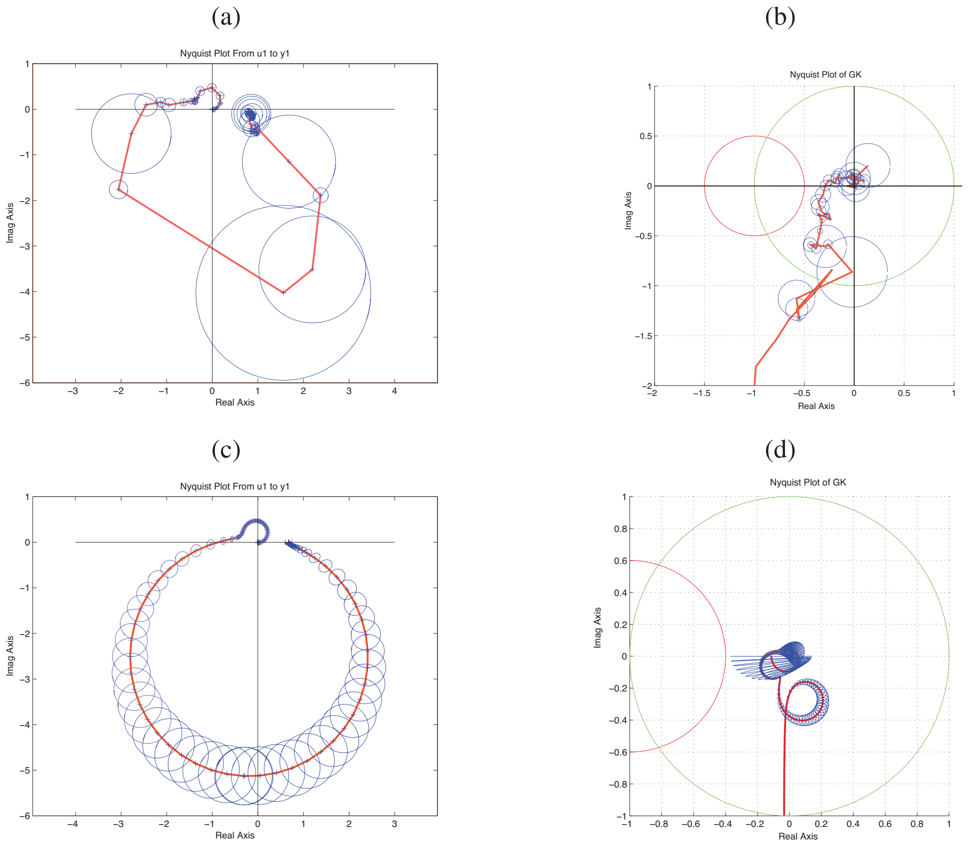


Fig. 3. (a) Nyquist diagram of the spectral model together with the uncertainty circles. (b) Nyquist plot of the designed open-loop system for the spectral model. (c) Nyquist diagram of the the identified parametric model together with the uncertainty ellipses. (d) Nyquist plot of the designed open-loop system for the parametric model.

basis functions and the desired open-loop transfer function. It has been shown, with some examples, that the new toolbox may provide better performance than the established robust control algorithms. Moreover, full compatibility of the toolbox with the identified uncertain models makes it appealing for data-driven approaches.

References

[1] G. Zames, "Feedback and optimal sensitivity: Model reference transformations, multiplicative seminorms, and approximate inverses," *IEEE Transactions on Automatic Control*, vol. 26, no. 2, pp. 301–320, 1981.

[2] D. McFarlane and K. Glover, "A loop shaping design procedure using H_∞ synthesis," *IEEE Transactions on Automatic Control*, vol. 31, no. 12, pp. 1799–1819, 1992.

[3] J. C. Doyle, K. Glover, , P. P. Khargonekar, and B. A. Francis, "State space solutions to standard H_2 and H_∞ control problem," *IEEE Transactions on Automatic Control*, vol. 34, no. 8, pp. 1228–1240, November 1989.

[4] P. Gahinet and P. Apkarian, "A linear matrix inequality approach to H_∞ control," *International Journal of Robust and Nonlinear Control*, vol. 4, no. 4, pp. 421–448, 1994.

[5] A. Packard and J. C. Doyle, "The complex structured singular value," *Automatica*, vol. 29, no. 1, pp. 71–109, 1993.

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

[7] 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.

[8] A. Karimi, "Frequency-domain robust controller design: A toolbox for MATLAB," Automatic Control Laboratory, EPFL, Switzerland, 2012. [Online]. Available: http://la.epfl.ch/FDRC_Toolbox

[9] 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.

[10] V. de Oliveira and A. Karimi, "Robust and gain-scheduled PID controller design for condensing boilers by linear programming," in *IFAC Conference in Advances in PID Control*, Brescia, Italy, 2012.

[11] M. Sadeghpour, V. de Oliveira, and A. Karimi, "A toolbox for robust PID controller tuning using convex optimization," in *IFAC Conference in Advances in PID Control*, Brescia, Italy, 2012.

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

[13] L. Ljung, *System Identification - Theory for the User*, 2nd ed. NJ, USA: Prentice Hall, 1999.

[14] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*. New Jersey: Prentice-Hall, 1993.

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

[16] T. E. Djaferis, *Robust Control Design: A Polynomial Approach*. Massachusetts, USA: Kluwer Academic Publishers, 1995.