Robust PID Design for a Servosystem Using mu Synthesis in MATLAB

Asparuh G. Markovski

***Abstract* — Robust PID design via μ synthesis for a laboratory model of a servo system manufactured by the Polish company Inteco is done. The main goal is to demonstrate the new potentials of the Robust Control Toolbox for MATLAB for tuning fixed structure controllers for objects with an uncertainty, introduced in year 2021. Thus, the new version of musyn function can be used for both "classical" non-parametric and parametric robust control synthesis. Here the controller is constructed as a tunablePID object, and the system has both uncertainty and tunable parameters (genss object). The control objectives are given as in the usual mu synthesis approach – in the frequency domain using penalty filters. Some of the model parameters are modelled with a structured uncertainty. Additionally, unstructured uncertainty is introduced due to the presence of nonlinearity of gap type. For both cases different PID controllers are tuned. Simulation experiments are performed with the model. The operability of the created regulator is tested on the experimental installation.**

***Index Terms* — Robust Control, mu synthesis, PID**

I. INTRODUCTION

Classical PID control and robust control have both their advantages and drawbacks. It is not uncommon for the proponents of the both sides to fairly criticize the opposite approach: from the one side, PID controller has an unmatched simplicity in its structure, clear physical meaning of its parameters, and can be easily tuned in many cases even by hand, does not require complex hardware with high computational capabilities. But, PID control is not directly applicable for MIMO systems – it requires additional schemes, for example, for decoupling, which is in fact part of the controller. In contrast, the classical H_∞ and μ synthesis work for MIMO systems by default, do not need additional decouplers, state observers and so on, allow synthesis based on multiple criteria at the same time – control limitation, reference model, etc., and, the most important, guarantee robust stability and performance over object uncertainty. As a drawback, the classical H_∞ and μ synthesis require model with uncertainties to be carefully built and produce controllers of high orders (though reducible) in the state space, which can not be easily retuned in real time and cannot be adjusted by hand at all.

It is very tempting to combine this two approaches. For example, in [1] and [2] the approach of H_∞ and μ synthesis has been tried for tuning of a PID controller; in [3] additional tunable variable is introduced which the PID coefficients depend on; for additional robustness mu synthesis can be combined with pole placement constraints [4]. But this task is not easy - H_∞ controller is synthesized via solving algebraic Riccati equations for the continuous or discrete-time case and constructed in state-space, and its extension – μ synthesis is not a convex optimization problem when done by DK iterations or via LMI approach. So, the procedure of robust PID tuning always includes some non-convex optimization problem with limitations and is difficult to be automatized – in many works it includes, for example, steps like checking if some polynomial is Hurwitz at every iteration [5], which require a special software written for every specific object.

It will not be an exaggeration to say that the first really applicable and automated tool for μ synthesis of fixed structure controllers is the new version of **musyn** function from Robust Control Toolbox in MATLAB from year 2021. This function provides interface between many types of models – for example models with both uncertainty and tunable parameters (**genss** models) from one side, and algorithm for μ synthesis via DK iterations from the other side. So, it can be used for obtaining full-order centralized controllers in state-space (classical μ synthesis), reduced order controllers in state-space (using **tunableSS** objects), PIDs (using **tunablePID** objects).

II. PHYSICAL MODELS USED

To demonstrate the methods for tuning PID coefficients to meet robust design specifications via μ synthesis using **musyn** function, a physical model of a servo system from the laboratories of the Department of Systems and Control at the Technical University – Sofia was used.

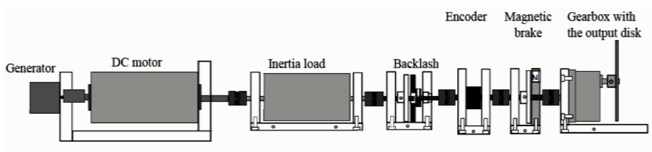

Fig. 1. Model of the servosystem

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 (gap nonlinearity), a magnetic brake, an incremental encoder, an output disk with a reducer [6]. The 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

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).

https://doi.org/10.47978/TUS.2022.72.02.001

the following dependences according to Fig. 2:

$$v(t) = Ri(t) + K_e\omega(t) \quad (1)$$

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

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.

