μ-Synthesis: Simple Controllers for Time Delay Systems

Marek Dlapa, Roman Prokop

***Abstract*--The contribution is focused on μ-synthesis methodology applied to continuous-time systems with time delay. Robust design is considered as a problem of minimization of the peak of the structured singular value denoted μ-function. The evolutionary μ-synthesis consists of the pole placement control design based on polynomial Diophantine equations and a Differential Migration procedure for an optimization evaluation. The results are compared with controllers designed via the *D-K* iteration as a standard method for μ-synthesis.**

***Index Terms*—structured singular value, Differential Migration, evolutionary algorithms, time delay, controller design**

I. Introduction

The robust control is one of the most frequent topics in the control theory. Recent years have brought non-traditional methods considering both structured and unstructured uncertainties.

Some of the methods are based on H_∞ approach in the ring of stable and proper transfer functions. These methods provide the measure that indicates robustness of designed controller. However, this measure evaluates only the robust stability. On the other hand, methods based on the Zames' small gain theorem [10] yield both robust stability, and performance conditions. One of them is the structured singular value - μ (e.g. [2]) which consider the robust stability and performance objectives simultaneously. Two methods for the μ-synthesis were derived: the *D-K* iteration [3] and μ-*K* iteration [6]. The *D-K* iteration yields an optimal controller minimizing peak of the μ-function. However, the controller is usually a high order transfer function and for further application it is simplified via some kind of approximation. If the simplification is too substantial it can cause degradation of frequency properties of the controller and whole feedback loop. This problem can be resolved by the evolutionary μ-synthesis treated in this contribution. In this method the controller is designed through the algebraic approach with the pole placement principle and the position of the poles is tuned through the newly developed evolutionary algorithm with the evaluation by the structured singular value.

In this paper the evolutionary μ-synthesis and *D-K* iteration is applied to a system with time delay and the resulting controllers are compared.

II. Outline of Pole Placement Design

The pole placement principle is one of traditional and well-known methods for the controller design (e.g. [4, 5, 7, 8]) which is simple for derivation and tuning. Consider a one degree of freedom (1DOF) structure of the closed loop system depicted in Fig. 1 with two external inputs – the reference w and disturbance v, respectively. The output and tracking error is according to Fig. 1 in the form

$$y(s) = \frac{bq}{d} w(s) + \frac{cfp}{d} v(s). \tag{1}$$

$$e(s) = w(s) - y(s) = \frac{afp}{d} \cdot \frac{h_w}{f_w} - \frac{cfp}{d} \cdot \frac{h_v}{f_v}, \tag{2}$$

where

$$afp + bq = d \tag{3}$$

is the characteristic polynomial of the closed loop system in Fig. 1.

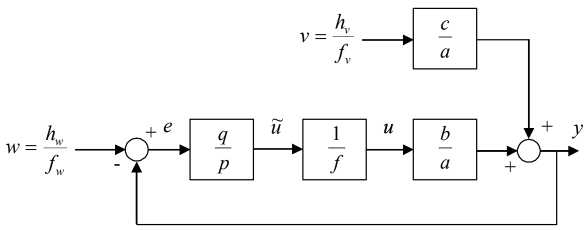

Fig. 1. The structure of the 1DOF system.

It can be proved that the asymptotic tracking of a reference is achieved if and only if the polynomial pfa is divisible by f_w and v is rejected if pfa is divisible by f_v. As a consequence, polynomials p, q are solutions to Diophantine equation (3).

It is also desirable that transfer function $\frac{q}{fp}$ is proper. The analysis of polynomial degrees in (3) for the most frequent case $f_w = f = s$ (stepwise reference) gives

$$\deg d = 2 \deg a \tag{4}$$

A frequent choice for polynomial d is then

Institute of Information Technologies, Tomas Bata University in Zlin, Mostni 5139, 760 01 Zlin, Czech Republic, fax: +420 57 603 3333, e-mail: m_dlapa@ft.utb.cz; prokop@ft.utb.cz

$$d(s)=\prod_{i=1}^{\deg d}(s+\alpha_i). \tag{5}$$

where $\alpha_i > 0$ are the tuning parameters of the controller design and d is a stable polynomial which ensures the stability of the system.

