Automatic Tuning of the Pulse-Step Model Predictive Controller

Jaroslav Sobota * **Miloš Schlegel** **

* *European Centre of Excellence*
NTIS - New Technologies for Information Society
Faculty of Applied Sciences, University of West Bohemia in Pilsen
Univerzitní 8, 301 00 Plzeň, Czech Republic,
(e-mail: jsobota@kky.zcu.cz)
** *(e-mail: schlegel@kky.zcu.cz)*

Abstract: This paper describes an autotuning procedure for single-loop model based predictive controller with manipulated value constraints focused on applications in process control. It is assumed that the controlled process is stable, linear and t-invariant FIR system described by three-parameter model derived from a simple identification experiment. Controller parameters are computed from the identified model. A single parameter intended for fine-tuning of the controller is available. Exact intervals are given to vary this parameter within.

Keywords: self-tuning regulators, predictive control, step function responses, constraints, quadratic performance indices.

1. INTRODUCTION

Automatic tuning is nowadays understood as a necessary part of any control algorithm which attempts to succeed in industrial practice. When user triggers a demand for (re)tuning by pressing a button, the tuning experiment is performed and the controller parameters are suggested. This automatic tuning can be referred to as tuning on demand or one-shot tuning (Åström et al., 1993).

This paper follows the authors' work focused on creation of a single-loop predictive controller with low memory and computation power requirements in order to allow its implementation in microcontrollers, compact controllers and PLCs. Only when these prerequisities are fulfilled it is possible to use the advantages of MPC also in the lower (process) levels of control systems, where traditional single-loop PID controllers dominate. In this paper, the most emphasis is put on automatic tuning of the single-loop pulse-step model predictive controller (Schlegel and Sobota, 2008).

The approach described in this paper thus differs from the most common use of MPC, where the optimization is performed in the upper levels of hierarchical control systems and the technology is optimized by feeding appropriate signals to setpoints of single-loop controllers in the process level (Maciejowski, 2002). The controller presented herein is quite unique because the vast majority of papers focused on automatic tuning of MPC controllers also follows the upper-level use of MPC (Al-Ghazzawi et al., 2001).

The presented control algorithm with autotuning feature is applicable to stable processes with monotonous step response, which are typical for process control. The most common constraints in simple control loops are the saturation limits on the control signal. These are considered and successfully dealt with when the pulse-step model predictive control algorithm is applied.

⋆ This paper was supported by the European Regional Development Fund (ERDF), project "NTIS - New Technologies for Information Society", European Centre of Excellence, CZ.1.05/1.1.00/02.0090.

2. SIMPLE PREDICTIVE CONTROL WITH CONSTRAINTS

Possibly the greatest advantage of MPC is its ability to include constraints directly into the design procedure. On the other hand, this ability implies the greatest disadvantage, which is the computation burden connected with constrained MPC.

2.1 The pulse-step control sequence

The application of pulse-step control sequence in model predictive control is an alternative to the well-known approaches to complexity and computational cost reduction in MPC algorithms. The classic complexity-reducing methods rely on the use of some blocking strategy (Tondel and Johansen, 2002), for example constant manipulated value or constant manipulated value differences over time intervals of specified length. Another possibility is the so-called functional predictive control (Richalet et al., 1987), where the control sequence is restricted to a linear combination of suitable base functions. Significant attention is also drawn by off-line explicit solution methods, where the most demanding on-line optimization is avoided (Tondel et al., 2003; Bemporad and Filippi, 2002; Kvasnica et al., 2004).

