Analytical PI Controller Tuning Using Closed-loop Setpoint Response

Wuhua Hu, Gaoxi Xiao *

School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798, Singapore

Abstract: Recently Shamsuzzoha and Skogestad proposed a proportional-integral (PI) tuning rule for a wide range of unidentified processes. The rule relies on a closed-loop setpoint response (CSR) of a process and was developed from extensive numerical experiments. This work analytically derives a similar PI tuning rule using the CSR method. Simulations indicate that the two rules perform similarly if the tuning parameter is selected properly for the analytical rule. Meanwhile, a guideline is proposed for choosing the P controller gain for the CSR experiment to result in a proper overshoot for obtaining good PI settings. Numerical examples are used to demonstrate the usefulness of the theoretical results.

1. Introduction

The proportional-integral (PI) controller has been widely applied in industry for its simplicity, robustness, and wide ranges of applicability in regulatory layer.1-5 PI controller tuning has been extensively studied in the last decades, generating a large number of PI tuning rules.3-6 Conventional PI controller tuning, however, requires trials and may experience instability during tuning experiment or process modeling, and the resulting closed-loop performance is satisfactory only for particular classes of processes.1 To overcome these problems, recently Shamsuzzoha and Skogestad proposed using closed-loop setpoint response (CSR) to set the PI parameters, which requires a single closed-loop experiment and gives fast and robust performance for a broad range of

* To whom correspondence should be addressed. Tel.: +65 6790 4552. Fax: +65 6793 3318. E-mail: egxxiao@ntu.edu.sg.
processes typical for process control. An earlier CSR method was considered by Yuwana and Seborg in 1982, but their method leads to a more complicated solution.

In the CSR method, one carries out a closed-loop experiment with a single proportional (P) controller and then utilizes the response information to derive the PI settings. Shamsuzzoha and Skogestad considered a special case of the simple internal model control (SIMC) tuning rule with its single tuning parameter, the closed-loop time constant \( \tau_c \), set as \( \tau_c = \theta \), where \( \theta \) is the effective time delay of a process. They derived a PI tuning rule by relating the closed-loop response quantities with the SIMC settings, including the peak time, overshoot and steady-state offset. The resulting PI tuning rule was tested on a broad range of processes and demonstrated to give comparable performance as the SIMC tuning rule.

While Shamsuzzoha and Skogestad developed the PI tuning rule from series of numerical experiments, analytical derivation of a similar rule using CSR method is of interest in this work. The derivation is based on an integral plus time delay (ITD) process and then extended to a first-order plus time delay (FOTD) process. The main idea is as follows. With the CSR method, a single P controller is applied to the process and a step test of setpoint change is performed. From the closed-loop response, the steady-state offset, peak time, and overshoot or rise time are recorded. These quantities, together with the applied proportional gain and setpoint change, are used to estimate the process parameters and consequently express the SIMC tuning rule in a new manner. The resulting PI tuning rule has a single tuning parameter \( \alpha \) which controls the trade-off between performance and robustness. This rule is tested on a broad range of processes typical for process control applications. The results indicate that the analytical rule gives comparable performance to Shamsuzzoha-Skogestad’s PI tuning rule if the detuning parameter \( \alpha \) is chosen properly. In a sense, the analysis and derived rule provide some kind of insight and support to the PI tuning rule proposed by Shamsuzzoha and Skogestad.
Throughout the paper, the notation “:=” means “is defined as” and is used to introduce new symbols when necessary.

2. Derivation of the PI Tuning Rule

The control system is described in Figure 1, where \( u \) is the manipulated control input, \( d \) the disturbance, \( y \) the controlled output, \( y_s \) the setpoint (reference) for the controlled output, \( c(s) \) the PI controller transfer function, and \( g(s) \) the process transfer function. The PI controller takes the form of

\[
c(s) = k_c \left(1 + \frac{1}{\tau_i s}\right),
\]

where \( k_c \) and \( \tau_i \) are the proportional (P) gain and the integral (I) time constant respectively.

**CSR Experiment.** When a single P controller \( (c(s) = k_{c0}) \) is applied to the process, a setpoint change is made. From the CSR experiment (see Figure 2), we record the following values:\(^1\)

- \( \Delta y_s \): Setpoint change
- \( \Delta y_p \): Peak output change
- \( \Delta y_{\infty} \): Steady-state output change after setpoint step test
- \( t_r \): Time from setpoint change to reach steady-state output for the first time
- \( t_p \): Time from setpoint change to reach peak output
- \( k_{c0} \): Controller gain used in experiment.

