International Journal of Automation, Control and Intelligent Systems
Vol. 1, No. 2, 2015, pp. 51-60
http://www.aiscience.org/journal/ijacis

A New Accurate Analytical Expression for Rise Time Intended for Mechatronics Systems Performance Evaluation and Validation

Ayman A. Aly[1, 2, *], Farhan A. Salem[1, 3]

[1]Mechatronics Engineering Program, College of Engineering, Taif University, Taif, Saudi Arabia
[2]Mechatronics Engineering Section, Dept. of Mechanical Engineering, Assuit University, Assiut, Egypt
[3]Alpha center for Engineering Studies and Technology Researches, Amman, Jordan

Abstract

Most Mechatronics systems are designed with synergy and integration to operate with exceptional high levels of accuracy and speed despite adverse effects of system nonlinearities, uncertainties and disturbances. Both rise time and peak time are used as a measure of swiftness of response, meanwhile, the closeness of the response to the desired response, is measured by the overshoot and settling time. Most used formulae and expressions for determining such performance specifications in texts lack accuracy, since it is more difficult to determine the exact analytical expressions of most used specifications. This paper proposes derivation of more accurate analytical expressions for rise time that can be applied to reflect the actual Mechatronics system rise time and used in accurate performance evaluation and verification.

Keywords

Mechatronics Systems, Performance, Rise Time, Analytical Expression

Received: June 18, 2015 / Accepted: June 27, 2015 / Published online: July 13, 2015

1. Introduction

Most used formulae and expressions for performance specifications in texts e. g. [1-8] lack accuracy. The determined performance specifications using these expressions, rarely accurate compared with actual results and measurements since it is more difficult to determine the exact analytical expressions of most used specifications and most introduced expressions are rough approximation of actual values. Mechatronics systems are supposed to operate with high accuracy and speed despite adverse effects of system nonlinearities and uncertainties, therefore, accuracy in Mechatronics systems performance is of concern, and the need for precise analytical expressions for mechatronics systems performance specifications calculation, is highly desired. This paper extends writer's previous work [9], and proposes derivation of accurate analytical expression for an important performance measure 'rise time' intended for research purposes in accurate verification/validation of Mechatronics systems performance evaluation as well as for the application in educational process.

2. Case Classification

The step response is the measured reaction of the control system to a step change in the input, step response has universal acceptance and popularity, because of simplicity of its form facilitates mathematical analysis, modeling, and experimental verifications, as well as it is easy to generate and has several measurement techniques for recording the time domain response. A typical step response and its associated performance specifications of second order systems are shown in Figure 1. The most used performance specifications are; Time constant T, Rise time T_R, Settling

* Corresponding author
E-mail address: draymanelnaggar@yahoo.com (A. A. Aly)

time T_s, Peak time, T_P, Maximum overshoot M_P, Maximum undershoot M_u, Percent overshoot OS%, Delay time T_d, The decay ratio D_R , Damping period T_O and frequency of any oscillations in the response, the swiftness of the response and the steady state error e_{ss}.

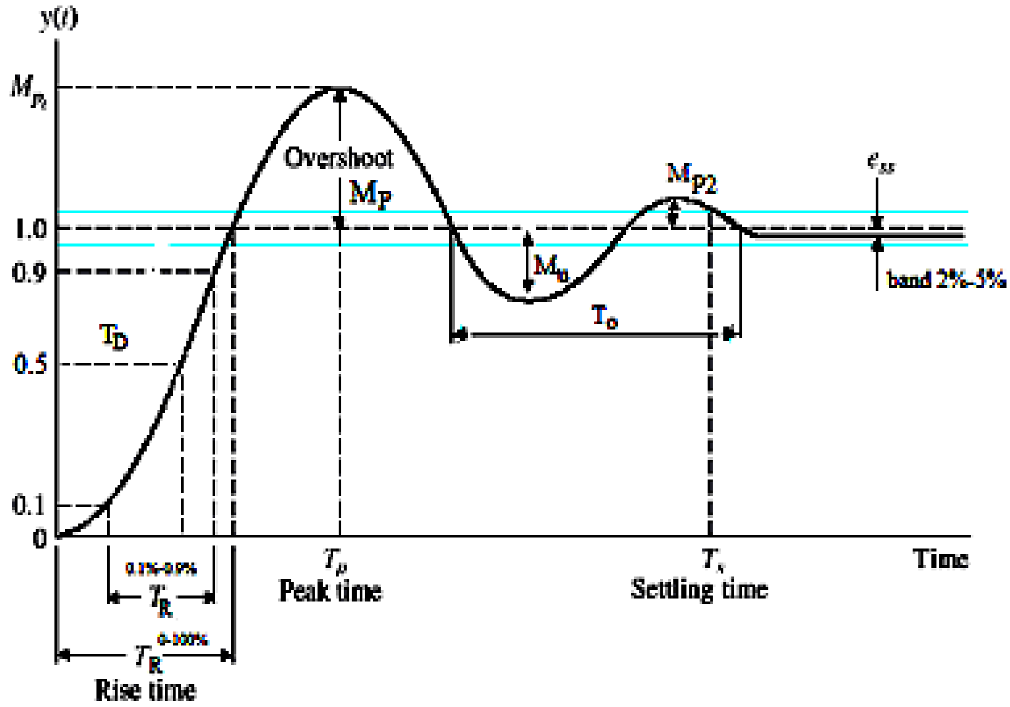

Figure 1. Second-order underdamped response specifications [9].