In general, the pulse-step control is a well known aggressive technique used for manual control in industrial practice. It is well-suited for the case when saturation constraints on the control signal must be dealt with. It results in

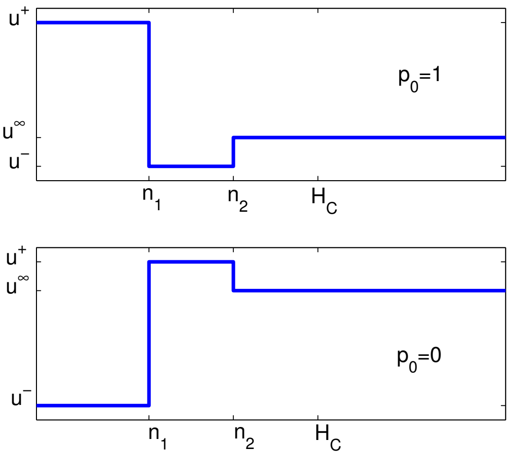

Fig. 1. Example of "pulse-step up" ($p_0 = 1$) and "pulse-step down" ($p_0 = 0$) control sequence

fast transitions when setpoint is changed, the manipulated value constraints are kept but fully exploited. The properties of pulse-step feedforward control in combination with the classical PID feedback control were first studied by Wallén and Åström (2002). The idea was incorporated into the MPC context by Schlegel and Sobota (2008), who demonstrated the properties of pulse-step model predictive controller (PSMPC) in detail.

As shown in Fig. 1, the pulse-step control sequence $u(k)$ begins with n_1 maximal (minimal) elements, followed by $n_2 - n_1$ minimal (maximal) elements according to the constraints $u^- \leq u(k) \leq u^+$. The remaining part of the control sequence is constant, $u(k) = u^\infty$ for $k \geq n_2$. The control horizon H_C determines the limit for n_1 and n_2, $0 \leq n_1 \leq n_2 \leq H_C - 1$ and of course u^∞ is subject to constraints $u^- \leq u^\infty \leq u^+$. So the whole control sequence is determined by only 4 variables p_0, n_1, n_2, and u^∞, where the parameter p_0 distinguishes between "pulse-step up" and "pulse-step down" control sequences.

2.2 Computing the control sequence

The controller output is restricted to control sequences in the pulse-step shape, the saturation constraints on the manipulated variable are present. The computation of the control signal to apply is based on the quadratic optimization criterion

$$I = \sum_{i=N_1}^{N_2} \left(\hat{y}(k+i|k) + d - w\right)^2 + \\ + \lambda \sum_{i=0}^{H_C-1} \Delta\hat{u}(k+i|k)^2 \to \min \qquad (1)$$

where $\hat{y}$ is the prediction of system output, d is the difference of the most current measurement and its predicted value and w is the desired output value (setpoint). Thus the optimized variables are p_0, n_1, n_2, $0 \leq n_1 \leq n_2 \leq H_C$, and $u^\infty \in \langle u^-; u^+\rangle$, which precisely define the pulse-step control sequence. The changes of the control signal $\Delta\hat{u}$ are summed and penalized in the criterion. This penalization

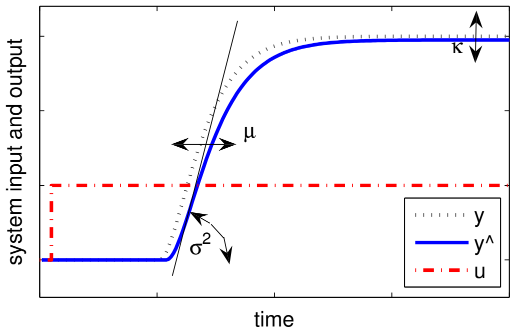

Fig. 2. Physical meaning of the characteristic numbers κ, μ, and σ^2

is accented or attenuated by the λ weighting coefficient. The constants N_1 and N_2 define the coincidence interval on which the output should follow the setpoint as precisely as possible.

The disturbance d is presumed to be constant over the whole prediction horizon. This presumption incorporates integrator into the structure of the controller, which ensures total compensation of arbitrary constant disturbance acting on the system. Unlike in the PID controllers, the manipulated value constraints are included in the design procedure and thus the integrator wind-up effect is avoided directly.

The prediction $\hat{y}$ employs a discrete step-response-model of the controlled system, which will be discussed later.

