Design of PI and PID Controllers for Fractional Order Time Delay Systems

M. M. Özyetkin, C. Yeroğlu, N. Tan and M. E. Tağlık

Inonu University, Dept. of Electrical and Electronics Engineering, 44280, Malatya, Turkey
(Tel: 04223774812; e-mail: mmozyetkin@inonu.edu.tr, cyeroğlu@inonu.edu.tr, ntan@inonu.edu.tr, metaglık@inonu.edu.tr)

Abstract: This paper deals with the computation of rational approximations of fractional derivatives and/or integrals. All rational approximations for fractional order of 0.1, 0.2,…, 0.9 are obtained using continued fraction expansion (CFE) method. Extension of the stability boundary locus approach to control systems with a fractional order transfer function is given for the computation of stabilizing PI and PID controllers using continuous approximations of fractional orders. Numerical examples are provided to illustrate the results and to show the effect of the order of approximation on the stability region.

Keywords: PI controller, PID controller, Fractional order system, Time delay, Stabilizing, Realization

1. INTRODUCTION

A system represented by differential equations where the orders of derivatives can take any real number not necessarily integer number can be considered as a fractional order system. The significance of fractional order representation is that it is more adequate to describe real world systems than those of integer order models (Nonnenmacher and Glöckle, 1991; Westerlund, 1994). It should be stated that there are many physical systems such as viscoelastic materials, electrochemical processes, long lines, dielectric polarizations, colored noise, cardiac behavior and chaos whose behaviors can be described using fractional order differential equations (Skars, et al., 1988; Hartley, et al., 1995; Hartley and Lorenzo, 2002; Goldberger, et al., 1985). In recent years, some important studies dealing with the application of the fractional calculus to the control systems have been done in (Podlubny, 1999; Hwang, and Cheng, 2006; Chen, et al., 2006; Tan, et al., 2009).

Fractional order control systems have transfer functions with fractional derivatives $s^\alpha$ and fractional integrals $s^{-\alpha}$ where $\alpha \in \mathbb{R}$. It is not easy to compute frequency and time domain behaviors of such fractional order transfer functions with available software packages. It is well known that the simulation programs have been prepared to deal with integer power of $s$ only. Also, the designed controllers can be implemented by using the same programs or hardware implementations can be done using electronic components which result an integer order transfer function. Although, there are some recent works dealing with implementation of a controller using fractance device (Nakagava and Sorimachi, 1992), this area needs further studies since an electronic component to implement fractional order systems is not commercially available. Therefore, the problem of integer order approximations of fractional order functions becomes a very important one to be solved. A fractional transfer function can be replaced with an integer order transfer function which has almost the same behaviors with the real transfer function but much more easier to deal with.

There are many methods (Podlubny, et al., 2002; Charef, 2006; Dorcak, et al., 2003; Krishna and Reddy, 2008; Varshney, et al., 2007; Chen, et al. 2004; Shyu, et al., 2009; Vinagre, et al., 2001) for computing integer order approximations of fractional order operators such as $s^\alpha$ or $1/s^\alpha$. However, it is interesting that the approximations for $s^{0.5}$ or $1/s^{0.5}$ are mostly given in the literature (Podlubny, et al., 2002; Charef, 2006; Dorcak, et al., 2003; Krishna and Reddy, 2008; Varshney, et al., 2007; Chen, et al. 2004; Shyu, et al., 2009; Vinagre, et al., 2001). Preparing a table which gives approximations for all $s^\alpha$ will be very helpful for the beginners in the field and accelerate further works. Also, it is necessary to point out that some methods of approximations give accurate frequency domain behavior while others can give accurate time responses. It is not possible to say that one is the best among others (Podlubny, et al., 2002). For computation of all stabilizing controllers, there is not any work which compares the stability regions obtained for different order of approximations.

The aim of this paper is to compute the rational approximations of $s^\alpha$ for all $\alpha = 0.1, 0.2,..., 0.9$ using continued fraction expansion method (Podlubny, et al., 2002) and to find all stabilizing PI and PID controllers for a fractional order transfer function with time delay. The computation of all stabilizing controllers is very important and many results have been obtained in (Tan, 2005; Söylemez, et al., 2003; Hamamci, 2007; Aoun, et al., 2004; Datta, et al., 2000; Ackermann and Kaesbauer, 2003). The stability regions for first, second, third and fourth order approximations are obtained using the stability boundary locus approach (Tan, 2005). Thus, a new method is proposed to compute parameters of controllers for fractional order time delay systems. It has been shown that both the stability regions obtained for the real fractional order time delay system and its fourth order integer approximation are the same. This has been illustrated via numerical examples. Besides, controllers which satisfying robust performance are selected from stability region.
The paper is organized as follows: Rational approximations of \( s^\alpha \) are given in Section 2. Computation of all stabilizing PI and PID controllers for fractional order transfer functions with time delay are given in Section 3 and Section 4, respectively. Finally, Section 5 includes concluding remarks.