2.1. Rise Time T_R

In control theory applications, rise time is defined as "the time required for the response to rise from *x%* to *y%* of its final value", with 0%-100% rise time common for underdamped second order systems, *5%-95%* for critically damped and *10%-90%* for overdamped [10]. The transient response of the system may be described in terms of two factors; *a)* The swiftness of response, as represented by the rise time and the peak time. *b)* The closeness of the response to the desired response, as represented by the overshoot and settling time. Therefore, the rise time yields information about the speed of the transient response. This information can help a designer determine if the speed and the nature of the response do or do not degrade the performance of the system [11]. It is, for example determines speed of rise of flow or pressure (e.g. volume or pressure control modes)

2.2. Rise Time for First Order Systems

First order systems without zeros and systems that can be approximated as first order systems are described by first order differential equation and transfer function, derived as given by Eq. (1). As shown in Figure 2, the response is characterized by time constant T, rise time T_R, settling time T_s and steady state error e_{ss}, where the only parameter required to characterize response is time constant T, and when first order system is subjected to a unity step input, $R(s) = 1/s$, as by Eq. (2) the response for these systems is either natural decay or growth generated by the system pole:

$$
\begin{aligned}
& a\cdot\frac{d}{dt}x(t)+\mathrm{b}\cdot x(t)=u(t)\\
& \mathrm{a}\cdot sX(s)+b\cdot X(s)=U(s)\\
& X(s)\left[\frac{\mathrm{a}}{b}\cdot s+1\right]=\frac{1}{b}U(s)\\
& X(s)\left[Ts+1\right]=\frac{1}{b}U(s)\\
& \mathrm{G(s)}=\frac{X(s)}{U(t)}=\frac{1/b}{Ts+1}=\frac{K_{DC}}{Ts+1}
\end{aligned}
\tag{1}
$$

$$C(s)=R(s)*G(s)=\frac{1}{s}\frac{K_{DC}}{Ts+1} \tag{2}$$

Taking the inverse transform, the (solution) step response is given by Eq. (3). Rise time is found by solving Eq. (3) for the difference in time, e.g. rise time for 10% to 90% criterion is found by solving Eq. (3) for the difference in time at *c(t)* = 0.9 and *c(t)* = 0.1, as given by Eq.(4)

$$c(s)=1-e^{-\alpha t} \tag{3}$$

$$
\begin{aligned}
T_R &= (1-e^{-\alpha t_{90}})-(1-e^{-\alpha t_{10}})\\
T_R &= \frac{2.3026}{a}-\frac{0.1054}{a}=\frac{2.1972}{a}
\end{aligned}
\tag{4}
$$

The rise time can be measured in terms of the *time constant*, and given by Eq. (5):

$$T_R = (1 - e^{\frac{t_{90}}{T}}) - (1 - e^{\frac{t_{10}}{T}}) = 2.3026T - 0.1054T = 2.1972T \quad (5)$$

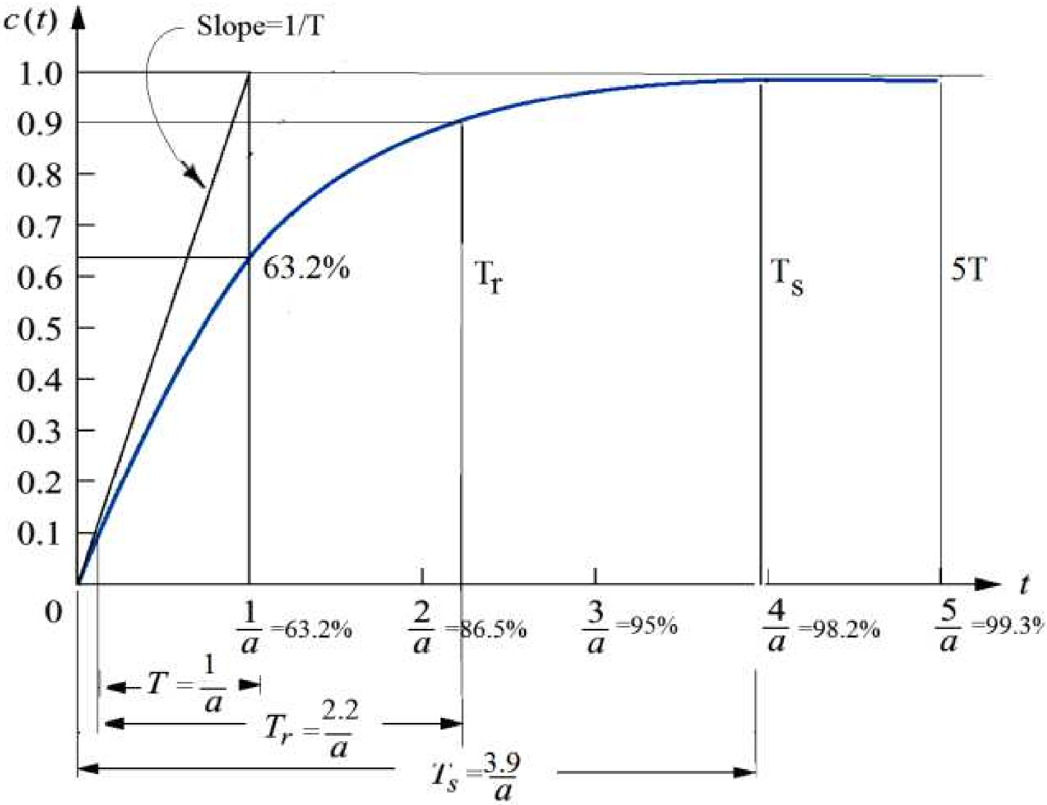

Figure 2 (a). First order system response to step, and performance specifications.

