Research article

ANALYTICAL ALGORITHM FOR COMPUTING GAIN AND PHASE MARGINS FOR DISCRETE TIME SYSTEM

Esmat Bekir
PhD Consulting Engineers, 5805 Serrania Avenue, Woodland Hills, 91367, California, USA

Received: 5 Sep 2018 Revised: 9 Nov 2018 Accepted: 20 Nov 2018 Online available: 25 Dec 2018
Handling Editor: Kemal Mazanoğlu

Abstract

This note describes an algorithm for computing the gain and phase margins for single input single output discrete time systems. It converts the complex computations into real computations. It results into Chebyshev polynomials from which we compute the desired margins. The algorithm is simple, easy to implement and devoid of complex computations.

Keywords: Gain margin; phase margin; control system stability; discrete time system; Chebyshev polynomials.

©2018 Usak University all rights reserved.

1. Introduction

The gain and phase margins of a control system play a central role in determining its stability [1,2]. The gain margin is the amount of amplification that the closed loop system can accommodate before becoming unstable and likewise for the phase margin. Analytical computation of these two margins for continuous time systems is a direct process as shown in references [3-5]. However, this process is a little more elaborate for discrete time systems. The intent here is to describe a simple algorithm for computing these margins for these systems.

A causal single input single output discrete time system is generally described by the proper rational polynomial:

\[ G(z) = \frac{B(z)}{A(z)} \]  

(1)
where \( A(z) \) and \( B(z) \) are polynomials of the time delay \( z \), and causality mandates that the degree of \( A(z) \) be at least as large as that of \( B(z) \). With no loss of generality, we assume this system is imbedded in the forward part of a feedback control loop. In such case the closed loop system must be analyzed to ensure its stability.

Because instability implies that the roots of the characteristic equations leave the unit circle, \( z \) is confined to \( z = e^{i\theta} \), where \( 0 \leq \theta \leq 2\pi \), \( \theta \) is in radians. The gain margin, \( G_m \), is defined as the real value providing

\[
G_m G(z_{cg} = \exp i\theta_{cg}) = -1
\]

(2)

where \( \theta_{cg} \) is the angle at which \( \angle G(z_{cg} = \exp i\theta_{cg}) = -\pi \) is for the smallest normalized frequency. (Normalized frequency is frequency normalized by the sampling rate.) Likewise the phase margin, \( P_m \), is defined as the phase lag that can be tolerated before the control system becomes unstable, i.e:

\[
P_m = \angle G(z_{cp} = \exp i\theta_{cp}) + \pi
\]

(3)

where \( \theta_{cp} \) is the angle at which \( |G(z_{cp} = \exp i\theta_{cp})| = 1 \) is for the smallest normalized frequency. One approach to compute these margins is to evaluate \( G(z) \) for many values of \( z \) and generate two lookup tables; \( z \) vs. \( |G(z)| \) and \( z \) vs. \( \angle G(z) \). From the 2nd table, we lookup \( z_{cg} \). From the 1st table, we lookup \( |z_{cg}| \) and use Eq. (2) to compute \( G_m \). Similarly from the 1st table, we lookup \( z_{cp} \). From the 2nd table we lookup \( \angle G(z_{cp}) \) and use Eq. (3) to compute \( P_m \). Evidently, this process uses complex computations and gives approximate margins, but it may be adequate for speedy computations. Matlab uses the H- theory and other methods to compute these margins [6]. Herein, we introduce a simplified analytical method that is free of complex computations to yield the same results.

To start, we introduce some mathematical background in which we define symmetric and asymmetric polynomials, describe their properties and provide alternative algebraic means for representing them. These types of polynomials play a central role in the computations of the gain and phase margins.