From this data, the following parameters are calculated

\[
M_p = \frac{\Delta y_p - \Delta y_{\infty}}{\Delta y_{\infty}}, \quad b = \frac{\Delta y_{\infty}}{\Delta y_s}.
\]

As recommended in (Shamsuzzoha and Skogestad, 2010), for deriving good PI settings the experiment should make \( M_p \) be larger than 10% and best around 30%. In case that it takes a long
time for the response to settle down, one may simply record the output, $\Delta y_u$, when the response reaches its first minimum and compute $\Delta y_\infty$ as $\Delta y_\infty = 0.45(\Delta y_p + \Delta y_u).$

The analytical derivation of the PI tuning rule proceeds as follows. Consider an ITD process

$$g(s) = \frac{ke^{-\theta s}}{s}, \quad (3)$$

where $k$ is the process gain and $\theta$ the time delay. With $c(s) = k_{c0}$ applied to the process, the closed-loop transfer function is obtained as

$$\tilde{g}(s) := \frac{g(s)c(s)}{1 + g(s)c(s)} = \frac{e^{-\theta s}}{s/K + e^{-\theta s}}, \quad (4)$$

where $K := kk_{c0}$. The time delay component in Eq. (4) is usually approximated by Padé approximation or Maclaurin expansion. Although Padé approximation is normally more accurate, Maclaurin expansion is adopted for the purpose of deriving a simple analytical PI tuning rule. Use Maclaurin expansion and approximate the numerator and denominator of $\tilde{g}(s)$ by the second-order polynomials respectively, yielding

$$\tilde{g}(s) \approx \frac{s^2 - \frac{1}{0.5\theta} s + \frac{1}{0.5\theta^2}}{s^2 + \left(\frac{1}{0.5K\theta^2} - \frac{1}{0.5\theta}\right)s + \frac{1}{0.5\theta^2}}. \quad (5)$$

Hence the characteristic polynomial of $\tilde{g}(s)$ is

$$f(s) := s^2 + \left(\frac{1}{0.5K\theta^2} - \frac{1}{0.5\theta}\right)s + \frac{1}{0.5\theta^2}. \quad (6)$$

The above $f(s)$ is in the standard second-order form, $s^2 + 2\zeta\omega_n s + \omega_n^2$, with

$$\omega_n = \frac{\sqrt{2}}{\theta}, \quad \zeta = \frac{1}{\sqrt{2}} \left(\frac{1}{K\theta} - 1\right). \quad (7)$$
In Eq. (7), $\zeta$ has a physical meaning of being the damping ratio of the closed-loop system. Equation (7) solves $K$ as

$$ K = \frac{1}{\left(\sqrt{2\zeta} + 1\right) \theta}. $$

Therefore the unit step setpoint response is

$$ Y(s) = \tilde{g}(s) y_s(s) \approx \frac{s^2 - \frac{1}{0.5\theta} s + \frac{1}{0.5\theta^2}}{s^2 + \frac{1}{0.5K\theta^2} - \frac{1}{0.5\theta}} \frac{1}{s} = \frac{a}{s} + \frac{b(s + \sigma) + c\omega_n}{s^2 + 2\zeta\omega_n s + \omega_n^2}, $$

where $\sigma := \zeta \omega_n$ and $\omega_n := \omega_n \sqrt{1 - \zeta^2}$, and the parameters $\zeta$ and $\omega_n$ are those defined in Eq. (7), and $a$, $b$ and $c$ are given by

$$ a = 1, \quad b = 0, \quad c = -\frac{1}{0.5K\omega_n\theta^2} = -\frac{2\zeta + \sqrt{2}}{\sqrt{1 - \zeta^2}}. $$

Assume that the initial states of the system and their derivatives are zero. By inverse Laplace transform, from Eq. (9) the time-domain response is derived as

$$ y(t) \approx 1 + ce^{-\sigma t} \sin \omega_n t. $$

From Eq. (11), the time-domain performance indices such as the rise time $t_r$, the peak time $t_p$, and the overshoot $M_p$ can all be estimated. Let the rise time be defined as the time for $y(t)$ reaching the steady-state value of one for the first time. This means

$$ y(t_r) = 1 = 1 + ce^{-\sigma t_r} \sin \omega_n t_r. $$

Equation (12) solves $t_r$ as
\[ t_r = \frac{\pi}{\omega_d} = \frac{\pi \theta}{\sqrt{2(1 - \zeta^2)}}. \]  

