POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Executive Summary of the Thesis

Problem-tailored model parametrization for the autotuning of event-based controllers

Laurea Magistrale in Automation and Control Engineering - Ingegneria dell'Automazione

Author: Marco Zamuner

Advisor: Prof. Alberto Leva

Academic year: 2022-2023

1. Introduction

With reference to model based (auto)tuning of industrial controller, we discuss the role of the procedure used to parametrize the chosen process model and makes that technique an integral part of the controller synthesis procedure together with the tuning rule of choice. Making reference to Event-Based (EB) digital controller realizations, we discuss the problem of extending tuning rules conceived in continuous time so as to also determine the controller parameters that refer to the event triggering mechanism.
The auto-tuning research is still an open field, also because control techniques have evolved beyond PID, while at the same time model identification has evolved and learning-base methodologies have emerged. We focus this research on model-based tuning, i.e., on tuning techniques that first use the collected process Input/Output (I/O) data to identify a process model and then exploit that model to tune the regulator. The accuracy in the parametrization of the process directly affects the quality of the tuning results and this is a critical aspect that is rarely taken into account; for this reason the first contribution is about the relationship between the Model Parametrization Procedure (MPP) and the tuning rule selected. We developed a technique that highlights the best Model Parametrization Procedure-Tuning Rule (MPP-TR) compound to help in the selection of the tuning rule for a specific application. The proposed technique is tested with different benchmark processes and in order to compare the tuning rules results we selected different tuning quality indices. The tuning quality indices are a powerful tool that allows to understand how a controller behaves [2, 5] and due to their clarity they can be implemented in an industrial scenario.
Another research area that is growing in recent times is EB control. The EB approach can be implemented in both wired and wireless systems, and is gaining particular importance in the latter case owing to the need for rapid plant reconfiguration that is typical of modern manufacturing. The advantages of an EB system are numerous; for example in modern control systems where the presence of potentially overloaded communication channels is a problem, an EB approach will reduce the traffic load on the network. For battery-operated devices this approach allows to save energy so improve the life cycle. Implementing a controller in an EB *scenario* is not as simple as designing the same controller in a

classical (fixed-rate) one because any mechanism to generate events has parameters, and therefore any tuning rule procedure should provide the event related parameters together with those relative to the control law.
It is clear that two (intertwined) problems are open. One is how to choose the MPP-TR compound for the control quality index of choice. The other is how to extend a tuning rule to the EB context. This work addresses the two problems individually, in a view of first solving them and then provide the foundations for future research to comprehend them jointly.

2. Background

In this section are introduced and explained the elements of the proposal.

2.1. Model-based PI and PID Tuning

The focus is on the model-based tuning to have uniformity on the results and on the MPPs. The availability of the process model allows to forecast the tuning results and to set up some tuning quality indices. To ease the comparison we work in normalized conditions and with normalized tuning rules, where possible. All the rules selected use the First Order Plus Dead Time process (FOPDT) to tune the parameters.

$$P(s) = \mu \frac{e^{-sD}}{1+sT}$$

The process in nominal conditions has unitary gain $\mu = 1$ and to make the results comparable we used the normalized time delay $\theta \in [0, 1]$

$$\theta = \frac{D}{D+T} \tag{1}$$

We selected twenty one rules from the hundreds collected in the "Handbook of PI and PID Controller Tuning Rules" [4]. The twelve PI tuning rules (hereinafter TR1 to TR12) are - in this order - IMC, Sko, Riv, Dsd, ABB, LSM, D&C, Mann, H&A, Lee, I&G and Smith. The nine PID tuning rules (TR13 to TR21) are - again, in this order - IMC, Connell, Moros, Liptàk, Sree, Wang, Frehauf, Riv and Lee.
Some tuning rules are parameter-free, some are not. We uniform the procedure to select the parameter λ, that can be interpreted as the desired closed-loop time constant

$$\lambda = \frac{T + D/5}{k_a}$$

k_a is the acceleration factor.
The PI controller has structure

$$C(s) = K_C \left(1 + \frac{1}{sT_i}\right)$$

For the PID we have two different control structures, the filterd one

$$C(s) = K_C \left(1 + \frac{1}{sT_i} + \frac{sT_d}{1 + sT_d/N}\right)$$

and the academic one

$$C(s) = K_C \left(1 + \frac{1}{sT_i} + sT_d\right)$$

2.2. Test Benchmark

To test the technique we decided to use the first five benchmark classes of the well known Åström benchmark [1]. They are well representative of most scenarios that can be found on a real implementation and are standard systems that are well suited for parametric studies.