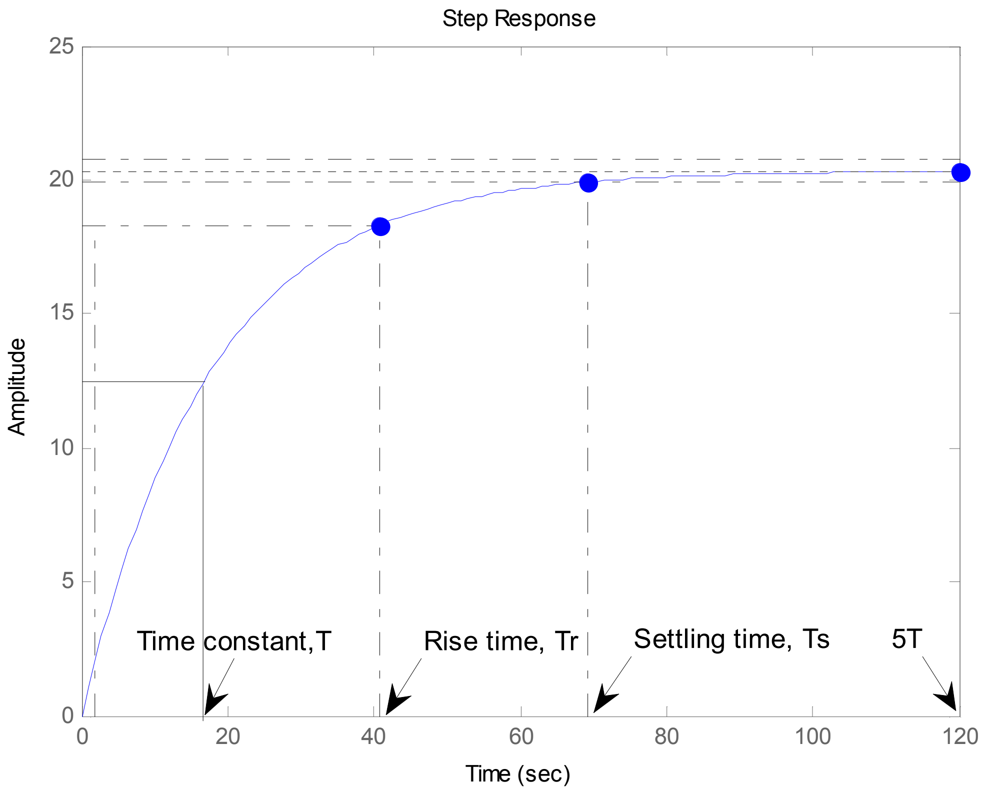

Figure 2 (b). Performance specifications of first order PMDC motor step response.

2.3. Rise Time for Second Order System

For second order systems, and systems that can be approximated as second order systems, when subjected to step input, *R(s) = A/s*, the response depends on pole location on complex plan given by Eq. (6), that in turns, depends on damping ratio ζ, and undamped natural frequency ω_n, where damping ratio determines how much the system oscillates as the response decays toward steady state and undamped natural frequency ω_n, determines how fast the system oscillates during any transient response, based on this, there are four cases of stable response to consider; undamped, underdamped, critically damped and overdamped response.

$$P = -\xi\omega_n \pm j\omega_n\sqrt{1-\xi^2} \tag{6}$$

For underdamped case; 0<ζ<1 and two complex conjugate poles given by Eq.(6), allow us to rewrite general form of second order system to have the form given by Eq.(7). To obtain inverse Laplace transform we need to expand by partial fractions and solve, this all gives:

$$\frac{C(s)}{R(s)} = \frac{\omega_n^2}{s^2+2\xi\omega_n s+\omega_n^2} = \frac{\omega_n^2}{(s+\xi\omega_n+j\sqrt{1-\xi^2})(s+\xi\omega_n-j\sqrt{1-\xi^2})} \tag{7}$$

$$\frac{C(s)}{R(s)} = \frac{1}{s}\frac{\omega_n^2}{(s^2+2\xi\omega_n s+\omega_n^2)} \Rightarrow$$

$$= \frac{s+2\xi\omega_n}{s^2+2\xi\omega_n s+(\omega_n\sqrt{1-\xi^2})^2} + \frac{1}{s} = \frac{s+2\xi\omega_n}{s^2+2\xi\omega_n s+\omega_d^{\,2}} + \frac{1}{s}$$

$$= \frac{s+2\xi\omega_n}{s^2+2\xi\omega_n s+(\omega_n\sqrt{1-\xi^2})^2} + \frac{1}{s} = -\frac{s+2\xi\omega_n}{s^2+2\xi\omega_n s+\omega_d^{\,2}} + \frac{1}{s}$$

$$= \frac{1}{s} - \frac{s+2\xi\omega_n}{(s+\xi\omega_n)^2+\omega_d^{\,2}} - \frac{\xi\omega_n}{\omega_d}\frac{\omega_d}{(s+\xi\omega_n)^2+\omega_d^{\,2}}$$

$$= \frac{1}{s} - \frac{s+2\xi\omega_n}{(s+\xi\omega_n)^2+\omega_d^{\,2}} - \frac{\xi}{\sqrt{1-\xi^2}}\frac{\omega_d}{(s+\xi\omega_n)^2+\omega_d^{\,2}}$$

Taking inverse Laplace transform, gives:

$$c(t) = 1 - e^{-\xi\omega_n t}(\cos\omega_d t + \frac{\xi}{\sqrt{1-\xi^2}}\sin\omega_d t) \tag{8}$$

This can be rewritten to have the following forms:

$$c(t) = 1 - \frac{1}{\sqrt{1-\xi^2}}e^{-\xi\omega_n t}\cos(\omega_d t - \phi) \tag{9}$$

$$c(t) = 1 - \frac{1}{\sqrt{1-\xi^2}}e^{-\xi\omega_n t}\sin(\omega_d t + \cos^{-1}\xi) \tag{10}$$

Eq.(8),(9)and(10) show that the damped natural frequency ω_d, given by $\omega_d = \omega_n\sqrt{1-\zeta^2}$,is the frequency at which the system will oscillate if the damping is decreased to zero.