(13)

With \( \frac{dy(t)}{dt} \big|_{t=t_p} = 0 \), the peak time \( t_p \) is solved as

\[ t_p = \frac{1}{\omega_d} (\pi + \arccos \zeta) = \frac{\theta}{\sqrt{2(1 - \zeta^2)}} (\pi + \arccos \zeta). \]  

(14)

Consequently the overshoot \( M_p \), which is defined in Eq. (2), is computed as

\[ M_p = \left( y(t_p) - 1 \right) \times 100\% = ce^{-\sigma \tau} \sin \omega_d t_p \times 100\% \]
\[ = \left( 2\zeta + \sqrt{2} \right) \exp \left( -\frac{\zeta}{\sqrt{1 - \zeta^2}} (\pi + \arccos \zeta) \right) \times 100\%. \]  

(15)

Note that the dimensionless scalars \( t_r / \theta \), \( t_p / \theta \) and \( M_p \) are all functions of \( \zeta \). The relationship between \( M_p \) and \( \zeta \) are shown in Figure 3. Hence \( \zeta \) can be read from the \( M_p - \zeta \) curve once \( M_p \) is measured from the CSR experiment. Or it can be solved from Eqs. (13) and (14) as

\[ \zeta = \left| \cos \frac{\pi t_p}{t_r} \right|. \]  

(16)

Although \( \cos \frac{\pi t_p}{t_r} < 0 \) comes from \( \pi \leq \frac{\pi t_p}{t_r} \leq \frac{3\pi}{2} \) as deduced from Eqs. (13) and (14), the absolute is taken to ensure a positive \( \zeta \) since \( t_p \) and \( t_r \) are measured values and the analysis here is approximate. Consequently \( \theta \) can be estimated from Eq. (14) as

\[ \theta = \frac{t_p \sqrt{2(1 - \zeta^2)}}{\pi + \arccos \zeta}. \]  

(17)
(The parameter $\theta$ may also be estimated from Eq. (13) in terms of $t_r$. The estimation in Eq. (17) is recommended since it avoids the measurement of $t_r$ if the damping ratio is read from the $M_p - \zeta$ curve.) The process gain is therefore solved from Eq. (8) as

$$k = \frac{1}{(\sqrt{2}\zeta + 1)\theta k_e 0}.$$  

Note that the SIMC tuning rule\textsuperscript{9} for an ITD process is equivalently expressed by

$$k_c = \frac{\alpha}{k\theta}, \quad \tau_i = \frac{4\theta}{\alpha},$$  

where $\alpha$ is a tuning parameter that corresponds to $\theta/(\tau_e + \theta)$ in the original SIMC rule. Applying the estimated processes parameters $\theta$ and $k$ respectively in Eqs. (17) and (18), the SIMC tuning rule for an ITD process becomes

$$k_c = \alpha(\sqrt{2}\zeta + 1)k_e 0,$$

$$\tau_i = \frac{4}{\alpha} \frac{\sqrt{2(1-\zeta^2)}}{\pi + \arccos \zeta} t_p,$$

where $\zeta = \left| \cos \frac{\pi t_p}{t_r} \right|.$

Alternatively, $\zeta$ can be read from the $M_p - \zeta$ curve shown in Figure 3. Equation (20) is the new PI tuning rule which requires no modeling of the process dynamics but only the peak time and rise time or overshoot as recorded from a single CSR experiment. This eases the PI controller tuning in practice.

Next, consider an FOTD process

$$g(s) = \frac{ke^{-\theta s}}{\tau s + 1}.$$  

(21)
where $k$ is the process gain, $\theta$ the time delay and $\tau$ the process time constant. During the transient of a setpoint response, since it involves mainly high frequency response, the transfer function can be approximated as

$$g(s) \approx \frac{k'e^{-\theta s}}{s}, \quad k' := \frac{k}{\tau}. \quad (22)$$

As the transient dynamics is of main interest, where quantities such as the rise time, peak time and overshoot are measured, approximate analysis can be made similarly to that for an ITD process. Therefore the time delay $\theta$ is estimated in Eq. (17) and the gain $k'$ is estimated in Eq. (8) with $K := k'k_{c0}$. At the steady state, the process gain $k$ satisfies

$$kk_{c0} = \frac{b}{1-b}, \quad (23)$$

where $b$ is given in Eq. (2). Equation (23), together with Eq. (8), solves

$$\tau = \frac{kk_{c0}}{K} = \frac{b}{1-b} (\sqrt{2\zeta} + 1) \theta. \quad (24)$$

