Available online at www.sciencedirect.com

ScienceDirect

IFAC-PapersOnLine 49-10 (2016) 182–187

Numerical Computation of $\mathcal{H}_\infty$ Optimal Controllers for Time Delay Systems Using YALTA ⋆

Mustafa Oğuz Yeğin and Hitay Özbay

Department of Electrical and Electronics Engineering, Bilkent University, 06800 Ankara, Turkey
(e-mail: yegin@ee.bilkent.edu.tr,hitay@bilkent.edu.tr).

Abstract: Numerical computation of $\mathcal{H}_\infty$ controllers for time delay systems has been a challenge since 1980s. Even though significant techniques are developed to obtain direct optimal controllers, application of these methods may require manual computation depending on the plant. In this paper, an alternative computational technique is proposed for direct optimal controllers originally obtained by Toker and Özbay (1995). The new controller expression contains finite dimensional transfer functions and an infinite dimensional term, which is stable. Thus it is suitable for finite dimensional approximations and practical non-fragile implementations. In this method, in order to eliminate manual computation of the plant factorization for neutral and retarded delay systems YALTA (a tool developed at INRIA) is used. The new controller computation is implemented in Matlab, and it is illustrated on an example.

Keywords: Robust control, $\mathcal{H}_\infty$ controller, infinite dimensional systems, time-delay systems, neutral and retarded systems

1. INTRODUCTION

Since 1980s, various methods have been developed for numerical computation optimal $\mathcal{H}_\infty$ controllers for time delay systems. First results on direct optimal controllers were given in Foias et al. (1986); Lypchuk et al. (1988); Zhou and Khargonekar (1987). See also Mirkin and Tadmor (2002) and its references. One of the most practically useful formulas for direct optimal controller was given in Toker and Özbay (1995), see also (Foias et al., 1996). The software implementing Toker-Özbay formula gives the optimal controller Bode plots and it shows an implementation where internal unstable pole-zero cancellations occur. Later Gümüşsoy (2012) has shown a reliable implementation of the Toker-Özbay controller using FIR blocks. In this paper, we give an alternative way to obtain a reliable implementation of using stable and finite dimensional terms in the controller. An alternative approach to design $\mathcal{H}_2$ and $\mathcal{H}_\infty$ controllers for time delay systems has been proposed in Oliveira and Geromel (2004). Moreover, fixed order $\mathcal{H}_\infty$ controller design for time delay systems has been considered in Gümüşsoy and Michiels (2011).

The computation of the optimal controller depends on inner-outer factorization of the plant. We provide here a new Matlab based tool to perform such factorization using YALTA, and then illustrating the controller in a new format with an example that shows how an approximate finite dimensional suboptimal controller may be obtained by using the results of Toker and Özbay (1995).

The rest of the paper is organized as follows. In the next section, we review plant factorization and a procedure for computing the optimal controller. We also give a new formula for implementing the controller in an internally stable manner. Then in Section 3, we describe the new Matlab program, which incorporates YALTA and implements the controller and its approximations given in Section 2. In Section 4, we provide a numerical example. Concluding remarks are made in the last section

2. AN ALTERNATIVE FORMULA FOR $\mathcal{H}_\infty$ OPTIMAL CONTROLLER

For general infinite dimensional systems, structure of $\mathcal{H}_\infty$ controllers have been investigated and numerical computational methods have been proposed based on state space and finite dimensional techniques. Most of the state space methods require solving of operator valued Riccati equations. One of the most widely used technique is based on inner-outer factorizations of the plant, (Toker and Özbay, 1995). For time delay systems such a factorization can be done by using DDE-BIFTOOL, QPMR, YALTA or similar tools, Avanessoff et al. (2008); Engelborghs et al. (2001); Vyhlìdal and Zítek (2009). Once the factorization is done, the method originally proposed by Toker and Özbay can be applied. In this section we provide the details of this algorithm which is separated into three steps.

⋆ This work is supported by the Scientific and Technological Research Council of Turkey (TÜBİTAK) under project EEEAG-115E820.