The algorithm used for solving the optimization task (1) combines brute force and the least squares method. The value u^∞ is determined using the least squares method for all admissible combinations of p_0, n_1, and n_2 and the optimal control sequence is selected afterwards. The computational cost is proportional to ${H_C}^2$. The selected sequence in the pulse-step shape is optimal in the open-loop sense. To convert from open-loop to closed-loop control strategy, only the first element of the computed control sequence is applied and the whole optimization procedure is repeated in the next sampling instant.

It is important to mention that the parameters N_1, N_2, H_C, and λ in the criterion (1) take the role of design parameters and significantly influence the closed loop characteristics. These parameters are the ones that should be tuned to obtain the required closed-loop performance.

2.3 The controlled process model

Apart from the controller parameters, the controlled system model is an essential part of each predictive controller. In the approach presented in this paper, the discrete step response is used. For stable, linear and t-invariant FIR systems with monotonous step response it is possible to use the moment model set approach (Schlegel and Večerek, 2005) and describe the system by only 3 characteristic numbers κ, μ, and σ^2.

As shown in Fig. 2, the characteristic numbers have a clear physical meaning, so it is possible to adjust them manually to fit the step response of the real system if necessary. The characteristic number κ is static gain, the number μ is known as resident time constant (it shifts the maximum of the impulse response along the time axis), and the parameter σ^2 changes the slope of the step response.

The characteristic numbers κ, μ, and σ^2 of the system in the form

$$P(s) = \frac{K}{\prod_{i=1}^{l}(\tau_i s + 1)} \cdot e^{-Ds} \qquad (2)$$

are defined as

$$\kappa = K, \qquad \mu = D + \sum_{i=1}^{l} \tau_i, \qquad \sigma^2 = \sum_{i=1}^{l} \tau_i^2. \qquad (3)$$

For details see (Schlegel and Večerek, 2005). Note that the transfer function in the form (2) can describe the most common systems in process control (temperature, flow, concentration, etc.).

In order to obtain the discrete step response of the controlled system, it is possible to approximate the system by first-order plus dead-time system

$$P_{FOPDT}(s) = \frac{K}{\tau s + 1} \cdot e^{-Ds}, \qquad (4)$$
$$\kappa = K,\ \mu = \tau + D,\ \sigma^2 = \tau^2$$

or second-order plus dead-time system

$$P_{SOPDT}(s) = \frac{K}{(\tau s + 1)^2} \cdot e^{-Ds}, \qquad (5)$$
$$\kappa = K,\ \mu = 2\tau + D,\ \sigma^2 = 2\tau^2$$

with the same characteristic numbers. The discrete step response to be used in the MPC controller is then generated from one of these transfer functions. But the modeling of the controlled system is by no means limited to the FOPDT and SOPDT systems, an arbitrary step response can be used if it is available.

The characteristic numbers can be further normalized in gain and time and then the system is described by characteristic numbers

$$\bar{\kappa} = 1,\ \bar{\mu} = 1,\ \bar{\sigma}^2 = \frac{\sigma^2}{\mu^2}, \qquad (6)$$

which in fact means that the system dynamics is described by only one number $\bar{\sigma}^2$. The sampling period must be normalized as well, $\bar{T}_S = \frac{T_S}{\mu}$.

Denormalization is very easy, the time constants, time delay and sampling period must be multiplied by μ. Therefore only normalized systems with $\kappa = 1$, $\mu = 1$ can be considered without the loss of generality.

The dynamics of typical systems in process control can be described by $\bar{\sigma}^2 \in \langle 0.2; 0.95 \rangle$.

3. CONTROLLER PARAMETERS

As was already mentioned, the controller parameters are N_1, N_2, H_C, and λ.

The control horizon H_C should be naturally as long as possible with respect to the computational power available. Therefore it is not a tuning parameter in the common sense.