The SIMC tuning rule\(^9\) for an FOTD process is equivalently expressed as

$$k_c = \frac{\alpha}{k\theta}, \quad \tau_r = \min \left\{ \tau, \frac{4\theta}{\alpha} \right\}, \quad (25)$$

where $\alpha$ is a tuning parameter. Applying the process parameter estimated in Eqs. (8), (17) and (24), the SIMC tuning rule for an FOTD process is rewritten as

$$k_c = \alpha(\sqrt{2\zeta} + 1)k_{c0},$$

$$\tau_r = \min \left\{ \frac{b}{1-b} (\sqrt{2\zeta} + 1), \frac{4}{\alpha} \right\} \times \frac{\sqrt{2(1-\zeta^2)}}{\pi + \arccos \zeta} t_p, \quad (26)$$

where $\zeta = \cos \frac{\pi t_p}{t_r}$. 
The damping ratio $\zeta$ can alternatively be read from the $M_p - \zeta$ curve as shown in Figure 3. In Eq. (26), the absolute are taken to ensure positive values in the presence of measurement and approximation errors. The tuning rule (26) covers the tuning rule (20), since $b/(1-b) \to \infty$ for an ITD process. Thus for either an ITD or an FOTD process, the PI tuning rule is given in Eq. (26).

The single tuning parameter $\alpha$ controls the trade-off between closed-loop performance and robustness. Hence an appropriate choice of $\alpha$ is important. Though it is difficult to derive any analytical guideline for determining $\alpha$ properly, it is clear that a larger $\alpha$ leads to more aggressive closed-loop response yet less robustness and vice versa. In applications, $\alpha$ can be detuned from a small (conservative) value until satisfactory performance is achieved. Extensive simulations indicate that it is almost sufficient to start $\alpha$ as 0.4 (which is conservative in most cases) and tune it at a step of 0.05 or 0.1 up if the response is too sluggish and down otherwise. The simulations also indicate that an acceptable $\alpha$ normally falls into the range of $[0.2, 0.6]$.

In comparison, using the CSR method, Shamsuzzoha and Skogestad concluded a similar PI tuning rule from series of numerical experiments that aims to match the SIMC rule.\(^1\) The rule takes the form of

\[
k_c = 2\alpha A k_{c,0},
\]

\[
\tau_I = \min \left\{ 0.86 A \left| \frac{b}{1-b} \right| t_p, \frac{1.22}{\alpha} t_p \right\},
\]

where $A = 1.152 M_p ^2 - 1.607 M_p + 1.0$, and $\alpha$ is a tuning parameter similar to the one in Eq. (26), which corresponds to $1/(2F)$ as adopted in the original tuning rule\(^1\). Comparing it with the new rule in Eq. (26), we see that these two rules are similar in form: in the proportional gains, the coefficient $2A$ in Eq. (27) is a function only of $M_p$ and so is the coefficient $\sqrt{2}\zeta + 1$ in Eq. (26) (as refers to the $M_p - \zeta$ curve in Figure 3 or Eq. (15)); and the integral gains are both functions of $t_p$. Nevertheless, the new rule does not
give the same relation between $\sqrt{2\zeta} + 1$ and $M_p$ as its counterpart $2A$ and $M_p$. Another difference is that the rule (27) adopts approximate relations of $\theta = 0.43t_p$ and $\theta = 0.305t_p$ (when $M_p$ varying from 0.1 to 0.6) in the first and second components of the $\min\{\bullet, \bullet\}$ function respectively, whereas the new rule (26) uses a common estimate of $\theta$ for both components as given in Eq. (17). Indeed similar approximate relations between $\theta$ and $t_p$ may be established using the $M_p - \zeta$ curve shown in Figure 3, subject to $\alpha = 0.5$.