2. RATIONAL APPROXIMATION

There are several methods for obtaining rational approximations of fractional order systems. For example Carlson’s method, Matsuda’s method, Oustaloup’s method, the Grünwald-Letnikoff approximation, Maclaurin series based approximations, time response based approximations etc. (Podlubny, et al., 2002). One of the most important approximations for fractional order systems is the CFE method. The CFE method is used for obtaining realization of \( s^\alpha \) (\( 0 < \alpha < 1 \)) in this paper. This method can be expressed in the form (Krisha and Reddy, 2008),

\[
(1 + x)^{\alpha} = \frac{1}{\alpha} \frac{\alpha + x}{1 + \frac{x}{1 + \frac{2x}{1 + \frac{3x}{1 + \frac{4x}{...}}}}}
\]

In this formulation \( x = s - 1 \) used for the computation of \( s^\alpha \). The calculated rational approximations for \( s^{0.1}, s^{0.2}, \ldots, s^{0.9} \) are presented in Table 1. Bode diagrams of real system and Bode diagrams of first, second, third, fourth order approximations for \( s^{0.3} \) are given in Fig. 1.

Using (1) first, second, third and fourth order integer approximations which are dependent on \( \alpha \) are obtained as follows:

First:

\[
(1 + \alpha)x + (1 - \alpha) = \frac{(1 + \alpha)x}{1 + \frac{(1 - \alpha)x}{1 + \frac{(1 - \alpha)x}{1 + \frac{(1 - \alpha)x}{...}}}}
\]

Second:

\[
(\alpha^2 + 3\alpha + 2)s^2 + (-2\alpha^2 + 8)s + (\alpha^2 - 3\alpha + 2)
\]

Third:

\[
(\alpha^3 + 6\alpha^2 + 11\alpha + 6)s^3 + (-3\alpha^3 - 6\alpha^2 + 27\alpha + 54)s^2 + (-3\alpha^3 - 6\alpha^2 + 27\alpha + 54)s + (\alpha^3 + 6\alpha^2 + 11\alpha + 6)
\]

Fourth:

\[
(\alpha^4 + 10\alpha^3 + 35\alpha^2 + 50\alpha + 24)s^4 + (-4\alpha^4 - 20\alpha^3 + 40\alpha^2 - 32\alpha + 384)s^3 + (6\alpha^4 - 150\alpha^3 + 864\alpha^2 + 144\alpha - 120\alpha + 384)s^2 + (-4\alpha^4 + 20\alpha^3 + 40\alpha^2 - 32\alpha + 384)s + (\alpha^4 + 10\alpha^3 + 35\alpha^2 + 50\alpha + 24)
\]

3. COMPUTATION OF STABILIZING PI CONTROLLERS USING THE STABILITY BOUNDARY LOCUS

Consider the SISO control system of Fig. 2 where

\[
G_p(s) = G(s)e^{-\alpha} = \frac{N(s)}{D(s)} e^{-\alpha}
\]

is the plant to be controlled and \( C(s) \) is a PI controller of the form

\[
C(s) = k_p + \frac{k_i}{s} = \frac{k_p s + k_i}{s}
\]

The problem is to compute the parameters of the PI controller of (8) which stabilize the system of Fig. 2.

![Bode Diagram for s^0.3](image)

Fig. 1. Bode diagrams for \( s^{0.3} \). Real system; First order approximation; Second order approximation; Third order approximation; Fourth order approximation

The closed loop characteristic polynomial \( \Delta(s) \) of the system of Fig.2, i.e. the numerator of \( 1 + C(s)G_p(s) \), can be written as

\[
\Delta(s) = sD(s) + (k_p + k_i)N(s)e^{-\alpha}
\]

Decomposing the numerator and the denominator polynomials of \( G(s) \) in (7) into their even and odd parts, and substituting \( s = jo \), gives

\[
G(j\omega) = \frac{N_j}{D_j} + j\omega N_o = \frac{D_j}{j\omega D_o}
\]