The controller parameters N_1, N_2 and λ are to some extent redundant. Increasing the N_1 parameter makes the controller more conservative because the coincidence points are more distant and this results in smoother and slower control action. From the controller perspective, it is not necessary to generate any rapid actions, there is enough time to reach the coincidence points. On the other hand, decreasing the N_2 parameter excludes the distant coincidence points from the criterion, putting more emphasis on the immediate future (again from the controller perspective), resulting in a swift and aggressive control action. But the conservativeness can also be influenced by the λ parameter, which penalizes the changes in the control signal. The problem is that unlike in the N_1 and N_2 parameters, there are no limits nor guidelines on how to adjust the λ coefficient, except that it is positive and that the greater the λ is, the more conservative control we get.

The agresivity of the pulse-step controller can be described by its tendency to use the limit values. The limits should be exploited to reach the setpoint or suppress disturbances faster, but they must be used sensibly. Too aggressive controller might run into an endless limit-limit cycle as a result of noise or discrepancy between the model and the real controlled system. On the other hand, too conservative pulse-step controller would always work in its linear mode, leaving its most powerful ability underused.

In the latter, the N_1 parameter will always be set to the first non-zero element of the discrete step response of the controlled system model and the N_2 parameter will be set to the first element reaching 95 % of the controlled system static gain. The main target will be the λ parameter and the goal will be to find such values of λ, which would result in a closed-loop with similar characteristics regardless of the controlled system dynamics ($\bar{\sigma}^2$) or sampling period (T_S). In other words, the λ parameter must be normalized.

3.1 Normalization of the λ weighting coefficient

The normalization procedure is based on an inverse optimization problem. The knowledge of the controlled system characteristic numbers and the pulse-step restriction on control sequences is used to estimate the optimal control sequence resulting from (1). Afterwards the parameter λ which leads to the particular estimated control sequence is found. Two methods for estimation of the optimal control sequence are presented further.

Normalization by degraded pulse-step control sequence The first method to estimate the optimal control signal is based on t-optimal constrained control of a FOPDT system, which is depicted in Fig. 3. When a transition of a FOPDT system to the setpoint u^∞ (0.8 in Fig. 3) is needed, the control signal is at its maximum until time T_R and then skips to the value u^∞ (remember, only systems with $K = 1$ are considered). The time T_R can be determined easily from the analytic formula for the FOPDT system step-response. Such control signal is in fact a degraded pulse-step control sequence where $n_1 = n_2$.

Now we can define auxiliary optimization problem:
We omit the constraints on the input signal, we use setpoint $w = 1$ and assume zero disturbance ($d = 0$)

