Analytical conversion of the Strejc model to the first order with time delay (FOTD) model

Anna Czemplik

Agenda

1. Context – control design methods
2. Definition of problem
3. Known accurate solutions
4. My proposal – method of simplification
5. Simplified formulas
6. Example of application

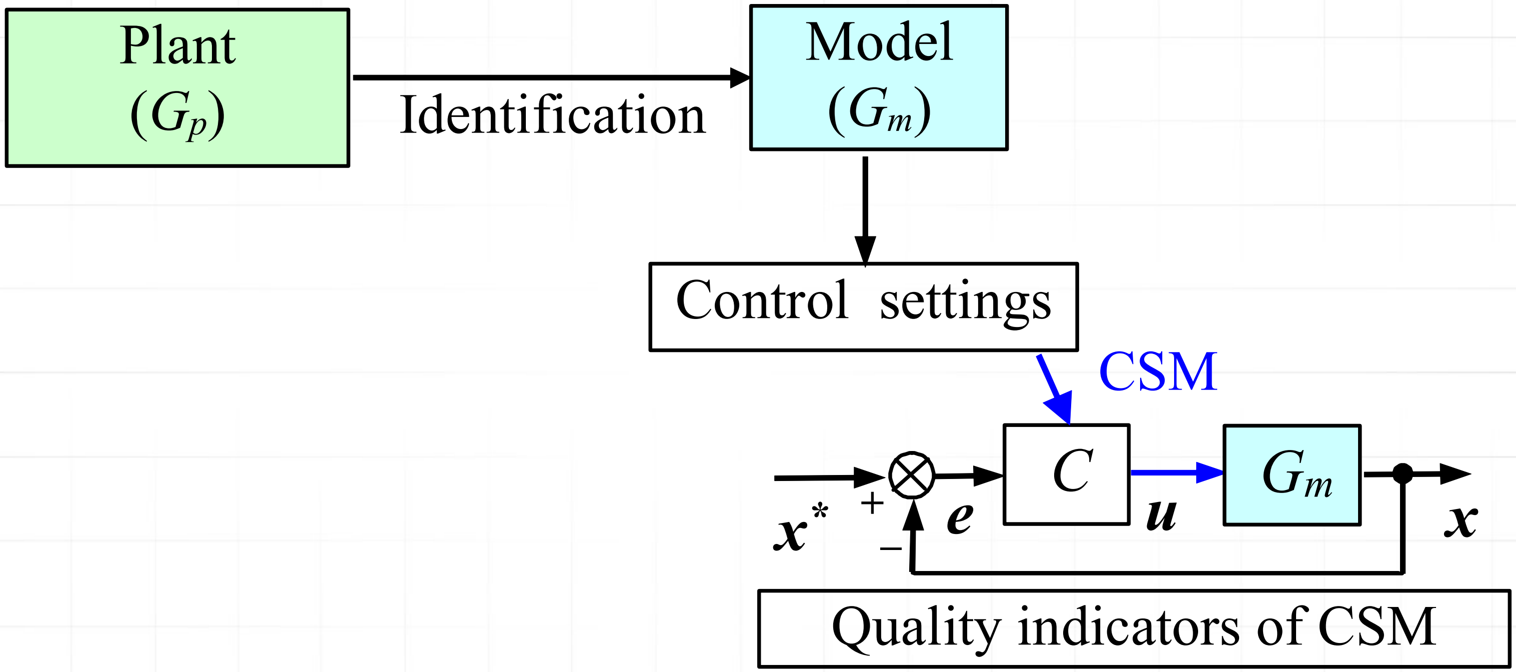
Conclusion

1.1. Control design methods

Control design method		
Model G_m (form, identification)	Formulas for control settings	Predicted quality indicators

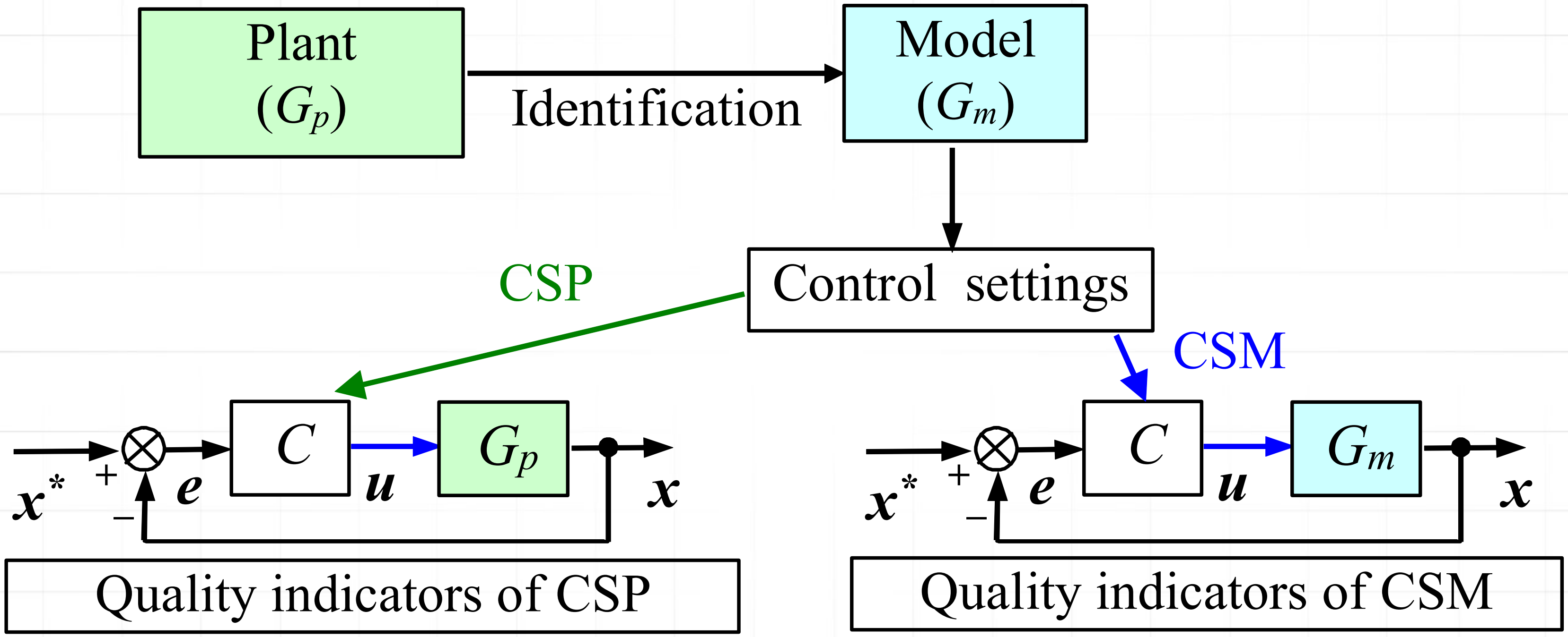
Ziegler-Nichols tuning method		
Model G_m	Formulas	Quality indicators
$\dfrac{k}{T_m s+1} e^{-sT_0}$, tangent	$K_p = 0.9\ T_m \,/\, (k \cdot T_0)$ $T_i = 3.33\ T_0$	Quarter-Decay Ratio (QDR)

1.2. Quality of CSM and CSP



Control System on the Model

1.2. Quality of CSM and CSP



Control System on the Plant

Control System on the Model

1.3. Forms of models

Aidan O'Dwyer, Handbook of PI and PID controller tuning rules, Imperial College Press 2009

Table 1. Models used in the design of control systems

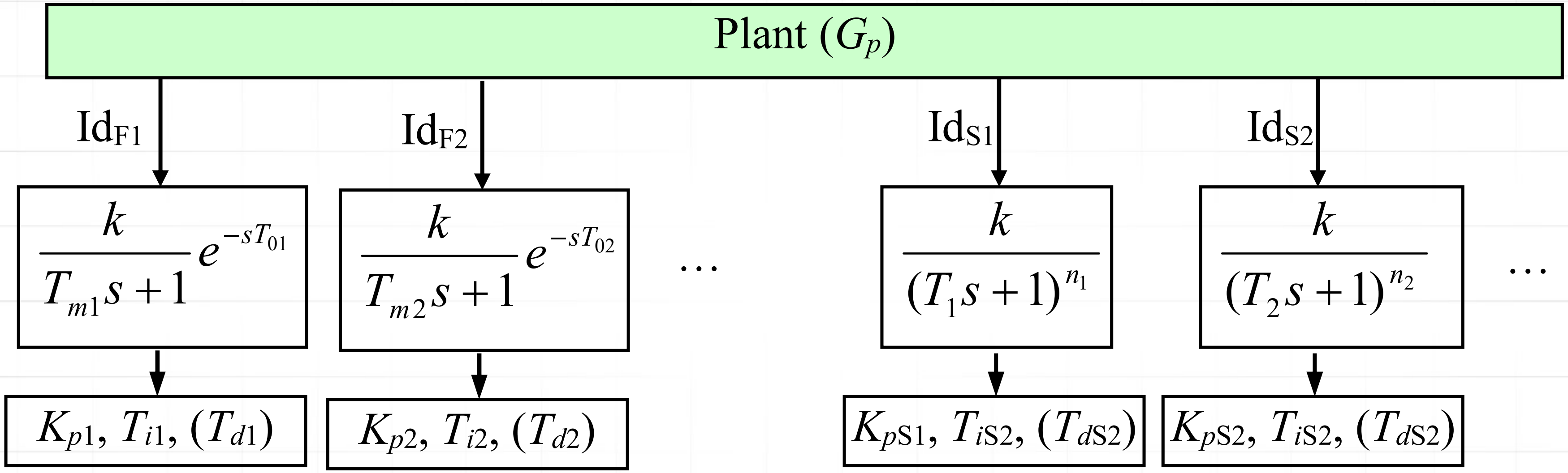
Model type	FOTD	SOTD	Other stable models	Non-model specific	Models with an integrator	Unstable	Total
Number of methods	649	291	103	169	339	182	1731

