Robust Control of a Servo System Using mu Synthesys and Regional Pole Placement Constraints

Asparuh G. Markovski

***Abstract* — Two approaches for synthesizing a robust regulator for a laboratory model of the servo system manufactured by Inteco are compared: mu synthesis using dksyn routine and H∞ design with regional pole placement constraints using h2hinfsyn, both from Robust Control Toolbox for MATLAB. Some of the inaccurate parameters are introduced as sources of structured uncertainty. Additionally, unstructured uncertainty has been introduced due to the presence of nonlinearity in the "gap" type object. Simulation experiments were performed with the model. The operability of the created regulator was tested on the experimental set-up.**

***Index Terms* — Robust Control, μ synthesis, H∞ design, pole placement**

I. Introduction

The only complete and practically usable tool for solving the problem of H∞ and μ optimal synthesis is the Robust Control Toolbox for MATLAB. Nevertheless, other attempts for creating software for robust control has been made: For example, to meet the need for highly efficient, portable programs in the field of automatic control theory, under the NICONET (Numerics In COntrol NETworks) project [1] an Internet-accessible virtual library of FORTRAN computing programs called SLICOT (A Subroutine LIbrary in systems and COntrol Theory) [2] has been developed by leading experts in management theory in the European Community. Another interesting software bunch is the LCT Toolbox [3] from the Catholic University of Leuven, which aim is to aggregate the existing software tools for robust control design in MATLAB in objects from the according classes along with the measurement information and results from the identification.

In this paper a brief comparison between two approaches for robust control synthesis is made: mu synthesis using dksyn routine and H∞ design with regional pole placement constraints using h2hinfsyn, both from Robust Control Toolbox for MATLAB. The work is illustrated by a physical model of servo system. A similar results obtained using author software, included in the SLICOT library, are described in [4].

II. Physical models used

To demonstrate the methods of robust synthesis using both approaches, a physical model of a servo system from the laboratories of the Department of Systems and Control at the Technical University - Sofia was used.

This work has been done in the laboratories in the Technical University in Sofia, Faculty of Automatics.

The author Asparuh Markovski is from the Faculty of Automatics, Technical University of Sofia, 1000 Sofia, Bulgaria (e-mail: agm@tu-sofia.bg).

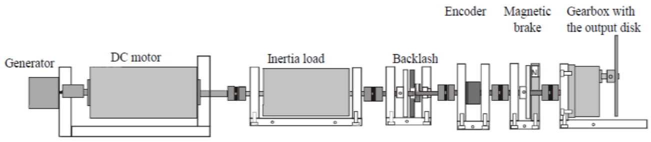

Fig. 1. Model of the servo system

The servo system (Fig. 1) consists of a DC motor, a tachogenerator, a load with a significant moment of inertia, an element with one revolution of dead zone, a magnetic brake, an incremental encoder, an output disk with a reducer [5]. All parts are rail mounted and can be easily moved. The shaft rotation angle is measured by both the incremental encoder and the tachogenerator. The motor is controlled by PWM so that the control signal is scaled, i.e. $|u(t)| \leq 1$. The system connects to a computer via an RT-DAC device and can be controlled via MATLAB. The dynamics of the system is described by the following dependences according to Fig. 2:

$$v(t) = Ri(t) + K_e\omega(t) \tag{1}$$

$$J\dot{\omega}(t) = K_m i(t) - \beta\omega(t) \tag{2}$$

where $v(t)$ is the input voltage, $i(t)$ is the current, $\omega(t)$ is the angular velocity of the shaft, R is the resistance of the motor coil, J is the inertial moment of all rotating parts, β is the coefficient of friction, $K_m i(t)$ is the electromagnetic torque and $K_e\omega(t)$ is the reaction of the motor coil.