Thus, the closed loop characteristic polynomial of (9) can be written as

\[
\Delta(j\omega) = [k_j N_j - k_o N_o \cos(\omega r) + \alpha (k_o N_j + k_i N_o) \sin(\omega r) - \omega^2 D_o]
\]

\[
+ j\omega [k_o N_j + k_i N_o \cos(\omega r) - (k_i N_j - \alpha^2 k_o N_o) \sin(\omega r) + \alpha D_o]
\]

\[
= R + j\omega L = 0
\]

Then, equating the real and imaginary parts of \( \Delta(j\omega) \) to zero, one obtains

\[
k_j (\alpha^2 N_j \cos(\omega r) + \alpha N_o \sin(\omega r)) + k_o (N_j \cos(\omega r) + \alpha N_o \sin(\omega r)) = \omega^2 D_o
\]

and

\[
k_j (\alpha^2 N_j \cos(\omega r) + \alpha N_o \sin(\omega r)) + k_i (\alpha N_j \cos(\omega r) - N_j \sin(\omega r)) = 0
\]

Let

\[
Q(\omega) = -\alpha^2 N_j \cos(\omega r) + \alpha N_o \sin(\omega r)
\]

\[
R(\omega) = N_j \cos(\omega r) + \alpha N_o \sin(\omega r), \quad X(\omega) = \omega^2 D_o
\]
Table 1. Rational Approximations for all $s^n$

<table>
<thead>
<tr>
<th>$s$</th>
<th>First order</th>
<th>Second order</th>
<th>Third order</th>
<th>Fourth order</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s^1$</td>
<td>$1.22s+1$</td>
<td>$1.351s^2+4.67s+1$</td>
<td>$1.444s^3+11.42s^2+10.33s+1$</td>
<td>$1.518s^4+21.529s^3+44.596s^2+18.222s+1$</td>
</tr>
<tr>
<td></td>
<td>$s+1/22$</td>
<td>$s^2+4.67s+13.51$</td>
<td>$s^3+10.33s+11.42s+1$</td>
<td>$s^4+18.222s^3+44.596s^2+21.529s+1.518$</td>
</tr>
<tr>
<td>$s^2$</td>
<td>$1.5s+1$</td>
<td>$1.83s^2+5.5s+1$</td>
<td>$2.1s^3+14.67s^2+12s+1$</td>
<td>$2.316s^4+29.33s^3+56s^2+21s+1$</td>
</tr>
<tr>
<td></td>
<td>$s+1.5$</td>
<td>$s^2+5.5s+10$</td>
<td>$s^3+12s+14.67s+2.1$</td>
<td>$s^4+29.33s^3+56s^2+21s+2.316$</td>
</tr>
<tr>
<td>$s^3$</td>
<td>$1.857s+1$</td>
<td>$2.512s^2+6.571s+1$</td>
<td>$3.071s^3+19.134s^2+14.128s+1$</td>
<td>$3.57s^4+40.63s^3+71.546s^2+24.57s+1$</td>
</tr>
<tr>
<td></td>
<td>$s+1.857$</td>
<td>$s^2+6.571s+10.25$</td>
<td>$s^3+14.128s^2+19.134s+3.071$</td>
<td>$s^4+71.546s^3+40.63s^2+24.57s+3.57$</td>
</tr>
<tr>
<td>$s^4$</td>
<td>$2.333s+1$</td>
<td>$3.5s^2+8s+1$</td>
<td>$4.576s^3+25.5s^2+17s+1$</td>
<td>$5.594s^4+57.53s^3+93.5s^2+29.33s+1$</td>
</tr>
<tr>
<td></td>
<td>$s+2.333$</td>
<td>$s^2+8s+10$</td>
<td>$s^3+17s^2+25.5s+4.576$</td>
<td>$s^4+93.5s^3+57.53s^2+29.33s+5.594$</td>
</tr>
<tr>
<td>$s^5$</td>
<td>$3s+1$</td>
<td>$5.5s^2+10s+1$</td>
<td>$7s^3+35s^2+21s+1$</td>
<td>$9s^4+84s^3+126s^2+36s+1$</td>
</tr>
<tr>
<td></td>
<td>$s+3$</td>
<td>$s^2+10s+15$</td>
<td>$s^3+21s^2+35s+7$</td>
<td>$s^4+36s^3+126s^2+84s+9$</td>
</tr>
<tr>
<td>$s^6$</td>
<td>$4x+1$</td>
<td>$7.428s^2+13r+1$</td>
<td>$11.143s^3+50.143s^2+27s+1$</td>
<td>$15.076s^4+128.143s^3+177.42s^2+46s+1$</td>
</tr>
<tr>
<td></td>
<td>$s+4$</td>
<td>$s^2+13r+17.428$</td>
<td>$s^3+27s^2+50.143s+11.143$</td>
<td>$s^4+46s^3+177.42s^2+128.143s+15.076$</td>
</tr>
<tr>
<td>$s^7$</td>
<td>$5.67s+1$</td>
<td>$11.77s^2+18s+1$</td>
<td>$18.933s^3+76.846s^2+37s+1$</td>
<td>$26.965s^4+209.38s^3+267.54s^2+62.67s+1$</td>
</tr>
<tr>
<td></td>
<td>$s+5.67$</td>
<td>$s^2+18s+11.77$</td>
<td>$s^3+37s^2+76.846s+18.933$</td>
<td>$s^4+62.67s^3+267.54s^2+209.38s+26.965$</td>
</tr>
<tr>
<td>$s^8$</td>
<td>$9s+1$</td>
<td>$21s^2+28s+1$</td>
<td>$36.27s^3+133s^2+57s+1$</td>
<td>$54.41s^4+386.91s^3+456s^2+96s+1$</td>
</tr>
<tr>
<td></td>
<td>$s+9$</td>
<td>$s^2+28s+21$</td>
<td>$s^3+57s^2+133s+36.27$</td>
<td>$s^4+96s^3+456s+386.91s+54.41$</td>
</tr>
<tr>
<td>$s^9$</td>
<td>$19s+1$</td>
<td>$50.1s^2+58s+1$</td>
<td>$93.026s^3+308.45s^2+117s+1$</td>
<td>$147.04s^4+959.64s^3+1042.36s^2+196s+1$</td>
</tr>
<tr>
<td></td>
<td>$s+19$</td>
<td>$s^2+58s+50.1$</td>
<td>$s^3+117s^2+308.45s+93.026$</td>
<td>$s^4+196s^3+1042.36s^2+959.64s+147.04$</td>
</tr>
</tbody>
</table>

