

# Interactive Software Tool for Design of Higher Derivative Degree PID Controllers<sup>\*</sup>

Pavol Bisták<sup>\*</sup> Mikuláš Huba<sup>\*</sup>

*<sup>\*</sup>Institute of Automotive Mechatronics  
Faculty of Electrical Engineering and Information Technology  
Slovak University of Technology in Bratislava, Slovakia  
(e-mail: pavol.bistak@stuba.sk, mikulas.huba@stuba.sk).*

---

**Abstract:** A new class recently developed controllers extends the family of PID regulators by several new features. The aim of this contribution is to present an interactive software tool for computer support of these controllers among researchers, students and broader research community by providing them with a graphical interface enabling to interact with the controllers and choose their tuning that best fulfils specified requirements. The main advantage of the new controllers, which cover the gap between traditional and fractional-order PID controllers, is that the more consistent solution to the noise filtering problem makes possible work with higher order derivatives and achieving broader spectrum of performance specifications. The paper briefly summarizes in a general way the design of PID controllers with higher derivative degree, noise attenuation and compensation of time delays. Based on this summary and further references the developed software tool is being described that takes the form of a virtual laboratory. The back-end of the tool relies on Matlab simulations, when it provides data of setpoint and disturbance step responses in the chosen loop, where the controlled system is represented by the first order time-delayed system distorted by a noise. The output data of Matlab simulations are processed with the front-end part of the software tool that uses possibilities of JavaScript programming language to interact with the control system and graphically visualize the results in different views within the Web browser environment.

Keywords: PID control, virtual laboratory, Matlab/Simulink, higher order derivatives, filtration, first order time-delayed plant

---

## 1. INTRODUCTION

It is well-known that the proportional-integral-derivative (PID) control plays a crucial role in the control engineering for a long time. Therefore one can find numerous methods showing how to calculate optimal PID controller parameters or how to tune them (O'Dwyer, 2009; Åström and Hägglund, 2006). For tuning of advanced PID controllers the first-order time-delayed (FOTD) system is often used. The software tool described in this paper is oriented to the tuning of PID controllers for the FOTD system with supporting use of higher order derivative terms.

By introduction of the higher order PID (HO-PID) controllers it is expected that the responses of a controlled system will be more dynamical, smoother and hopefully the settling time will decrease. But the question of realizability must be solved by an appropriate filtration. In practical applications filters have been always used for the noise attenuation even for controllers without the derivative part. This has been considered to be especially problematic, since it could amplify the noise. Therefore, similarly as in Huba (2015), the design procedure of the HO-PID controllers takes into account the filter presence from very beginning and the implementation filters of the chosen order are consider as an integral part of the design.

The HO-PID design procedure is based on the multiple real dominant pole (MRDP) method (Vítečková and Víteček, 2010; Viteckova and Vitecek, 2016) that is extended by the filter design. When a filter is included in the control circuit the dynamics of the circuit is slowed down. Therefore the MRDP method must be used again to recalculate the optimal tuning of the HO-PID control taking into account an extension of the original time-delay. The theoretical background of the HO-PID controllers for integrative plants with time-delays can be found in Huba (2018). In Huba and Vrančič (2018) this design has been generalized for an arbitrary FOTD system (even unstable). This can be advantageous in comparison with the classical Smith predictor which is not applicable for integrative and unstable plants. Unfortunately, the HO-PID control method could not benefit from the frequency domain approach because it based on time domain principles when introducing an equivalent time delay. From the user point of view the design of HO-PID controller is not very complicated as it is necessary to choose only 3 parameters: the derivative order of the HO-PID controller, the order of the filter, and the time constant of the filter or alternatively the equivalent time delay.

To disseminate the new HO-PID design method we decided to create the interactive software tool named HO-PID Designer that will enable users to interact with the design process: to choose parameters, see responses to a set-

---

<sup>\*</sup> Supported by APVV SK-IL-RD-18-0008 and VEGA 1/0745/19.

point and disturbance steps and also evaluate performance of the controller using noise characteristics. The HO-PID tool offers a virtual laboratory with all calculations carried out on the server side and the client side is responsible only for setting parameters and visualization of results. This client-server architecture is utilized by different virtual and remote laboratories (De La Torre et al., 2015; Sáenz et al., 2015; Kalúz et al., 2015; Jáno and Žáková, 2011) as it allows an easy conversion of the virtual laboratory to a remote laboratory.