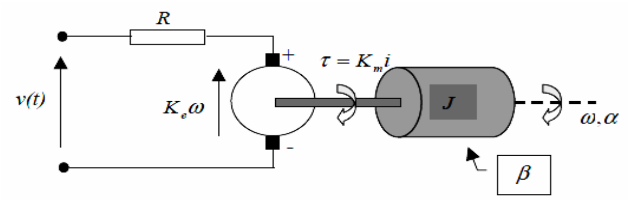

Fig. 2. Model of the dynamics of the servosystem

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

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

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:

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

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

From (4) a description in state space can be obtained:

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

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

Linearization at the zero point with the gap element included would lead to an uncontrollable model, so approximation with a small static gain of the gap is used. It is also possible to use the command **linmod** in MATLAB with simulated excitation.

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

a) Model with structured uncertainty, 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_m;

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

$$G_{nm} = (I + W_{mult}\,\Delta_{mult})\,G_m \quad (6)$$

The estimates of the maximum multiplicative uncertainty are as follows:

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

The process of the uncertainty modelling is shown in [7]. It should be noted that the amount of the unstructured uncertainty, described by W_{mult} is not too significant.

Two different cases are shown: position control (the shaft angle α is controlled) and velocity control (the shaft angular velocity ω is controlled). The synthesis scheme for position control is shown in Fig. 3. Here $W_p = \frac{0.01s+7}{s+0.14}$ is the performance filter, and $W_u = \frac{0.5s+5}{s+2.5}$ is the control penalty. Like in [3], other design filters can be used, such as reference model, etalon closed-loop system model, noise model.

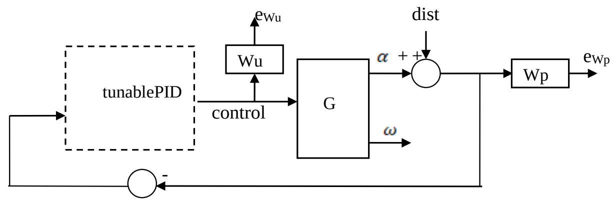

Fig. 3. Synthesis scheme for position control

III. Position Control PID Tuning via μ Synthesis for G_M and G_{NM}

The **tunablePID** object in MATLAB represents parallel structure PID with D filter in the form $K_p + \frac{K_I}{s} + \frac{K_D s}{T_f s+1}$. Anti-windup mechanism is also applied. So, the tunable parameters are K_p, K_I, K_D and T_f. For position control we actually need PD only control, but we will demonstrate that it will be found automatically. The connection between tunablePID, G (which can be G_m or G_{nm}), W_p and W_u is a generalized continuous-time state-space model with 2 outputs, 1 input and 5 states. The goal is to tune the tunablePID to minimize the influence of the load disturbance *dist* over the shaped output position angle e_{Wp} and the penalized control e_{Wu}. The additional minus sign in the feedback is added because the μ regulator assumes positive feedback.

First we make the synthesis for the G_m model (structured uncertainty). After 5 DK iterations the function **musyn** returns the following values for the tuned coefficients:

$$K_p = 0.051, K_I = 9.11e-6, K_D = 0.0556, T_f = 0.0325$$

For position control **musyn** automatically finds that K_I should be (nearly) 0. In this case we could also set PD only control in advance.

The maximal value of the μ norm achieved is 0.913, so robust performance according to W_p and W_u is achieved. The μ norm upper and low limits are shown in Fig. 4.

If we focus on robust stability only, it can be checked using **robuststab** command. The μ norm for the robust stability in this case is shown on Fig. 5. The system can tolerate up to 930% increasing the uncertainty, so the uncertainty stability margin is pretty high.