10.1016/j.ifacol.2016.07.523

2.1 Plant Factorization for Time Delay Systems

We consider the following inner-outer factorization for the plants for which $\mathcal{H}_\infty$ controllers are to be computed:

$$P(s) = \frac{M_n N_o}{M_d} \tag{1}$$

where M_n and M_d are inner functions and N_o is an outer function. The right half plane zeros of the plant appear in M_n and the right half plane poles of the plant appear in M_d. In this paper we will assume that M_d is finite dimensional. Hence, the plant has finitely many unstable poles. We will use YALTA to compute the poles and zeros of the plant in the right half plane. We consider plants in the form

$$P(s) = \frac{t(s) + \sum_{\kappa=1}^{N'} t_\kappa(s) e^{-\kappa s h}}{p(s) + \sum_{k=1}^{N} q_k(s) e^{-k s h}} = \frac{n(s)}{d(s)} \tag{2}$$

where t, p, t_κ, q_k are polynomials with

$$deg(p(s)) \geq deg(q_k(s)), \qquad deg(p(s)) \geq deg(t(s))$$

and $deg(p(s)) \geq deg(t_\kappa(s))$. Note that additionally, YALTA allows numerator and denominator of the plant to have fractional powers, hence such plants can also be incorporated into the $\mathcal{H}_\infty$ optimal controller computation that is discussed in the paper. However, we will restrict our attention to non-fractional usual time delay systems.

2.2 Computation of the $\mathcal{H}_\infty$ Optimal Performance Level

In this section we summarize computation steps for the $\mathcal{H}_\infty$ optimal controller and the optimal performance level for the mixed-sensitivity minimization problem:

$$\gamma_{opt} = \inf_{(C,P)\text{ stable}} \left\| \begin{bmatrix} W_1(1+PC)^{-1} \\ W_2 PC(1+PC)^{-1} \end{bmatrix} \right\|_\infty$$

where W_1 and W_2 are sensitivity and multiplicative uncertainty weights, respectively. We assume $W_1(s) = nW_1(s)/dW_1(s)$ for polynomials nW_1 and dW_1 with $deg(dW_1) = n_1 \geq deg(nW_1)$.

The following is a summary of computation of γ_{opt} and the optimal controller C_{opt} in Toker-Özbay formula. We start with the following notation

$$E_\gamma(s) = \frac{W_1(s)W_1(-s)}{\gamma^2} - 1 = \frac{nE_\gamma(s)}{\gamma^2 dW_1(s) dW_1(-s)},$$

$\alpha_1, \ldots, \alpha_l$: unstable poles of the plant $P(s)$

$\beta_1, \ldots, \beta_{n_1}$: zeros of $E_\gamma(s)$ in $\mathbb{C}_+$ and on the positive Im-axis.

n : $l + n_1$

$$\mathcal{R}(\gamma) = \begin{bmatrix} V_n & D_n V_n \\ D_n V_n J_n & V_n J_n \end{bmatrix} \tag{3}$$

$J_n = diag\{(-1)^i\}$ is an $n \times n$ diagonal matrix

$V_n = \begin{bmatrix} V_\alpha^n \\ V_\beta^n \end{bmatrix}$, combination of Vandermonde matrices

$$V_\alpha^n = \begin{bmatrix} 1 & \alpha_1 & \cdots & \alpha_1^{n-1} \\ \vdots & \vdots & \cdots & \vdots \\ 1 & \alpha_l & \cdots & \alpha_l^{n-1} \end{bmatrix}, \text{ similarly for } V_\beta^n$$

$$D_n = \begin{bmatrix} D_l & 0 \\ 0 & D_{n_1} \end{bmatrix} \text{ where}$$

$$D_l = diag\{M_n(\alpha_1)F(\alpha_1), ..., M_n(\alpha_l)F(\alpha_l)\}$$

$$D_{n_1} = diag\{M_n(\beta_1)F(\beta_1), ..., M_n(\beta_{n_1})F(\beta_{n_1})\}$$

$$F_\gamma(s) = \gamma M_1(s) \widehat{G}_\gamma(s) \text{ where}$$
$$M_1 = \frac{dW_1(-s)}{dW_1(s)}$$

and outer function $\widehat{G}_\gamma(s)$ is the spectral factor of

$$\left(W_1(-s)W_1(s) - W_2(-s)W_2(s)E_\gamma(s)\right)^{-1}.$$

With the above definitions, γ_{opt} is defined as the maximum γ value that makes $\mathcal{R}(\gamma)$ in (3) singular. This method allows us to compute γ_{opt} within any given tolerance specification, provided that an interval in which the optimal performance level lies is known.