Our previous software tools, virtual and remote laboratories have been exclusively based on the Matlab platform (i.e. both client and server needed the Matlab installation). The novelty of the solution presented in this paper is that the front-end part (client side) is realized in the Web page form so the user can run the HO-PID Designer tool from his/her Web browser. It uses the websocket technology to connect to the Matlab running on the server side. So, from the implementation point of view, the HO-PID Designer can be considered as an extension of the pure Matlab tool presented in Bisták and Huba (2019). It also covers the functionality of the Matlab tool for integral plants with time-delays (Bisták, 2018). The HO-PID Designer extends this previous tools and papers from the user point of view when it offers to evaluate the quality of HO-PID controllers for FOTD systems on the base of displayed noise characteristics.

The paper is organized as follows. Section 2 includes explanation of the HO-PID controller design for FOTD systems. Section 3 introduces performance measures applied to evaluate performance of the HO-PID controllers. Section 4 describes the architecture of the HO-PID Designer software tool and section 5 presents its Web user interface. Main paper results are summarized by Conclusions.

## 2. DESIGN OF HIGHER ORDER PID (HO-PID) CONTROLLERS

The ideal PID controller is usually expressed as

$$C(s) = K_c \frac{1 + T_i s + T_i T_D s^2}{T_i s} \quad (1)$$

Its extension (Huba and Vrančič, 2018) will be proposed for the FOTD plant model

$$S(s) = \frac{Y(s)}{U(s)} = \frac{K_{sm} e^{-T_{dm} s}}{s + a_m}$$

or for  $a_m \neq 0$ ,  $T_m = 1/a_m$ ,  $K_m = K_{sm}/a_m$  (2)

$$S(s) = \frac{K_m e^{-T_{dm} s}}{T_m s + 1}$$

Where appropriate, the model index  $m$  will be omitted in the paper. The ideal PID controller with higher order derivatives denoted as  $PID^m$  can be defined as

$$C^m(s) = K_c \frac{1 + T_i s + T_i T_{D1} s^2 \dots + T_i T_{Dm} s^{m+1}}{T_i s} \quad (3)$$

To be realizable the controller has to be supplemented by an additional filter.

Analyses of the setpoint and disturbance step responses of the  $PID_n^m$  controller according to Fig. 1 (Huba and Vrančič, 2018)) considers a FOTD system (2).  $PID_n^m$  is realized by the product of the ideal  $PID^m$  controller expressed by the transfer function  $C^m(s)$  (3) and the

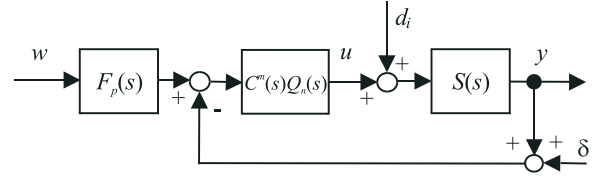


Fig. 1.  $PID_n^m$  controller composed of an ideal  $C^m(s)$  controller augmented by the filter  $Q_n(s)$  and extended by the prefilter  $F_p(s)$ .  $S(s)$  = FOTD plant.

transfer function of the filter  $Q_n(s)$ . The loop may be extended by the prefilter  $F_p(s)$ . The reference setpoint is denoted as  $w$ ,  $d_i$  represents the input disturbance and  $\delta$  measurement noise.

$m + 2$  unknown  $PID^m$  parameters may be calculated from the loop characteristic quasi-polynomial. Without filtration ( $n = 0$ ) it can be expressed as

$$P(s) = T_i s(s+a) e^{T_d s} + K_c K_s [1 + s T_i (1 + T_{D1} s + \dots + T_{Dm} s^m)] \quad (4)$$

In the multiple real dominant pole method,  $(m + 3)$ -tuple real dominant pole  $s_0$  has to fulfill requirements

$$\left[ P(s) = 0; \frac{dP(s)}{ds} = 0; \dots; \frac{d^{m+2}P(s)}{ds^{m+2}} = 0 \right]_{s=s_0} \quad (5)$$

Possible prefilter

$$F_p(s) = \frac{b_{m+1} T_i T_{Dm} s^{m+1} + \dots + b_2 T_i T_{D1} s^2 + b_1 T_i s + 1}{T_i T_{Dm} s^{m+1} + \dots + T_i T_{D1} s^2 + T_i s + 1} \quad (6)$$

has to avoid overshooting by canceling numerator of the setpoint-to-output transfer function. By cancelling  $p$  dominant closed loop poles  $s_0$  (coefficients  $b_i$ ,  $i = 1, \dots, p$ ;  $p \leq m + 1$ ) it may also speed up the transients.