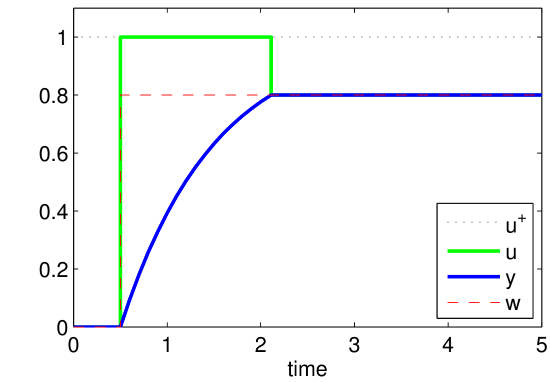

Fig. 3. Time-optimal control of a first order system

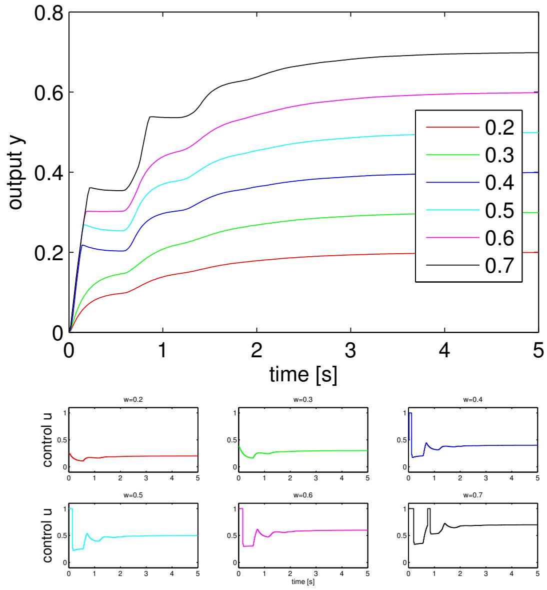

Fig. 4. Step responses and control signals for various setpoint step changes

and zero initial state and we find a control signal in the shape shown in Fig. 3 which minimizes the criterion (1). The width T_R is fixed, it is computed from the dynamics of the system and rounded to the nearest integer multiple of the sampling period. Therefore there are only 2 variables to optimize: u_{max} and u^{∞}. The ratio between the optimal values of the variables ($\frac{u_{max}}{u^{\infty}}$) corresponds with the agresivity of the controller, it rises when the parameter λ is decreased.

And we define inverse optimization problem:
We want to find such λ, which will result in u_{max} and u^{∞} having a given ratio when solving the previous optimization problem.

Such a value of λ can be found using the least squares method and we can use it as a parameter for the PSMPC controller. For example if we find λ resulting in $u_{max}/u^{\infty} = 3$, we in fact tell the PSMPC algorithm to use the saturation limits only when the setpoint step change is bigger than 1/3 (the ratio u^{+}/u^{∞} must be at most 3). This is depicted in Fig. 4. For smaller values of u^{+}/u^{∞} we get higher λ, resulting in a more conservative

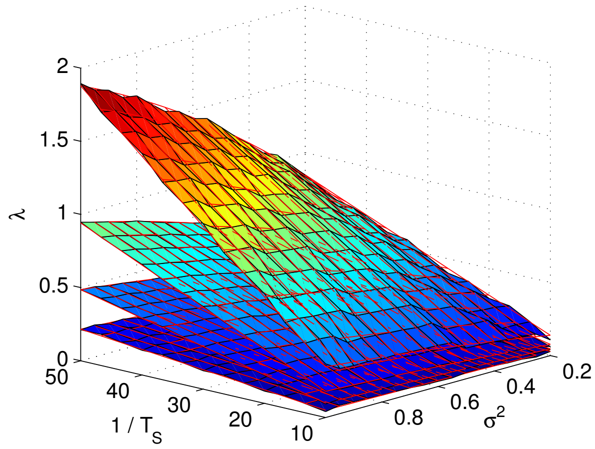

Fig. 5. Values of λ resulting from normalization based on degraded pulse-step control sequence; top-down: λ_C, λ_M, λ_A, λ_{AA}

controller, which uses the saturation limit only for even bigger setpoint changes. In this way, we can normalize the agresivity of the PSMPC controller in the sense that it has the same tendency to use the limit values of the control signal regardless of the system dynamics or sampling period. Suitable values for normalization are displayed in Table 1 along with the notation used further.

Table 1. Normalization by $\frac{u^{+}}{u^{\infty}}$ ratio

$\frac{u^{+}}{u^{\infty}}$	1.5	2.0	2.5	3.0
Controller agresivity	conservative	medium	aggressive	very aggressive
Notation for resulting λ	λ_C	λ_M	λ_A	λ_{AA}

By computing the λ parameters for each combination of sampling frequency (T_S^{-1}) and system dynamics ($\bar{\sigma}^2$), we get surfaces in the T_S^{-1}–$\bar{\sigma}^2$–λ space, which are depicted in Fig. 5.

But even for the most conservative λ_C, the controller might be too aggressive in some cases (due to noise or model discrepancies) and in such a case another normalization method must be used.