As an example consider a nominal plant transfer function

$$P(s)=\frac{b(s)}{a(s)}=\frac{b_0}{s^2+a_1 s+a_0} \tag{6}$$

and

$$w=\frac{h_w}{f_w}=\frac{1}{s}, \tag{7}$$

$$v=\frac{h_v}{f_v}=\frac{1}{s}. \tag{8}$$

Then equation (3) has the form

$$\begin{aligned}&\left(s^2+a_1 s+a_0\right)s\left(s+p_0\right)+b_0\left(q_2 s^2+q_1 s+q_0\right)=\\&=s^4+d_3 s^3+d_2 s^2+d_1 s+d_0.\end{aligned} \tag{9}$$

and by simple equating the coefficients at the like power of s at the left and right of (9) it can be obtained

$$\begin{aligned}p_0&=d_3-a_1\\q_2&=\left(d_2-a_0-a_1 p_0\right)/b_0\\q_1&=\left(d_1-a_0 p_0\right)/b_0\\q_0&=d_0/b_0\end{aligned} \tag{10}$$

Then the resulting controller is proper and has a "generalized PID" structure in the form

$$Q=\frac{q_2 s^2+q_1 s+q_0}{s\left(p_1 s+p_0\right)} \tag{11}$$

III. μ-SYNTHESIS

Consider a family of plant transfer functions $\widetilde{P}$. Suppose that the nominal plant transfer function is P and consider the perturbed plant transfer function in the form of $\widetilde{P}=(1+\Delta W_2)P$, where W_2 is a fixed stable transfer function, the weight, and Δ is a variable stable transfer function satisfying $\|\Delta\|_\infty \le 1$.

Let S denote perturbed transfer function from the reference input w to tracking error e. Let W_1 denote a weighting function and let the following performance condition be defined as

$$\|W_1 S\|_\infty < 1. \tag{12}$$

If condition (12) holds, then the behaviour of the closed loop can be changed through W_1.

The closed-loop feedback system can be transformed to that in Fig. 3, where **M** is the lower linear fractional transformation (LFT) on **G**(s) and controller K(s), i.e.,

$$\mathbf{M}=\mathbf{F}_l(\mathbf{G},K)=\mathbf{G}_{11}+\mathbf{G}_{12}K\left(1-G_{22}K\right)^{-1}\mathbf{G}_{21}, \tag{13}$$

the other element of the system is

$$\widetilde{\Delta}=\begin{bmatrix}\delta_1 & 0 & 0\\ 0 & \delta_2 & 0\\ 0 & 0 & \delta_n\end{bmatrix} \tag{14}$$

where **G**(s) is a generalized plant including the nominal plant and weighting functions, which can be parted to

$$\mathbf{G}(s)=\begin{bmatrix}\mathbf{G}_{11} & \mathbf{G}_{12}\\ \mathbf{G}_{21} & G_{22}\end{bmatrix} \tag{15}$$

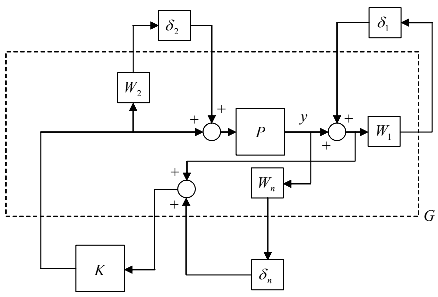

Fig. 2 The structure of the feedback system with multiplicative uncertainty.

The evolutionary μ-synthesis is applied to a 1DOF system for the interconnection shown in Fig. 2. The D-K iteration is applied to the same structure.

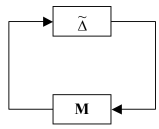

Fig. 3 The transformed closed-loop control system.