2.3 Computation of the $\mathcal{H}_\infty$ Optimal Controller

By using the Toker-Özbay formula it can be shown that C_{opt} is in the form

$$C_{opt} = \frac{W_1(s)}{\gamma_{opt}^2 d_\infty} \left(\frac{\widehat{G}_\gamma(s) N_o^{-1}(s)}{1 + H_n(s) + H_d(s)} \right)$$

where

$$H_n(s) + H_d(s) = \frac{R_o(s)}{d_\infty} \left(\frac{K_{opt}(s)}{\gamma} + M_n(s) M_1(s) \hat{G}_\gamma(s) \right) - 1, \tag{4}$$

$d_\infty = \gamma_{opt}^{-1} R_o(\infty) K_{opt}(\infty)$, $R_o(s) = \frac{nW_1(s) dW_1(s)}{nE_\gamma(s) M_d(s)}$ and K_{opt} is defined as

$$K_{opt}(s) = \frac{[s^0 \; s^1 \; \cdots \; s^{n-1}]\Psi_1}{[s^0 \; s^1 \; \cdots \; s^{n-1}]\Psi_2}$$

where the vectors Ψ_1 and Ψ_2 satisfy

$$\mathcal{R}(\gamma_{opt}) \begin{bmatrix} \Psi_1 \\ \Psi_2 \end{bmatrix} = 0.$$

In the decomposition (4), H_n and H_d are selected in such a way that the order of H_n is the same as the number of unstable poles of K_{opt}, if any; the poles of $H_n(s)$ are precisely the unstable poles of K_{opt}. Then, we obtain $H_d \in \mathcal{H}_\infty$. If we define

$$C_0(s) := (1 + H_n(s))^{-1}, \quad C_1(s) := \frac{W_1(s)}{\gamma_{opt}^2 d_\infty} \widehat{G}_\gamma(s) N_o^{-1}(s)$$

the resulting optimal controller is

$$C_{opt}(s) = \left(\frac{C_0(s)}{1 + C_0(s) H_d(s)} \right) C_1(s). \tag{5}$$

Note that C_0 is finite dimensional, and C_1 is outer.

Thus, an internally stable implementation (non-fragile) of the optimal controller is given above. Moreover, a finite dimensional approximately optimal controller can be obtained by approximating H_d by a rational transfer function, and approximating $\widehat{G}N_o^{-1}$ if this term is infinite dimensional.

2.4 Stable Implementation and Approximation of the Controller

Considering the controller expression (5), it is possible to obtain a finite dimensional controller with guaranteed feedback stability and performance bounds by replacing stable infinite dimensional terms by their finite dimensional approximations. More precisely, we define

$$C_a(s) := \left(\frac{C_0(s)}{1+C_0(s)H_{da}(s)}\right) C_{1a}(s), \tag{6}$$

where $H_{da} \in \mathcal{H}_\infty$ is a finite rational approximation of H_d, and C_{1a} is defined as

$$C_{1a}(s) := \frac{W_1(s)}{\gamma_{opt}^2 d_\infty}\widehat{G}_\gamma(s)N_{oa}^{-1}(s) \tag{7}$$

where N_{oa} is a finite dimensional outer function approximating N_o.