### 2. Symmetric and asymmetric polynomials

The polynomial,

\[
P(z) = \frac{1}{2}(p_1 z^n + p_2 z^{n+1} + \ldots + p_{n+1} z^n + p_1 z^{-n})
\]

(4)

is symmetric if \( P(z) = P(1/z) \Rightarrow p_k = p_{-k}; k = 1, \ldots, n \). Thus Eq. (4) becomes as follows:

\[
P(z) = \frac{1}{2}[p_1(z^n + z^{-n}) + p_2(z^{n+1} + z^{-n+1}) + \ldots + p_{n+1}]
\]

(5)

The polynomial,

\[
Q(z) = \frac{1}{2i} \left( q_1 z^n + q_2 z^{n+1} + \ldots + q_{n+1} z^{-n} \right)
\]

(6)

is asymmetric if \( Q(z) = -Q(1/z) \Rightarrow q_k = -q_{-k}; k = 1, \ldots, n \). Thus Eq. (6) becomes:

\[
Q(z) = \frac{1}{2i} [q_1(z^n - z^{-n}) + q_2(z^{n+1} - z^{-n+1}) + \ldots + q_n(z - z_{-1})], \quad q_{n+1} = 0
\]

(7)
Since we are confining our analysis to \( z = e^{i\theta} \) one can see that:

\[
\begin{align*}
  z^k + z^{-k} &= 2\cos k\theta \\
  z^k - z^{-k} &= 2i\sin k\theta
\end{align*}
\]

which implies that \( P(\theta) \) and \( Q(\theta) \) take following trigonometric forms:

\[
\begin{align*}
P(\theta) &= p_1 \cos n\theta + p_2 \cos (n-1)\theta + \ldots + 0.5p_{n+1} \\
Q(\theta) &= q_1 \sin n\theta + q_2 \sin (n-1)\theta + \ldots + q_n \sin \theta
\end{align*}
\]

Thus, on the unit circle the symmetric and asymmetric polynomials are purely real polynomials. Our ultimate goal is to represent \( P(\theta) \) and \( Q(\theta) \) in power series form of \( x = \cos \theta \). First, we need to convert the sines in Eq. (10) to cosines as shown below. We observe that,

\[
\sin k\theta = 2 \sin \theta \cos(k-1)\theta + \sin(k-2)\theta
\]

and substituting from Eq. (11) for the first term of \( Q(\theta) \) in Eq. (10) gives:

\[
\begin{align*}
Q(\theta) &= 2q_1 \sin \theta \cos(n-1)\theta + q_2 \sin (n-1)\theta \\
&\quad + (q_1 + q_2) \sin (n-2)\theta + \ldots + q_n \sin \theta
\end{align*}
\]

Continuing the substitution from Eq. (11) into Eq. (12) results in,

\[
Q(\theta) = 2 \sin \theta [q_1' \cos(n-1)\theta + q_2' \cos(n-2)\theta + \ldots + q_{n-1}' \cos \theta + 0.5q_n']
\]

which makes use of following recursion conditions:

\[
q_j' = q_{j-2} + q_j, \quad j = 3, 4, \ldots, n, \quad q_1' = q_1, \quad q_2' = q_2
\]

The above polynomials take the more concise forms as follow:

\[
\begin{align*}
P(\theta) &= p\ c \\
Q(\theta) &= 2\sin \theta \ q\ c_o
\end{align*}
\]

where \( p, q, c \) and \( c_o \) are vectors given below:

\[
\begin{align*}
p &= [p_1 \ p_2 \ldots \ 0.5p_{n+1} ] \\
q &= [q_1' \ q_2' \ldots \ 0.5q_n'] \\
c &= [\cos n\theta \ \cos(n-1)\theta \ \ldots \ \cos \theta \ \ 1]^T \\
c_o &= [\cos(n-1)\theta \ \cos(n-2)\theta \ \ldots \ \cos \theta \ \ 1]^T
\end{align*}
\]