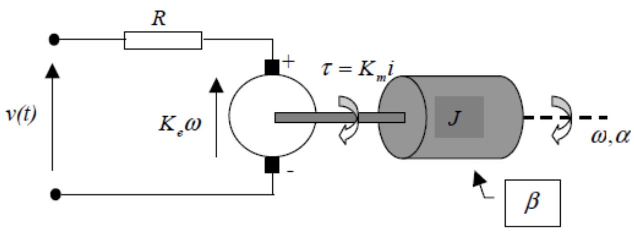

Fig. 2. Model of the dynamics of the servo system.

Combine the electrical (1) and mechanical (2) equations:

$$T_s\dot{\omega}(t) = -\omega(t) + K_{\mathrm{sm}}v(t) \tag{3}$$

where $T_s = \frac{\mathrm{RJ}}{\beta R + K_e K_m}$, $K_{\mathrm{sm}} = \frac{K_m}{\beta R + K_e K_m}$. The model is linear because the element with a gap (dead zone) is not included, as well as some nonlinearities due to friction. The system is then described by the following transfer functions:

$$W_{v\to\omega}(s) = \frac{\omega(s)}{v(s)} = \frac{K_{\mathrm{sm}}}{T_s s+1}, W_{v\to\alpha}(s) = \frac{\alpha(s)}{v(s)} = \frac{K_{\mathrm{sm}}}{s(T_s s+1)} \tag{4}$$

https://doi.org/10.47978/TUS.2021.71.02.014

From (4) a description in state space is obtained:

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix}=\begin{bmatrix}0 & 1\\ 0 & -\frac{1}{T_s}\end{bmatrix}\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix}+\begin{bmatrix}0\\ \frac{K_s}{T_s}\end{bmatrix}u \qquad (5)$$

$$y=\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix}$$

Linearization at the zero point with the gap element included would lead to an uncontrollable model, so a slightly different linear model is used for μ synthesis. It is obtained with the command in MATLAB linmod with simulated excitation. It is also possible to linearize not in the dead zone, or to approximate it with a small static gain.

Two types of models with uncertainty have been obtained for the purposes of robust synthesis:

a) Structured uncertainty model, where K_s is an uncertainty parameter with nominal value 186 rad/(s*V) and an uncertainty interval [166, 206] and T_s is an uncertainty parameter with a nominal value of 1.04 s and an uncertainty interval [0.84, 1.24]. This model is named G_{1m};

b) The logarithmic amplitude frequency responses are obtained for different values of these parameters. The maximum interval between them is modeled with unstructured uncertainty with maximum relative error $W_{\text{mult}}(\omega)=\begin{bmatrix}W_{\text{mult1}} & 0\\ 0 & W_{\text{mult2}}\end{bmatrix}$ and unmodeled dynamics $\Delta_{\text{mult}}=\begin{bmatrix}\Delta_{\text{mult1}} & 0\\ 0 & \Delta_{\text{mult2}}\end{bmatrix}$. Then the model with a multiplicative uncertainty is represented as follows:

$$G_{1nm}=(I+W_{mult}\,\Delta_{mult})\,G_{1m} \qquad (6)$$

The estimates of the maximum multiplicative uncertainty are as follows:

$$W_{\text{mult1}}=\frac{0.245s+0.0372}{s+0.39},\; W_{\text{mult2}}=\frac{0.462s+0.114}{s+0.924} \qquad (7)$$

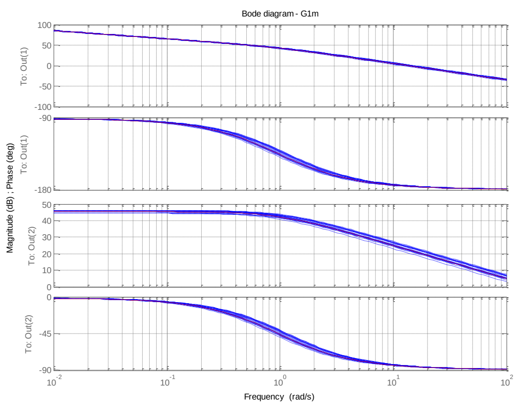

Fig. 3. Logarithmic frequency responses of G_{1m} for 20 arbitrary admissible realizations of the parameters with uncertainty

In Fig. 3 the logarithmic frequency responses of G_{1m} for 20 arbitrary admissible realizations of the parameters with uncertainty are shown. The figure for G_{1nm} has almost the same appearance and is not shown.