FOTD $$G_m(s) = \frac{k}{T_m s + 1} e^{-sT_0}$$

$$\left(\text{or} \quad G_m(s) = k \frac{T_z s + 1}{T_m s + 1} e^{-sT_0} \right)$$

Strejc model $$G_m(s) = \frac{k}{(Ts+1)^n}$$

1.4. Identification of models

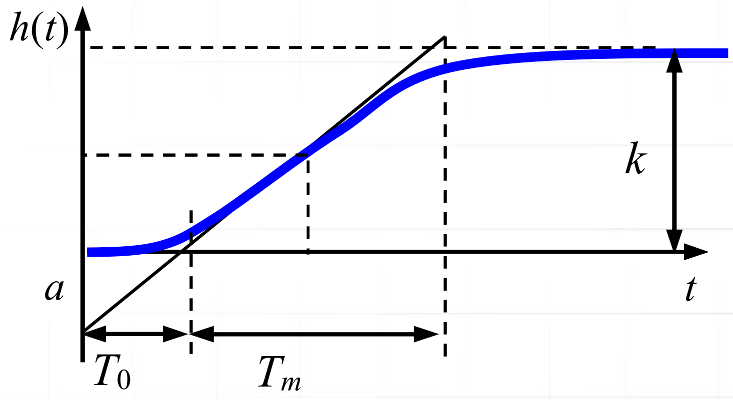


1.4. Identification of models

FOTD model

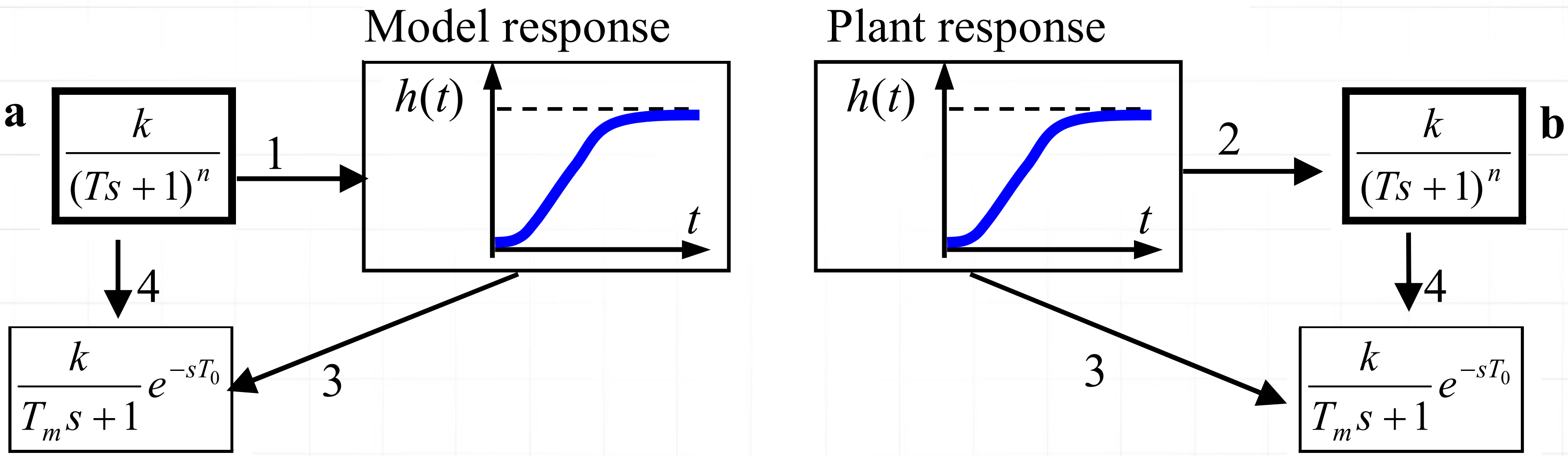
$$G_m(s) = \frac{k}{T_m s + 1} e^{-sT_0}$$

Tangent method

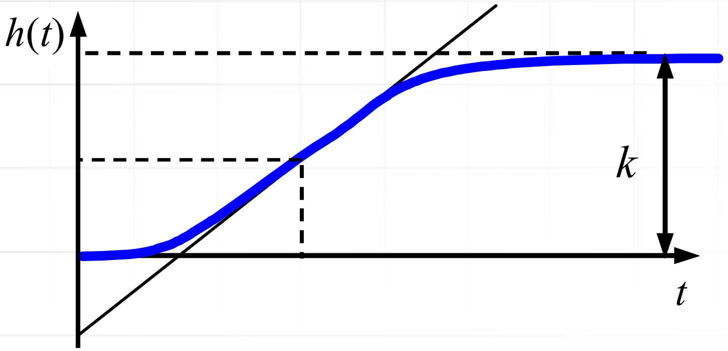


Analytical calculation of FOTD parameters

Strejc model $$G_m(s) = \frac{k}{(Ts+1)^n}$$



3.1. Step response of the Strejc model

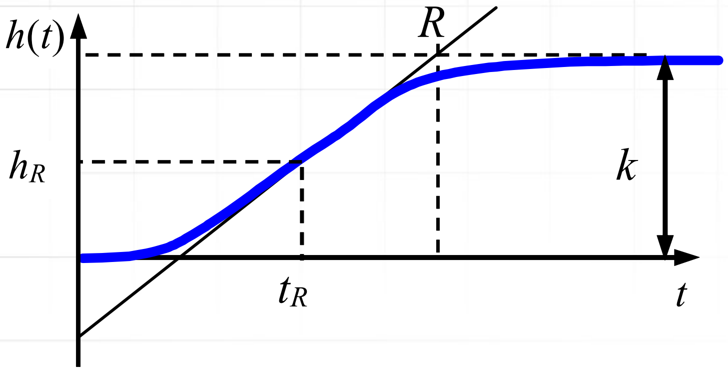


$$h(t) = k\left(1 - \sum_{i=1}^{n} \frac{1}{(i-1)!T^{i-1}} t^{i-1} e^{-t/T}\right)$$

$$g(t) = \dot{h}(t) = \frac{k}{(n-1)!T^{n}} t^{n-1} e^{-\sigma t}$$

(the derivative function of step response)

3.2. Parameters of step response



Inflection point:

$$t_R = (n-1)T$$

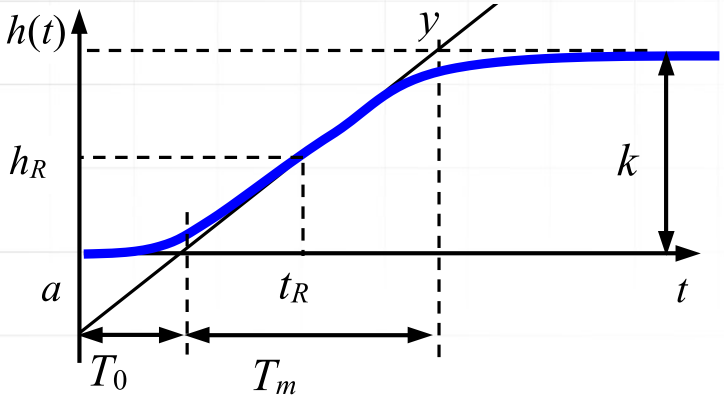
$$h_R = h(t_R) = k\left(1 - \frac{1}{e^{(n-1)}}\sum_{i=1}^{n}\frac{(n-1)^{i-1}}{(i-1)!}\right)$$

Tangent line:

$$y = R \cdot t - a$$

$$R = g(t_R) = k\frac{(n-1)^{n-1}}{(n-1)!\,e^{n-1}T}$$

3.2. Parameters of step response



Inflection point:

$$t_R = (n-1)T$$

$$h_R = h(t_R) = k\left(1 - \frac{1}{e^{(n-1)}}\sum_{i=1}^{n}\frac{(n-1)^{i-1}}{(i-1)!}\right)$$