Eq.(8) shows that performance of second order system depends on damping ratio ζ and undamped natural frequency ω_n. Plots of the step response as functions of the normalized time $\omega_n t$ for various damping ratio values of $0 \le \zeta \le 1.5$ are illustrated in Figure 3 (a), the curves show that he response becomes more oscillatory as ζ decreases in value, up to ζ=1, when $\zeta \ge 1$, the step response does not exhibit any overshoot or oscillatory behavior, also when ζ between *0.5* and *0.8* the system reaches final value more rapidly. Plots of the step response for various ω_n are illustrated in Figure 3 (b), the responses show that ω_n has a direct effect on the rise time, delay time, and settling time but does not affect the overshoot [9].

It is difficult to determine precise analytical expressions for rise time T_R[11] for second order systems. Different approximate formulae for the rise time appear in different texts[1-8]. One reason for that is because of different definitions of the rise time [7], as well as the required accuracy. An alternative measure to represent the rise time is as the reciprocal of the slope of the step response at the instant that the response is equal to 50% of its final value [3][9], that is at delay time T_D. The exact values of rise time for given range of damping ratio, can be determined directly from the responses of Figure 1, or rise time is found by solving Eq. (8) for the difference in time e.g. rise time for 10% to 90% is found by solving Eq. (8) for the difference in time at *c(t)* = 0.9 and *c(t)* = 0.1. For underdamped case; 0% to 100% of its final value, the rise time can be obtained by equating Eq.(8) with unity and solve for time *t*, that is rise time, this shown in equations below. Also MATLAB code can be written to return actual values of rise time, and plot it again given range of zetas, an example code, is written and applied in this paper.

$$c(t) = 1 - e^{-\xi\omega_n T_R}(\cos\omega_d T_R + \frac{\xi}{\sqrt{1-\xi^2}}\sin\omega_d T_R) = 1 \tag{11}$$

Since $e^{-\xi\omega_n T_R} \neq 0$, we have:

$$\cos\omega_d T_R + \frac{\xi}{\sqrt{1-\xi^2}}\sin\omega_d T_R = 0 \Rightarrow \tan\omega_d T_R = -\frac{\sqrt{1-\xi^2}}{\xi}$$

$$\tan\omega_d T_R = -\frac{\omega_n\sqrt{1-\xi^2}}{\xi\omega_n} \Rightarrow T_R = \frac{1}{\omega_d}\tan^{-1}\left(\frac{\omega_d}{\xi\omega_n}\right)$$

$$T_R = \frac{\pi-\phi}{\omega_d} = \frac{\pi-\tan^{-1}(\sqrt{1-\xi^2}/\xi)}{\omega_n\sqrt{1-\xi^2}} \quad \rightarrow \quad 0<\xi<1$$

Where, referring to Figure 4 (a), ϕ is defined by the following Eqs.:

$$\phi = \cos^{-1}(\xi) \Leftrightarrow \phi = \sin^{-1}(\sqrt{1-\xi^2}) \Leftrightarrow \phi = \tan^{-1}(\frac{\sqrt{1-\xi^2}}{\xi})$$

In the limit as $\xi \to 0$, this equation can be approximated as:

$$T_R = \frac{\pi - \pi/2}{\omega_n} = \frac{\pi}{2\omega_n}$$

In the limit as $\xi \to 1$, this equation can be approximated as:

$$T_R = \frac{\pi - 0}{\omega_n\sqrt{1-\xi^2}} = \frac{\pi}{\omega_n\sqrt{1-\xi^2}} \quad (12)$$

These equations imply that rise time increases as damping approaches unity. An approximation techniques can be used to estimate approximate values; by plotting normalized time $\omega_n T_R$ versus range of $0 \le \zeta \le 1.5$, and then approximate the curve by a straight line or over the range of $0 < \zeta < 1$. We first designate $\omega_n T_R$ as the normalized time variable and select a value for ζ. Using the computer, we solve for the values of $\omega_n T_R$ that yield $c(t) = 0.9$ and $c(t) = 0.1$. Subtracting the two values of $\omega_n T_R$ yields the normalized rise time, $\omega_n T_R$, for that value of ζ [4], continuing in like fashion with other values of ζ, and we obtain the results plotted in Figure 4(b). for $\omega_n=1$, this plot shows that increase in damping ratio leads to increase in the rise time that is not desirable .

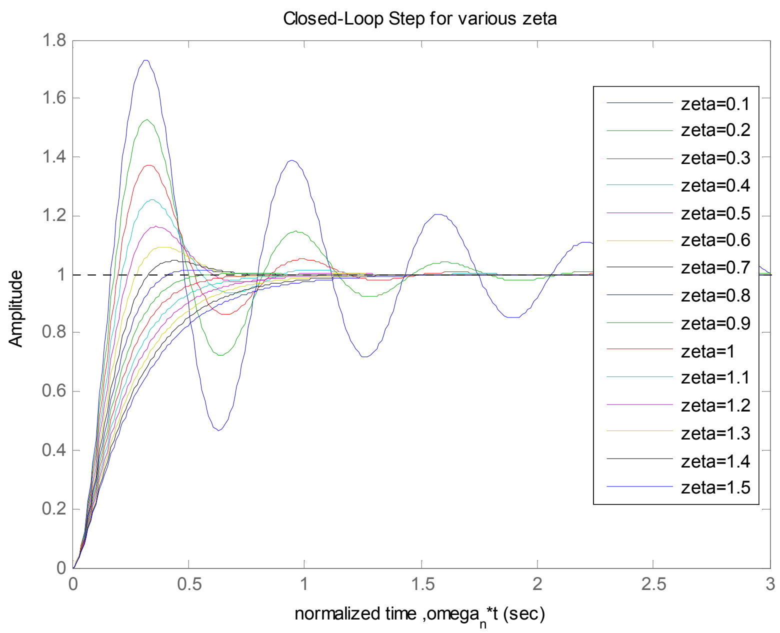

Figure 3 (a). Plots of the step response for various $0 \le \zeta \le 1.5$ with $\omega_n=10$.