μ regulators for the two models described above are synthesized in accordance with the scheme of Fig. 4:

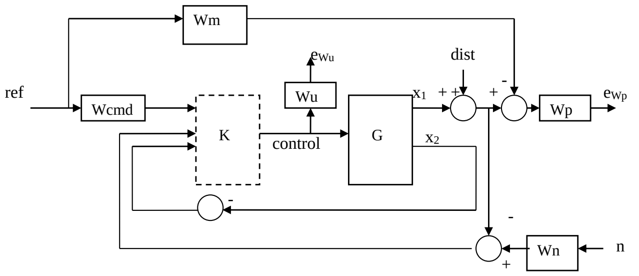

Fig. 4. Synthesis scheme

The weighting filters are as follows: $W_{cmd}=1$ is the reference model; $W_m=\frac{225}{s^2+21s+225}$ is the etalon closed-loop system model; $W_u=\frac{0.02s+0.2}{0.04s+1}$ is the control penalty; $W_p=\frac{5s+50}{10s+1}$ is the performance filter; $W_n=\frac{0.001s}{s+1}$ is the noise model.

III. SYNTHESIS OF μ REGULATORS KD_{MUM} FOR G_{1M} AND KD_{MUNM} FOR G_{1NM}

So, the first approach is to synthesize μ regulator using the MATLAB function dksyn, where the K step (H_∞ design) is made by γ iterations of obtaining successive suboptimal H_∞ regulators, estimated by solving Riccati equations approach; the model with uncertainty is first discretized and the regulator is in a discrete form.

The first μ regulator Kd_{mum} is synthesized for the discretized with Ts=0.002 s model with structured uncertainty G_{1m}. It achieves robust stability with a maximum value of μ norm 0.1964 (Fig. 5a) and the robust performance μ norm achieved is 0.843 (Fig. 5b).

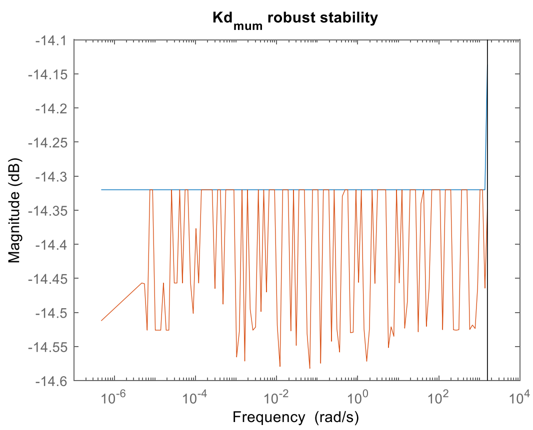

Fig. 5a. Robust stability for Kd_{mum} with G_{1m}

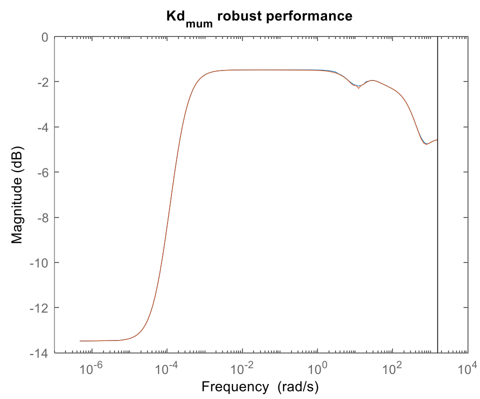

Fig. 5b. Robust performance for Kd_{mum} with G_{1m}

The step responses at reference 1 rad/s for 20 arbitrary admissible realizations of the uncertain parameters in the linear model are shown in Fig. 6: on the left is the response to the reference, on the right - to the load disturbance.