The structured singular value of matrix **M**, denoted $\mu_{\widetilde{\Delta}}(\mathbf{M})$, is defined as

$$\mu_{\widetilde{\Delta}}(\mathbf{M})=\frac{1}{\min\left\{\overline{\sigma}\left(\widetilde{\Delta}\right):\det\left(I-\mathbf{M}\widetilde{\Delta}\right)=0\right\}}, \tag{16}$$

and if no $\widetilde{\Delta}$ exists which makes $\mathbf{I}-\mathbf{M}\widetilde{\Delta}$ singular, let $\mu_{\widetilde{\Delta}}(\mathbf{M})=0$ [1], where $\overline{\sigma}\left(\widetilde{\Delta}\right)$ denotes the maximum singular value. The control objective is to find a stabilizing controller **K** minimizing H_∞ norm of $\mu_{\widetilde{\Delta}}(\mathbf{M})$, i.e.,

$$\min_{\mathbf{K}\ \text{stabilizing}\ \mathbf{G}}\left\|\mu_{\widetilde{\Delta}}\left[\mathbf{F}_l(\mathbf{G},\mathbf{K})\right]\right\|_\infty. \tag{17}$$

The following result is used for the robust performance test (see [1]):
The feedback system has the robust performance, i.e., expression (12) holds, if and only if

$$\mu_{\tilde{\Delta}}(\mathbf{M}) < 1 \tag{18}$$

at all frequencies.

IV. DIFFERENTIAL MIGRATION

The philosophy of this optimization algorithm is based on migration of individuals to cluster centres along perturbed trajectories. The clusters are the individuals concentrated around their centres, which are individuals with the lowest cost value. The individuals then jump to the cluster centres. The jump size is a random value from the interval defined by the jump range, which is a predefined parameter. The perturbation causes that the individuals migrate in subspaces perpendicular to the dimension which is reset. It is based on a simple idea that the local extremes are usually placed symmetrically along the axes of the migration space. The perturbation also suppresses degeneration of the population. These facts cause better robustness of the algorithm in comparison with algorithm without the perturbation.

Verifications and tests proved that the DM (Differential Migration) is 60% faster at the same robustness than the older evolutionary algorithms, e.g. Differential Evolution [11, 12] and SOMA (Self-Organizing Migration Algorithm) [13]. The DM algorithm was applied to 2DOF and 1DOF systems with parametric uncertainties in [14] and [15] with reliable results.

The simplified flow chart of the DM is outlined in Fig. 4. The first step of the algorithm is parameters definition. It is necessary to define individual, i.e., the range of optimized parameters, its type (continuous, discrete, integer), jump range – JR, accepted error – AE, maximum number of migration loops – ML, coefficient of perturbation – PRT, cost function – CF and cluster distance – CD.

First a random population of individuals is generated. Each individual has assigned random parameters from the range defined in the previous step. Then the clusters are searched. The first cluster centre has assigned the first individual in the population. Then the next individual is selected and its pertinence to an existing cluster is checked by the condition

$$\sqrt{\sum_{i=1}^{n}\left(\frac{\mathrm{IND}_i - \mathrm{CC}_i}{\mathrm{UB}_i - \mathrm{LB}_i}\right)^2} \le \mathrm{CD}\,, \tag{19}$$

Parameters:

CD	cluster distance,
IND_i	i-th coordinate of the individual,
CC_i	i-th center of the cluster coordinate,
UB_i	upper bound of i-th dimension,
LB_i	lower bound of i-th dimension.

If condition (19) holds and CF of the individual is higher than CF of the centre then the next centre is chosen. Otherwise the current centre is replaced by the individual. If pertinence to all the clusters is checked and the individual does not belong to an existing cluster, then the individual is chosen to be a new cluster centre. Note that the clusters are not an exactly defined set of individuals. Therefore it is not possible to consider parameter CD as distance of two exactly defined sets. The CD is just a parameter of the algorithm not a distance in mathematical sense.