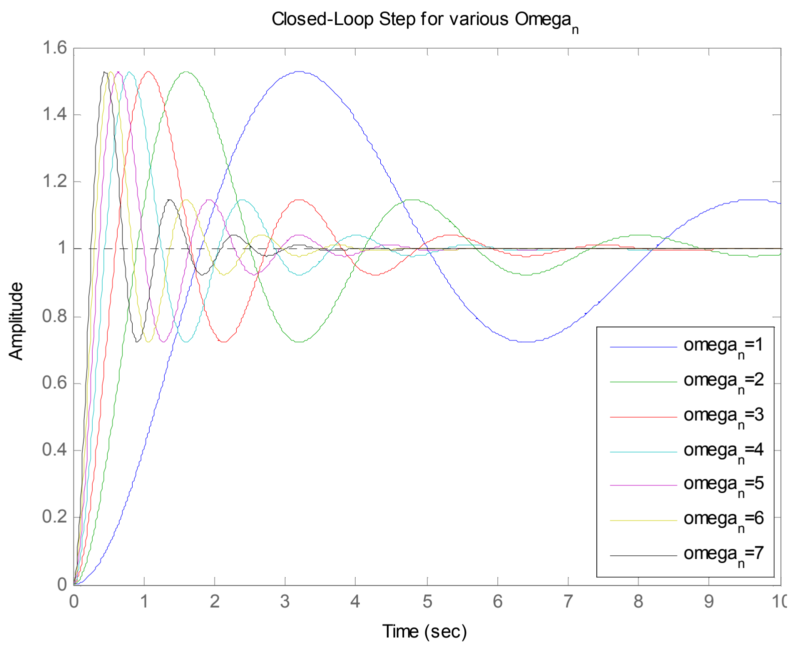

Figure 3 (b). Plots of the step response for various ω_n with $\zeta=0.2$.

3. Deriving Analytical Expressions for Rise Time T_R

3.1. Expressions for Rise Time T_R of 0% to 100% Criterion

Applying curve fitting to curve shown in Figure 4(b), to derive an approximate third order approximation given by Eq.(13):

$$T_R \cong \frac{1.765\xi^3 - 0.417\xi^2 + 1.039\xi + 1}{\omega_n} \Rightarrow 0 < \xi < 0.9 \tag{13}$$

Quadratic approximation can result in expressions given by Eq.(14)

$$\begin{aligned} T_R &\cong \frac{2.230\xi^2 - 0.078\xi^2 + 1.12}{\omega_n} \Rightarrow 0 < \xi < 0.9 \\ T_R &\cong \frac{2.917\xi^2 - 0.4167\xi + 1}{\omega_n} \Rightarrow 0 < \xi < 1 \end{aligned} \tag{14}$$

Referring to [3] the rise time T_R, for second order underdamped system, can be approximated as a straight line given by given by Eq.(15):

$$T_R \cong \frac{0.8 + 2.5\xi}{\omega_n} \Rightarrow 0 < \xi < 1 \tag{15}$$

Referring to [4] the linear approximation of rise time is given by Eq.(16):

$$T_R = \frac{0.6 + 2.16\xi}{\omega_n} \Rightarrow 0.3 \le \xi \le 0.8 \tag{16}$$

Referring to [5] rise time is given by given by Eq.(17):

$$T_R = \frac{2.2}{\xi\omega_n} \tag{17}$$

Referring to [6] rise time is given by given by Eq.(18):

$$\begin{aligned} T_R &= \frac{1.2 - 0.45\xi + 2.6\xi^2}{\omega_n} \quad \Rightarrow \xi < 1.2 \\ T_R &= \frac{4.7\xi - 1.2}{\omega_n} \quad \Rightarrow \xi > 1.2 \end{aligned} \tag{18}$$

All these equations shows that rise time is proportional to ζ and inversely proportional ω_n. based these equations, it can be stated that, the maximum overshoot and the rise time conflict with each other, as overshoot increases, the rise time decreases, this means that we can *note* make both the maximum overshot and the rise time smaller simultaneously, if one is made smaller the other will become larger.

Most of these derived expression are rough approximations and mostly has huge deviation at actual values, this is shown in Figure 4(c) that shows the plots of different approximation for rise time against damping ratio (PO) .

Analyzing actual curve rise time against damping ratio, shown in Figure 4(b), show that the curve can be fit as ramp in some regions and of second order in others. applying curve fitting and trial and error approaches, a better and more accurate expressions, can be suggested for $0 \le \zeta < 0.4$, for $0.4 \le \zeta < 1.2$, and for $\zeta > 1.2$, suggested analytical expressions are given by Eq.(19):

$$\begin{aligned} T_R &= \frac{1.2 - 0.2\xi + 3\xi^2}{\omega_n} \quad \Rightarrow 0 < \xi < 0.4 \\ T_R &= \frac{1.26 - 0.51\xi + 2.58\xi^2}{\omega_n} \quad \Rightarrow 0.4 \le \xi < 1.2 \\ T_R &= \frac{4.67\xi - 1.2}{\omega_n} \quad \Rightarrow \xi > 1.2 \end{aligned} \tag{19}$$

Plotting actual rise time and a rise time obtained using suggested expressions, both versus normalizes time, are shown in Figure 4(d), analysis of both plots show that the suggested expressions match the actual values with maximum upper error of 0.06 seconds for $0.34 < \zeta < 0.4$, and maximum upper error of 0.026 seconds for $0.6 < \zeta < 0.7$. this can lead us to conclude, that the suggested expressions can be used to analytically calculate rise time with error of $\pm$ 0.02 seconds.