$S(o) = \alpha N_s \cos(o) + \omega^2 N_o \sin(o)$

$U(o) = \alpha V_o \cos(o) - N_o \sin(o)$ and $Y(o) = -\alpha D_e$

Then, (12) and (13) can be written as

$k_p Q(o) + k_i R(o) = X(o)$  \hspace{3cm} (16)

$k_p S(o) + k_i U(o) = Y(o)$

From these equations

$k_p = \frac{X(o)U(o) - Y(o)R(o)}{Q(o)U(o) - R(o)S(o)}$  \hspace{3cm} (17)

and

$k_i = \frac{Y(o)Q(o) - X(o)S(o)}{Q(o)U(o) - R(o)S(o)}$  \hspace{3cm} (18)

Substituting (14) and (15) into (17) and (18), it can be found that

$k_p = (\omega^2 N_D_e + N_D_s) \cos(o) + \omega(N_D_s - N_D_e) \sin(o)$  \hspace{3cm} (19)

and

$k_i = \omega^2 (N_D_e - N_D_s) \cos(o) - \omega(N_D_s + \omega N_D_e) \sin(o)$  \hspace{3cm} (20)

The stability boundary locus, $l(k_p,k_i,o)$, can be constructed in the $(k_p,k_i)$-plane using (19) and (20). Once the stability boundary locus has been obtained then it is necessary to test whether stabilizing controllers exist or not since the stability boundary locus, $l(k_p,k_i,o)$ and the line $k_1 = 0$ may divide the parameter plane $(k_p,k_i)$-plane into stable and unstable regions. Here, the line $k_1 = 0$ is the boundary line obtained from substituting $o = 0$ into (9) and equating it to zero since a real root of $\Delta(s)$ of (9) can cross over the imaginary axis at $s = 0$ (Ackermann and Kaesbauer, 2003).