Even though Eqs. (15) and (16) are amenable to real number computations, they are inadequate for computing their roots. They need to be transformed into powers of \( x = \cos \theta \). One may realize that the entries of \( c \) and \( c_o \) in Eq. (18) are nothing but Chebyshev polynomials, \( T_n(x) = \cos n\theta \), can be transformed to the desired powers as given in references [7,8]. For example, in a matrix form, \( c \) in Eq. (18) can be represented by:

\[
c = V x
\]

where \( x \) is a column vector given as:
\[ \mathbf{x} = [x^n \ x^{n-1} \ldots \ 1]^T \]  

(20)

and \( V \) is the transformation matrix. Numerical values of \( V \) for few degrees are listed in references [7,8]. However this approach requires either to store the above matrix or to generate it in the computer; both are not very attractive. A more appealing approach is using the identity, \( T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x), \ T_0(x) = 1, \ T_1(x) = x, \) where \( x = \cos \theta. \) Applying this identity repetitively, degree of the \( T_n \) term goes down while power of \( x \) goes up. An in-place procedure [8],

\[
\begin{align*}
&\text{for } i = n - 1 \text{ down to 1 and for } j = 1, \ldots, i \\
&p(j + 2) = p(j + 2) - p(j); \\
&p(j) = 2p(j);
\end{align*}
\]

(21)

to the \( n^{th} \) degree polynomial whose coefficients are as given in Eq. (17) yields the desired transformation.

Now, we describe the method for computing the gain margin.

3. Procedure for computing the gain margin

To carry out the computation of \( G_m \) in real arithmetic, we multiply both terms of Eq. (2) by \( A(1/z) \), which will now become as follows:

\[
G_m B(z)A(1/z) + A(z)A(1/z) = 0
\]

(22)

Evidently,

\[
D(z) = A(z)A(1/z)
\]

(23)

is real and

\[
N(z) = B(z)A(1/z)
\]

(24)

is complex number for which the real and imaginary parts are as follow:

\[
N_r(z) = [N(z) + N(1/z)]/2
\]

(25)

\[
N_i(z) = [N(z) - N(1/z)]/2i
\]

(26)

The modified characteristic equation now becomes:

\[
G_m(N_r + iN_i) + D = 0
\]

(27)

Therefore, the real and imaginary parts of Eq. (27) are written as:

\[
G_mN_r + D = 0
\]

(28)

\[
G_mN_i = 0
\]

(29)

The above equation shows that the gain margin is determined at \( z_{cg} \) at which \( N_i = 0, \) and for this value of \( z, \) the gain margin is found by computing following expression.

\[
G_m = -D(z_{cg}) / N_r(z_{cg})
\]

(30)
The polynomial $N_i$, in Eq. (26), is asymmetric and its roots can be found as follows: Let $n$ and $m$ be the degrees of $A$ and $B$. Eqs. (24) and (26) yield:

$$N_i(z)2/ i = N(1/z) - N(z) = B(1/z) A(z) - B(z) A(1/z)$$

(31)

To make the above in positive powers of $z$, let $B(1/z) = z^{-m} \bar{B}(z), A(1/z) = z^{-n} \bar{A}(z), C(z) = \bar{B}(z)A(z)$, and $\bar{C}(z) = \bar{A}(z)B(z)$. (Note that the coefficients of $\bar{B}(z)$ and $\bar{A}(z)$ are the reverse of $B(z)$ and $A(z)$ respectively.) This implies Eq. (32),

$$N_i(z)2/ i = z^{-m}C(z) - z^{-n}\bar{C}(z)$$

(32)

where $C(z)$ is given as:

$$C(z) = \bar{B}(z)A(z) = \sum_{j=1}^{n+m+1} c_j z^{m+n+1-j}$$

(33)

Coefficients of $\bar{C}(z)$ are the reverse of $C(z)$, $\bar{c}_j = c_{m+n+2-j}$, $j = 1, ..., m + n + 1$. Thus:

$$\bar{C}(z) = \sum_{j=1}^{n+m+1} \bar{c}_j z^{m+n+1-j} = \sum_{j=1}^{n+m+1} c_{m+n+2-j} z^{m+n+1-j} = \sum_{k=1}^{n+m+1} \bar{c}_k z^{-1+k}$$