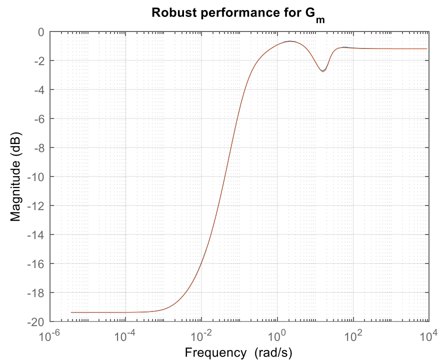

Fig. 4. Robust performance for *Gm*

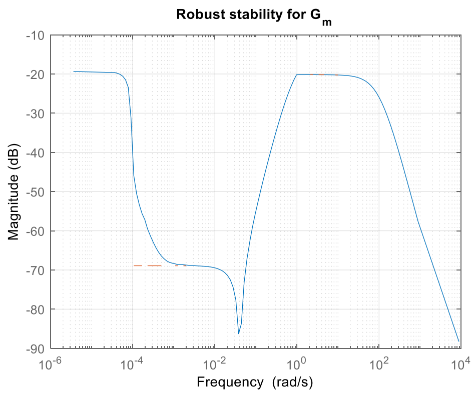

Fig. 5. Robust stability for *Gm*

The step responses for 20 arbitrary admissible realizations of the uncertain parameters in the linear model are simulated in Fig. 6: on the left is the response to the reference position, on the right - to the load disturbance.

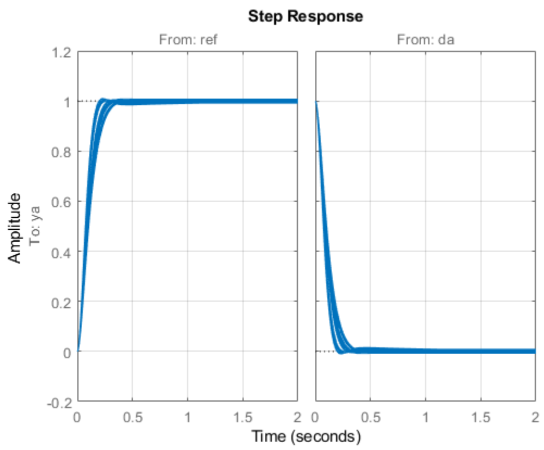

Fig. 6. Simulated position step responses for *Gm*

The work of the real laboratory set-up (with an additional load) is shown in Fig. 7.

The second tunablePID is tuned for the G_{nm} model (structured and unstructured uncertainty). After 6 DK iterations the function **musyn** returns the following values for the tuned coefficients:

$$K_p = 0.0524, K_I = 5e-6, K_D = 0.0579, T_f = 0.0288$$

The maximal value of the µ norm achieved is 1.088 – the robust performance is slightly unmet due to the higher uncertainty, as is shown in Fig. 8.

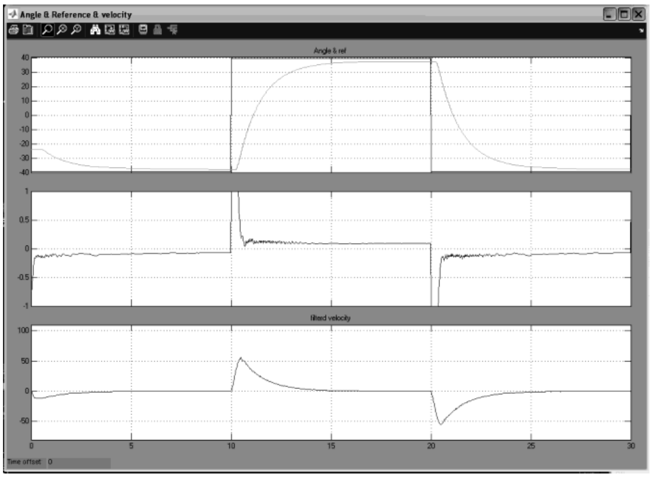

Fig. 7. Real position step responses for controller tuned for *Gm.* In the upper subfigure – reference and actual angle, in the middle – the control, in the down side – the filtered velocity