Under the controller C_a defined above the resulting sensitivity is

$$S_a = \frac{M_d(1+C_0H_{da})}{M_d(1+C_0H_{da})+M_nN_oC_0C_{1a}} \tag{8}$$

where $P = M_nN_o/M_d$ is the plant factorization. Let $T_a = 1 - S_a$, $S_{\rm opt} = (1 + PC_{opt})^{-1}$ and $T_{\rm opt} = 1 - S_{\rm opt}$. Then, we have the following result on feedback system stability and performance bound.

Proposition. The feedback system (C_a, P) is stable if

$$\delta := (\delta_1 + \delta_2) < 1 \tag{9}$$

where

$$\delta_1 := \|S_{\rm opt}\frac{C_0(H_{da}-H_d)}{1+C_0H_d}\|_\infty \tag{10}$$

$$\delta_2 := \|T_{\rm opt}(N_{oa}^{-1}N_o - 1)\|_\infty\ . \tag{11}$$

Moreover, in this case the resulting performance level is estimated by the following inequality

$$\gamma_a := \left\| \begin{bmatrix} W_1S_a \\ W_2T_a \end{bmatrix} \right\|_\infty \le \gamma_{\rm opt}\ \frac{1+\varepsilon}{1-\delta} \tag{12}$$

where

$$1+\varepsilon := \max\{\|\frac{1+C_0H_{da}}{1+C_0H_d}\|_\infty\ ,\ \|N_{oa}^{-1}N_o\|_\infty\}. \qquad \square$$

The above result gives a guideline on how H_d and N_o should be approximated for feedback system stability and near optimal performance.

3. DESCRIPTION OF THE MATLAB PROGRAM

The software provides a GUI for user to enter necessary inputs. The initial state of the GUI contains explanation of each input.

As the user enters all inputs (W_1, W_2, P and allowable precision level as well as the initial bounds for γ_{opt}), the software computes $M_n(s)$, $M_d(s)$ and $N_o(s)$ by using YALTA, and displays $M_n(s)$ and $M_d(s)$ in a new panel.

Initially, the program calculates singular values of $\mathcal{R}(\gamma)$ for each γ point in the given range $[\gamma_{min}\ ,\ \gamma_{max}]$. The algorithm continues with searching peaks of negative of singular values of $\mathcal{R}_\gamma$. By using `findpeaks` command in Matlab, the local minimum points of singular values and corresponding γ values are stored. The software iteratively selects minimum and maximum γ closer to resulting γ, until the corresponding singular value is less than the defined tolerance level. If the iteration number exceeds 4 and the resulting singular value does not drop 4/5 of the resulting value obtained in previous iteration, the next γ in stored data is chosen and the iterative algorithm is applied from the beginning. The implementation of this algorithm is given in Fig. 1. Parameters defined as `xval` and `yval` are γ values and corresponding singular values of $\mathcal{R}(\gamma)$ respectively. Also `ind` represents the chosen peak index starting from rightmost peak of negative values of singular values of $\mathcal{R}(\gamma)$.

Lastly, the software directly follows Toker-Özbay formula to compute infinite dimensional optimal controller, and provides finite dimensional functions defined in the formula. The Nyquist plot illustrates stability, performance plot shows that the computed γ_{opt} is consistent. Also the Bode plot of the optimal controller is given. Additionally, approximation of $H_d(s)$ and $N_o(s)$ may be entered manually to find a suboptimal controller.

4. EXAMPLE

In this section, a nominal plant is chosen as

$$P_o(s) = \frac{e^{-\tau s}}{s+1+4e^{-hs}}. \tag{13}$$

The uncertainty weight

$$W_2(s) = \frac{1.667s^3+6.333s^2+4.001s+0.0004}{s+4},$$

bounds the error caused by parametric uncertainty in h, and assumes increasing high frequency neglected dynamics for $\omega > 6$ rad/sec (see Fig. 2). Since the relative degree of the plant is 1 and the relative degree of W_2^{-1} is 2 the optimal controller is strictly proper with relative degree $2 - 1 = 1$. The sensitivity weight is chosen so that the feedback system tracks step-like reference inputs,

$$W_1(s) = \frac{\varepsilon s+1}{s+\varepsilon}\ \text{ where } \varepsilon = 0.0001.$$

Firstly, by varying τ and h, their effects on γ_{opt} are observed. Then, by using constant τ and h, optimal and suboptimal controllers are computed numerically.