**Choice for the P Controller Gain $k_{c_0}$.** An overshoot of around 30% is recommended for the CSR experiment giving good PI settings.\(^1\) This is confirmed by the simulations with the proposed PI tuning rule (The simulation results are not shown for brevity.). Normally such an overshoot is achieved by detuning the P controller gain $k_{c_0}$ via trials and errors. The detuning process can be time consuming and may disturb the process much as are undesirable in applications. Therefore an efficient way for determining $k_{c_0}$ is important for the CSR experiment. We present a method to generate $k_{c_0}$’s that can reduce the number of times of detuning $k_{c_0}$. The method is developed based on the PI tuning rule proposed by Shamsuzzoha and Skogestad, avoiding the errors involved in the above analysis.

The method requires a foregoing CSR experiment. Suppose that we apply a P controller gain of $k_{c_0}^0$ in a CSR experiment and it results in an overshoot $M_p^0$ that is larger than 10% but not around 30%. Let the target overshoot be $M_p^\ast$ and the target P controller gain be $k_{c_0}$. Note that Shamsuzzoha-Skogestad’s PI tuning rule aims to match the SIMC rule which keeps a constant P gain $k_c$ regardless of the overshoot resulted from the CSR experiment. Ideally, $k_c$ should be the same as determined with different overshoots from various CSR experiments. That is, it should have
\[
2\alpha(1.152(M_p^0)^2 - 1.607M_p^0 + 1.0)k_c^0 = 2\alpha(1.152(M_p^*)^2 - 1.607M_p^* + 1.0)k_c^0,
\]

which solves

\[
k_c^0 = \frac{1.152(M_p^0)^2 - 1.607M_p^0 + 1.0}{1.152(M_p^*)^2 - 1.607M_p^* + 1.0}k_c^0.
\]  

(29)

Equation (29) gives a general guideline for choosing the P controller gain for the next CSR experiment. If \( M_p^* \) is set as 30\%, then the gain for the next CSR experiment is recommended as

\[
k_c^0 = 1.609\times[1.152(M_p^0)^2 - 1.607M_p^0 + 1.0]k_c^0.
\]  

(30)

If the gain does not result in a desired overshoot, the formula (30) can be applied repeatedly until the overshoot reaches around 30\%. Such a repeating process converges and ultimately gives a P controller gain that results in the exact overshoot of 30\%. With the monotonic relationship between \( M_p \) and \( k_c^0 \), this can be understood from Eq. (29): The gain \( k_c^0 \) will be adjusted until \( M_p^0 \rightarrow M_p^* \) and thus \( k_c^0 \rightarrow k_c^0 \). The observation has been justified by extensive simulations and typical results are shown in the next section.

3. Simulation Results

Although the PI tuning rule in Eq. (26) was derived for ITD and FOTD processes, it turns out to be effective for a wide range of processes. Simulations were carried out on various processes and typical results are summarized in Table 1. (For all the processes being studied, we adopt the same numbering as that in (Shamsuzzooha and Skogestad, 2010)\(^1\) to achieve good consistency and easy reference.) In the simulations, the damping ratios \( \zeta \)'s were read from the \( M_p - \zeta \) curve or computed using the rise time \( t_r \)'s and the peak time \( t_p \)'s. Typical simulation results are shown in Figure 4, where in each case a unit step change was applied in both the setpoint and the disturbance.
The results indicate that the proposed PI tuning rule leads to similar closed-loop performance and robustness (in terms of peak sensitivity) in each case as compared to the corresponding result in (Shamsuzzoha and Skogestad, 2010), if a proper $\alpha$ is chosen. It is observed that for each process, the PI settings work well in both situations when the damping ratios $\zeta$’s are computed by the formula and read from the $M_p - \zeta$ curve. Overall when $\zeta$’s were read from the $M_p - \zeta$ curve, the PI settings are more aggressive, giving rise to faster setpoint responses with larger overshoots and faster load responses yet less deviations. This is also reflected from the larger peak sensitivities as referred to Table 1.

The closed-loop response normally changes smoothly as $\alpha$ changes. The simulations indicate that it is good to start $\alpha$ as 0.4 and then adjust it, say, at a step of 0.05 or 0.1, until a satisfactory response is attained. Typical closed-loop responses for the proposed PI tuning rule when different $\alpha$’s were applied are shown in Figure 5. The results confirm that a larger $\alpha$ leads to more aggressive response with less robustness and vice versa. In comparison, the tuning rule proposed in (Shamsuzzoha and Skogestad, 2010) has an advantage that a constant value of $\alpha$ at 0.5 is almost sufficient to give satisfactory closed-loop performance for various processes, which can clearly be seen from the PI settings in Table 1 and the closed-loop responses shown in Figure 4.