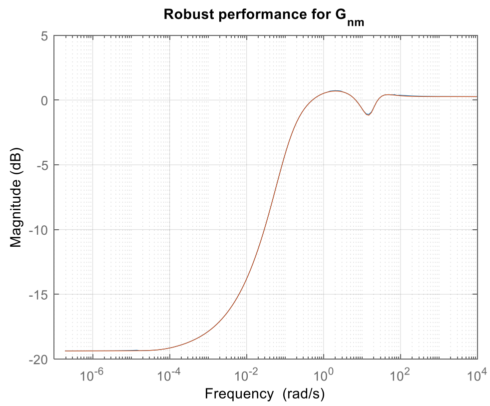

Fig. 8. Robust performance for *Gnm*

The robust stability only µ norm is low enough – the system can tolerate up to 349% of the uncertainty:

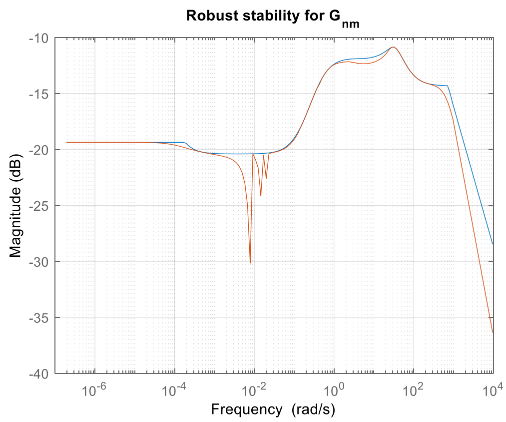

Fig. 9. Robust stability for *Gnm*

IV. Velocity Control PID Tuning via µ Synthesis for G_M and G_{NM}

The synthesis scheme for this case is shown in Fig. 10. Here $W_p = \frac{0.01s+10}{s+0.2}$ and W_u is the same.

First we make the velocity control synthesis for the G_m model. After 5 DK iterations the function **musyn** returns the following values for the tuned coefficients:

$$K_p = 0.376, K_I = 0.00294, K_D = -3.29, T_f = 0.103$$

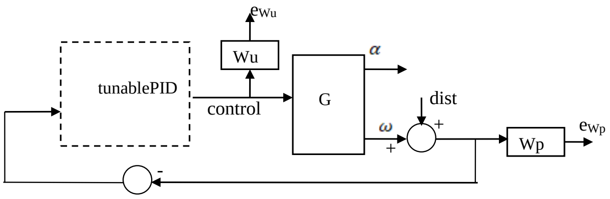

Fig. 10. Synthesis scheme for velocity control

According to musyn, the maximal value of the μ norm achieved is 0.781 as shown in Fig. 11.

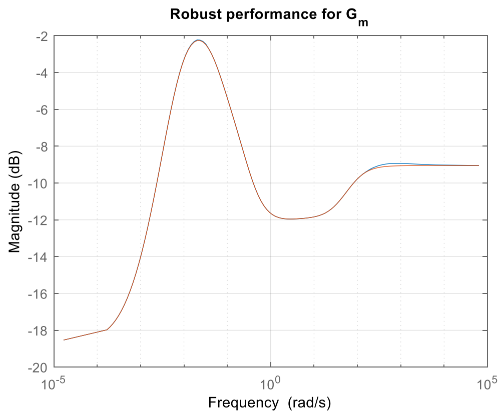

Fig. 11. Robust performance for *Gm*

As for the robust stability, μ norm is low enough – the system can tolerate up to 936% of the uncertainty.

The step responses for 20 arbitrary admissible realizations of the uncertain parameters in the linear model are simulated in Fig. 12: on the left is the response to the reference velocity, on the right - to the load disturbance.

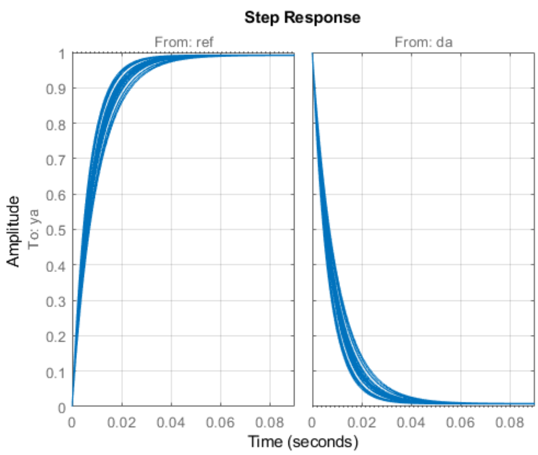

Fig. 12. Simulated velocity step responses for *Gm*

The experiment with the real object controlling velocity (with an additional load) are shown in Fig. 13.