3.2. Expressions for Rise Time T_r of 10% to 100% Criterion

Applying the same procedure, expressions given by Eq.(20) are proposed. Plotting actual rise time and a rise time obtained using proposed expressions, both versus normalizes time, are shown in Figure 5,

$$\begin{aligned} T_R &= \frac{1.15 - 0.21\xi + 2.55\xi^2}{\omega_n} \quad \Rightarrow 0 < \xi < 0.4 \\ T_R &= \frac{1.26 - 0.55\xi + 2.58\xi^2}{\omega_n} \quad \Rightarrow 0.4 \le \xi < 0.85 \\ T_R &= \frac{1.2 - 0.45\xi + 2.58\xi^2}{\omega_n} \quad \Rightarrow 0.85 \le \xi < 1.2 \\ T_R &= \frac{4.69\xi - 1.19}{\omega_n} \quad \Rightarrow \xi > 1.2 \end{aligned} \tag{20}$$

4. Testing Proposed Expressions Against MATLAB

To test the proposed expressions, against actual values, a MATLAB code is written to calculate rise time for a given *I* or *II* order systems. Rise time is to be calculated, applying each of the following: MATLAB control toolbox , proposed expressions, rise time as the reciprocal of the step response slope at delay time T_D , Rise time for the difference in time for 10% to 90% by solving Eq. (8)

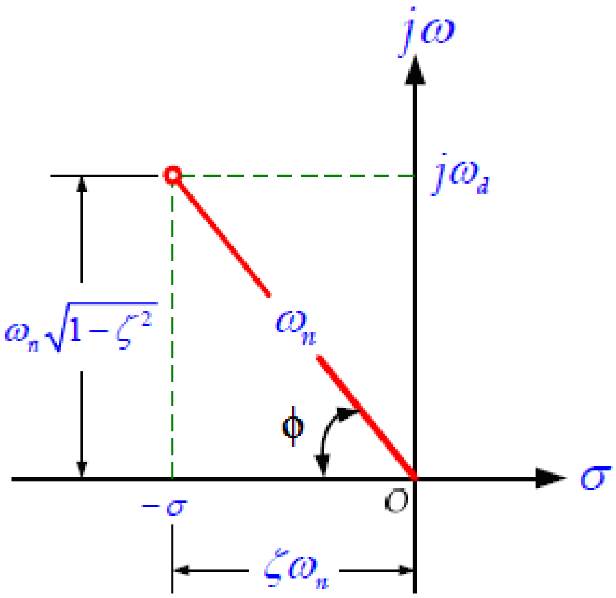

Figure 4 (a). Definition of angle ϕ.

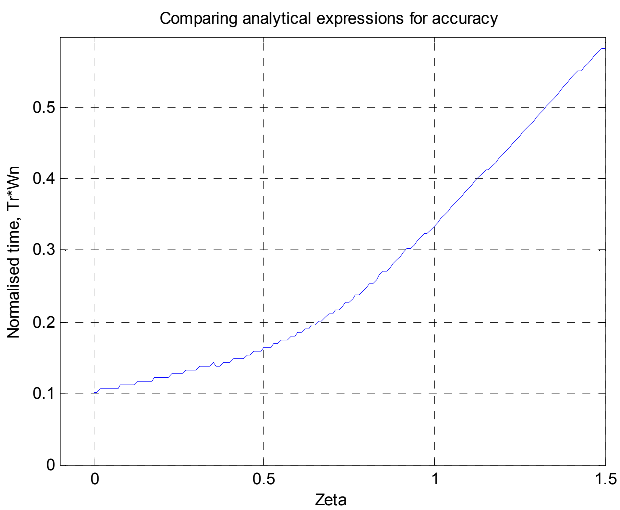

Figure 4 (b). Normalized rise time versus ζ, for second-order system.

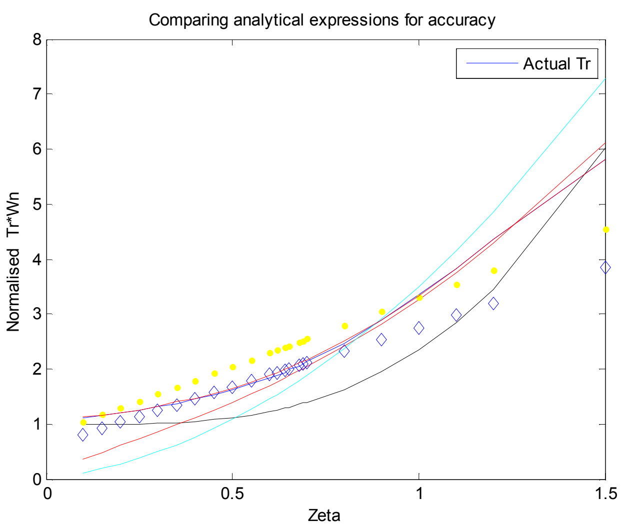

Figure 4 (c). Plots of different approximation for rise time.