Normalization by constant control sequence In order to obtain the values of λ for even more conservative controllers, we use similar approach. The difference is that we omit the nonlinear part of pulse-step control ($n_1 = n_2 = 0$) and we simply find a value of λ resulting in a constant control signal of given amplitude u^{∞}. Suitable values for normalization are displayed in Table 2 along with the notation used further. Controllers resulting from the use of λ_{C1} to λ_{C4} are conservative and thus robust and noise-insensitive. Again, the values of λ can be displayed in the T_S^{-1}–$\bar{\sigma}^2$–λ space, see Fig. 6.

It is evident from Fig. 5 and 6 that the ranges within which the λ parameter should be tuned differ enormously

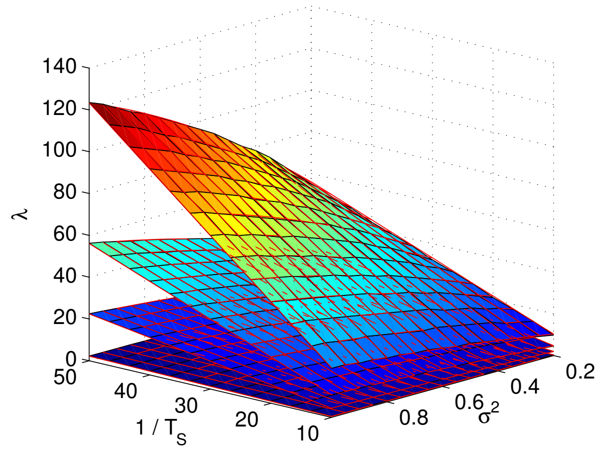

Fig. 6. Values of λ resulting from normalization based on constant control sequence; top-down: λ_{C4}, λ_{C3}, λ_{C2}, λ_{C1}

for various sampling frequencies and system dynamics as was mentioned in Section 3.

Explicit formulas for the λ parameter The computed penalization coefficients λ lead to very good closed-loop performance but for implementation in simple devices, it is necessary to derive explicit formulas to compute the λ parameter from. Thus the surfaces displayed in Fig. 5 and 6 must be interpolated by an appropriate function.

For each fixed $\bar{\sigma}^2$, λ is a linear function of sampling frequency T_S^{-1}. The slope of such linear dependence is a quadratic function of the $\bar{\sigma}^2$ parameter. Therefore the relation between λ, T_S^{-1}, and $\bar{\sigma}^2$ can be approximated by

$$\lambda = \frac{k_1\left(\bar{\sigma}^2\right)^2 + k_2\bar{\sigma}^2 + k_3}{T_s} + q \tag{7}$$

Table 3 shows the coefficients for various levels of controller agresivity. The approximating functions are displayed in Fig. 5 and 6 in red color.

For implementation in target device it is sufficient to incorporate 8 formulas (7) into the controller and the intervals to vary the λ penalizing coefficient within can be computed as soon as the controlled system model is known. It is also possible to perform some mapping of the interval $\langle -1; 1\rangle$ to the interval given by $\langle \lambda_{AA}; \lambda_{C4}\rangle$, so that the user can tune the controller from the most conservative (1, i.e. λ_{C4}) to the most aggressive (-1, i.e. λ_{AA}).

For systems with static gain different from 1, the λ parameter must be denormalized in gain by multiplying by κ^2. Denormalization in time is not necessary.

Table 2. Normalization by constant control signal

u^∞	1.25	1.0	0.75	0.5
Controller conservativeness	low	medium	high	very high
Notation for resulting λ	λ_{C1}	λ_{C2}	λ_{C3}	λ_{C4}

4. AUTOMATIC TUNING OF CONTROLLER PARAMETERS

The methods described so far can be used to create an autotuning procedure for the PSMPC controller, which can be used in typical process control applications, where the controlled processes are monotonous and can be described by a transfer function in the form of (2) with satisfactory precision.

4.1 Process identification

The characteristic numbers κ, μ and σ^2 can be obtained easily from a very short and simple experiment. The controlled process is excited by a rectangular pulse and the characteristic numbers are computed from the measured response. This identification technique has been widely accepted in industrial practice for PID controllers tuning purposes (Schlegel et al., 2003).