When $\tau = 0.1$ and $h \in [0,\ 3]$, stability analysis by using YALTA shows that nominal plant is stable for $h \in [0,\ 0.4708)$. Furthermore, there are two unstable poles for the interval $h \in [0.4708,\ 2.0931)$ and four unstable poles in $h \in [2.0931,\ 3]$. The root loci shows how location of poles varies as h increases.

By applying only the first and second steps of the software, γ_{opt} is obtained for different values of h in the interval $[0\ ,\ 3]$. The resulting γ_{opt} shown in Fig. 4 are consistent with the locations of unstable poles: it discontinuously increases as number of unstable poles increases and decreases as natural frequency of conjugate unstable pole

```
if iterNu==1
  xvalue = xval;
  yvalue=abs(yval);
  plot(xvalue,yvalue)
  newFunc=-yval;
  figure;
  plot(xval,newFunc)
  [~,b]=findpeaks(newFunc);
  gamma_opt=xval(b);
  try
    minValCoeff=0.5*(gamma_opt(ind:ind+1)*[1;-1]);
  catch
    if gamma_opt(ind)>1e-2
      minValCoeff = 1e-2;
    else
      minValCoeff = 3*gamma_pos/4;
    end
  end
  gamma_pos = gamma_opt(ind);
  temp = yval(b);
else
  [~,b]=min(yval);
  gamma_pos=xval(b);
end
if yval(b)<EPS
  break;
else
  if iterNu>4 && yval(b)>4*temp/5
    iterNu = 1;
    ind = ind+1;
    gamma_pos = gamma_opt(ind);
    try
      minValCoeff=0.5*(gamma_opt(ind:ind+1)*[1;-1])
    catch
      if gamma_opt(ind)>1e-2
        minValCoeff = 1e-2;
      else
        minValCoeff = 3*gamma_pos/4;
      end
    end
  end
  gmin=gamma_pos-minValCoeff
  gmax=gamma_pos+minValCoeff
end
```

Fig. 1. *Implementation of* γ_{opt} *computation*

pairs decrease, since $W_2(j\omega)$ is a monotone increasing function with respect to ω, as shown in Fig. 2.

To test reliability of the results, effect of τ on γ_{opt} has been investigated when $h = 2.7$. As expected, γ_{opt} increases with increasing τ (see Fig. 5). Moreover, h is more dominant compared to τ in the sense of affecting γ_{opt}

A suboptimal controller is obtained for the plant (13), where $h = 2.7$ and $\tau = 0.1$ (see below).

By using YALTA, the plant factorization results are obtained as

$$M_n(s) = e^{-0.1s} \quad , \quad N_o(s) = \frac{M_d(s)}{s+1+4e^{-2.7s}}$$

$$M_d(s) = \frac{(s^2 - 0.6654s + 0.9881)(s^2 - 0.1602s + 9.221)}{(s^2 + 0.6654s + 0.9881)(s^2 + 0.1602s + 9.221)}$$