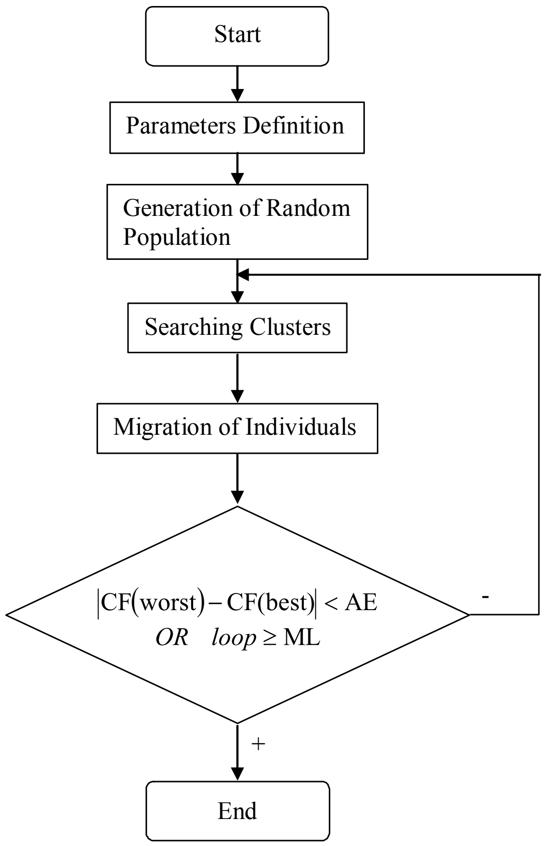

Fig. 4 The simplified flowchart of the Differential Migration.

Then the individuals migrate to all cluster centres. For each individual and cluster centre the direction vector $\overrightarrow{\mathrm{DV}}$ is calculated by expression

$$\overrightarrow{\mathrm{DV}} = \overrightarrow{\mathrm{CC}} - \overrightarrow{\mathrm{IND}}\,. \tag{20}$$

For each coordinate a random number from the range $\langle 0;1 \rangle$ is generated. If the number is lower than PRT then the coordinate remain unchanged, otherwise 0 is assigned so that individual migrates in the subspace perpendicular to the reset dimension. The individuals then jump in the directions defined by $\overrightarrow{\mathrm{DV}}_{\mathrm{P}}$. Location of the jump is obtained from the expression

$$\overrightarrow{\text{jump point}} = \overrightarrow{\text{IND}} + \overrightarrow{\text{DV}}_{\text{P}} \cdot k; \quad k \in \text{JR}, \tag{21}$$

where k is a random number from JR. If the CF of the $\overrightarrow{\text{jump point}}$ is lower than CF of the old $\overrightarrow{\text{IND}}$ then the $\overrightarrow{\text{jump point}}$ is assigned to $\overrightarrow{\text{IND}}$ in the new population. If any of the parameters exceeds predefined range, then a random number from the range is assigned to this parameter.
The algorithm repeats cluster searching and migration until number of loops exceeds ML, or if CF of the worst and best individual is lower than AE (see Fig. 4).

V. Application to a Time Delay System

The design of the feedback controller is based on evolutionary approach with evaluation by μ-function. The DM searches a suitable pole placement, so that $\left\| \mu_{\widetilde{\Delta}} \{\mathbf{M}[Q(d)]\} \right\|_\infty$ is minimal, where $\mathbf{M}$ is a generalized plant for a controller Q which is obtained from the expressions (10) for the nominal plant. The individual consists of parameters α_i defining the nominal feedback system poles. The location of the poles is constrained at real axis in the left half-plane, so that the closed-loop system is aperiodic.

Consider the family of plant transfer functions

$$\widetilde{P} = \left\{ \frac{e^{-\tau s}}{s^2 + 2s + 1} : 0 \le \tau \le 10 \right\} \tag{22}$$

To apply the multiplicative perturbation described above to the plant family (22), weight W_2 must be established. The nominal plant can be defined as

$$P = \frac{1}{s^2 + 2s + 1} \cdot \tag{23}$$

Weight W_2 must satisfy

$$\left| \frac{\widetilde{P}(j\omega)}{P(j\omega)} - 1 \right| \le \left| W_2(j\omega) \right|, \quad \forall \omega, \tau, \tag{24}$$

that is,

$$\left| e^{-\tau j\omega} - 1 \right| \le \left| W_2(j\omega) \right|, \ \forall \omega, \tau . \tag{25}$$

Weight W_2 can be chosen as the envelope curve of function $\left| e^{-\tau j\omega} - 1 \right|$:

$$W_2(s) = \frac{22s}{10s + 1} \tag{26}$$

In Fig. 5 the Bode plot of weight W_2 and function $\left| e^{-\tau j\omega} - 1 \right|$ for $\tau = 10$ is shown.