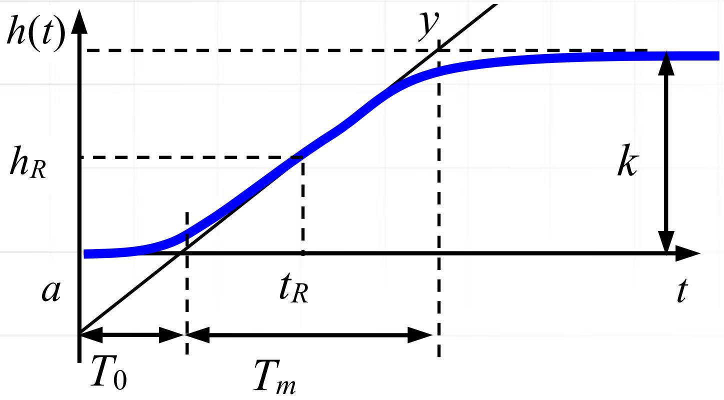
Tangent line:

$$y = R \cdot t - a$$

$$R = g(t_R) = k\frac{(n-1)^{n-1}}{(n-1)!\,e^{n-1}T}$$

$$R = \frac{k}{T_m} = \frac{a}{T_0} = \frac{h_R}{t_R - T_0}$$

3.2. Parameters of step response



Inflection point:

$$t_R = (n-1)T$$

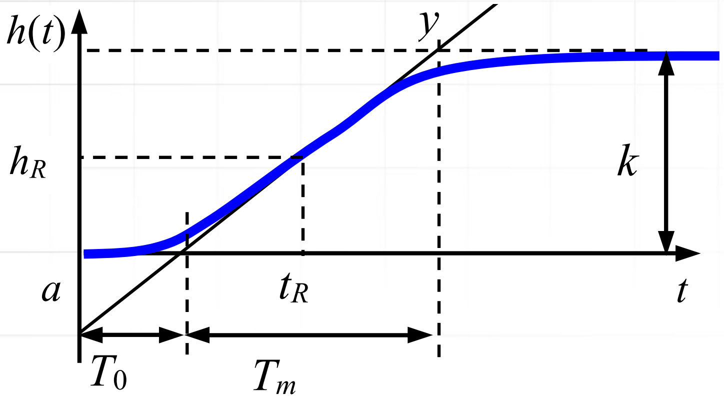
$$h_R = h(t_R) = k\left(1 - \frac{1}{e^{(n-1)}}\sum_{i=1}^{n}\frac{(n-1)^{i-1}}{(i-1)!}\right)$$

T_m and T_0 parameters in the step response of the Strejc model

$$T_m = \frac{k}{R} = k_{Tm} T \text{ , where: } \quad k_{Tm} = \frac{(n-1)!\, e^{n-1}}{(n-1)^{n-1}}$$

$$T_0 = t_R - \frac{h_R}{R} = k_{T0} T \text{ , where: } \quad k_{T0} = \frac{(n-1)^n - (n-1)!\, e^{n-1} + (n-1)!\, S_{n-1}}{(n-1)^{n-1}}, \quad S_{n-1} = \sum_{i=1}^{n}\frac{(n-1)^{i-1}}{(i-1)!}$$

4.1. Simplification

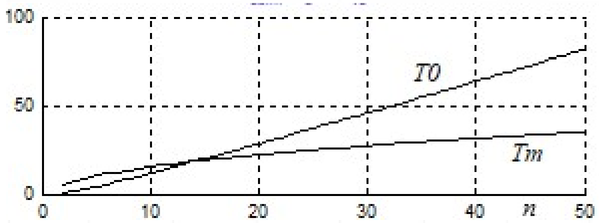


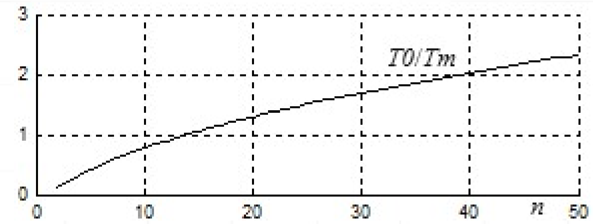
Accurate formulas:

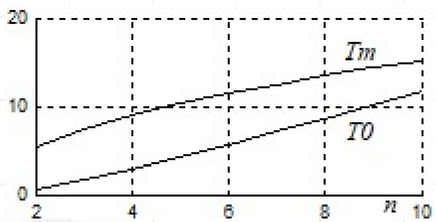
$$T_m = k_{Tm} T = \frac{(n-1)!\, e^{n-1}}{(n-1)^{n-1}} T$$

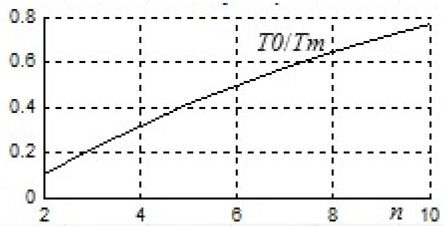
$$T_0 = k_{T0} T = \frac{(n-1)^n - (n-1)! e^{n-1} + (n-1)! S_{n-1}}{(n-1)^{n-1}} T,$$

$$S_{n-1} = \sum_{i=1}^{n} \frac{(n-1)^{i-1}}{(i-1)!}$$







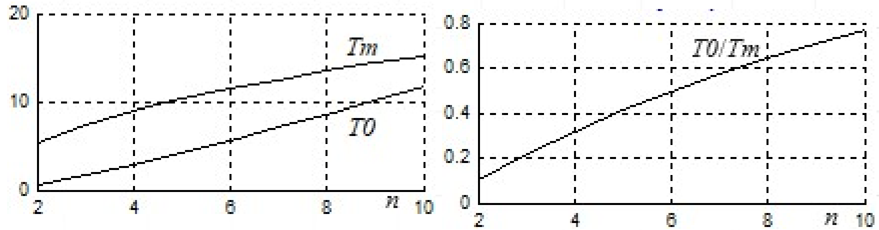


Assumptions:
- simplify formulas
- accuracy for small n

4.2. Simplification methods

1) Analytical formulas that simplify expressions

2) Optimization with simple functions



3) Classical interpolation of functions in the assumed range of n values

$$\begin{bmatrix} F_0(n_0) & F_1(n_0) & .. & F_n(n_0) \\ F_0(n_1) & F_1(n_1) & \dots & F_n(n_1) \\ \dots & \dots & & \dots \\ F_0(n_k) & F_1(n_k) & \dots & F_k(n_k) \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \dots \\ a_k \end{bmatrix} = \begin{bmatrix} f(n_0) \\ f(n_1) \\ \dots \\ f(n_k) \end{bmatrix}$$

$$\mathbf{VA} = \mathbf{f} \longrightarrow \mathbf{A} = \mathbf{V}^{-1}\mathbf{f}$$

$(n_0, n_1, \dots, n_k)$ – nodes of interpolation

$\mathbf{B}(n) = [F_0(n), F_1(n), \dots F_k(n)]$ – base of functions

$\mathbf{A} = [a_0, a_1, \dots, a_n]^T$ – vector of coefficients

$a_0F_0(n_i) + a_1F_1(n_i) + \dots + a_nF_n(n_i)$ – interpolating polynomial

$f(n_i)$ – value from accurate formulas

4) Approximation in the assumed range of n values

Levenberg-Marquardt algorytm (LMA) (Mathematica)

$(n_0, n_1, \dots, n_k)$ – nodes of interpolation

$\mathbf{B}(n) = [F_0(n), F_1(n), \dots F_k(n)]$ – base of function

4.3. Formulas for calculating T_m and T_0 on the base T and n

Various variants of functions and nodes were examined.

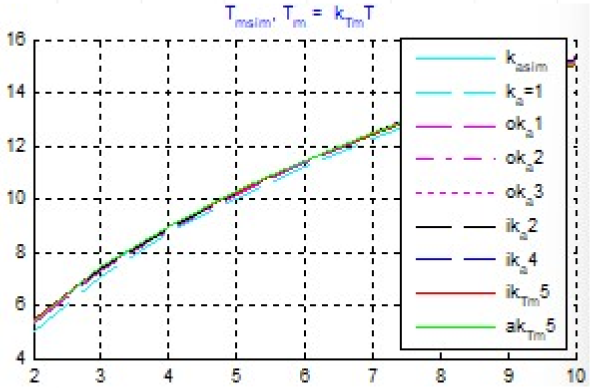
The three sets of formulas were selected, hereafter referred to as:

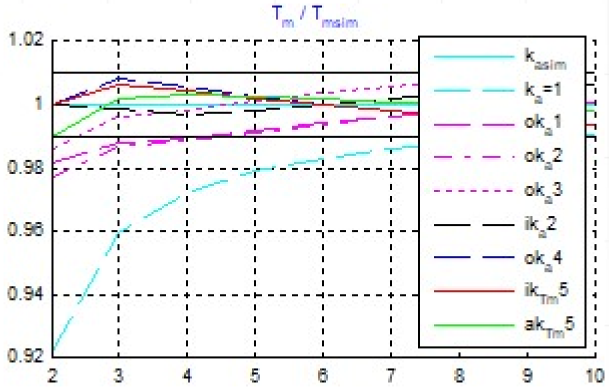
- **easy** – solutions obtained in the simplest way (substitution of analytical formulas)
- **use** – solutions proposed for use in design methods
- **best** – solutions with a wider range of application