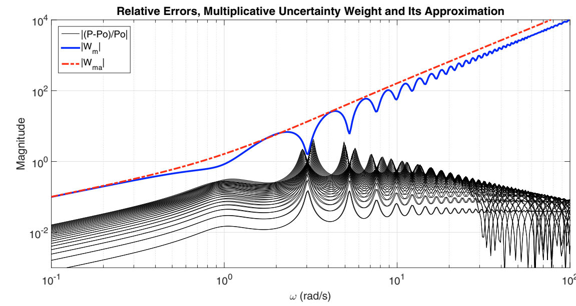

Fig. 2. *Relative errors and multiplicative uncertainty weight*

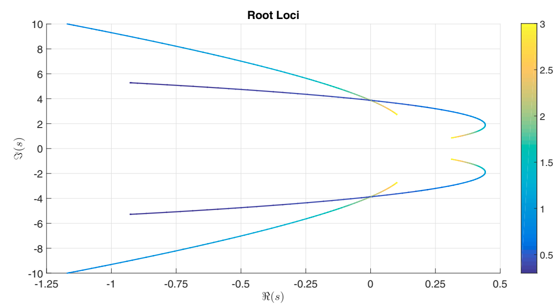

Fig. 3. *Root loci of denominator with respect to h*

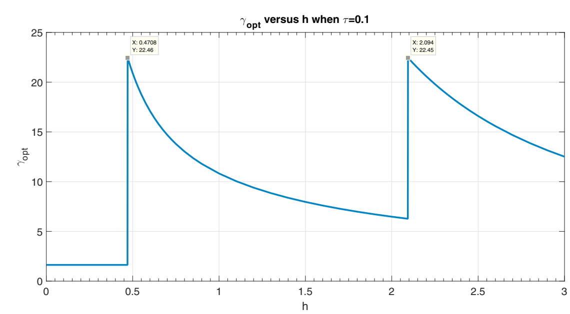

Fig. 4. γ_{opt} *versus* h *when* $\tau = 0.1$

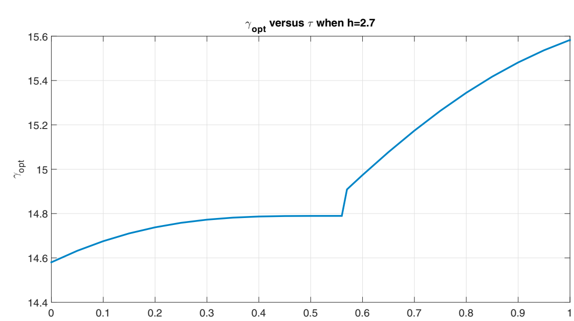

Fig. 5. γ_{opt} *versus* τ *when* $h = 2.7$

With such plant and weights, γ_{opt} is computed as 14.6755356 with a tolerance of 10^{-6}. In particular, the program generates

$$H_n(s) = \frac{53243(s + 0.1031)(s^2 - 3.53s + 10.29)}{(s - 4.107)(s - 0.09333)(s^2 - 1.258s + 1.545)}.$$

The infinite dimensional term $H_d(s)$ is approximated by the following fourth order rational stable transfer function by using `fitfrd` command of Matlab

$$H_{da}(s) = \frac{-43578(s+3.252)(s^2+1.217s+1.611)}{(s+2.99)(s+1.076)(s^2+0.9903s+0.7462)}.$$

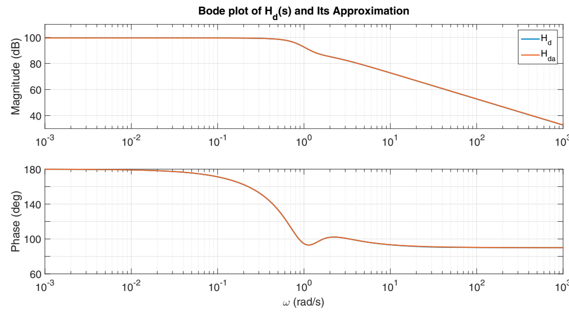

Fig. 6. *Bode diagrams of* $H_d(s)$ *and* $H_{da}(s)$

Since $N_o(s)$ is infinite dimensional, it also needs to be approximated to find a finite dimensional suboptimal controller. We use a 12^{th} order transfer function ($N_{oa}(s)$) by using `fitfrd` command again to approximate $N_o(s)$ so that the performance degradation is at an acceptable level. The resulting finite dimensional controller $C_a(s)$ is 16^{th} order and it turns out that it is a stable transfer function (see Bode plots in Fig. 8).