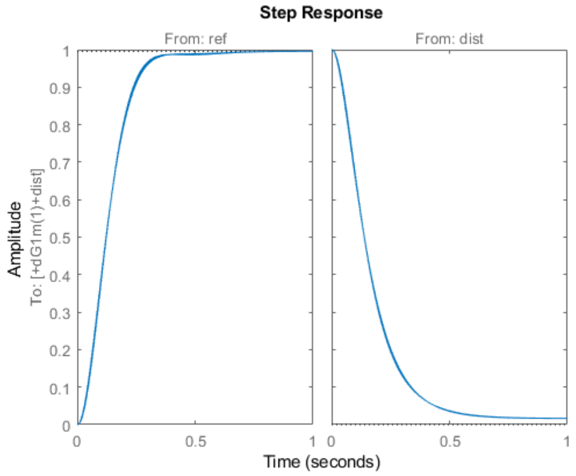

Fig. 6. Step responses for Kd_{mum} with G_{1m}

The second μ regulator Kd_{munm} is synthesized for the discretized with Ts=0.002 s model with unstructured uncertainty G_{1nm}. It achieves robust stability with a maximum value of μ norm 0.021 (Fig. 7a) and robust performance with μ norm 0.752 (Fig. 7b).

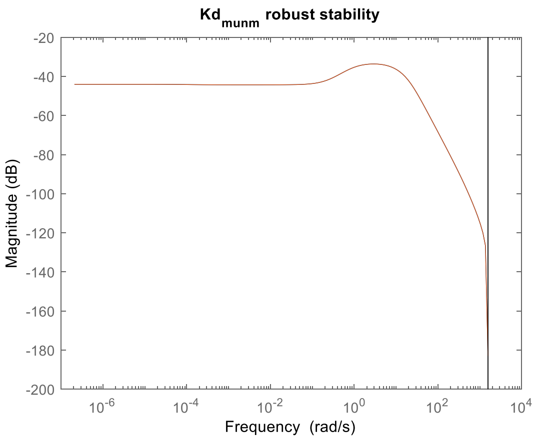

Fig. 7a. Robust stability for Kd_{munm} with G_{1nm}

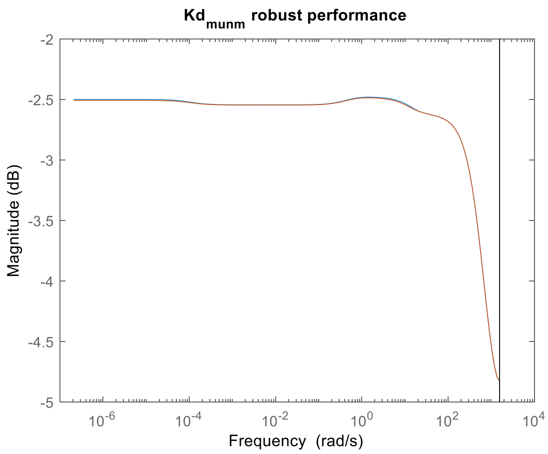

Fig. 7b. Robust performance for Kd_{munm} with G_{1nm}

The simulation results are similar to those for for Kd_{mum} with G_{1m} and are not shown.