Finally, four examples are presented to validate the method proposed for choosing P controller gain for the CSR experiment. The target overshoot is set as $M_p^* = 30\%$. Hence the P controller gain $k_{c_0}$ is recommended as that in the formula (30). The formula was applied repeatedly to update $k_{c_0}$ until the overshoot of the CSR converges to 30%. The four examples are with the processes E1, E17, E21 and E24 as given in Table 1. As reported in (Shamsuzzoha and Skogestad, 2010), for processes E17 and E24, the P gains of the PI settings are almost the same in spite of the CSR’s having various overshoots; whereas for processes E1 and E21, the P gains vary significantly when CSR having different overshoots. Note that the former and latter cases correspond to the cases that
are consistent and inconsistent with the assumption of the analysis that led to the proposed formula (30). The CSR experiments were carried out with repeated application of the formula (30) and the results are shown in Figure 6. From the results, we see that in the cases of E17 and E24, both P controller gains converge quickly to the ideal ones giving target overshoots of 30%. In either case, it requires only one round of detuning $k_{c0}$ before reaching an overshoot within 25%-35%. In contrast, in the cases of E1 and E21, both P controller gains converge much more slowly but to the ideal values ultimately. It takes four and six rounds of detuning $k_{c0}$ before reaching an overshoot within 25%-35% for E1 and E21, respectively. Nevertheless, the number of rounds of detuning $k_{c0}$ remains acceptably small. These results demonstrate the effectiveness and usefulness of the proposed method in determining a proper P controller gain for the CSR experiment.

4. Conclusions

An analytical PI tuning rule was derived for ITD and FOTD processes using the CSR method. The rule expresses the PI parameters in terms of the steady-state offset, peak time, and overshoot or rise time as recorded in a CSR experiment. The rule turns out to be applicable to a broad range of processes typical for process control, and it gives comparable performance to the PI tuning rule proposed in (Shamsuzzoha and Skogestad, 2010)\(^1\) when a tuning parameter is properly chosen. Meanwhile, a method was proposed for choosing the P controller gain for the CSR experiment to result in a preferred overshoot of around 30%. The presented analysis and derived rule provide some insight and support to the PI tuning rule proposed by Shamsuzzoha and Skogestad.

Acknowledgment

Comments from S. Skogestad, M. Shamsuzzoha, and the anonymous reviewers are gratefully acknowledged.
References


Figure 1. Block diagram of feedback control system.

Figure 2. Setpoint response with P control.\textsuperscript{1}
Figure 3. $M_p - \zeta$ curve: relation between the overshoot $M_p$ and the damping ratio $\zeta$. 
Figure 4. Responses for PI control of typical processes: solid black line—Shamsuzzoha-Skogestad’ method, dotted red line—proposed method with $\zeta$ being computed by the formula, dashdot green line—proposed method with $\zeta$ being read from the $M_p$-$\zeta$ curve.
Figure 5. Effect of detuning $\alpha$: responses for PI control of $g(s) = 1/[(s + 1)(0.2s + 1)]$ (E1), with unit setpoint change at $t = 0$ and unit load disturbance at $t = 5$. 
Figure 6. Detuning process of the P controller gain $k_c$ using the proposed method. The target overshoot is 30%, the arrow directions indicate the detuning directions of $k_c$’s relative to the respective initial values.
Table 1. PI controller settings for Shamsuzzoha-Skogestad’s (Shams-Skog is used as a shorthand in the table.) and proposed rules. The PI settings and peak sensitivities in regular and bold fonts were obtained by the proposed method with \( \zeta \)'s as computed by the formula and read from the \( M_p - \zeta \) curve, respectively. For convenience, the values of the tuning factor \( F \) (as adopted in the original Shamsuzzoha-Skogestad’s rule) which corresponds to \( 1/(2\alpha) \) are also listed.