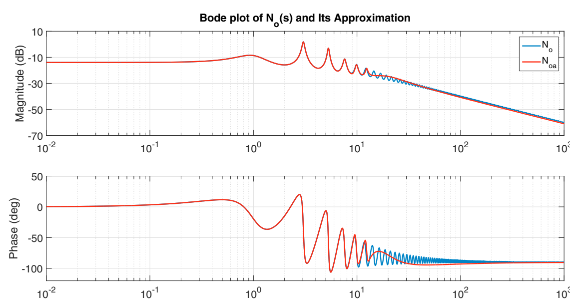

Fig. 7. *Bode diagrams of* $N_o(s)$ *&* $N_{oa}(s)$

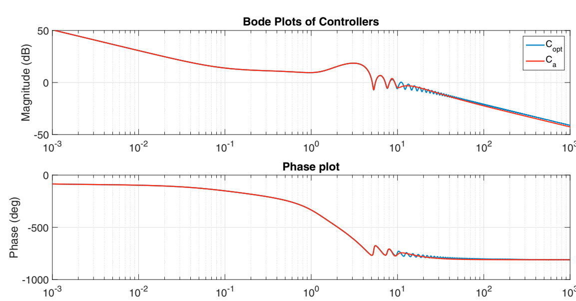

Fig. 8. *Bode diagrams of optimal & suboptimal controllers*

As mentioned before, when $\tau = 0.1$ and $h = 2.7$, the plant has four unstable poles. By observing that the Nyquist plots of PC_{opt} and PC_a encircle -1 four times in the CCW direction (see Fig. 9 and Fig. 10), we conclude that $C_a, C_{opt} \in \mathcal{H}_\infty$.

Performance of suboptimal controller can be observed from Fig. 11, where $\psi(j\omega)$ is defined as

$$\psi(\omega) := \left\| \begin{bmatrix} W_1(j\omega)S_a(j\omega) \\ W_2(j\omega)T_a(j\omega) \end{bmatrix} \right\|$$

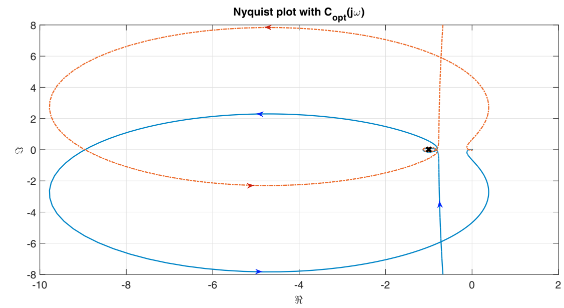

Fig. 9. *Nyquist plot of* $C_{opt}P_o$

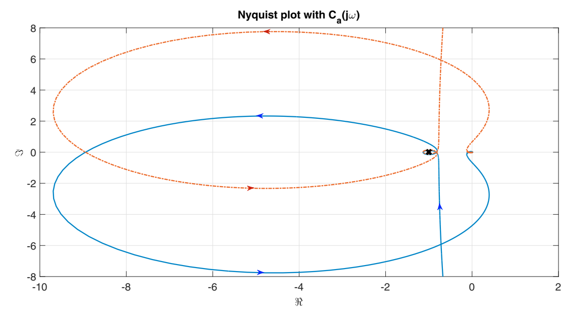

Fig. 10. *Nyquist plot of* C_aP_o

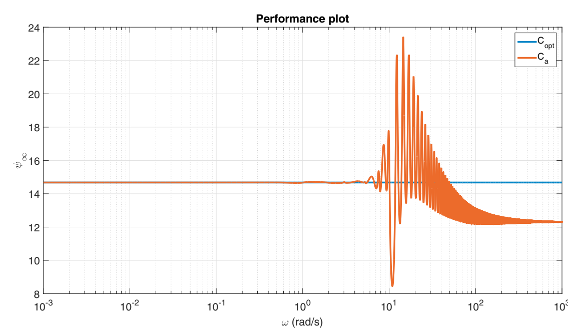

Fig. 11. *Performance plot of suboptimal controller:* $\psi(\omega)$ *versus* ω.

where $S_a = (1 + P_oC_a)^{-1}$ and $T_a = 1 - S_a$.