In order to get a proper transfer function,  $PID^m$  controller has to be supplemented with an additional  $n$ th order binomial filter

$$Q_n(s) = 1/(T_f s + 1)^n; \quad n = 1, 2, \dots; \quad n \geq m \quad (7)$$

when the HO-PID denoted also as  $PID_n^m$  can be characterized as

$$PID_n^m : C_n^m(s) = C^m(s) Q_n(s) \quad (8)$$

For determining its filtration time constant  $T_f$  the "half rule" principle has been used in Skogestad (2003). Here the term of an equivalent time delay  $T_e$  is used in approximating the delay of  $Q_n(s)$ . By comparing multiple real dominant poles of characteristic polynomials corresponding to two basic types of delays - the first one consisting of  $PID^m$  control combined with the noise filter  $Q_n(s)$  and a dead-time free first order plant  $S(s)$  (with  $T_d = 0$ ); the second one combining  $PID^m$  control with an FOTD plant  $S(s)$  having a dead time  $T_d = T_e$  - analytical formulas for determining  $T_f$  may be derived.

Thus, by specifying the derivative degree  $m$ , an equivalent time delay value  $T_e$  and the low-pass filter order  $n \geq m$ , all  $PID_n^m$  parameters may be determined. Then, by choosing number of the cancelled dominant poles  $p$ , the prefilter  $F_p(s)$  can be specified. To show the procedure of the controller parameters derivation the next sub-chapter briefly reviews the simplest situation with  $m = 0$  (filtered PI-controller). Derivations of all  $PID_n^m$  parameters up to  $m = 2$  may be found in Huba and Vrančič (2018).

## 2.1 FPI - Filtered PI controller design

The simplest choice  $m = 0$  yields a PI control

$$C(s) = K_c \frac{1 + T_i s}{T_i s} = K_c + \frac{K_i}{s}; F_p(s) = \frac{bT_i s + 1}{T_i s + 1} \quad (9)$$

characterized by the transfer functions

$$F_s(s) = \frac{Y(s)}{W(s)} = \frac{K_s K_c (bT_i s + 1)}{T_i s (s + a) e^{T_i a s} + K_s K_c (T_i s + 1)} \quad (10)$$

$$F_d(s) = \frac{Y(s)}{D_i(s)} = \frac{K_s T_i s}{T_i s e^{T_i a s} (s + a) + K_c K_s (T_i s + 1)}$$

and quasi-polynomial

$$P(s) = T_i s e^{T_i a s} (s + a) + K_c K_s (T_i s + 1) \quad (11)$$

Its triple real dominant pole (TRDP)  $s_o$   $P(s)$  (11) (Vítečková and Víteček, 2010) satisfying  $P(s_o) = 0$ ,  $\dot{P}(s_o) = 0$  and  $\ddot{P}(s_o) = 0$  yields the "optimal" set of parameters  $K_{co}$ ,  $T_{io}$ , and with  $p = 1$  the prefilter tuning  $b_1$  canceling one of the dominant poles  $s_o$ .

$$s_o = -\frac{A + 4 - S}{2T_d}, \quad A = aT_d, \quad S = \sqrt{A^2 + 8}$$

$$K_o = K_{co} K_s T_d = (S - 2)e^{(S - A - 4)/2}$$

$$\tau_{io} = \frac{T_{io}}{T_d} = \frac{2(2 - S)}{A^2 + 2A + 28 - (A + 10)S} \quad (12)$$

$$b_1 = \frac{1}{\tau_{io} T_d s_o} = \frac{A^2 + 2A + 28 - (A + 10)S}{(S - 2)(S - A - 4)}$$