5.1. Time constant $T_m = k_{Tm}T$

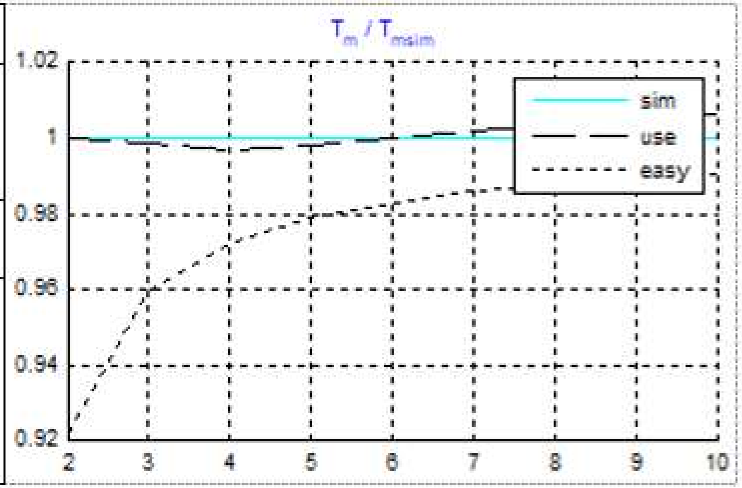
Symbol	Formula	Methods of simplify
1) k_{asim}	$k_{Tm} = \frac{(n-1)!e^{n-1}}{(n-1)^{n-1}}$	Accurate formula
2) k_a=1 **easy**	$k_{Tm} = \sqrt{2\pi(n-1)}$	Stirling formula $n! \approx \sqrt{2\pi n} \cdot n^n / e^n$
3) ok_a1	$k_{Tm} = 1.01\left(1 + \frac{e^{-n}}{\sqrt{2\pi}}\right) \cdot \sqrt{2\pi(n-1)}$	Optimization k_a ($k_{Tm} = k_a \cdot \sqrt{2\pi(n-1)}$)
4) ok_a2	$k_{Tm} = 1.01\left(1 + \frac{e^{-n}}{e}\right) \cdot \sqrt{2\pi(n-1)}$	Optimization k_a ($k_{Tm} = k_a \cdot \sqrt{2\pi(n-1)}$)
5) ok_a3	$k_{Tm} = \left(1.02 + e^{-(n+1)}\right) \cdot \sqrt{2\pi(n-1)}$	Optimization k_a ($k_{Tm} = k_a \cdot \sqrt{2\pi(n-1)}$)
6) ik_a2 **use**	$k_{Tm} = \left(a_0 + a_1 e^{-n}\right) \cdot \sqrt{2\pi(n-1)}$	Interpolation k_a ($n = [2,6]$), $a \approx [1.016, 0.509]$
7) ik_a4	$k_{Tm} = (a_0 + a_1 / n) \cdot \sqrt{2\pi(n-1)}$	Interpolation k_a ($n = [2,6]$), $a \approx [0.983, 0.203]$
8) $ik_{Tm}5$	$k_{Tm} = a_0 + a_1\sqrt{n-1}$	Interpolation k_{Tm} ($n = [2,6]$), $a \approx [0.307, 2.412]$
9) $ak_{Tm}5$	$k_{Tm} = a_0 + a_1\sqrt{n-1}$	Aproximation k_{Tm} ($n = 2 \div 10$), $a \approx [0.249 ,2.442]$

5.1. Time constant $T_m = k_{Tm} T$ (chosen)





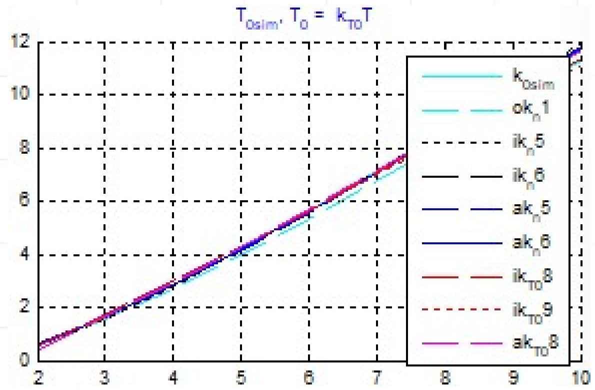
Symbol	Formula
sim (k_{asim})	$k_{Tm} = \frac{(n-1)!\,e^{n-1}}{(n-1)^{n-1}}$
easy (k_a=1)	$k_{Tm} = \sqrt{2\pi(n-1)}$
use (ik_a2)	$k_{Tm} = \left(a_0 + a_1 e^{-n}\right)\cdot\sqrt{2\pi(n-1)}$ $a \approx [1.016,\ 0.509]$

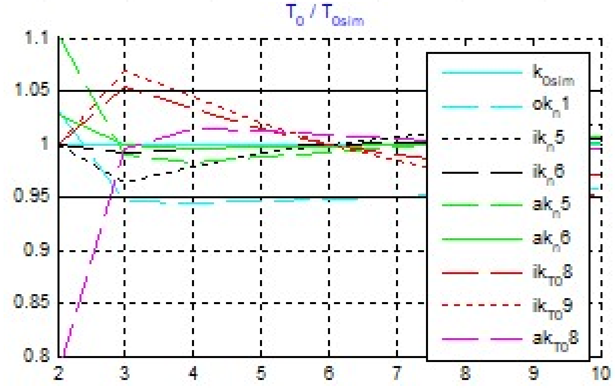


5.2a. Transport delay $T_0 = k_{T0}T$

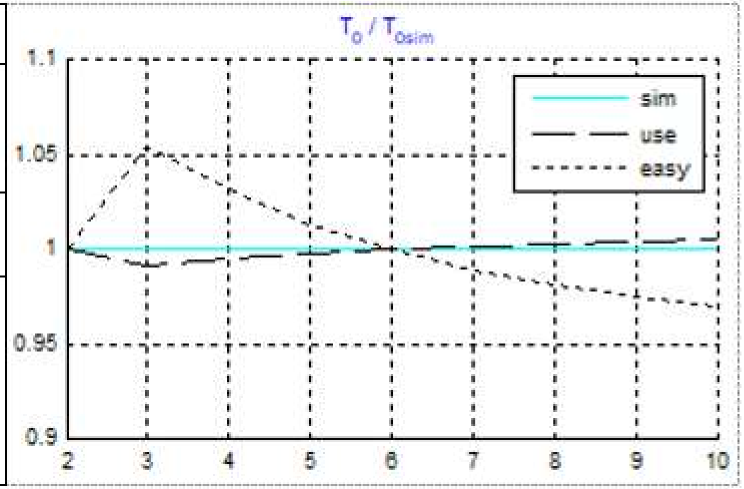
Symbol	Formula	Methods of simplify
1) k_{0sim}	$k_{T0} = \dfrac{(n-1)^n - (n-1)!e^{n-1} + (n-1)!S_{n-1}}{(n-1)^{n-1}}$	Accurate formula, $S_{n-1} = \sum_{i=1}^{n} \dfrac{(n-1)^{i-1}}{(i-1)!}$
	$k_{T0} = \dfrac{(n-1)(n-k_n)}{n}$	Maclaurin series: $S_{n-1} \approx e^{n-1} - k_n R_n$ Lagrange rest: $R_n = k_n \dfrac{(n-1)^n}{n!}$
2) ok_n1	$k_n = 1.25(1+e^{-n})\sqrt{n-1}$	Optimization k_n
3) ik_n5	$k_n = a_0 + a_1\sqrt{n-1}$	Interpolation k_n, $n = [2,6]$, $a \approx [0.474, 0.963]$
4) ik_n6 **use**	$k_n = a_0 + a_1\sqrt{n}$	Interpolation k_n, $n = [2,6]$, $a \approx [-0.189, 1.149]$
5) ak_n5	$k_n = a_0 + a_1\sqrt{n-1}$	Aproximation k_n, $n = 2 \div 10$, $a \approx [0.345, 1.032]$
6) ak_n6	$k_n = a_0 + a_1\sqrt{n}$	Aproximation k_n, $n = 2 \div 10$, $a \approx [-0.236, 1.172]$
7) $ik_{T0}8$ **easy**	$k_{T0} = a_0 n + a_1\sqrt{n}$	Interpolation k_{T0}, $n = [2,6]$, $a \approx [0.916, -1.096]$
8) $ik_{T0}89$	$k_{T0} = a_0(n-1) + a_1\sqrt{n-1}$	Interpolation k_{T0}, $n = [2,6]$, $a \approx [0.789, -0.507]$
9) $ak_{T0}8$	$k_{T0} = a_0 n + a_1\sqrt{n}$	Aproximation k_{T0}, $n = 2 \div 10$, $a \approx [0.968, -1.213]$

5.2a. Transport delay $T_0 = k_{T0}\, T$ (chosen)