Since approximation of $N_o(s)$ is a 12^{th} order function, the approximation error around 14.57 rad/s reaches its maximum value. Therefore, the performance can be improved by using a higher order approximation of $N_o(s)$. Applying the result given by the main result of Section 2.4, we find that

$$\delta_1 = 0.0339, \ \delta_2 = 0.0368,$$

$$\delta = 0.0707, \ \varepsilon = 0.5731.$$

Since $\delta < 1$, the finite dimensional controller C_a stabilizes P. Furthermore, the approximation error is bounded by

$$\frac{\gamma_a - \gamma_{opt}}{\gamma_{opt}} = \frac{1+\varepsilon}{1-\delta} = 1.69$$

that is, γ_a is within 69% of γ_{opt}. Indeed, from From Fig. 11, we see that the actual relative error is 59%,

$$\frac{23.40 - 14.68}{14.68} = 0.59 \leq 0.69.$$

5. CONCLUSION

The paper proposes a software, which is implemented in Matlab, that finds $\mathcal{H}_\infty$ optimal controller directly by using Toker-Özbay formula and allows a large set of plants to be entered as input by using YALTA. There is an additional iterative algorithm applied on Toker-Özbay's formula, to find γ_{opt} according to desired tolerance and given interval. Many examples have been solved to test reliability of the software. As a result, by combining one of the most widely used formulas for numerical computation of $\mathcal{H}_\infty$ controllers and YALTA, the developed software is able to find optimal controllers for a wide set of time delay systems. Also implemented in this software is approximation of the optimal controller by identifying its stable infinite dimensional parts. Moreover, an approximation error bound is derived for the performance deviation under proposed controller approximation scheme.

REFERENCES

Avanessoff, D., Fioravanti, A., and Bonnet, C. (2008). YALTA: a Matlab toolbox for the $\mathcal{H}_\infty$-stability analysis of classical and fractional systems with commensurate delays. In *IFAC Symposium on System, Structure and Control*, 839–844. Grenoble, France.

Engelborghs, K., Luzyanina, T., and Samaey, G. (2001). DDE-BIFTOOL v.2.00: a Matlab package for bifurcation analysis of delay differential equations. Technical report, Katholieke Universiteit Leuven.

Foias, C., Özbay, H., and Tannenbaum, A. (1996). Robust control of infinite dimensional systems: Frequency domain methods. In *Lecture Notes in Control and Information Sciences*, volume 209. Springer, London.

Foias, C., Tannenbaum, A., and Zames, G. (1986). Weighted sensitivity minimization for delay systems. *IEEE Trans. Automatic Control*, 31, 763–766.

Gümüşsoy, S. (2012). Coprime-inner/outer factorization of siso time-delay systems and fir structure of their optimal $\mathcal{H}_\infty$ controllers. *Int. J. Robust Nonlin. Cont.*, 22, 981–998.

Gümüşsoy, S. and Michiels, W. (2011). Fixed Order H-infinity Control For Interconnected Systems Using Delay Differential Algebraic Equations *SIAM J. Control and Optimization*, 49, 2212–2238.

Lypchuk, T., Smith, M., and Tannenbaum, A. (1988). Weighted sensitivity minimization: General plants in $\mathcal{H}_\infty$ and rational weights. *Linear Algebra and its Applications*, 109, 71–90.

Mirkin, L. and Tadmor G. (2002). $\mathcal{H}_\infty$ Control of System with I/O Delay: A review of some problem-oriented methods, *IMA Journal of Mathematical Control and Information*, 19, 185–199.

de Oliveira, M.C. and Geromel, J.C. (2004). Synthesis of non-rational controllers for linear delay systems. *Automatica*, 40, 171–188.

Toker, O. and Özbay, H. (1995). $\mathcal{H}_\infty$ Optimal and suboptimal controllers for infinite dimensional SISO plants *IEEE Trans. on Automatic Control*, 40, pp. 751–755.

Vyhlìdal, T. and Zítek, P. (2009). Mapping based algorithm for large-scale computation of quasi-polynomial zeros. *IEEE Transactions on Automatic Control*, 54, 171–177.

Zhou, K. and Khargonekar, P. (1987). On the weighted sensitivity minimization problem for delay systems. *Systems & Control Letters*, 8, 307–312.