Example 1:

Consider the control system of Fig. 2 with the fractional order transfer function

$G_p(s) = \frac{1}{s^{1.3}+1} \alpha^s = \frac{1}{s^{1.3}+1} e^{-1}$  \hspace{3cm} (21)

The aim is to compute all stabilizing PI controllers for the real system of (21) and its approximate models. Thus, whether the stability regions obtained from approximate models correspond to the actual stability region or not will be investigated. Substituting $s = j\omega$ in (21) and using $j\omega^2 = \omega \cos(2\alpha) + j\sin(2\alpha)$  \hspace{3cm} (22)

The following equations can be found

$k_p = (0.454\omega^{1.3} - 1) \cos(o) + 0.891\omega^{1.3} \sin(o)$  \hspace{3cm} (23)

and

$k_i = 0.891 \omega^{2.3} \cos(o) + \omega - 0.454 \omega^{2.3} \sin(o)$  \hspace{3cm} (24)

All stabilizing PI controllers using the first, second, third and fourth order approximations of $s^{0.3}$ have been computed and all of them are shown in Fig. 3 together. For example, for his first order approximation

$k_p = (-1.46\omega^{3.46} \cos(o) + \omega) - 1.86\omega^{3.46} \sin(o)$  \hspace{3cm} (25)

and

$k_i = -1.86\omega^{3.46} \cos(o) - (-1.46\omega^{3.46} \sin(o)$  \hspace{3cm} (26)

is obtained. Using the stability boundary locus approach, the stability region for first order approximation can be computed. Similarly, the stability regions for other order of approximation can be computed.

From the Fig. 3, one can observe that fourth order approximation exactly matches the real system. Third order approximation also nearly reaches the exact system's stability region. However, first and second order systems do not give exact stability region. From this observation, one can say that while analyzing and designing fractional order control systems, using third or fourth order approximate integer models will meet the necessary requirements. Even, it can be said that for designing a PI controller which provide required performance, the first order integer approximate may be enough. Because the controllers near the boundary of the stability region will not give a good step response.
For example searching over the stability region, the step responses shown in Fig. 4 for \( k_p = 0.5 \) and \( k_i = 0.5 \) have been obtained. Clearly, these values are common for all approximate and real system’s stability regions. From Fig. 4, it can be seen that step responses of second, third and fourth orders are very close to each other. However, the step response of first order is slightly different.

\[
\begin{align*}
\omega^2(N_o D_o - N_o D_o) \cos(\omega r) - \alpha(N_o D_o + \omega^2 N_o D_o) \sin(\omega r) \\
= -k_i \omega^2 (N_i^2 + \omega^2 N_i^2) - (N_o^2 + \omega^2 N_o) \\
\end{align*}
\]

(28)

For the computation of stabilizing boundary locus in the \((k_p, k_d)\) plane in terms of \(k_i\), it has been calculated that \(k_p\) is again the same as (19) and

\[
\begin{align*}
\omega^2(N_o D_o - N_o D_o) \cos(\omega r) - \alpha(N_o D_o + \omega^2 N_o D_o) \sin(\omega r) \\
= +k_i \omega^2 (N_i^2 + \omega^2 N_i^2) + \omega^2 (N_o^2 + \omega^2 N_o) \\
\end{align*}
\]

(29)

Thus, using (19), (28) and (29) the stability regions in the \((k_p, k_i)\) plane and \((k_p, k_d)\) plane can be obtained. Using these stability regions, a new approach for computation of all stabilizing \(k_p\), \(k_i\) and \(k_d\) values is given as illustrated in the following example (Tan, et. al., 2006).

Example 2:

Consider a system with the transfer function

\[
G_p(s) = \frac{-5s+1}{(s^2+1)(2s+1)} e^{-0.6s}
\]

(30)

which is a process with a right half plane zero. The aim is to find the limiting values of the parameters of a PID controller such that the resulting closed loop system is stable. To obtain the stability region in the \((k_p, k_d)\) plane in terms of \(k_d\), it can be found (19) and (28) that

\[
\begin{align*}
\left(0.309\omega^2 - 2.3775\omega^2 - 0.951\omega^2 - 2.5\omega^2\right)\sin(0.6\omega) \\
+ (0.951\omega^2 + 0.7725\omega^2 - 0.951\omega^2 - 2.5\omega^2)\sin(0.6\omega) \\
= -(1+0.25\omega^2) \\
\end{align*}
\]

(31)

and

\[
\begin{align*}
(0.951\omega^4 + 0.7725\omega^4 - 0.951\omega^4 - 2.5\omega^4)\cos(0.6\omega) \\
- (0.309\omega^4 - 2.3775\omega^4 - 0.951\omega^4 - 2.5\omega^4)\cos(0.6\omega) \\
= -(1+0.25\omega^4) \\
\end{align*}
\]

(32)