<table>
<thead>
<tr>
<th>Case</th>
<th>Process model</th>
<th>( k_{r_o} )</th>
<th>( M_p )</th>
<th>( t_c )</th>
<th>( t_p )</th>
<th>( b )</th>
<th>Method</th>
<th>( \alpha )</th>
<th>( F )</th>
<th>( k_r )</th>
<th>( \tau_s )</th>
<th>( M_s )</th>
</tr>
</thead>
<tbody>
<tr>
<td>E1</td>
<td>( 1/(s+1)(0.2s+1) )</td>
<td>15.0</td>
<td>0.327</td>
<td>0.227</td>
<td>0.373</td>
<td>0.938</td>
<td>Shams-Skog</td>
<td>0.5</td>
<td>1.0</td>
<td>8.968</td>
<td>0.910</td>
<td>1.75</td>
</tr>
<tr>
<td></td>
<td>Proposed</td>
<td>0.4</td>
<td>1.25</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>9.467</td>
<td>1.122</td>
<td>1.73</td>
</tr>
<tr>
<td>E6</td>
<td>( (0.17s+1)^2 ) ( s(s+1)^2(0.028s+1) )</td>
<td>0.80</td>
<td>0.301</td>
<td>3.187</td>
<td>5.002</td>
<td>1.000</td>
<td>Shams-Skog</td>
<td>0.5</td>
<td>1.0</td>
<td>0.496</td>
<td>12.205</td>
<td>1.77</td>
</tr>
<tr>
<td></td>
<td>Proposed</td>
<td>0.5</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.522</td>
<td>12.293</td>
<td>1.80</td>
</tr>
<tr>
<td></td>
<td>Shams-Skog</td>
<td>0.5</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.642</td>
<td>11.979</td>
<td>1.99</td>
</tr>
<tr>
<td>E8</td>
<td>( 1/s(s+1)^2 )</td>
<td>0.58</td>
<td>0.307</td>
<td>4.038</td>
<td>6.210</td>
<td>1.000</td>
<td>Shams-Skog</td>
<td>0.5</td>
<td>1.0</td>
<td>0.357</td>
<td>15.152</td>
<td>1.75</td>
</tr>
<tr>
<td></td>
<td>Proposed</td>
<td>0.5</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.338</td>
<td>15.187</td>
<td>1.71</td>
</tr>
<tr>
<td></td>
<td>Shams-Skog</td>
<td>0.5</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.463</td>
<td>14.890</td>
<td>2.01</td>
</tr>
<tr>
<td>E11</td>
<td>( (-s+1)e^{-s} ) ( (6s+1)(2s+1)^2 )</td>
<td>1.40</td>
<td>0.344</td>
<td>9.391</td>
<td>13.674</td>
<td>0.583</td>
<td>Shams-Skog</td>
<td>0.5</td>
<td>1.0</td>
<td>0.817</td>
<td>9.609</td>
<td>1.59</td>
</tr>
<tr>
<td></td>
<td>Proposed</td>
<td>0.35</td>
<td>1.43</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.585</td>
<td>7.004</td>
<td>1.49</td>
</tr>
<tr>
<td></td>
<td>Shams-Skog</td>
<td>0.5</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.765</td>
<td>9.012</td>
<td>1.56</td>
</tr>
<tr>
<td>E12</td>
<td>( (6s+1)(3s+1)e^{-0.5s} ) ( (10s+1)(8s+1)(s+1) )</td>
<td>15.0</td>
<td>0.310</td>
<td>0.609</td>
<td>0.844</td>
<td>0.938</td>
<td>Shams-Skog</td>
<td>0.5</td>
<td>1.0</td>
<td>9.191</td>
<td>2.059</td>
<td>1.74</td>
</tr>
<tr>
<td></td>
<td>Proposed</td>
<td>0.45</td>
<td>1.11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>10.125</td>
<td>2.281</td>
<td>1.86</td>
</tr>
<tr>
<td></td>
<td>Shams-Skog</td>
<td>0.5</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>10.769</td>
<td>2.250</td>
<td>1.95</td>
</tr>
<tr>
<td>E13</td>
<td>( (2s+1)e^{-s} ) ( (10s+1)(0.5s+1) )</td>
<td>4.75</td>
<td>0.302</td>
<td>1.687</td>
<td>2.183</td>
<td>0.826</td>
<td>Shams-Skog</td>
<td>0.5</td>
<td>1.0</td>
<td>2.942</td>
<td>5.327</td>
<td>1.76</td>
</tr>
<tr>
<td></td>
<td>Proposed</td>
<td>0.35</td>
<td>1.43</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3.082</td>
<td>5.332</td>
<td>1.82</td>
</tr>
<tr>
<td></td>
<td>Shams-Skog</td>
<td>0.5</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2.665</td>
<td>4.978</td>