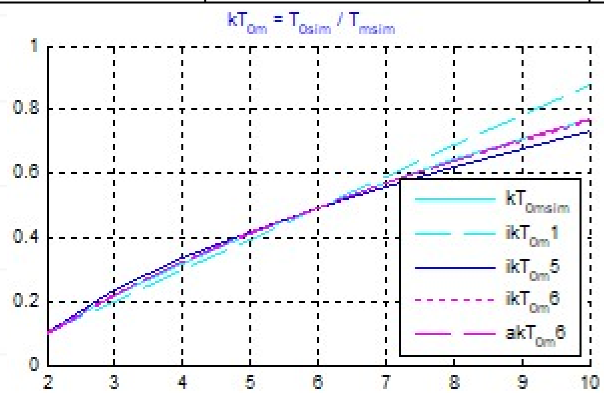


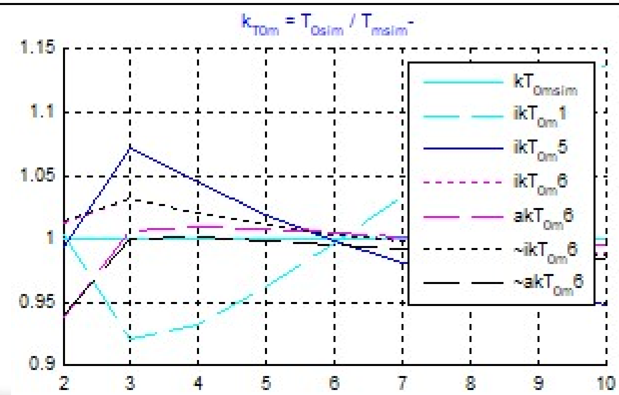
Symbol	Formula
sim (k_{0sim})	$k_{T0} = \dfrac{(n-1)^n - (n-1)!\,e^{n-1} + (n-1)!\,S_{n-1}}{(n-1)^{n-1}}$
easy ($ik_{T0}8$)	$k_{T0} = a_0 n + a_1\sqrt{n}$, $a \approx [0.916, -1.096]$
use (ik_n6)	$k_{T0} = \dfrac{(n-1)(n-k_n)}{n}, k_n = a_0 + a_1\sqrt{n}$ $a \approx [-0.189, 1.149]$



5.2b. Ratio T_0 / T_m

Symbol	Formula	Methods of simplify
1) kT_{0msim}		Accurate formulas
2) $ikT_{0m}1$	$k_{T0m} = a_0 + a_1 n$	Interpolation k_{T0m}, $n = [2,6]$, $a \approx [-0.091, 0.097]$
3) $ikT_{0m}5$	$k_{T0m} = a_0 + a_1\sqrt{n-1}$	Interpolation k_{T0m}, $n = [2,6]$, $a \approx [-0.212, 0.315]$
4) $ikT_{0m}6$	$k_{T0m} = a_0 + a_1\sqrt{n}$	Interpolation k_{T0m}, $n = [2,6]$, $a \approx -0.429, 0.376]$
5) $aikT_{0m}6$	$k_{T0m} = a_0 + a_1\sqrt{n}$	Aproximation k_{T0m}, $n = 2 \div 10$, $a \approx [-0.447, 0.385]$
6) $\sim ikT_{0m}6$ **best**	$k_{T0m} = a_0 + a_1\sqrt{n}$	Interpolation k_{T0m}, $n = [2,6]$, $a \approx [-0.428, 0.377]$
6) $\sim akT_{0m}6$	$k_{T0m} = a_0 + a_1\sqrt{n}$	Aproximation k_{T0m}, $n = 2 \div 10$, $a \approx [-0.44, 0.38]$

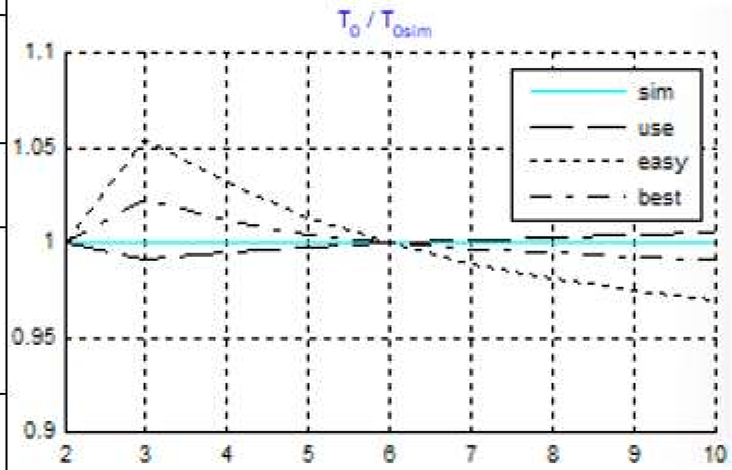




5.2. Transport delay T_0 (chosen)

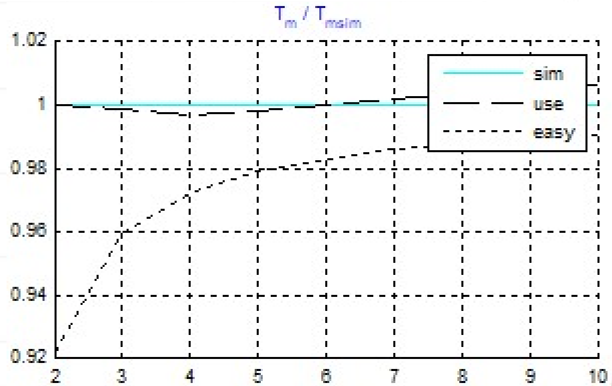
$$T_0 = k_{T0}\,T \;\text{ or }\; T_0 = k_{T0m}\,T_m$$

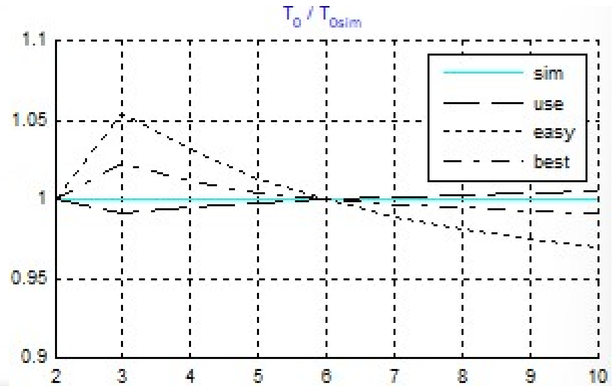
Symbol	Formula
sim (k_{0sim})	$k_{T0} = \dfrac{(n-1)^n - (n-1)!\,e^{n-1} + (n-1)!\,S_{n-1}}{(n-1)^{n-1}}$
easy ($ik_{T0}8$)	$k_{T0} = a_0 n + a_1\sqrt{n}$, $a \approx [0.916, -1.096]$
use (ik_n6)	$k_{T0} = \dfrac{(n-1)(n-k_n)}{n}, k_n = a_0 + a_1\sqrt{n}$ $a \approx [-0.189, 1.149]$
best ($ikT_{0m}6$)	$k_{T0m} = a_0 + a_1\sqrt{n}$, $a \approx [-0.428, 0.377]$ plus T_m (use)



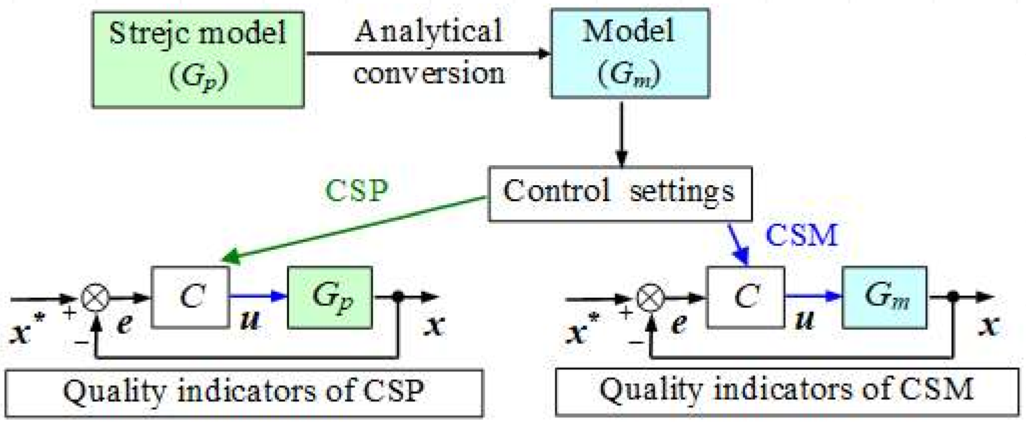
5.3. Conversion $(T, n) \rightarrow (T_0, T_m)$