$$P_1(s) = \frac{1}{(s+1)^\alpha}, \ \alpha = 2, 3, 4, 5, 6, 7, 8;$$
$$P_2(s) = \frac{1}{(s+1)(1+\alpha s)(1+\alpha^2 s)(1+\alpha^3 s)}$$
$$\alpha = 0.05 : 0.05 : 0.95;$$
$$P_3(s) = \frac{1-\alpha s}{(s+1)^3}, \ \alpha = 0.1 : 0.1 : 5;$$
$$P_4(s) = \frac{e^{-s}}{1+s\alpha}, \ \alpha = 0.1 : 0.1 : 10;$$
$$P_5(s) = \frac{e^{-s}}{(1+s\alpha)^2}, \ \alpha = 0.1 : 0.1 : 10;$$

2.3. Event-based Realization

An EB controller updates its control law only when an event is triggered, this means that the system has changed and the old control action does not meet the specifications.
In an EB implementation an event can occur at any time, and this brings about some problems such as the well known Zeno behaviour. In order to avoid such problems, and thanks to the inherently clocked nature of any industrial (digital) control system, we decided to resort to *periodic* EB control. In periodic EB systems event can occur only at integer multiples of a time quantum q, hence such systems can be treated as fixed-rate ones where "some steps are skipped" thanks to the event-triggering mechanism. We

decided to compute q, that taking the above viewpoint comes to play the role of the sampling time, constraining the phase margin reduction caused by the digital controller realization to be less than a prescribed $\Delta\varphi_m$, k_c is a parameter that ranges from 1/2 to 3/2 and ω_c is the nominal cutoff frequency.

$$q = \frac{\Delta\varphi_m}{k_c\omega_c} \tag{2}$$

There are different event-triggering mechanism and the choice of one with respect to another depends on the specific application. An example can be the Send on Delta (SoD) policy where the event is triggered when in a multiple of q the difference in magnitude between the current measurement y and the past one y_{old} is greater than a certain threshold δ_y.
Another element to consider in an EB control is how to express the tuning rule in discrete-time. In literature there are different discretization methods, we decided to use Backward Euler. The specific EB realization selected is based on multitransmission [3]. This technique considers the controller a switching system and the switching signal is σ that is a boolean variable. The switching signal σ will determine the controller status, hence for $\sigma = 1$ the controller is in Running mode (R-mode) and for $\sigma = 0$ it is in Holding mode (H-mode). When an event is detected the controller switches from H-mode to R-mode and the event generator transmits the present $y(k)$ value and the last v_p past values $(y(k-1)\dots y(k-v_p))$ with these values the controller is able to update its control law. The detection of an event will make the controller transmit for the further $v_f - 1$ instants then it returns in H-mode. The paper reports that the state of the closed-loop system, represented in the closed-loop switching dynamic matrix A_σ, in H-mode, can only expand up to the multiplying factor

$$E_{H,\xi} = \sqrt{1 + (n_C - 1) \min_{i=1,\dots,n_P} \left(\frac{1-\lambda_{Pi}}{\rho_{Pi}}\right)^2}$$

where n_C is the order of the controller, λ_{Pi} is the set of the eigenvalues of the process and ρ_{Pi} is the set of residues. The key point is that the state of closed-loop system in R-mode is contracting subjected to the same expansion ratio and it achieves the stability in a minimum number of steps identified by Δ_R.
Δ_R is the parameter that is used to determine the minimum time of transmission from the last event received.