IV. SYNTHESIS OF H_∞ REGULATORS $K005_{MD}$ FOR G_{1M} AND $K005_{NMD}$ FOR G_{1NM}

In this chapter the H_∞ synthesis is made by the Robust Control Toolbox function *h2hinfsyn*. It employs LMI techniques. According to Fig. 8, it works as follows:

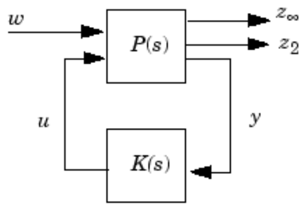

Fig. 8. Work scheme of *h2hinfsyn*

keeps the H_∞ norm of the transfer function G from w to z_∞ below the value specified;
keeps the H_2 norm of the transfer function H from w to z_2 below the value specified;
minimizes a trade-off criterion of the form $W_1G^2 + W_2H^2$, where the weights $W_1$and W_2 are given by the user;
places the closed-loop poles in the LMI region specified by the user.

In this chapter *h2hinfsyn* is used to make H_∞ only design, but with restrictions to poles of the closed-loop system. The H_∞ synthesis is made for the nominal realizations of the continuous-time G_{1m} and G_{1nm}, and then the corresponding regulators are discretized.

The pole placement restriction for the both cases is that the real part of the continuous closed loop system pole has to be less or equivalent to -0.05, which is (roughly) equivalent to the closed loop poles of the discrete time system to be in a circle with radius 0.999. Thus, we try to compensate the possible lack of robust stability by explicit moving away from the instability region.

The first H_∞ regulator $K005m_d$ is synthesized for the discretized with Ts=0.002 s model with structured uncertainty G_{1m}. It achieves robust stability with a maximum value of μ norm 0.1923 (Fig. 9a) and the robust performance μ norm achieved is 0.8783 (Fig. 9b).

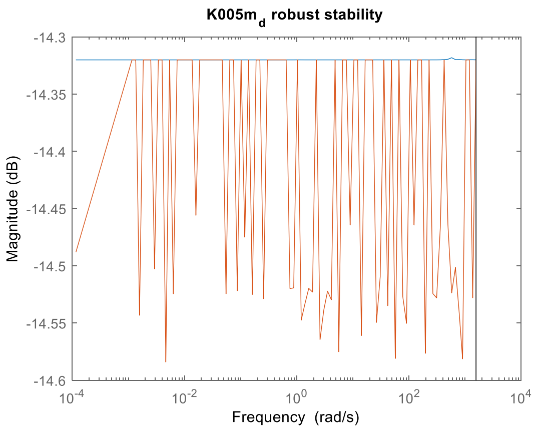

Fig. 9a. Robust stability for $K005m_d$ with G_{1m}

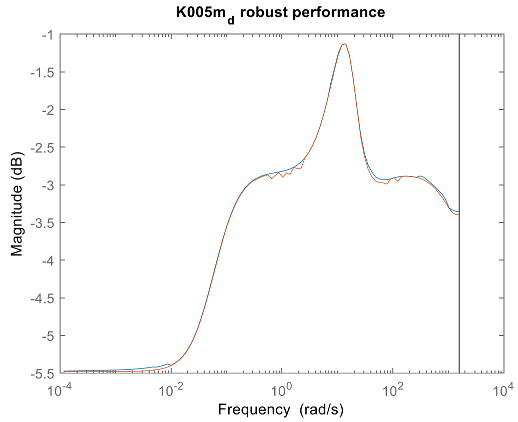

Fig. 9b. Robust performance for $K005m_d$ with G_{1m}

The step responses at reference 1 rad/s for 20 arbitrary admissible realizations of the uncertain parameters in the linear model are shown in Fig. 10: on the left is the response to the reference, on the right - to the load disturbance.