(34)

Substituting for the $C(z)$ and $\bar{C}(z)$ polynomials from Eqs. (33) and (34) into Eq. (32) yields:

$$N_i(z)2/ i = z^{-m} \sum_{j=1}^{n+m+1} c_j z^{m+n+1-j} - z^{-n} \sum_{j=1}^{n+m+1} c_j z^{-1+j} = \sum_{j=1}^{n+m+1} c_j (z^{n+1-j} - z^{-n+1+j})$$

(35)

The above summation can be simplified if it is split into three summations:

$$N_i(z)2/ i = \sum_{j=1}^{n-m} c_j (z^{n+1-j} - z^{-n-1+j}) + \sum_{j=n-m+1}^{n} c_j (z^{n+1-j} - z^{-n-1+j})$$

$$+ \sum_{j=n+1}^{n+m+1} c_j (z^{n+1-j} - z^{-n-1+j})$$

(36)

Applying $k = 2n+2-j$ to the last term, noting that $j = n+1$ term vanishes, gives:

$$\sum_{j=n+2}^{n+m+1} c_j (z^{n+1-j} - z^{-n-1+j}) = \sum_{k=n}^{n+m+1} c_{2n+2-k} (z^{n-1+k} - z^{n+1-k})$$

(37)

Substituting back into Eq. (36) gives:

$$N_i(z)2/ i = \sum_{j=1}^{n-m} c_j (z^{n+1-j} - z^{-n-1+j}) + \sum_{j=n-m+1}^{n} (c_j - c_{2n+2-j}) (z^{n+1-j} - z^{-n-1+j})$$

(38)

Thus, Eq. (38) can be simplified to Eq. (39).

$$N_i(z) = -\sum_{j=1}^{n} q_j \sin(n + 1 - j) \theta$$

(39)

$$q_j = c_j, \quad j = 1, ..., n - m$$

$$q_j = c_j - c_{2n+2-j}, \quad j = n - m + 1, ..., n$$

Precisely following the above steps for $N_r$, we reach to following solution.
\[ N_r(z) = \sum_{j=1}^{n+1} q_j \cos(n + 1 - j) \theta \]

\[ q_j = c_j, \quad j = 1, \ldots, n - m \]

\[ q_j = c_j + c_{2n+2-j}, \quad j = n - m + 1, \ldots, n \]

\[ q_j = c_{n+1}, \quad j = n + 1 \]

(40)

Therefore, the procedure for determining the gain margin is summarized below:

i. Compute the coefficients \( c_j \) of the \( C \) polynomial from the coefficients of \( A \) and \( B \) using Eq. (33).

ii. Construct polynomial \( N_i \) from Eq. (39) and solve for its roots to get \( z_{cg} \).

iii. Construct polynomial \( D \) from Eq. (23) and \( N_r \) from Eq. (40), evaluate at \( z_{cg} \).

iv. Determine the gain margin from Eq. (30).

Next, we describe the process for computing the phase margin.

4. Procedure for computing the phase margin

Computing the phase margin, \( P_m \), is governed by Eq. (3). It shows that the amplitude of \( G(z) \) equals to 1, at \( z_{cp} \) for which \( G(z_{cp})G(1/z_{cp}) = 1 \). Thus, we solve Eq. (41).

\[ W(z_{cp}) = A(z_{cp})A(1/z_{cp}) - B(z_{cp})B(1/z_{cp}) = 0 \]

(41)

Similar to Eqs. (22)-(27), we see the equation given below.

\[ G(z) = (N_r + iN_i) / D \]

(42)

Hence, \( \angle G \) becomes,

\[ \angle G = \arctan(N_i/N_r) \]

(43)

where \( N_r \) and \( N_i \) are computed at \( z_{cp} \). The phase margin is then found as:

\[ P_m = \angle G + \pi \]

(44)

The procedure for determining \( P_m \) is summarized here:

i. Construct \( W \) from Eq. (41) and solve for its roots to get \( z_{cp} \).

ii. Evaluate \( N_r \) and \( N_i \) at \( z_{cp} \).

iii. Compute the phase margin from Eq. (44).