3. The Proposed MPP-TR Selection Technique

The proposed evaluation technique is composed by two parts one offline that evaluates the tuning quality indices and computes the MPP-TR tables using directly the benchmark classes and one online that can be used directly on the field to determine which rule is the best. The online technique cannot be used standalone we need the tables computed in the offline technique.
The two MPPs selected are the Method of Areas (M_1) and the Sundaresan an Krishnaswamy method or percentage method (M_2). These two methods require a process step response to compute the equivalent FOPDT model but how they compute the parameters is very different. M_1 symbolically computes the two integrals

$$A_0 = \int_0^\infty (y_{us}(\infty) - y_{us}(t))dt,$$
$$A_1 = \int_0^{A_0/y_{us}(\infty)} y_{us}(t)dt$$

where $y_{us}(\infty) = lim_{t\to\infty} y_{us}(t)$, then set

$$\mu = y(\infty), \quad T = e\frac{A_1}{\mu}, \quad D = \frac{A_0}{\mu} - T$$

M_2 selects two instants t_1 and t_2 that correspond to the 35.3% and the 85.3% of the total step amplitude. The parameters are

$$\mu = y(\infty),\ T = \frac{2}{3}(t_2 - t_1),\ D = 1.3t_1 - 0.29t_2$$

The tuning quality indices selected covers two different control scenarios, set-point tracking and disturbance rejection. These scenarios require different control tuning hence we selected different indices that exploits better the control problems.
The indices for set-point tracking are:
- ISE, integral square error;
- Maximum overshoot;
- 99% settling time.

For disturbance rejection we have:
- ISE, integral square error;
- Maximum absolute error.

3.1. Evaluation Technique

The offline part of the proposed technique is summarized below.

1. Select a benchmark class;
2. Make the class parameter change in a predefined range;
3. Compute the equivalent FOPDT model through an MPP;
4. Compute for each model the normalized estimated delay, θ_{si} as (1);
5. Tune the controller for each model and tuning rule;
6. Compute the indices;
7. For each MPP create a table of the best tuning rule per index.

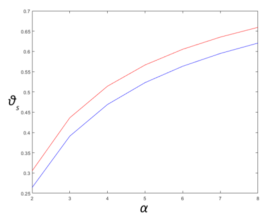
Each table of the MPP-TR compound is codified, through interpolation or a threshold method, in the rule $nI[A, C]$ where n is the benchmark class, I is the index selected and $[A, C]$ gives the aggressive (A) or conservative (C) approach. All the tuning rules that are too unstable or too low damped for a certain θ_s are discarded from the pool of results, we set as minimum acceptable phase margin $\varphi_m = 20°$.
The online part of the technique given an unknown problem is:

1. Perform the process response;
2. Compute the equivalent FOPDT and θ with all MPPs hence there will be n_{MPP} models and n_{MPP} θ;
3. For each MPP:
    (a) Identify in which benchmark class the process can be approximated and select which table to use;
    (b) Decide the θ approach (aggressive, conservative...);
    (c) Select the tuning rule from the table and save the corresponding index value;
4. Select the best index and get the optimal MPP-TR compound.

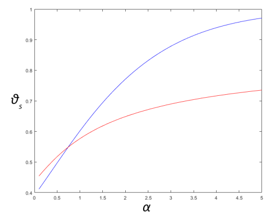
We assume to know in which benchmark class the process belongs to.

3.2. Benchmark Assessment

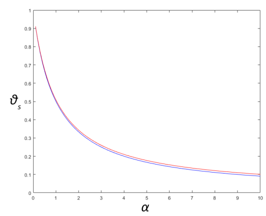
Here we propose some results obtained from the evaluation of the proposed technique.
The two MPPs return different θ_s from the same step response as you can see in the figures 1 hence the procedures are not directly comparable. For this reason in the technique evaluation the two normalized delays are always computed and the greater θ_s matches the most conservative implementation, the smaller one matches the most aggressive.
Usually M_2 is more conservative than M_1 but for benchmark class 3 this does not happen because the percentage method considers the undershoot, caused by the unstable zero, as a pure dead time reducing the accuracy of the estimation.
In figure 1c the two methods can be considered equivalent because they are parametrizing directly the FOPDT. We computed the deviation from the benchmark class parameters, $D = 1$ and $T = \alpha$ and the results are that M_1 does not deviate from the FOPDT. M_2 is less accurate and struggles a lot in the computation of the delay where the deviation ranges from 1% to 11%.



(a) class 1



(b) class 3



(c) class 4

Figure 1: θ_s comparison with respect to benchmark class parameter, the colors are blue for M_1 and red for M_2

Now we show some MPP-TR tables. We have two tables for each index, one for M_1 and one for M_2. In red are highlighted the best tuning rules the others are grayed. Each table has the value of θ_s on the horizontal axis and the tuning rules on the vertical axis; a black bold line marks the separation between the PI and PID tuning rules. The performance indices are computed in response to a unit set-point step and to a unit load disturbance step. We simulated the system for 500 seconds to have the results as uniform as possible.