For the filter (7) in the feedforward loop path and  $T_d = 0$  the open-loop is described as

$$F_o(s) = K_c \frac{1 + T_i s}{T_i s} \frac{K_s}{(s + a)(1 + T_f s)^n} \quad (13)$$

The closed loop characteristic polynomial

$$P(s) = T_i s (s + a)(1 + T_f s)^n + K_c K_s (T_i s + 1) \quad (14)$$

yields the triple real dominant pole (TRDP)  $s_n$

$$s_n = -\frac{n(A_f - S_n) + 4}{2(n + 2)T_n}; \quad A_f = aT_f; \quad S_n = \sqrt{A_f^2 + 8 \frac{1 - A_f}{n(n + 1)}} \quad (15)$$

A simple filtering properties evaluation is possible by keeping the closed loop dynamics nearly constant, when the dominant closed loop poles in (12) and (15) remain in a constant position, i.e.  $s_o = s_n$ , when

$$-\frac{A - S + 4}{2T_d} = -\frac{n(A_n - S_n) + 4}{2(n + 2)T_f}$$

$$A = aT_d, \quad S = \sqrt{A^2 + 8} \quad (16)$$

$$A_f = aT_f, \quad S_n = \sqrt{A_f^2 + 8 \frac{1 - A_f}{n(n + 1)}}$$

In other words, the *equivalent dead time*  $T_e$  characterizes impact of  $Q_n(s)$  on the noise filtration and on the dynamics of the closed loop responses. From (16) with  $T_d = T_e$  follows

$$T_f = T_e \frac{A + (n + 1)(4 - S) - \sqrt{n[nA^2 + 4(n + 1)(3 - S)]}}{(1 + n)[A^2 + 4A + 6(n + 2) - (A + 2n + 4)S]} \quad (17)$$

$T_e$  has the same impact on the dominant poles determining the closed loop performance as  $Q_n(s)$  with the time constant  $T_f$ . However, with respect to dead time additivity, the controller (12) may be tuned according to

$$T_d = T_{dm} + T_e \quad (18)$$

when  $T_e$  and  $n$  are the tuning parameters used for modifying the noise attenuation in (12).

The below described software tool is implemented for the  $PID_n^m$  controllers with the derivative degree up to the  $m = 2$ . Derivation of their parameters follows similar steps as shown above.

## 3. HO-PID CONTROLLERS PERFORMANCE MEASURES

All designed HO-PID ( $PID_n^m$ ) controllers will be evaluated by means of performance measures focused both on speed of the transients and their shapes. The speed of transients is quantified by integral of absolute error (IAE)

$$IAE = \int_0^\infty |e(t)| dt; \quad e = w - y \quad (19)$$

where  $e$  is the control error. The index  $s$  will denote the setpoint analysis ( $IAE_s$ ) and the index  $d$  will be used for the disturbance analysis ( $IAE_d$ ) based on the unit steps. An ideal setpoint step response of the FOTD plant has a one-pulse shape. Deviations of output  $y$  from such an ideal shape may be quantified by a relative total variance

$$TV_1(y_d) = \sum_i |y_{i+1} - y_i| - |2y_m - y_\infty - y_0| \quad (20)$$

A similar formula can be applied for the input  $u$  and denoted as  $TV_1(u_d)$  where again the indices  $s$  and  $d$  will denote the setpoint and the disturbance.

In order to demonstrate the impact of chosen parameters ( $m$  and  $T_e$ ), the noise characteristics are introduced in the  $IAE - TV_1(u)$ , or  $IAE - TV_1(y)$  planes. They relate the dependence of the speed of transients (expressed in terms of  $IAE$  values) to the excessive signal changes at the plant input and output (expressed in terms of the relative total variance). The relation  $IAE$  vs.  $TV_1(u)$  is called Speed to Effort (SE) characteristic and  $IAE$  vs.  $TV_1(y)$  has been named Speed to Wobbling (SW) characteristic. In Matlab/Simulink, the measurement noise impact has been emulated by using the "Uniform Random Number" block with the sampling period  $T_s = 0.001T_{dp}$  and chosen amplitudes  $|\delta|$  according to the control circuit block diagram (Fig. 1).

The noise characteristics and the graphical user interface description will be discussed in the 5<sup>th</sup> chapter.