The stability regions for \(k_d = 0\) and \(k_d = 0.6\) are shown in Fig. 5. For the computation of stabilizing regions in the \((k_p, k_d)\) plane for fixed \(k_i\), it has been calculated that \(k_p\) is the same as (31) and from (29), one can obtain

\[
\begin{align*}
(0.951\omega^2 + 0.7725\omega^2 - 0.951\omega^2 - 2.5\omega^2)\cos(0.6\omega) \\
- (0.309\omega^2 - 2.3775\omega^2 - 0.951\omega^2 - 2.5\omega^2)\cos(0.6\omega) \\
= -(1+0.25\omega^2) \\
\end{align*}
\]

(33)

The stability regions for \(k_i = 0.1\) and \(k_d = 0.6\) are shown in Fig. 6. Since it is known that the stability region in the \((k_i, k_d)\) plane for a fixed value of \(k_p\) is a convex polygon (Datta, et al., 2000), it is possible to obtain this polygon using Figs. 5 and 6. From Fig. 5 it can be seen that there are two stability regions in the \((k_p, k_i)\) plane obtained for \(k_d = 0\) and \(k_d = 0.6\), therefore, one can obtain equations of two straight lines which contribute to the boundary of the stability region for each value of \(k_p\). For example, it is clear from Fig. 5 that
the line $k_p = 1$ intersect with the boundary of the stability region for $k_d = 0$ when $k_i = 0$ and $k_i = 0.802$. Similarly, the line $k_p = 1$ intersect with boundary of the stability region for $k_d = 0.6$ when $k_i = 0$ and $k_i = 1.065$. So, one straight line passes through the points $(k_i,k_d) = (0.802,0)$ and $(k_i,k_d) = (1.065,0.6)$ and other straight line passes through the points $(k_i,k_d) = (0.0)$ and $(k_i,k_d) = (0.6,0.6)$ for $k_p = 1$. Thus the equations of these two straight lines are

\[ l_1; k_d = 2.2814k_i - 1.83 \quad \text{and} \quad l_2; k_i = 0 \]

Similarly, from Fig. 6, it is also clear that one straight line going through the points $(k_i,k_d) = (0.1,-1.6)$ and $(k_i,k_d) = (0.6,-0.46)$ and the other going through $(k_i,k_d) = (0.1,3.036)$ and $(k_i,k_d) = (0.6,3.185)$ for $k_p = 1$. The equations of these two straight lines are

\[ l_3; k_d = 2.28k_i - 1.828 \quad \text{and} \quad l_4; k_d = 0.298k_i + 3 \]

As seen from (34) and (35), lines $l_1$ and $l_3$ are the same.

These four lines are shown in Fig. 7 where the shaded region is the stability region in the $(k_i,k_d)$ plane for $k_p = 1$. Repeating the procedure of different values of proportional gain, $k_p$, the stabilizing $k_p$, $k_i$ and $k_d$ values can be shown in the three-dimensional plot of Fig. 8. The step responses of the system for different values of $k_p$, $k_i$ and $k_d$, which are obtained from the stability domain of Fig 8, using Simulink are shown in Fig. 9.

It can be seen that all stabilizing PID controllers can be computed graphically with the proposed approach once the stability regions in the $(k_p,k_i)$ plane and $(k_p,k_d)$ plane have been plotted. Since a graphical approach is adopted, generating the set of stabilizing PID controllers without human interactions seems to be difficult. However, one can easily overcome this difficulty with the aid of specialized software programs such as MATLAB. It is also clear that to obtain the complete stability domain, one must resort to a grid on $k_p$. However, the range of $k_p$ over which gridding needs to be done can be reduced using Figs. 5 and 6. It will be a very good result to obtain the complete stability domain without gridding but this seems impossible for PID.

\[ \begin{align*}
\text{Stability Region} \\
\end{align*} \]

5. CONCLUSIONS

In this paper, the integer order approximations of fractional order operators have been first studied. Equations for first, second, third and fourth order approximations have been derived. A table showing first, second, third and fourth order approximations of $s^a$ for $a = 0.1,0.2,\ldots,0.9$ has been provided which will be very important for the studies in the field. Then, the extension of the stability boundary locus method for computation of all stabilizing PI and PID controllers to
the control systems with fractional order transfer functions with time delay has been given. It has been seen that approximate models are quite appropriate for the computation of stability regions. Especially, third and fourth order approximations are exactly matches to the real system. The CFE method has been used for obtaining approximate transfer functions. Other approximate methods can be investigated for the future work.

![Step responses for different values of \(k_p\), \(k_i\) and \(k_d\)](image)

**Fig. 9:** Step responses for different values of \(k_p\), \(k_i\) and \(k_d\)

**REFERENCES**