4.2 Controller parameters

As soon as the characteristic numbers κ, μ and σ^2 are measured, the FOPDT or SOPDT approximation is computed. It is up to the user to determine which model suits the controlled system best. The corresponding discrete step response can be generated afterwards and the λ parameter is evaluated as described above. Analysis of the linear part of the PSMPC controller, simulations and practical experiments show that λ_C is a safe and robust starting point. The agresivity can be increased manually if the noise is not the issue and the model fits the real system well.

5. EXAMPLE

The properties of the pulse-step model predictive controller will be illustrated here. Consider the controlled system described by the transfer function

$$P(s) = \frac{1}{(0.2s+1)^5} \tag{8}$$

and manipulated value constraint $u \in \langle 0, 1\rangle$.

The sampling period of $T_S = 0.02$ s is used. The controlled system is approximated by the SOPDT system

$$P(s) = \frac{1}{(0.3162s+1)^2} \cdot e^{-0.3675s} \tag{9}$$

whose discrete step response is used for predicting the system output. The other parameters of the PSMPC controller are $H_C = 10$, $N_1 = 21$, $N_2 = 92$.

Table 3. Coefficients for explicit computation of the λ parameter

Agresivity level	k_1	k_2	k_3	q
λ_{AA}	-0.0012	0.0042	0.0012	0.0085
λ_A	-0.0028	0.0099	0.0026	0.0010
λ_M	-0.0055	0.0194	0.0052	-0.0117
λ_C	-0.0110	0.0389	0.0104	-0.0383
λ_{CC1}	-0.0179	0.0501	0.0092	0.1507
λ_{CC2}	-0.1399	0.4720	0.1171	-0.0559
λ_{CC3}	-0.3433	1.1753	0.2969	-0.4003
λ_{CC4}	-0.7500	2.5819	0.6565	-1.0892

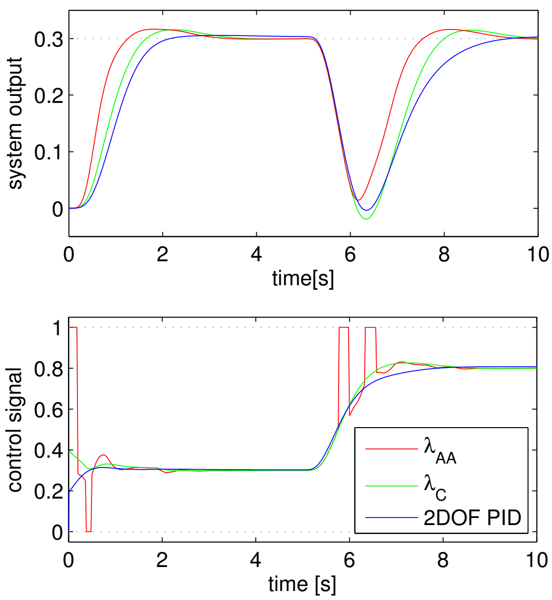

Fig. 7. Comparison of aggressive PSMPC, conservative PSMPC and 2DOF PID

Fig. 7 compares the behavior of pulse-step predictive controller to the classical PID controller. A setpoint step change of 0.3 occurs at time $t = 0$ and a constant load disturbance of -0.4 acts on the system from time $t = 5$s onward. One response belongs to the aggressive PSMPC controller ($\lambda = \lambda_{AA}$), the other is conservative ($\lambda = \lambda_C$) and the third response is a 2DOF PID control loop as a benchmark. The PID controller was tuned in the virtual PID laboratory `www.pidlab.com` (Čech and Schlegel, 2006) with respect to the following design specifications: gain margin $G_m = 2$, phase margin $P_m = 60°$, and restriction on the peak of the sensitivity function $M_S < 1.4$. The step response reaches the setpoint faster with the predictive controller. On the other hand, small but acceptable overshoot occurs. Notice the pulse-step shape of the control sequence of the aggressive controller. The manipulated value constraints are kept and fully exploited when setpoint changes. The input disturbance is also rejected faster by the predictive controller. The aggressive one uses the limit values and thus returns to the desired value significantly faster. On the other hand, the most aggressive controller might be too aggressive when measurement noise is present. In such a case, the more conservative (and thus robust) parameters must be used.