Set	$T_m = k_{Tm} T$	$T_0 = k_{T0} T$ lub $T_0 = k_{T0m} T_m$
sim	$k_{Tm} = \dfrac{(n-1)!e^{n-1}}{(n-1)^{n-1}}$	$k_{T0} = \dfrac{(n-1)^n - (n-1)!e^{n-1} + (n-1)!S_{n-1}}{(n-1)^{n-1}}$, $S_{n-1} = \sum_{i=1}^{n} \dfrac{(n-1)^{i-1}}{(i-1)!}$
easy - - - -	$k_{Tm} = \sqrt{2\pi(n-1)}$	$k_{T0} = a_0 n + a_1\sqrt{n}$ $a \approx [0.916, -1.096]$
use — —	$k_{Tm} = (a_0 + a_1 e^{-n}) \cdot \sqrt{2\pi(n-1)}$ $a \approx [1.016, 0.509]$	$k_{T0} = \dfrac{(n-1)(n-k_n)}{n}$, $k_n = a_0 + a_1\sqrt{n}$ $a \approx [-0.189, 1.149]$
best -·-·-	$k_{Tm} = (a_0 + a_1 e^{-n}) \cdot \sqrt{2\pi(n-1)}$ $a \approx [1.016, 0.509]$	$k_{T0m} = a_0 + a_1\sqrt{n}$, $a \approx [-0.428, 0.377]$





6.1. Control project (PID tunning)



Ziegler-Nichols tuning method

Model G_m

FOTD, tangent method

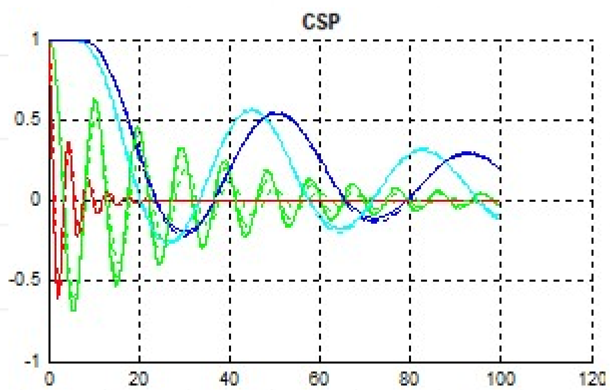
Formulas

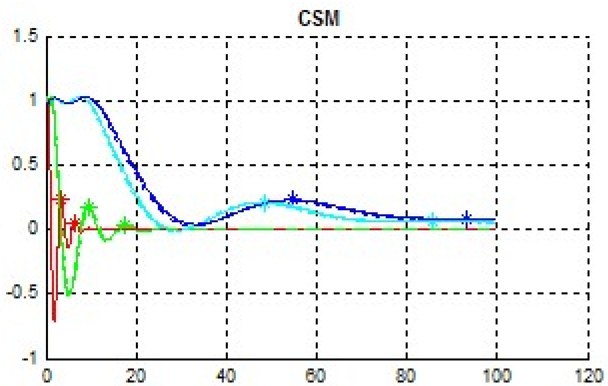
$K_p = 0.9\ T_m\ /\ (k \cdot T_0)$

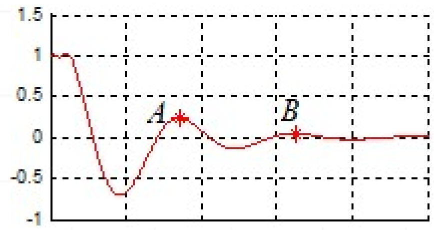
$T_i = 3.33\ T_0$

Quality indicators

Quarter-Decay Ratio (QDR)



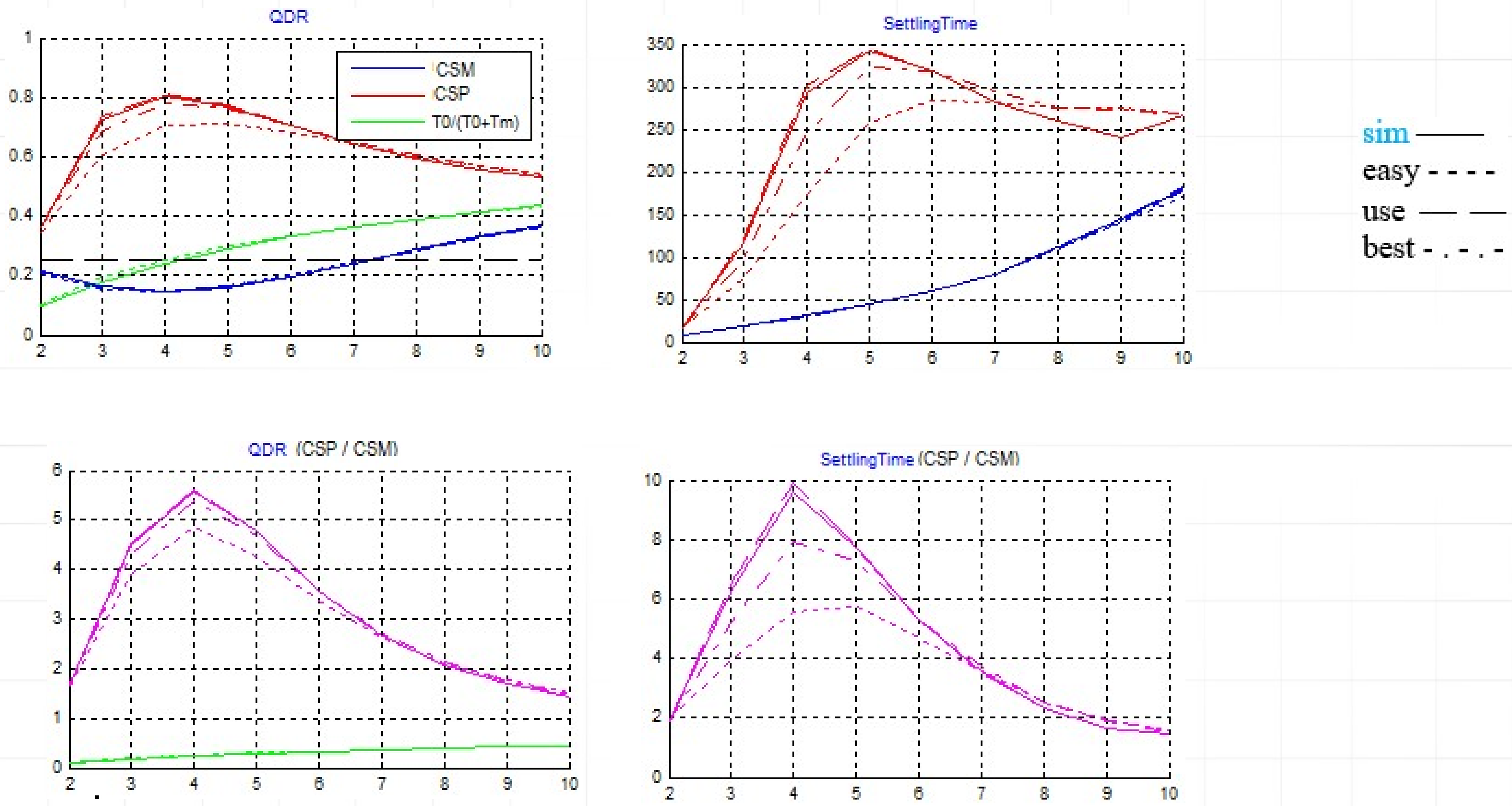




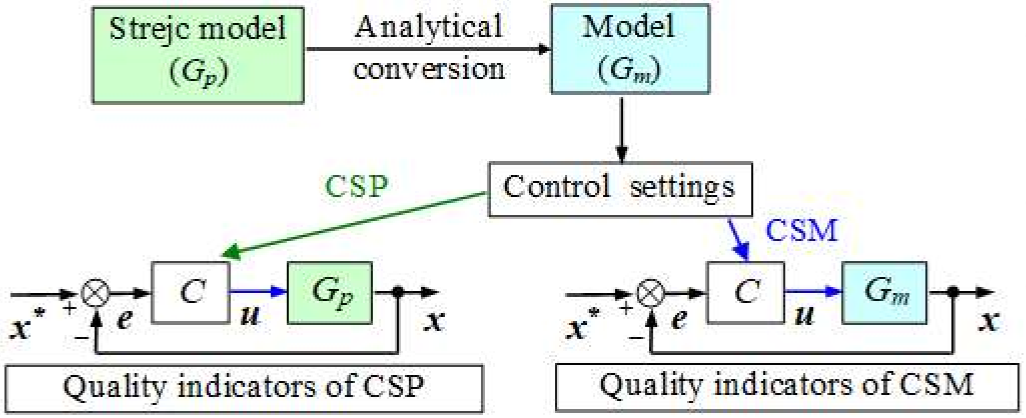
QDR: $B/A < 0.25$

(n: 2 3 ... 9 10)