<td>1.64</td>
</tr>
<tr>
<td>E17</td>
<td>( e^{-s}/5s+1 )</td>
<td>4.00</td>
<td>0.298</td>
<td>2.123</td>
<td>3.024</td>
<td>0.800</td>
<td>Shams-Skog</td>
<td>0.5</td>
<td>1.0</td>
<td>2.493</td>
<td>6.484</td>
<td>1.56</td>
</tr>
<tr>
<td></td>
<td>Proposed</td>
<td>0.4</td>
<td>1.25</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2.133</td>
<td>4.954</td>
<td>1.48</td>
</tr>
<tr>
<td></td>
<td>Shams-Skog</td>
<td>0.5</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2.573</td>
<td>5.819</td>
<td>1.59</td>
</tr>
<tr>
<td>E21</td>
<td>( e^{-s} )</td>
<td>0.30</td>
<td>0.300</td>
<td>1.001</td>
<td>2.001</td>
<td>0.231</td>
<td>Shams-Skog</td>
<td>0.5</td>
<td>1.0</td>
<td>0.186</td>
<td>0.321</td>
<td>1.53</td>
</tr>
<tr>
<td></td>
<td>Proposed</td>
<td>0.4</td>
<td>1.25</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.193</td>
<td>0.288</td>
<td>1.66</td>
</tr>
<tr>
<td></td>
<td>Shams-Skog</td>
<td>0.5</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>E24</td>
<td>( e^{-s}/s )</td>
<td>0.80</td>
<td>0.302</td>
<td>2.282</td>
<td>3.282</td>
<td>1.000</td>
<td>Shams-Skog</td>
<td>0.5</td>
<td>1.0</td>
<td>0.496</td>
<td>8.008</td>
<td>1.70</td>
</tr>
<tr>
<td></td>
<td>Proposed</td>
<td>0.5</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.509</td>
<td>8.064</td>
<td>1.72</td>
</tr>
<tr>
<td></td>
<td>Shams-Skog</td>
<td>0.5</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.642</td>
<td>7.861</td>
<td>2.03</td>
</tr>
<tr>
<td>E29</td>
<td>( (-s+1)e^{-2s} ) ( (s+1)^2 )</td>
<td>0.4</td>
<td>0.304</td>
<td>8.812</td>
<td>11.981</td>
<td>0.286</td>
<td>Shams-Skog</td>
<td>0.5</td>
<td>1.0</td>
<td>0.247</td>
<td>2.546</td>
<td>1.70</td>
</tr>
<tr>
<td></td>
<td>Proposed</td>
<td>0.35</td>
<td>1.43</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.225</td>
<td>2.301</td>
<td>1.72</td>
</tr>
<tr>
<td></td>
<td>Shams-Skog</td>
<td>0.5</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.224</td>
<td>2.299</td>
<td>1.72</td>
</tr>
<tr>
<td>E32</td>
<td>( (s^2+2s+9) ) ( (-2s+1)(s+1)e^{-2s} ) ( (s^2+0.5s+1)(5s+1)^2 )</td>
<td>0.12</td>
<td>0.301</td>
<td>10.623</td>
<td>15.055</td>
<td>0.519</td>
<td>Shams-Skog</td>
<td>0.5</td>
<td>1.0</td>
<td>0.074</td>
<td>8.674</td>
<td>1.55</td>
</tr>
<tr>
<td></td>
<td>Proposed</td>
<td>0.4</td>
<td>1.25</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.065</td>
<td>6.806</td>
<td>1.61</td>
</tr>
<tr>
<td></td>
<td>Shams-Skog</td>
<td>0.5</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.077</td>
<td>7.809</td>
<td>1.64</td>
</tr>
<tr>
<td>E33</td>
<td>( e^{-s}/5s-1 )</td>
<td>4.00</td>
<td>0.300</td>
<td>2.527</td>
<td>3.677</td>
<td>1.333</td>
<td>Shams-Skog</td>
<td>0.5</td>
<td>1.0</td>
<td>2.487</td>
<td>7.866</td>
<td>2.33</td>
</tr>
<tr>
<td></td>
<td>Proposed</td>
<td>0.6</td>
<td>0.83</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2.878</td>
<td>5.402</td>
<td>2.92</td>
</tr>
<tr>
<td></td>
<td>Shams-Skog</td>
<td>0.5</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3.855</td>
<td>7.070</td>
<td>3.24</td>
</tr>
</tbody>
</table>

* For a pure time delay process, the analytical \( \zeta \) is zero and hence invalid. For this case, \( \zeta \) has to be read from the \( M_p - \zeta \) curve.