## 4. ARCHITECTURE OF HO-PID DESIGNER

As one can see from the chapters above the theory of the HO-PID controllers is a complex one therefore we have decided to create a software tool that will enable students and researchers to test this new class of PID controllers in a user friendly way. We have designed the Web graphical user interface that is based on Hypertext Markup Language (HTML), Cascading Style Sheets (CSS) and JavaScript (JS) programming language that are nowadays standard tools for building front-end applications in the Web environment. For the communication with the server side the WebSocket technology has been used. Websockets are network software elements for data transmission on the highest communication level in the ISO/OSI communication model. From the user point of view they allow a two-way continuous communication similarly to the classical

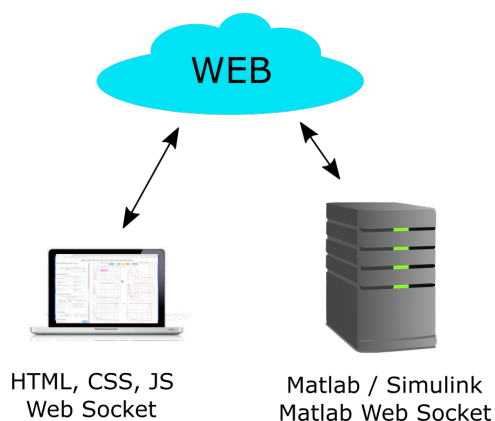


Fig. 2. Web client-server architecture of the HO-PID Designer software tool

sockets. The back-end of the HO-PID Designer has been created by the server where Matlab/Simulink application was running. All calculations of HO-PID coefficients, pre-filter coefficients and filter time constants as well as simulations of the control system have been running at the server side based on through the Websockets received parameters from the Web client side. The results of simulations have been sequentially transmitted to be progressively displayed on user Web interface. According to the selected options corresponding graphs have been plotted on the HO-PID Designer Web page. The corresponding client-server architecture is depicted in Fig. 2.

The WebSocket technology is implemented in the JavaScript language but Matlab does not directly support Websockets. Therefore we have used the MatlabWebSocket library (Jebej, 2016) that consists of the WebSocket server and the Matlab client build on Java-WebSocket technology. We have implemented *onTextMessage* method from the *WebSocketServer* class that reacts to the messages sent by a client and can send the simulation data back to the client via the *send* method. The HO-PID Designer tool shares the architecture with the remote laboratory described in (Bisták, 2019) where readers can find a more detailed explanation.

## 5. HO-PID WEB USER INTERFACE

The designed Web user interface of the HO-PID tool is displayed in Fig. 3. It consists of two main parts: the parameters setting column placed to the left and the visualization panel given by the set of graphs placed in the rest of the window. At the top of the parameters column there is depicted the block diagram of the control circuit. Below the controlled system parameters ( $K_s, a, T_d$ ) can be set. Notice the the  $a$  can be also negative corresponding to the unstable system. On the right side of the Controlled System section a user can set input signals like desired value, input disturbance and noise magnitude. Last two parameters in this section define the length of simulation time and the sampling period. Below the Controlled System section there is the Controllers and Filters section that serves for setting of the parameters of the HO-PID controllers, especially the derivative degree of the  $PID_n^m$  controller, then the order of the filter and the equivalent time-delay. There is also a possibility to enter the set of

equivalent time-delays. This set is then used when calculating and drawing the noise characteristics (SW, SE).

At the bottom part there is the Visualisation section that enables to choose which graphs to be displayed. Setpoint step responses are drawn on the upper left part of the graph panel. Disturbance step responses are shown on the upper right part. Each time the time response of the output signal is displayed together with the time response of the control variable. The left bottom part is reserved for the noise characteristics corresponding to the setpoint responses and the right bottom part presents the noise characteristics related to the disturbance time responses. Both types of the noise characteristics use the set of equivalent time-delays entered in the Controllers and Filters panel. When selecting the Speed to Wobbling characteristic, it is displayed together with the Speed to Effort characteristic.

The last section in the bottom of the left column is about data retrieval from the server. If there is a big amount of data, it is recommended to use the Depreciation parameter that specifies how many times the received set of simulation data will be reduced. The original (not reduced) set of simulation data can be acquired by the Download button any time when the simulation has finished.