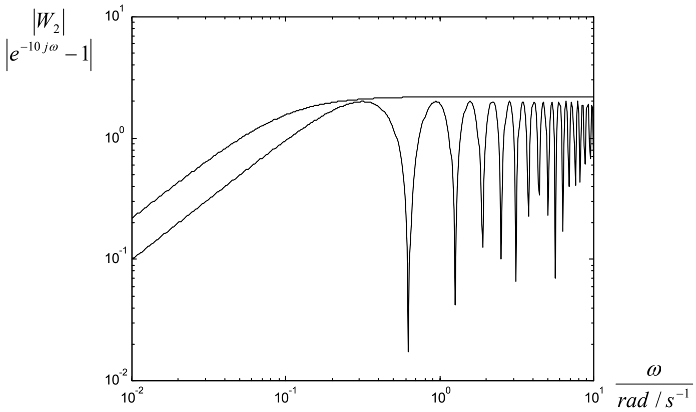

Fig. 5 Weight W_2 (dashed) and function $\left| e^{-\tau j\omega} - 1 \right|$ (solid).

Weight W_1 in the performance condition (12) can be easily derived from the sensitivity transfer function S and complementary sensitivity function T by the following procedure. Let

$$\begin{aligned} \widetilde{S} &= 1 - \widetilde{T} = 1 - \frac{1}{(\tau_1 s + 1)(\tau_2 s + 1)} = \\ &= \frac{\tau_1 \tau_2 s^2 + (\tau_1 + \tau_2)s}{\tau_1 \tau_2 s^2 + (\tau_1 + \tau_2)s + 1} \end{aligned} \tag{27}$$

denote the required shape of S. Weight W_1 is chosen so that it is less than $1/\widetilde{S}$. However, W_1 cannot be equal to $1/\widetilde{S}$ since this would cause the performance condition unachievable. Thus for $\tau_1 = \tau_2 = 20$ weight W_1 can be defined as follows

$$W_1 = \frac{10s + 1}{s + 0.0001} \cdot 0.016 . \tag{28}$$

The Bode plot of W_1, and $1/\widetilde{S}$ is shown in Fig. 6. One can see that W_1 is a simplified envelope curve of $1/\widetilde{S}$.

Weight of noise is

$$W_n = \frac{s/0.001 + 1}{s/10 + 1} 0.0016. \tag{29}$$

The cost function is $\left\| \mu_{\widetilde{\Delta}} \{\mathbf{M}[Q(d(\overrightarrow{\text{IND}}))]\} \right\|_\infty$ where $\mathbf{M}$ is the LFT on generalized plant $\mathbf{G}$ and controller $K = Q$.
Parameters of the DM optimization:
the individual was defined as 4 real parameters in the range of 0 to 20, jump range $\text{JR} = \langle 0.2; 4 \rangle$, accepted error = 0, maximum number of migration loops ML = 10, number of

the individuals NP = 30, const. of perturbation PRT = 0.6, cost function CF = $\left\| \mu_{\tilde{\Delta}} \{\mathbf{M}[Q(d(\overrightarrow{\mathrm{IND}}))]\} \right\|_{\infty}$, cluster distance CD = 0.25.

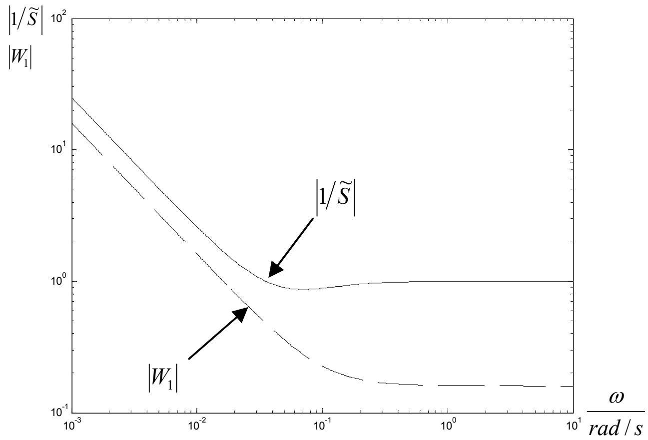

Fig. 6 Weight W_1(dashed) and $1/\tilde{S}$ (solid).