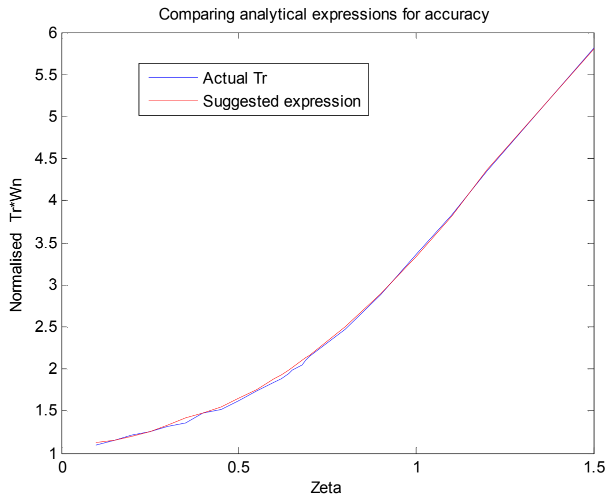

Figure 4 (d). Rise time plot of actual and using suggested expressions.

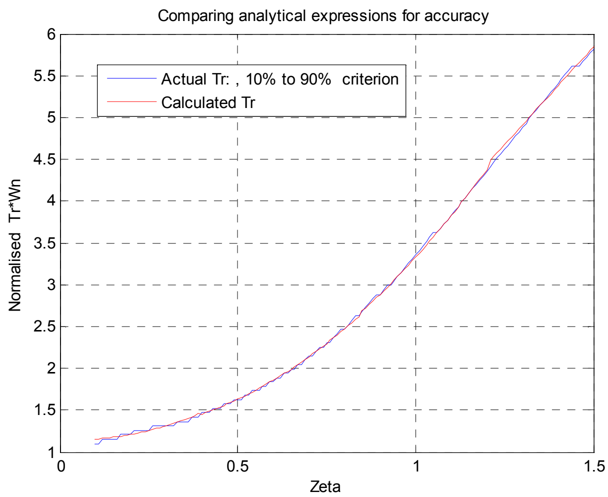

Figure 5. Rise time, *10%-90%;* comparing actual normalized rise time against obtained using derived expressions.

Testing for *I* or *II* order systems given by Eq.(21-23), will result in rise time values actual and calculated given in tables 1-3. Analysis of plots of both actual rise time and calculated using suggested expressions, both versus normalizes time shown in Figures 4,5, as well as ,calculated data in tables1-3, show that the suggested expressions match the actual values with very small deviation from actual value, this can lead us to conclude, that the suggested may be used to reflect the actual value and can be applied in accurate calculation, evaluation and verification of rise time with error of ± 0.01 seconds

$$G(s) = \frac{6}{s^2 + 2s + 6} \tag{21}$$

$$G(s) = \frac{4}{s^2 + 2.4s + 4} \tag{22}$$

$$G(s) = \frac{1}{s+2} \tag{23}$$

Table 1. Testing for II order system given by Eq.(21).

	Proposed TR (0.1-0.09)	Proposed TR (0.0-1)	MATLAB. Step properties :TR (0.0-0.9)	MATLAB (step properties:0-1)	TR at TD slope
Calculated T_R	0.59827	0.60494	0.6040	0.605	0.642
Actual T_R	0.58837	0.60394	0.60394	0.60394	0.60394
Deviation	0.0099	0.0010	0.0001	0.0011	0.0381

Table 2. Testing for II order system given by Eq.(22).

	Proposed TR (0.1-0.09)	Proposed TR (0.0-1)	MATLAB Step properties :TR (0.0-0.9)	MATLAB (step properties:0-1)	TR at TD slope
Calculated T_R	0.92940	1.39015	0.928	1.4	1.25
Actual T_R	0.91803	1.39015	0.91803	1.39015	1.39015
Deviation	0.0114	0	0.01	0.0098	0.1402

Table 3. Testing for I order system given by Eq.(23).

	Proposed TR (0.1-0.09) TR= 2.1972*T	MATLAB (step properties)	Rise time at TD
T_R	1.0986	1.1	0.89

5. Conclusions

More precise analytical expressions for accurate calculation of "rise time" with minimum deviation from actual values are derived and tested. The suggested expressions can be applied to analytically calculate rise time with deviation to ± 0.02 seconds at actual value. Suggested expressions are intended to be used in systems dynamics performance analysis, controller design verification, and related sciences, as well as for the application in educational process.