At the top of the graph panel there is the control panel that includes buttons for connecting to the server, starting simulation, pausing the simulation, stopping it and disconnecting from the server. It is possible to hide this panel not to block the view of the upper graphs. After the simulation starts the graphs are plotted progressively as data are being transferred. The JavaScript Plotly library has been used to draw interactive graphs. The interactivity has been extended in the case of noise characteristics when by pointing the mouse to a certain point of the characteristic not only the coordinates of the point are displayed but the values of the controllers, prefilter coefficients and filter time constants are shown as well and can be copied by a user for an external processing.

Each graph can be displayed in the new window by using the button in the right down corner of the graph. In such a way a user can see the details of the graphs. For instance, Fig. 4 shows the detail of the Speed to Wobbling characteristic for the setpoint when all derivative degrees were selected ( $m = 0, 1, 2$ ) and additionally the option  $m = 0, n = 0, T_e = 0$  was also checked that corresponds to the non-filtered  $PI$  controller. This  $PI$  controller is represented on the graph by just one point (the cross) as there is no set of  $T_e$  values in this case ( $T_e = 0$ ). It can be seen that this classical  $PI$  controller without filters is slower than the  $PID_4^2$  controller and its wobbling is several times higher. From this particular plot one can notice how filtration decreases the wobbling of the output signal. The noise characteristics give a user chance to choose such a controller that will be fast enough and still not too much noisy as a compromise solution. By inspecting the noise characteristics one can discover the significance of the HO-PID control design method that relates the fast behavior with effort and wobbling. It shows how the speed of responses depends on the limits of these two measures.

Typical scenario for using the HO-PID Designer tool by students can be as follows. A student opens a Web page