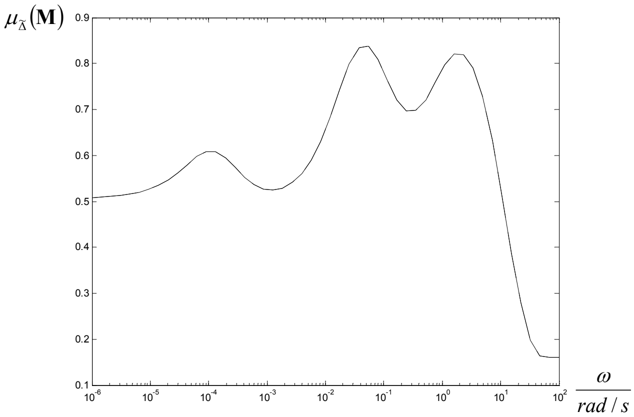

Fig. 7 The resulting structured singular value plot for the controller designed through the evolutionary approach.

The μ-plot of the controller designed through the evolutionary approach does not exceed 1 (Fig. 7) and control objectives are met. The μ-plots of simplified controllers obtained via the *D-K* iteration are shown in Fig. 8. One can see that the μ-plot is getting worse with the decreasing order of the controller. The μ-plot for the controller approximated by the 3rd order transfer function is considerably worse than in the evolutionary approach.

The pole placement design gives:

$$\alpha_1 = 6.09; \alpha_2 = 1.79; \alpha_3 = 0.80; \alpha_4 = 0.03; \qquad (30)$$

and the feedback controller is:

$$Q = \frac{q}{pf} = \frac{1.079s^2 + 1.599s + 0.281}{s^2 + 7.712s} \qquad (31)$$

The controller obtained via the *D-K* iteration was approximated by the 4th order transfer function. Since the parameters at low powers of s were almost equal to zero the resulting controller is the 3rd order transfer function:

$$K = \frac{0.950s^3 + 9.727s^2 + 1.475s + 0.015}{s^3 + 40.09s^2 + 0.687s}. \qquad (32)$$

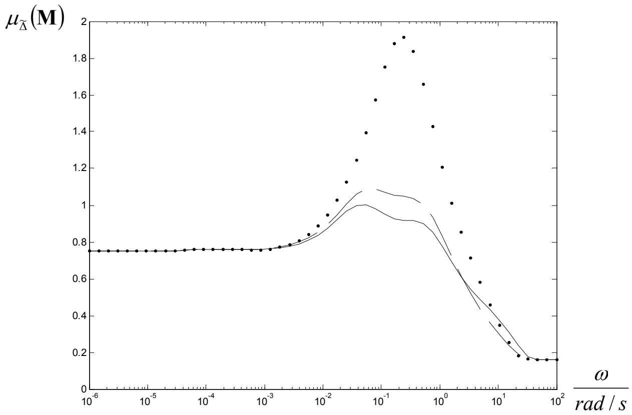

Fig. 8 The μ-plot for the controllers designed via the *D-K* iteration approximated by 6th(solid), 4th (dashed) and 3rd (dot) order transfer function.

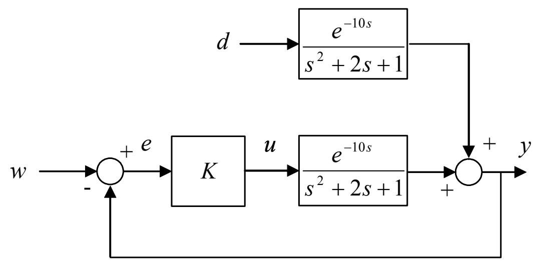

Fig. 9 The feedback system used in simulation.