The Applied MATLAB Code

```
clc, clear all, close all,
num=input(' Enter Numerator: ');
den=input(' Enter Denominator: ');
sys1=tf(num, den);
q=length(den);
if q==2
 root=roots(den);
 poles=abs(root);
 if ((poles < 0)||(real(poles)<0))
  uu=' The system is stable';
 else
  uu=' The system is unstable';
 end
 Time_constant = -1/( poles);
 % rise time calculations, Tr Calculations
 rise_time_1_9=2.1972/poles;
 rise_time_0_1=4.6/poles;
 Ess= 1/(1+ dcgain(sys1));
 home,
 printsys(num,den,'s')
 disp( ' '),
 sys1;
 step(sys1)
 disp('============================');
 disp( ' Stability analysis: ')
 disp('--------------------------');
 fprintf( ' The system pole is: P1 = %2.2f , %s \n' ,poles,uu)
 disp('--------------------------');
 disp('============================');
 disp(' Rise time,Tr Calculations :')
 disp('--------------------------');
 fprintf( 'Calculated Rise time,(0.1-0.09 criterion), Tr = %2.5f Seconds \n' ,rise_time_1_9)
 fprintf( 'Calculated Rise time,(0.0- 1.0 criterion), Tr = %2.5f Seconds \n' ,rise_time_0_1)
 disp('============================');
 disp( ' ')
elseif q==3
 disp( ' ')
% disp(' Please wait: it takes 3-5 seconds ')

 zeta=den(2)/(2*sqrt(den(3)*den(1)));
 omega_n=sqrt(den(3)/den(1));
 omega_d=omega_n*sqrt(1-zeta^2);
 system_poles=roots(den);
 if system_poles < 0
  uu= ' The system is stable';
 else
  uu= ' The system is unstable';
 end
 Time_constant=1/(zeta*omega_n)
 zeta1=[0:0.01:1.5]';
```

```
sys={};
for i=1:length(zeta1)
  sys{i}=tf(omega_n*omega_n, [1, 2*zeta1(i)*omega_n, omega_n*omega_n]); % omega_n=1
  sys1=tf(omega_n*omega_n, [1, 2*zeta*omega_n, omega_n*omega_n]);
end
[y,t]=step([sys{:}]);
% Calculating Rise time ,Tr ; 10%:90%
A1=logical( y(:,:)>=0.1);
A9=logical( y(:,:)>=0.9);
time_1=[];
time_9=[];
Rise_time_1_9=[];
for i=1:length(zeta1)
  tmp1= t(A1(:,i));
  tmp9= t(A9(:,i));
  time_1(i)=tmp1(i);
  time_9(i)=tmp9(i);
end
Rise_time_1_9=time_9-time_1;
answer3=[ zeta1, Rise_time_1_9'];
%   format bank
% P=zeta
 a3=find(answer3(:,1)==zeta);

actual_Tr_1_9=answer3(a3,2);
if ((0< zeta) && (0.4 > zeta))
  Tr_calcul_1_9=(1.15-0.21*zeta+2.55.*zeta^2)/omega_n ;
elseif 0.4 <= zeta && (zeta < 0.85)
 Tr_calcul_1_9=(1.26-0.55*zeta+2.58.*zeta^2)/omega_n ;
elseif 0.85 <= zeta && (zeta <= 1.2)
Tr_calcul_1_9=(1.20-0.45*zeta+2.58.*zeta^2)/omega_n ;
else
  Tr_calcul_1_9=(4.69*zeta-1.19)/omega_n;
end
% Calculating Rise time ,Tr ; 0%:100%
A0=logical( y(:,:)>=0);
A1=logical( y(:,:)>=0.999);
time_0=[];
time_1=[];
Rise_time_0_1=[];
for i=1:length(zeta1)
  tmp0= t(A0(:,i));
  tmp1= t(A1(:,i));
  time_0(i)=tmp0(i);
  time_1(i)=tmp1(i);
end
Rise_time_0_1=time_1-time_0;
answer4=[ zeta1, Rise_time_0_1'];
a4=find(answer4(:,1)== zeta);
actual_Tr_0_1=answer4(a4,2);
if ((0< zeta) && (0.4 > zeta))
  Tr_calcul_0_1 = (1.2 -0.2*zeta+3.*zeta^2)/omega_n ;
elseif 0.4 <= zeta && (zeta < 1.2)
 Tr_calcul_0_1=(1.26-0.51*zeta+2.58.*zeta^2)/omega_n ;
else
  Tr_calcul_0_1=(4.67*zeta-1.2)/omega_n;
end
Tr_calcul_0_1;
y=step(sys1,t);
DC_gain2=y(length(t));
home, printsys(num,den,'s')
disp(' '), sys1; step(sys1)
disp(' ')
disp(' Time constant,T Calculations :')
disp('=====================');
disp( ' Stability analysis: ')
disp( [system_poles(1,1);system_poles(2,1)])
fprintf( '       %s  \n' ,uu),
disp('------------------------');
fprintf( ' The damping ratio,Zeta = %2.5f Seconds \n' ,zeta)
fprintf( ' The UNdamped natural frequency, Omega_n= = %2.5f rad/s \n' ,omega_n )
fprintf( ' The Damped natural frequency, Omega_d= = %2.5f rad/s \n' ,omega_d )
disp('------------------------');
fprintf( ' Time constant,T = %2.5f Seconds \n' ,Time_constant)

disp(' Rise time,Tr Calculations :')
disp('=====================');
fprintf( 'Actual Rise time,(0.1-0.09 criterion), Tr = %2.5f Seconds \n' ,actual_Tr_1_9)
fprintf( 'Calculated Rise time,(0.1-0.09 criterion), Tr = %2.5f Seconds \n' ,Tr_calcul_1_9)
disp('----------------------');
fprintf( 'Actual Rise time,(0.0- 1.0 criterion), Tr = %2.5f Seconds \n' ,actual_Tr_0_1)
fprintf( 'Calculated Rise time,(0.0- 1.0 criterion), Tr = %2.5f Seconds \n' ,Tr_calcul_0_1)
disp('=====================');
end
```

References

[1] Ashish Tewari, Modern Control Design with MATLAB and SIMULINK, John Wiley and sons, February 2002.

[2] LTD, 2002 England Katsuhiko Ogata, modern control engineering, third edition, Prentice hall, 1997.

[3] Farid Golnaraghi Benjamin C. Kuo, Automatic Control Systems, John Wiley and sons INC, 2010.

[4] Norman S. Nise control system engineering, Sixth Edition John Wiley & *Sons,* Inc, 2011.

[5] Gene F. Franklin, J. David Powell, and Abbas Emami-Naeini, Feedback Control of Dynamic Systems, 4th ed., Prentice Hall, 2002.

[6] http://courses.engr.illinois.edu/ece486/lab/estimates/estimates.html.

[7] Bill Goodwine, Engineering Differential Equations Theory and Applications, Springer 2011.

[8] Dale E. Seborg, Thomas F. Edgar, Duncan A. Mellichamp ,Process dynamics and control, second edition, Wiley 2004.

[9] Farhan A. Salem, Precise Performance Measures for Mechatronics Systems, Verified and Supported by New MATLAB Built-in Function, International Journal of Current Engineering and Technology, Vol.3, No.2, June, 2013.

[10] Levine, William S. (1996), *The control handbook*, Boca Raton, FL: CRC Press, p. 548, ISBN -8493-8570-9.

[11] Norman S. Nise, Control system engineering, 6 edition, John Wiley & *Sons,* 2011.