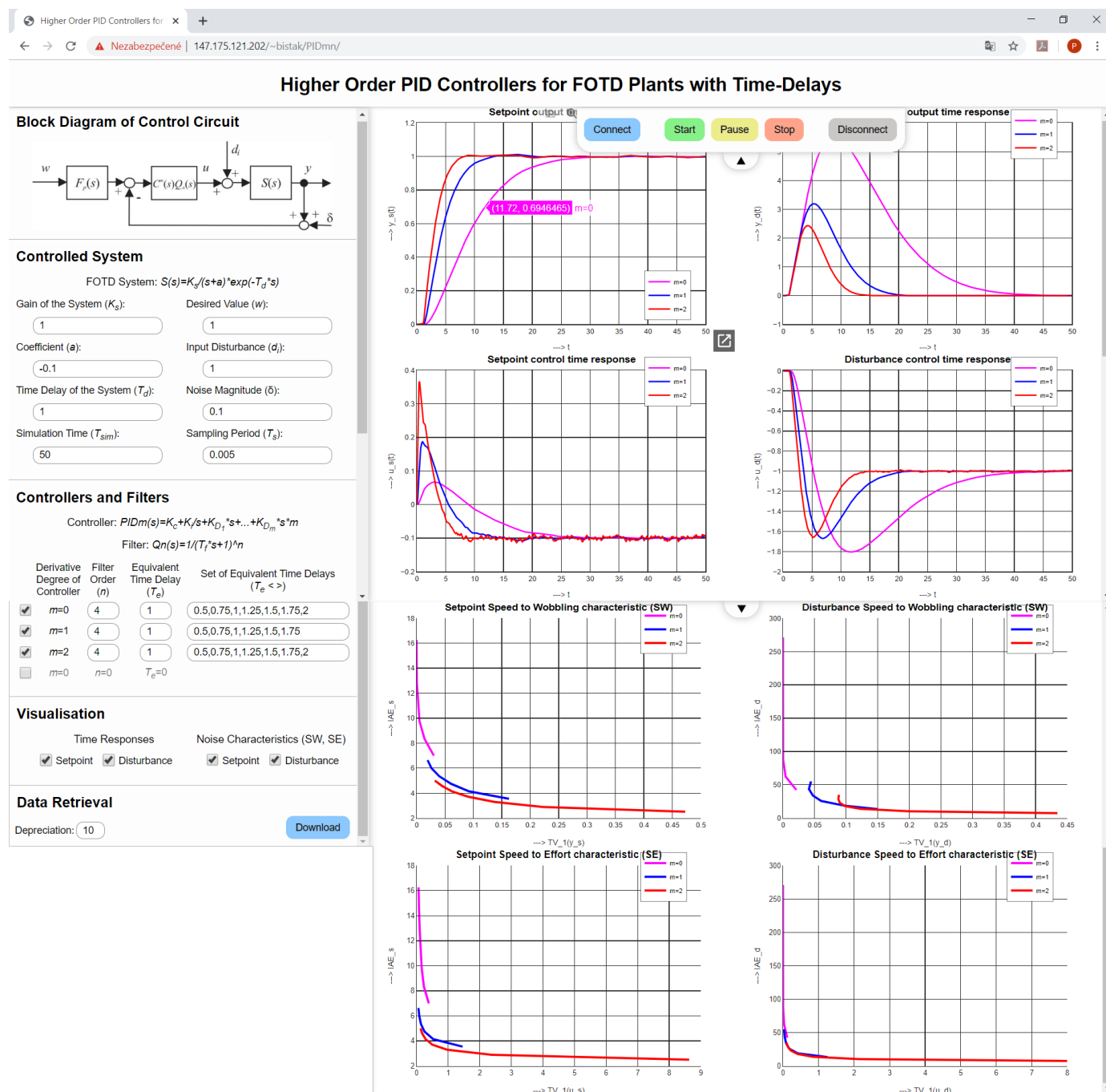


Fig. 3. HO-PID Web user interface

with HO-PID Designer, sets controlled system parameters (even unstable systems could be chosen), input signals (setpoint and disturbance) and selects controller and filter parameters. For the value given in the Equivalent Time Delay column the time responses will be plotted. The Set of Equivalent Time Delays values determine the points of the corresponding noise characteristics displayed below the time responses. In the Visualisation section the student can select graphs that will be plotted. From the graph panel the student can activate the detailed view of the chosen noise characteristic and evaluate each point of it by pointing the mouse to it. Then the parameters of the corresponding controller will be displayed and the student can copy them for further utilization in simulations or for

the application to a real control circuit. If the student is not satisfied with the results she/he can select a different point of the noise characteristic (taking into account the relation speed vs. effort/wobbling) or change the controller and filter parameters and plot a new set of responses and noise characteristics. Thus the student can realize the necessity of a compromise choice.

## 6. CONCLUSION

The presented HO-PID Designer software tool has been created to help with dissemination of the recently developed class of PID controllers with higher derivative degree. After the brief revision of the theoretical principles of the



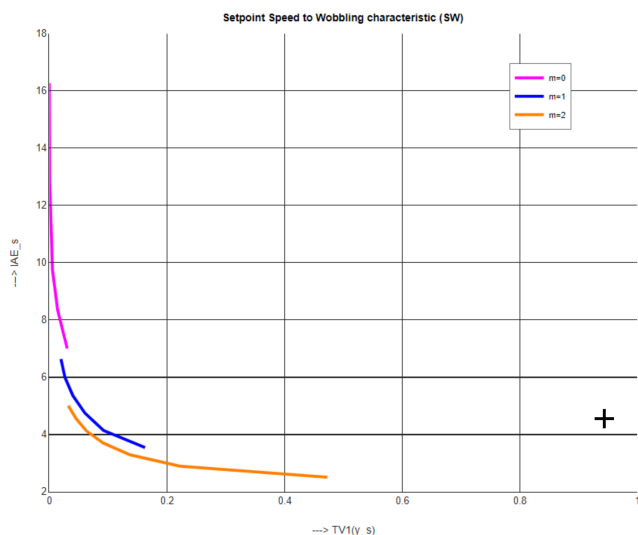


Fig. 4. Detailed view of the Setpoint Speed to Wobbling graph with additional non-filtered PI controller represented by the cross

controller design readers should understand all parameters that can be chosen through the Web interface of HO-PID Designer.

Using this tool students and researchers achieve a simplified way to tuning numerous HO-PID controller parameters according to their preferences. Of course, this was not able to explain all possible situations provided by Matlab/Simulink environment but for starting the process of getting familiar with the HO-PID controllers design it could yield a satisfactory support. The developed tool supports simultaneously several types of new controllers, therefore it was possible to demonstrate just a limited set of parameters and features that could be interesting in the learning process.