6.1. Control project (PID tunning)



6.2. Control project (PID tunning)

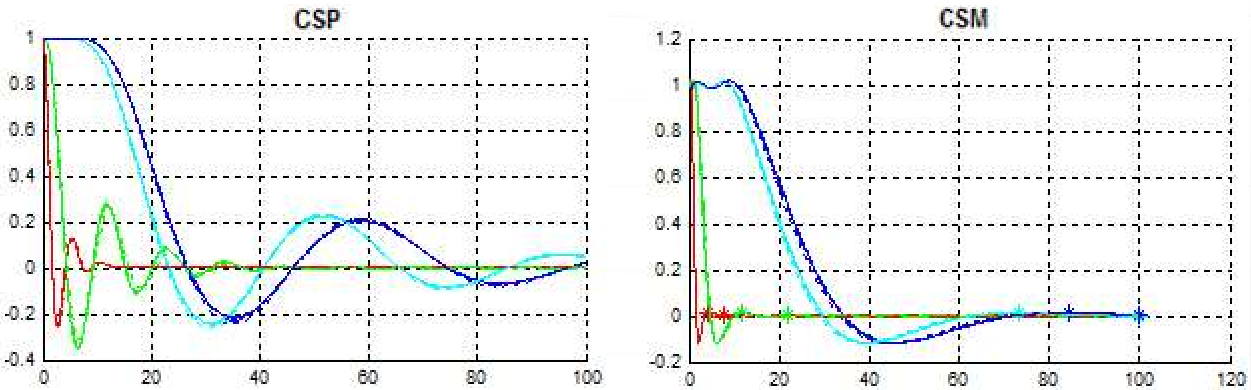


CHR tuning method

Model G_m
FOTD, tangent method

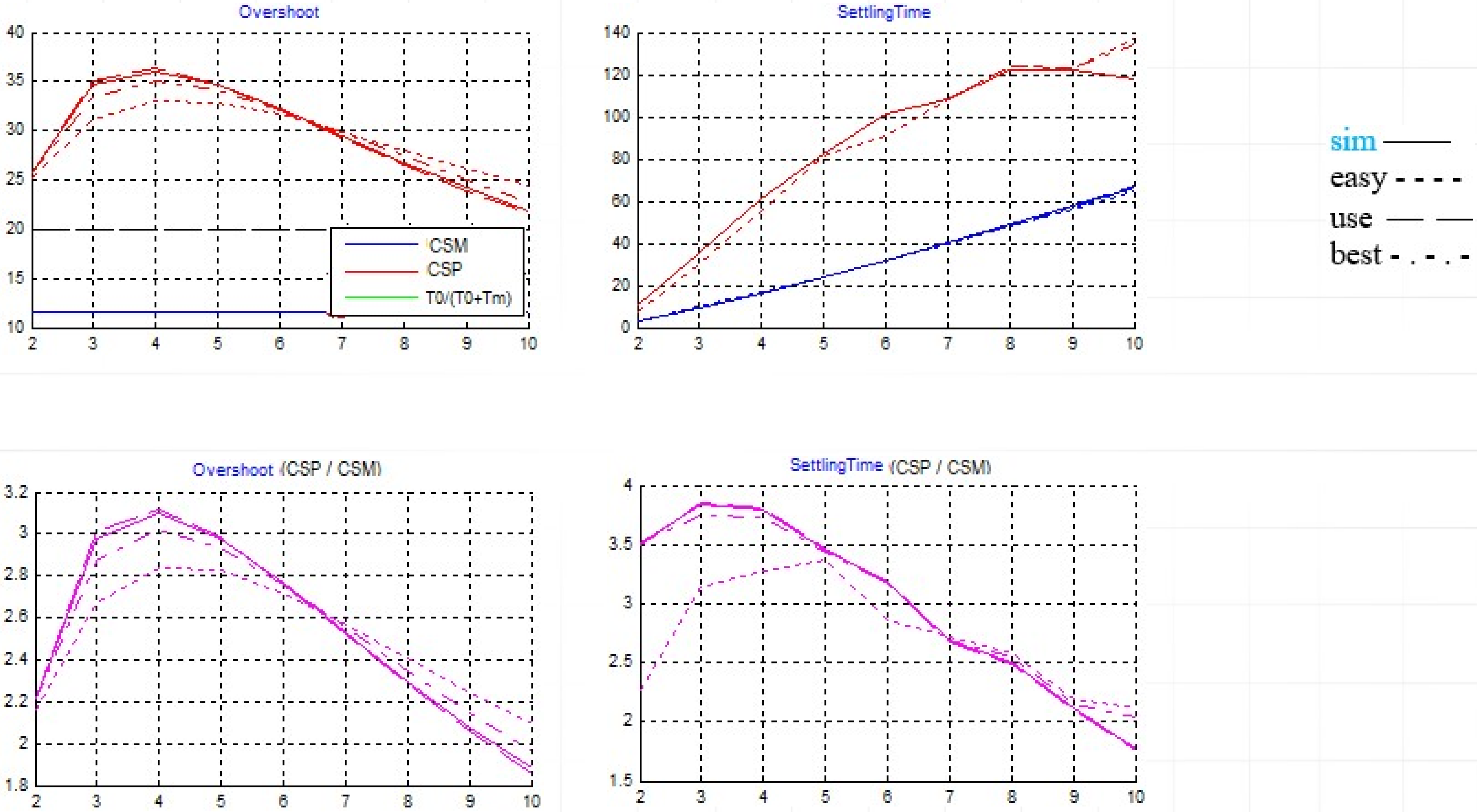
Formulas
$K_p = 0.6\, T_m / (k \cdot T_0)$
$T_i = T_0$

Quality indicators
Overshoot 20%

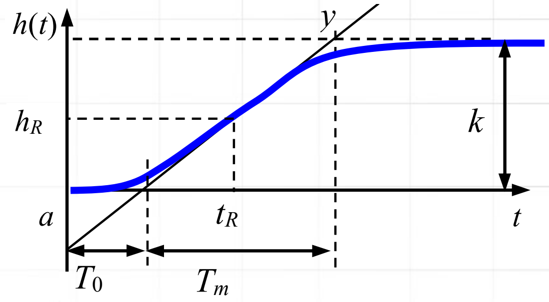


(n: 2 3 ... 9 10)

6.2. Control project (PID tunning)



6.3. Identification of Strejc model



$$G_m(s) = \frac{k}{(Ts+1)^n}$$

Id1) Measurement of T_0 and T_m, and table

n	2	3	4	5	6	7	8	9	10
T_0 / T_m	0.104	0.218	0.319	0.410	0.493	0.570	0.642	0.709	0.773
t_R / T	1	2	3	4	5	6	7	8	9
h_R	0.264	0.323	0.353	0.371	0.384	0.394	0.401	0.407	0.413

Id2) Measurement time t_R and slope R, and the Stirling formula: $n = 1 + \frac{2\pi R^2 t_R^2}{k^2}$ $\quad T = \frac{t_R}{n-1}$

Id3) Measurement of T_0 and T_m, and simplified formulas

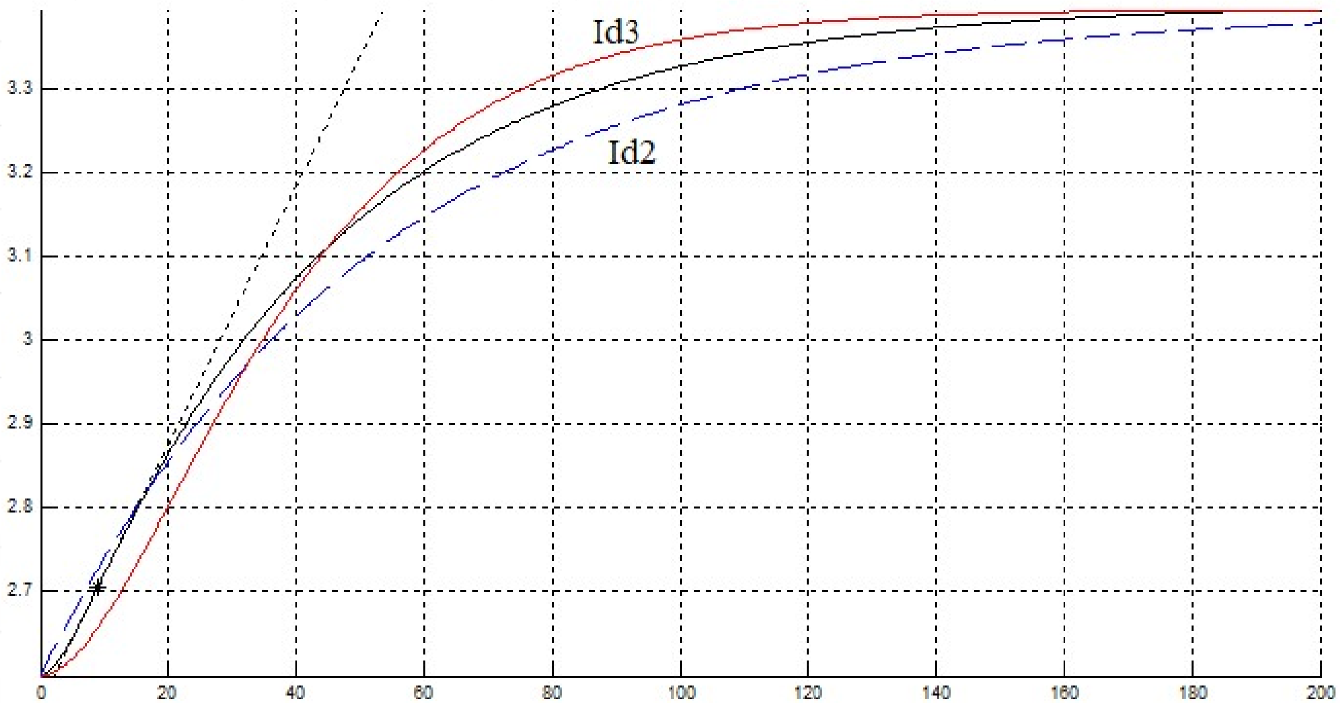
	$T_m = k_{Tm} T$	$T_0 = k_{T0m} T_m$
best - . - . -	$k_{Tm} = \left(a_0 + a_1 e^{-n}\right) \cdot \sqrt{2\pi(n-1)}$ $a \approx [1.016, 0.509]$	$k_{T0m} = c_0 + c_1 \sqrt{n}$, $c \approx [-0.428, 0.377]$