6. CONCLUSION

The described autotuning method for the pulse-step model predictive controller was successfully implemented into compact controllers and PLCs. Along with the quality of the control loop, the autotuning feature is supposed to catalyze the spreading of MPC in the lower levels of control systems.

Thanks to the autotuner, the comissioning of the PSMPC controller is straightforward. The controller parameters are computed from a simple experiment and the resulting closed-loop performance is very high for the typical processes. But there is still a room for a human expert, who can easily adjust the controller parameters (which all have a clear physical meaning) and even improve the overall performance of the control loop.

The PSMPC controller is a part of the Matlab/Simulink compatible RexLib function block library, which is available for open public and whose general description was published by Balda et al. (2005).

ACKNOWLEDGEMENTS

This paper was supported by the European Regional Development Fund (ERDF), project "NTIS - New Technologies for Information Society", European Centre of Excellence, CZ.1.05/1.1.00/02.0090. This support is gratefully acknowledged.

REFERENCES

Al-Ghazzawi, A., Ali, E., Nouh, A., and Zafiriou, E. (2001). On-line tuning strategy for model predictive controllers. *Journal of Process Control*, 11(3), 265–284.

Åström, K.J., Hägglund, T., Hang, C., and Ho, W. (1993). Automatic tuning and adaptation for PID controllers - a survey. *Control Engineering Practice*, 1(4), 699 – 714.

Balda, P., Schlegel, M., and Štětina, M. (2005). Advanced control algorithms + Simulink compatibility + Real–time OS = REX. In *Preprints of the 16th IFAC World Congress*. Prague, Czech Republic.

Bemporad, A. and Filippi, C. (2002). Suboptimal explicit RHC via approximate multiparametric quadratic programming. In *Technical report AUT02-07*. Automatic Control Laboratory, ETH Zurich.

Čech, M. and Schlegel, M. (2006). Fractional-order PID controller design on internet: www.PIDlab.com. In *Proceedings of 7th International Carpathian control conference*. VŠB - Technická univerzita, Ostrava.

Kvasnica, M., Grieder, P., and Baotić, M. (2004). Multi-Parametric Toolbox (MPT).

Maciejowski, J. (2002). *Predictive control with constraints*. Prentice-Hall.

Richalet, J., el Ata-Doss, S.A., Arber, C., and Kuntze, H. (1987). Predictive functional control: application to fast and accurate robots. In *Proceedings of the 10th IFAC World Congress*. Munich, Germany.

Schlegel, M., Balda, P., and Štětina, M. (2003). Robust PID autotuner: method of moments. *Automatizace*, 46(4), 242–246.

Schlegel, M. and Sobota, J. (2008). Simple pulse-step model predictive controller. In *Preprints of the 17th IFAC World Congress*. Seoul, Korea.

Schlegel, M. and Večerek, O. (2005). Robust design of Smith predictive controller for moment model set. In *Preprints of the 16th IFAC World Congress*. Prague, Czech Republic.

Tondel, P. and Johansen, T. (2002). Complexity reduction in explicit model predictive control. In *IFAC World Congress Proceedings*. Barcelona, Spain.

Tondel, P., Johansen, T., and Bemporad, A. (2003). An algorithm for multi-parametric quadratic programming and explicit MPC solutions. *Automatica*, 39, 489–497.

Wallén, A. and Åström, K. (2002). Pulse-step control. In *Proceedings of 15th Triennial World Congress*. Barcelona, Spain.