As concerning technological point of view the tool is based on Matlab, Websockets and Web technologies. In comparison with the previous solutions when the front-end has been created also within the Matlab/Simulink the HO-PID Designer uses basic Web technologies (HTML, JavaScript and CSS) to create an interactive user interface in the form of a Web page. For an end user this has an advantage that no Matlab/Simulink installation is necessary at the client side. In order not to violate Matlab/Simulink license rules we provide a limited access to the HO-PID Designer Web page through our university Moodle e-learning system as our university administrates the Matlab/Simulink campus license.

For the further development of the tool we plan to add one more type of the HO-PID controller with the derivative degree of four where the analytical solutions have already been derived. Similarly, the HO-PID control method for second order systems with time-delays is prepared for implementation. In principle the HO-PID design method can be applied also to higher order systems but the complexity of analytical solutions is growing rapidly. As concerning robustness the HO-PID control has been worked out also for this issue and will be included in the next versions of the HO-PID Designer tool.

## REFERENCES

- Karl Johan Åström and Tore Hägglund. *Advanced PID Control*. ISA - The Instrumentation, Systems and Automation Society, 2006. ISBN 978-1-55617-942-6.
- P. Bisták. Disturbance Analysis Virtual Laboratory for PID Controllers with Higher Derivative Degrees. In *2018 16th International Conference on Emerging eLearning Technologies and Applications (ICETA)*, pages 69–74, Nov 2018.
- P. Bisták. 3D three-tank temote laboratory based on Matlab and Websockets. In *2019 5th Experiment International Conference (exp.at'19)*, pages 85–89, June 2019.
- P. Bisták and M. Huba. Analysis of Higher Derivative Degree PID Controllers via Virtual Laboratory. In *2019 27th Mediterranean Conference on Control and Automation (MED)*, pages 256–261, July 2019.
- L. De La Torre, M. Guinaldo, R. Heradio, and S. Dormido. The ball and beam system: A case study of virtual and remote lab enhancement with Moodle. *IEEE Transactions on Industrial Informatics*, 11(4):934–945, 2015. doi: 10.1109/TII.2015.2443721.
- M. Huba. Filter choice for an effective measurement noise attenuation in PI and PID controllers. In *IEEE/IES International Conference on Mechatronics, ICM2015*, Nagoya, Japan, 2015.
- M. Huba. Extending spectrum of filtered controllers for IPDT plant models. In *Proceedings of the 29th International Conference on Cybernetics and Informatics, K and I 2018*, Lazy pod Makytou, Slovakia, Jan 2018.
- M. Huba and D. Vrančič. Comparing filtered PI, PID and PIDD2 control for the FOTD plants. *IFAC-PapersOnLine*, 51(4):954 – 959, 2018. ISSN 2405-8963. 3rd IFAC Conference on Advances in Proportional-Integral-Derivative Control PID 2018.
- Jebej. MatlabWebSocket. <https://github.com/jebej/MatlabWebSocket>, 2016. Accessed: 2019-02-11.
- M. Jáno and K. Žáková. SciLaB based remote control of thermo-optical plant. *International Journal of Online Engineering*, 7(4):10–15, 2011. doi: 10.3991/ijoe.v7i4.1783.
- M. Kalúz, J. García-Zubía, M. Fikar, and L. Čirka. A flexible and configurable architecture for automatic control remote laboratories. *IEEE Transactions on Learning Technologies*, 8(3):299–310, 2015. doi: 10.1109/TLT.2015.2389251.
- O'Dwyer. *Handbook of PI & Pid Controller Tuning Rules*. World Scientific Publishing, 2009.
- J. Sáenz, J. Chacón, L. De La Torre, A. Visioli, and S. Dormido. Open and low-cost virtual and remote labs on control engineering. *IEEE Access*, 3:805–814, 2015. doi: 10.1109/ACCESS.2015.2442613.
- S. Skogestad. Simple analytic rules for model reduction and PID controller tuning. *Journal of Process Control*, 13(4):291 – 309, 2003.
- M. Vítečková and A. Víteček. 2DOF PI and PID controllers tuning. In *9th IFAC Workshop on Time Delay Systems*, volume 9, pages 343–348, Praha, 2010.
- M. Viteckova and A. Vitecek. 2DOF PID controller tuning for integrating plants. In *2016 17th International Carpathian Control Conference (ICCC)*, pages 793–797, May 2016.