One observes that polynomial, \( W \), is symmetric. Again its roots can be found by converting it to a Chebyshev polynomial and determine its roots.

5. Application of the procedure

To expedite the derivation of this algorithm, we treated the discrete time transfer function (DTTF), from the outset, as a theoretical mathematical entity without regards to where it is originated from or how its results would be applicable. Now, we address some applicable practical issues. In practice, DTTF usually results from sampling linear continuous time system at a constant time interval \( T \) and transforming them to DTTF. There are various transformation methods, and one of them is the bilinear transformation. The Laplace variable \( s = i\omega \) is now transformed into the delay \( z = e^{i\omega T} = e^{i\omega T} \). Hence, \( \theta \) represents the
frequency $\omega$ normalized by the sampling rate, $1/T$; and the frequency, $\omega$, is obtained by dividing $\theta$ by $T$. Now, from Eqs. (2) and (3), the gain and phase cross over frequencies are given by $\omega_{cg} = \theta_{cg}/T$ and $\omega_{cp} = \theta_{cp}/T$. To describe the procedure, we shall use following example.

$$G(z) = \frac{0.04798z + 0.0464}{z^3 - 1.41z^2 + 0.1808z + 0.36}, \quad T = 0.1$$

(45)

The 4th element of denominator, $A(4)$, will be used as a variable that ranges from 0.3 to 0.36. The gain and phase margins are computed and plotted, against the variable $A(4)$, using the above analytical algorithm vs. the Matlab 'margin' function as shown in Figs. 1 and 2. Both figures show that, practically, differences between the results of methods are eye indistinguishable.

![Fig. 1 Gain margin vs A(4) graph.](image1)

![Fig. 2 Phase margin vs A(4) graph.](image2)
Figs. 3 and 4 depict the differences between these two solutions for margin parameters against the variable $A(4)$. Both figures show that these differences are minor. For example, when $A(4) = 0.36$, the cross over frequencies and the gain and phase margins found by the analytical algorithm are,
\[
\omega_{cg} = 4.1959 \text{ rad/s} \quad G_m = 1.2179 \\
\omega_{cp} = 3.9846 \text{ rad/s} \quad P_m = 6.6869 \text{ deg}
\]
while those given by the Matlab function 'margin' are as follows:
\[
\omega_{cg} = 4.1960 \text{ rad/s} \quad G_m = 1.2181 \\
\omega_{cp} = 3.9846 \text{ rad/s} \quad P_m = 6.6861 \text{ deg}
\]

![Fig. 3 Gain margin error vs A(4) graph.](image)

![Fig. 4 Phase margin error vs A(4) graph.](image)
As seen, the results of both methods match very closely. However, this analytical method is programmable and can be implemented in any language and can be imbedded in any design. Despite the appearance of trigonometric functions everywhere, they are only used when computing the cross over frequencies and in computing the phase margin in Eq. (43).

6. Summary and conclusions

This paper has presented a numerical algorithm for computing the gain and phase margins for a discrete time system. The algorithm entirely uses real computations. Inputs to this algorithm are just the coefficients of the numerator and denominator of the polynomials of the discrete system. The method is programmable and non-iterative; therefore, it can be imbedded in procedures for computing the design and stability of discrete time systems. Computationally, when compared with the Matlab function ‘margin’, it provides almost identical results.

Acknowledgement

The author is very grateful to the anonymous reviewers for thoroughly reviewing and editing the paper and for Prof. N. Bekir and A. Bekir for editing the paper and suggesting many improvements.

References