The last tunablePID is tuned for the G_{nm} model for velocity control. After 6 DK iterations the function **musyn** returns the following values for the tuned coefficients:

$$K_p = 0.69, K_I = 0.0268, K_D = -0.688, T_f = 0.68$$

The maximal value of the μ norm achieved is 0.69 in the μ synthesis process; it is even less then in the case of structured only uncertainty, probably due to the peculiarities of the DK synthesis procedure. In the case of robust stability only, the system can tolerate up to 230% of the modelled uncertainty.

The work of the real laboratory set-up with this regulator (with an additional load) is shown in Fig. 13.

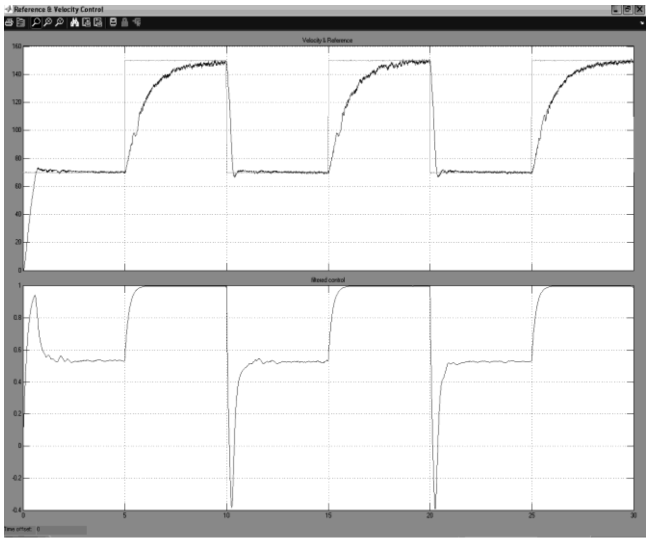

Fig. 13. Real velocity step responses for the controller tuned for *Gm*. In the upper subfigure – reference and actual angular velocity, in the down side – the control

V. CONCLUSION

The results show that the new **musyn** function in Robust Control Toolbox in MATLAB can be used for PID regulators tuning in the same manner as, for example, the classical Ziegler-Nichols, Åström – Hägglund, etc. methods. Thus, this allows the engineer to set design specifications in the frequency domain using penalty filters, which allows control objectives to be set in a precise way. If the main concern is the robust performance, this is the initial goal of the optimization procedure, and robust stability is automatically guaranteed. Of course, the μ synthesis is conservative by nature – the controller is tuned according to the worst possible uncertainty of the model. One can analyze the result and decide if this worst possible uncertainty is realistic and, possibly, use the robust controller as a starting point for further tuning the regulator.

REFERENCES

[1] Ming Tzu Ho, "Synthesis of H_∞ PID controllers: A parametric approach", Automatica vol. 39 no. 6, June 2003, 1069-1075, https://doi.org/10.1016/S0005-1098(03)00078-5

[2] Sariyildiz Emre, Haoyong Yu, Kouhei Ohnishi, "A Practical Tuning Method for the Robust PID Controller with Velocity Feed-Back", Machines vol. 3, https://doi.org/10.3390/machines3030208

[3] Verma Bharat, Prabin Kumar Padhy, "Robust Fine Tuning of Optimal PID Controller With Guaranteed Robustness", IEEE Transactions on Industrial Electronics, vol. 67, issue 6, 2019, https://doi.org/10.1109/TIE.2019.2924603

[4] Markovski A. G., "Robust control of a servosystem using mu synthesys and regional pole placement constraints", Publishing House of Technical University of Sofia, vol. 71, book 2, ISSN 2738-8549, 2021, https://doi.org/10.47978/TUS.2021.71.02.014

[5] Xian Hong Li, Hai Bin Yu, Ming Zhe Yuan, "Design of Optimal PID Controller with ε-Routh Stability for Different Processes", Mathematical Problems in Engineering, 2013

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

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