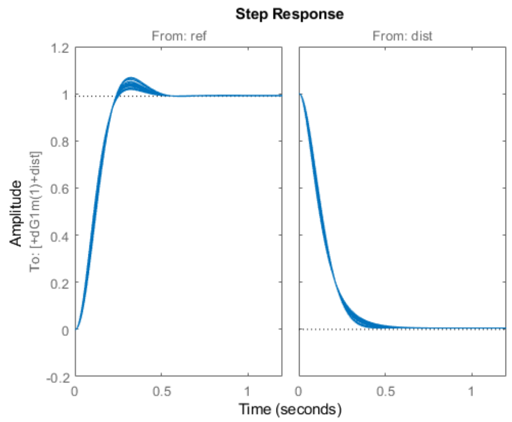

Fig. 10. Step responses for $K005m_d$ with G_{1m}

The overshoot is higher than in the case of μ design. Checking arbitrary realizations of the uncertain parameters shows that the absolute value of the closed-loop discrete time system poles is about 0.9999, i. e. the system is stable (the robustness is already proven by the Fig. 9b)

The second H_∞ regulator $K005nm_d$ is synthesized for the discretized with Ts=0.002 s model with unstructured uncertainty G_{1nm}. It achieves robust stability with a maximum value of μ norm 0.0353 (Fig. 11a) and robust performance with μ norm 0.034 (Fig. 11b).

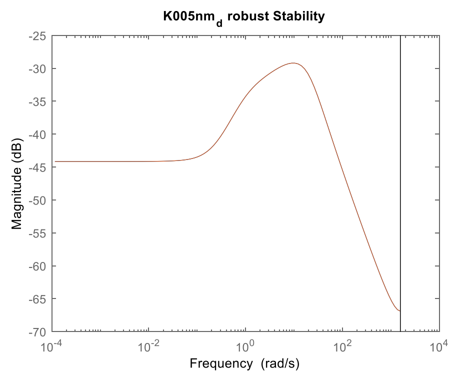

Fig. 11a. Robust stability for $K005nm_d$ with G_{1nm}

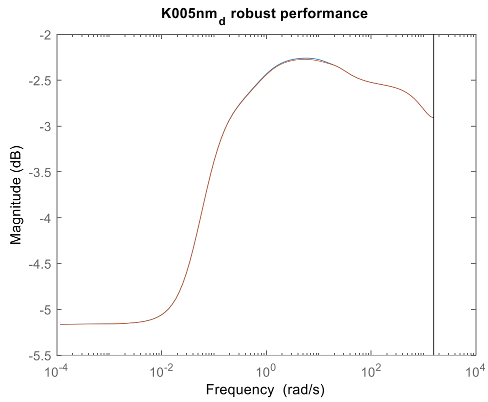

Fig. 11b. Robust performance for $K005nm_d$ with G_{1nm}

Similar to the Fig. 10 are the results of the simulation of $K005nm_d$ with G_{1nm} and are not shown. The absolute value of the closed-loop discrete time system poles is also about 0.9999.

V. Conclusion

From the experiments made with the real object (Fig. 12) it can be seen that workable regulators are obtained. Using Robust Control Toolbox functions *robuststab* and *robustperf* they are proven to be robust. The order of the μ controllers is not very high - they have 9 states for both models, and for $Kd1nm$ it can be reduced to 5. The order of $K005m_d$ is 7 and for $K005nm_d$ is 9.

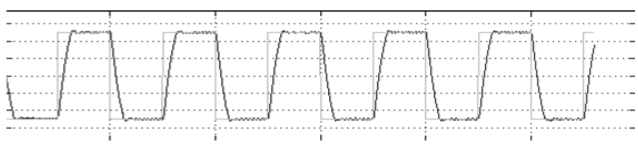

Fig. 12. Step responses for *Kd1m*. In light - reference angular velocity, in dark – actual angular velocity.

References

[1] NICONET e. V., http://www.niconet-ev.info/en/

[2] SLICOT - Subroutine Library in Systems and Control Theory www.slicot.org

[3] LCToolbox - an open-source linear control toolbox for MATLAB https://meco.pages.gitlab.kuleuven.be/lc_toolbox/

[4] Markovski, A. G, "Robust control of servosystem using MATLAB and SLICOT," in Proceedings of TU-Sofia, Bulgaria, 2019, Bulgaria, ISBN ISSN 1311-0829

[5] Modular Servo System – User's Manual, InTeCo Ltd