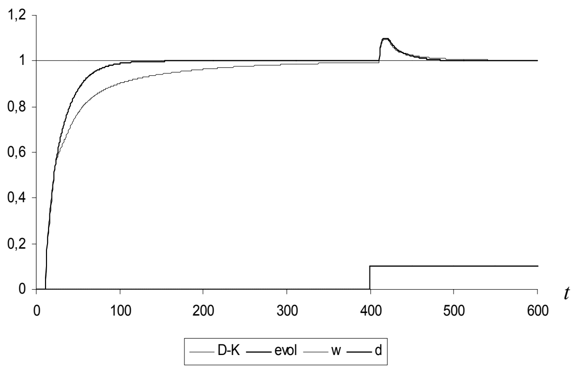

Fig. 10 The simulation of the feedback system obtained via the *D-K* iteration and evolutionary approach.

The simulation was applied to the feedback system in Fig. 9. The stepwise response of the reference signal and disturbance is in Fig. 10. It can be seen that the evolutionary

μ-synthesis yields faster tracking and disturbance rejection than controller K obtained via the D-K iteration.

VI. Conclusion

The paper presents comparison of two different approaches to the continuous time controller design for time-delay systems. The developed method is based on some optimization of μ-function as a multivariable one. The optimal value of μ-function was obtained by the Differential Migration algorithm. The D-K iteration principle was applied as the second method for a comparison. The evolutionary μ-synthesis minimizes μ function directly for the controller with a given simple structure while the D-K iteration converges to the final value of μ via scaling matrices D, D^{-1} that increase the order of the resulting controller. That is why the μ plot of the simplified controllers obtained through the D-K iteration have higher peak value than the evolutionary one. The simulation results confirm that the simplified controller obtained by the evolutionary approach gives better responses from the point of view of tracking as well as disturbance rejection.

VII. Acknowledgements

This work was supported by the grant of FRVS area G1, no. 34/2002, and project CEZ MSM 281 100001

VIII. References

[1] J.G. Balas and A. Packard, *The control handbook*, CRC Press, Inc., 1996, pp. 671-687.
[2] J.C. Doyle, “Analysis of feedback systems with structured uncertainties,” *Proceedings of IEEE, Part-D*, 1982, 129, pp. 242-250.
[3] J.C. Doyle, “Structure uncertainty in control system design,” *Proceedings of 24th IEEE Conference on decision and control*, 1985, pp. 260-265.
[4] V. Kucera, *Analysis and design of discrete linear control systems*, Academia, 1991.
[5] V. Kucera, “Diophantine equations in control – a survey,” *Automatica*, **Vol. 29**, No. 6, pp. 1361-75, 1993.
[6] J. L. Lin, I. Postlethwaite and D. W. Gu, “μ-K iteration: a new algorithm for μ synthesis,” *Automatica*, **29**, 219-224, 1993.
[7] R. Prokop and J. P. Corriou, “Design and analysis of simple robust controllers,” *Int. J. Control*, **Vol. 66**, No. 6, pp 905-921, 1997.
[8] R. Prokop, P. Dostal and A. Meszaros, “Continuous-time control via proper regulators,” *Journal A*, **Vol. 33**, No. 4, 1992.
[9] M. Vidyasagar, *Control system synthesis – A factorization approach*, MIT Press Cambridge, 1985.
[10] G. Zames, “Feedback and optimal sensitivity: model reference transformations, multiplicative seminorms, and approximate inverse,” *IEEE Trans. Auto. Control*, **26**, 1981, pp. 301-320.
[11] R. Storn, “System design by constraints adaptation and Differential Evolution”, *IEEE Trans. on Evolutionary Computation*, **Vol. 3**, 1999, No. 1, pp 22-23.
[12] Internet: http://www.icsi.berkeley.edu/~storn/code.html
[13] M. Dlapa, I. Zelinka and R. Prokop, “Evolutionary algorithms in fuzzy logic control,” *Proceedings of Process Control ’01*, 2001.
[14] M. Dlapa and R. Prokop, “Evolutionary μ-Synthesis for Systems with Parametric Uncertainties”, *Proceedings of Conference on Control Application*, Glasgow, 2002.
[15] M. Dlapa and R. Prokop, “Evolutionary μ-Synthesis: A Simple Controller for Feedback Loop”, *Proceedings of 10th Mediterranean Conference on Control and Automation*, Lisbon, 2002.