$$\frac{T_0}{T_m} = c_0 + c_1 \sqrt{n} \longrightarrow n = \left(\frac{T_0 - c_0 T_m}{c_1 T_m}\right)^2$$

$$T = \frac{T_m}{\sqrt{2\pi(n-1)}}$$

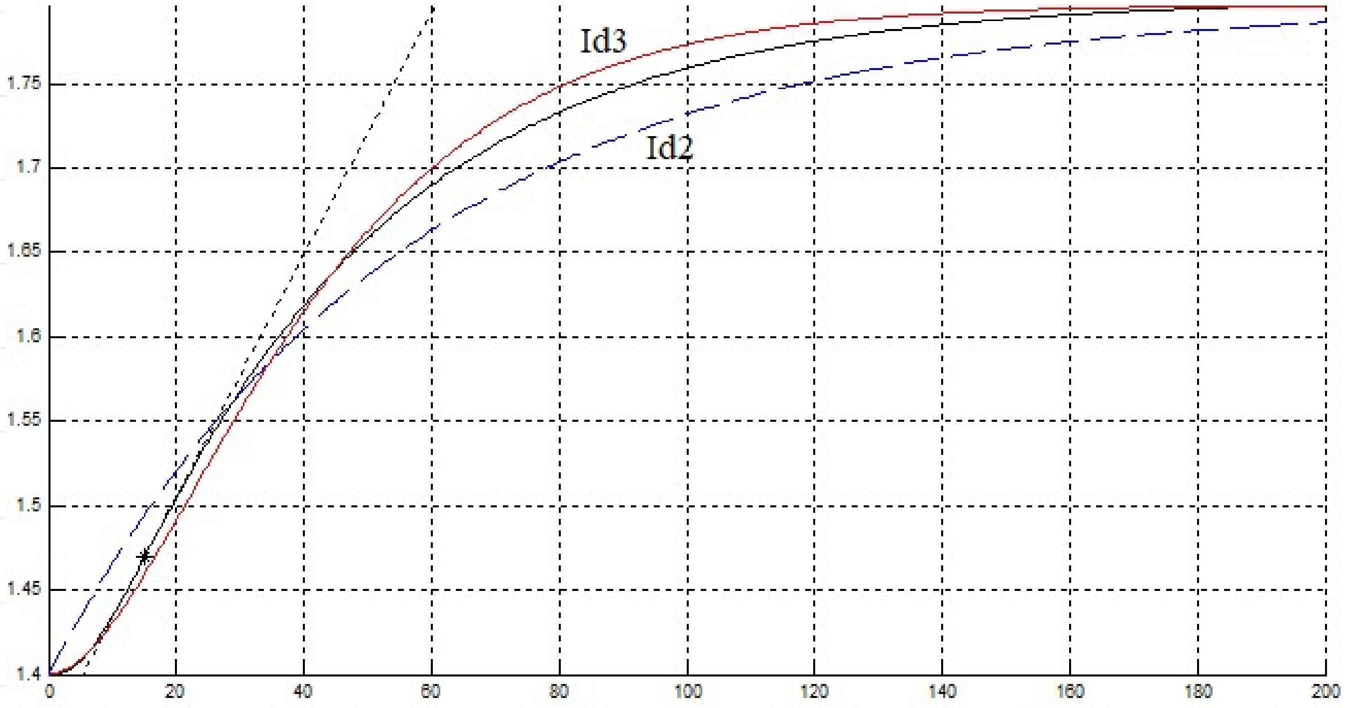
6.3. Identification of Strejc model

$$G_{31} = \frac{7.93(4s+1)}{(40.39s+1)(5.14s+1)(2.26s+1)}$$



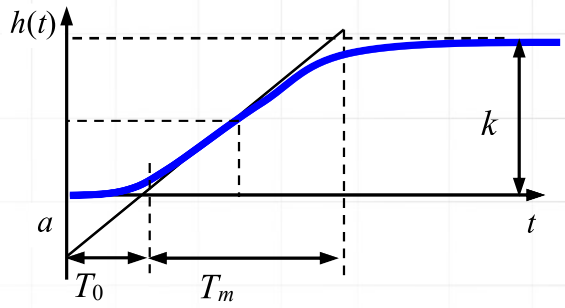
6.3. Identification of Strejc model

$$G_{30} = \frac{3.97}{(40.39s+1)(5.14s+1)(2.26s+1)}$$



6.3. Identification of Strejc model

$$G_m(s) = \frac{k}{(Ts+1)^n}$$



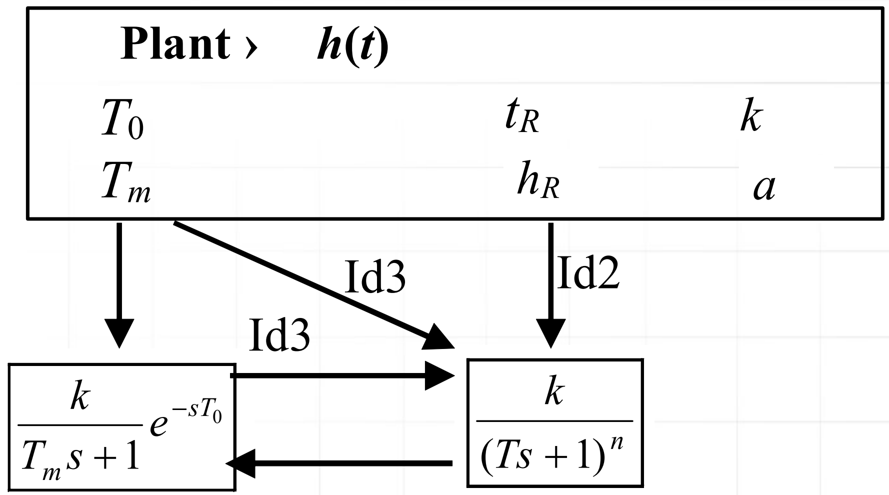
Id1) T_0 and T_m, and table

Id2) Time t_R and slope R, and the Stirling formula:

$$n = 1 + \frac{2\pi R^2 t_R^2}{k^2} \qquad T = \frac{t_R}{n-1}$$

Id3) T_0 and T_m, and the simplified formula:

$$n = \left(\frac{T_0 - c_0 T_m}{c_1 T_m}\right)^2 \qquad T = \frac{T_m}{\sqrt{2\pi(n-1)}}$$



Conclusion

We can perform analytical conversion of Strejc model to FOTD model

Set	$T_m = k_{Tm}T$	$T_0 = k_{T0}T$ lub $T_0 = k_{T0m}T_m$	
sim	$k_{Tm} = \frac{(n-1)!e^{n-1}}{(n-1)^{n-1}}$	$k_{T0} = \frac{(n-1)^n - (n-1)!e^{n-1} + (n-1)!S_{n-1}}{(n-1)^{n-1}}$, $S_{n-1} = \sum_{i=1}^{n} \frac{(n-1)^{i-1}}{(i-1)!}$	accurate
easy	$k_{Tm} = \sqrt{2\pi(n-1)}$	$k_{T0} = a_0 n + a_1\sqrt{n}$ $a \approx [0.916, -1.096]$	simplified
use	$k_{Tm} = \left(a_0 + a_1 e^{-n}\right)\cdot\sqrt{2\pi(n-1)}$ $a \approx [1.016, 0.509]$	$k_{T0} = \frac{(n-1)(n-k_n)}{n}$, $k_n = a_0 + a_1\sqrt{n}$ $a \approx [-0.189, 1.149]$	simplified
best	$k_{Tm} = \left(a_0 + a_1 e^{-n}\right)\cdot\sqrt{2\pi(n-1)}$ $a \approx [1.016, 0.509]$	$k_{T0m} = a_0 + a_1\sqrt{n}$, $a \approx [-0.428, 0.377]$	simplified

$(T, n) \leftrightarrow (T_m\ T_0)$

The next problems to be developed are:

– conversion of any n order inertial model (no formula for inflection point)

– adaptation of choosen control design methods to different models

Analytical conversion of the Strejc model to the first order with time delay (FOTD) model

Thanks for your attention