In the tables there is not a single tuning rule that overcomes the others for all indices. There are some indices for example maximum absolute error and the two ISEs where few rules are better than the others. 99% settling time and maximum overshoot present more rules and in some cases there is a chess-pattern where the rule changes frequently.
Comparing the two parametrization procedures for the same index we observe that there are similar tuning rules but for different values of θ_s. As expected the evaluation for benchmark class 3 shows completely different tuning rules for the two procedures because of the estimation the process parameters.

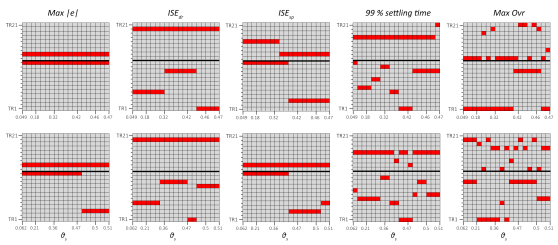


Figure 2: Best tuning rule per index, benchmark class 2. Red best tuning rule, Gray others

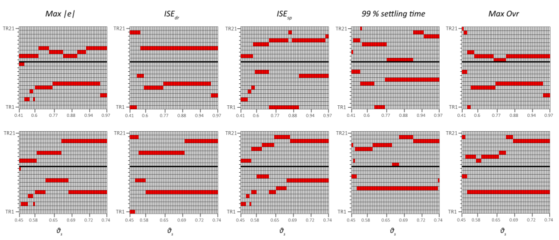


Figure 3: Best tuning rule per index, benchmark class 3. Red best tuning rule, Gray others

Figure 4 finally shows an example in which the selected MPP-TR compound is compared with another one in a load disturbance rejection case with the ISE as quality index: as can be seen, a proper compound selection does help.

4. Going Digital and then Event-Based

For the computation of the event parameters we followed the technique proposed in the paper [3]. We considered the process in structurally nominal condition hence the process is not identified through an MPP. We set $\mu = 1$, $T = 1$ and D comes from θ that varies from 0 to 1.

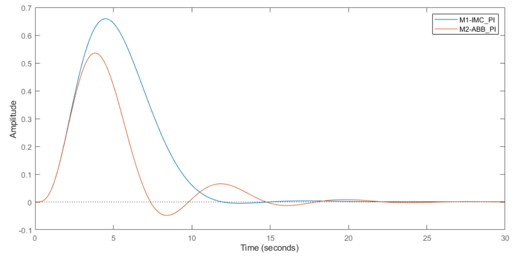


Figure 4: Example of the MPP-TR technique applied, process benchmark class 2, $\theta_{s1} = 0.46$, $\theta_{s2} = 0.51$

The technique is:

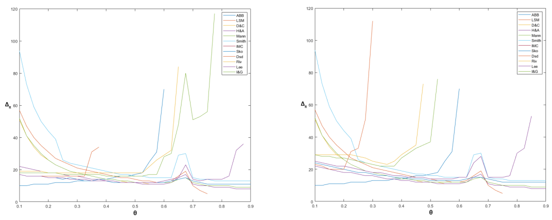
1. Tune the continuous-time PI(D);
2. Form a continuous-time loop with the tuned controller and the normalized FOPDT, compute ω_c and the sampling time q as in (2);
3. Discretize the controller and an approximation of the process both at step q and compute the closed-loop switching dynamic matrix A_σ for both R-mode and H-mode;
4. Compute the change of base T_H and $\tilde{A_R} = T_H^{-1} A_R T_H$,
5. Evaluate the maximum multiplicative expansion that in our case is $E_{H,\xi} = \sqrt{n_c}$;
6. Repeatedly check the condition $E_{H,\xi} \|\tilde{A_R}^k\|_2 < 1, \; \forall k \geq \Delta_R$, until a suitable Δ_R is found.

For all controllers we set $\Delta\varphi_m = 5°$, $k_c = 0.5$, θ ranges from 0.1 to 0.9 and k_a varies from 0.5 to 2.

4.1. Event-Based Realization Results

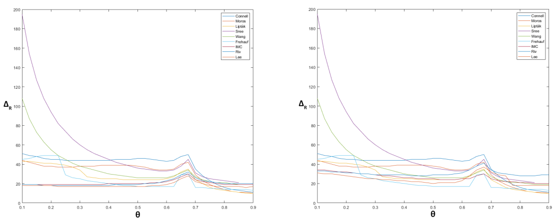
We observe that not all controllers are able to satisfy the iterated condition $E_{H,\xi} \|\tilde{A_R}^k\|_2 < 1, \; \forall k \geq \Delta_R$. The tuning rules that failed to converge are unstable or present too many oscillations.
We can observe that for $\theta \in [0.6, 0.7]$ there is a spike. The spike is caused by the coupling of discretization method and the Padé approximation of the process because the discretized process becomes singular for $\theta = 2/3$ hence the condition $E_{H,\xi} \|\tilde{A_R}^k\|_2 < 1$ cannot be satisfied.
In the following figures we evidence some common behaviours. When a tuning rule converges we see that Δ_R decreases as θ increases and Δ_R increases as k_a grows. For the unstable tuning rules the plots are truncated, we can detect the instability because Δ_R increases rapidly.

The PI tuning rules develop the instability more than the PID ones because of the higher complexity of the latter. We highlight that the parameter-free rules usually have higher initial Δ_R and the difference between the parameter-free rules to the others is evident for PI and less clear for PID. We can see that as k_a increases, more λ-dependent rules become unstable.



(a) $k_a = 1$ (b) $k_a = 2$

Figure 5: Δ_R comparison of PI tuning rules, k_a fixed. Δ_R vertical axis, θ horizontal axis



(a) $k_a = 0.5$ (b) $k_a = 1.5$

Figure 6: Δ_R comparison of PID tuning rules, k_a fixed. Δ_R vertical axis, θ horizontal axis

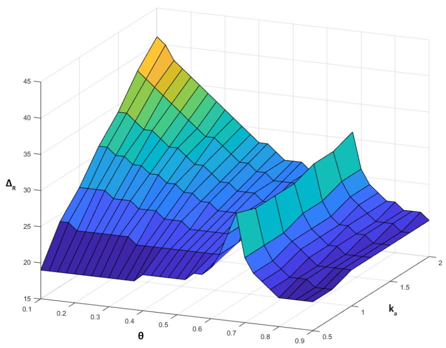


Figure 7: Surface plot of IMC-PID

We can take the results of each tuning rule and interpolate one function for Δ_R and one for q. These functions can be used on the field to quickly determine the event parameters for the controller.

We can use together the MPP-TR tables and the interpolating functions, but some compatibility problems could arise therefore further research should be done, for example compute the MPP-TR tables directly in discrete time.

5. Conclusions

The work objective was to provide a technique to choose the best MPP-TR compound given a tuning performance index and to extend a tuning rule to the EB context.
We studied the two problems individually; for the first problem we saw that there is not a tuning rule that exploits better than others all the quality indices and benchmark tests therefore provide the optimal MPP-TR compound will ease the design of the controller. For the second problem the computation of the event related parameters showed the limits that some tuning rules have with the chosen EB realization.
In this work we studied these two problems individually in a view to provide the foundations for future research to study them jointly and to expand to other tuning rules and MPPs.

References

[1] Karl Johan Åström and Tore Hägglund. Benchmark systems for pid control. *IFAC Proceedings Volumes*, 33(4):165–166, 2000.

[2] A. Leva and F. Donida. Quality indices for the autotuning of industrial regulators. *IET Control Theory & Applications*, 3(2):170–180, 2009.

[3] Alberto Leva, Federico Terraneo, and Silvano Seva. A multitransmission event-based architecture for energy-efficient autotuning wireless controls. *IEEE Transactions on Control Systems Technology*, 30(4):1510–1524, 2021.

[4] Aidan O'dwyer. *Handbook of PI and PID controller tuning rules*. World Scientific Publishing Company, 2006.

[5] Silvano Seva, Chiara Cimino, and Alberto Leva. On the criticality of the model parametrisation method in industrial autotuning controllers. In *2021 60th IEEE Conference on Decision and Control (CDC)*, pages 1137–1142. IEEE, 2021.