The Role of Positive Real Functions in A/D Transfer Function Matching

Bill R. McEachen

August 1988

A paper submitted in partial fulfillment of the requirements for the Masters in Electrical and Computer Engineering

George Mason University

Fairfax, VA

Abstract

Many adequate methods exist for conversion of transfer functions in the continuous domain to discrete equivalents with preservation of magnitude. Frequency response in the true sense however is composed of both magnitude and phase components with characteristics input response in the time domain dependent on both parts. In an effort to eliminate acceptance of prediction error associated with phase distortion between continuous and discrete "equivalents", this work strives for a transformation yielding complete transfer function (frequency response) match. Correlation is shown between operating on positive real continuous transfer functions and successful duplication of frequency response in the discrete domain. This work continues an earlier research effort.

Contents

1 List of Tables and Figures 4

2 Introduction 5

3 Discussion 6

4 Background 7

5 Modified Approach 8

6 Error Analysis 12

7 Results 13

8 Applications 14

9 Conclusions and Recommendations 21

10 Acknowledgments 22

11 References 23

12 Appendix A 24

13 Appendix B 25

14 Appendix C 27

15 About the Author 30

1 List of Tables and Figures

- Figure 1 Divergence Behavior of the Base Method
- Figure 2 Modification to the Base Method
- Figure 3 Problem PR3 Frequency Response
- Figure 4 Problem PR4 Frequency Response
- Figure 5 Problem 9 Frequency Response
- Figure 6 Problem 11 Frequency Response
- Table 1 Representative Problems
- Table 2 Discrete Equivalents
- Figure 7 Representative Results
- Table 3 Possible Cases
- Figure 8 Analog vs Orig vs Modified
- Figure 9 Analog vs Bilinear vs modified z-transform

2 Introduction

The objective of this work was to both assess and continue the progress of the research started over 10 years ago by Beale and Cook (1). That research dealt with complete transfer function mapping between the continuous and discrete domains. This work attempts to better explain the intermediate results of ref(1), with the desired goal having been to assess the validity of the base approach. Unless conclusive investigation indicated irreparable flaws, the base transformation method would be left intact.

Every effort was made to prevent this work from becoming bogged down by its programming aspects. However it was realized that automation of the transformation procedure of ref(1) was necessary to allow efficient examination of various aspects of the problem. The code allows for automatic generation of the discrete equivalent to an input continuous transfer function H(s) and performs all calculations needed to perform adequate error analysis on the numerous iterations. All written code used for this report is available though it is not optimized by any means and is slanted towards this author's particular situation/approach. Note that the deviations of the method in ref(1) from a standard bilinear transformation could be incorporated into commercial software such as MATLAB relatively easily. Note that the open source alternative Octave is used in this paper [**https://www.gnu.org/software/octave/**].

3 Discussion

For completeness, a brief check was carried out on ref(1)'s approach to p- to z-plane in terms of changes to natural frequency and damping ratio. After some investigation it was concluded that the approach of ref(1) was proper. It should be noted that the allowance for adjustment of an H(s) pole discussed in ref(1) was not replicated in this work, the seed H(s) being taken as inflexible.

The major questions outstanding in ref(1) are:

- for what class of function(s) does the procedure prove useful?
- why were some of the paper's results less than optimal and could anything be done to ameliorate the situation?

This follow up work shows:

- at least one class of function exists for which the procedure in ref(1) works extremely well,
- how the procedure in ref(1) exhibits a near universal inconsistency with typical non-positive real functions, and
- a proposed modification to the method in ref(1) for such "ill-conditioned" functions.

At the start of this research, this author thought that the heart of the inconsistency lay either with an inadequacy of the bilinear portion of the base transformation or with the specific application of the all-pass filtering. Subsequent work played down these ideas, focusing instead on the function type as the dominant factor affecting transformation success.

4 Background

In ref(1), transformation is made from the s-plane to the z-plane via an intermediate frequency conversion followed by the bilinear transformation. The frequency conversion serves to map the continuous range of interest (0 to the half-sampling rate Ws/2) to one extending from 0 to infinity (i.e. all possible frequencies) on the second plane. As a quick example, a term $s^2 + 2nw + w^2$ yields a p-term $p^2+2vq+q^2$ where v=n, q=tan(wT/2), T is the sampling period, n the damping ratio and w natural frequency. The bilinear transformation term $(z-1)/(z+1)$ then replaces p to complete the conversion. After tranforming H(s) to H(z), any unconverted zeros (those stemming from infinite s-plane zeros) must then be set. Finally, an all-pass filter (APF) multiplier is introduced out of necessity to "tune" the discrete phase response as closely as possible to the continuous response.

The problem with this transformation developed towards the higher end (usually within the last 1/5th) of the frequency range of interest, with divergence of the phase portion of the frequency response exhibited. Note that with the APF term, the procedure of ref(1) has the phase terminating at a multiple of pi (180 deg) at Ws/2. This clearly has drawbacks when trying to match an arbitrary continuous phase characteristic not at all guaranteed to exhibit the same terminal phase value. Unless the continuous phase should go to -180 degrees at Ws/2 as does the discrete version with APF, a fixed phase error is introduced from Ws/2 back to the last intersection of the continuous and discerte phase curves as illustrated in Figure 1. This inherent characteristic is not satisfactory. In problems artificially constructed for optimal continuous phase characteristic for application of the method of ref(1), the transfer function match was significantly worse than expected. The modification to the procedure of ref(1) described next is an attempt to exploit the predictable aspects of this phase divergence.

5 Modified Approach

At its basic level the modified approach uses knowledge about the true desired sampling rate Ws to shift the divergence region out beyond Ws/2. This shift is accomplished by generating the discrete frequency response at a slightly higher sampling rate Ws*, while utilizing the transformation information only up to the original Ws/2 cutoff (the function transformation still using the original sampling period). The basis of the modification is acceptance of a slight degradation of the phase match from w=0 to the divergence point to gain elimination of the error contributed by the divergence region. In most cases the tradeoff is beneficial as can be seen in Figure 2. The effect on magnitude match is not as predictable however. Typical improvement of the performance index (discussed in the next section) is on the order of 1.5 to 3 depending on the H(s) characteristic, with Ws* usually within 15 percent of Ws. As a check, all problems were run at the higher sampling frequency of the modified runs to ensure that the improvement was not merely due to the faster sampling alone. The modification does not seem to be effective when the denominator and numerator powers differ by two or more, which is good in that the original method of ref(1) is least applicable when this difference is one. Table 1 lists some of the problems examined, with Table 2 listing some of the associated H(z) forms after transformation.

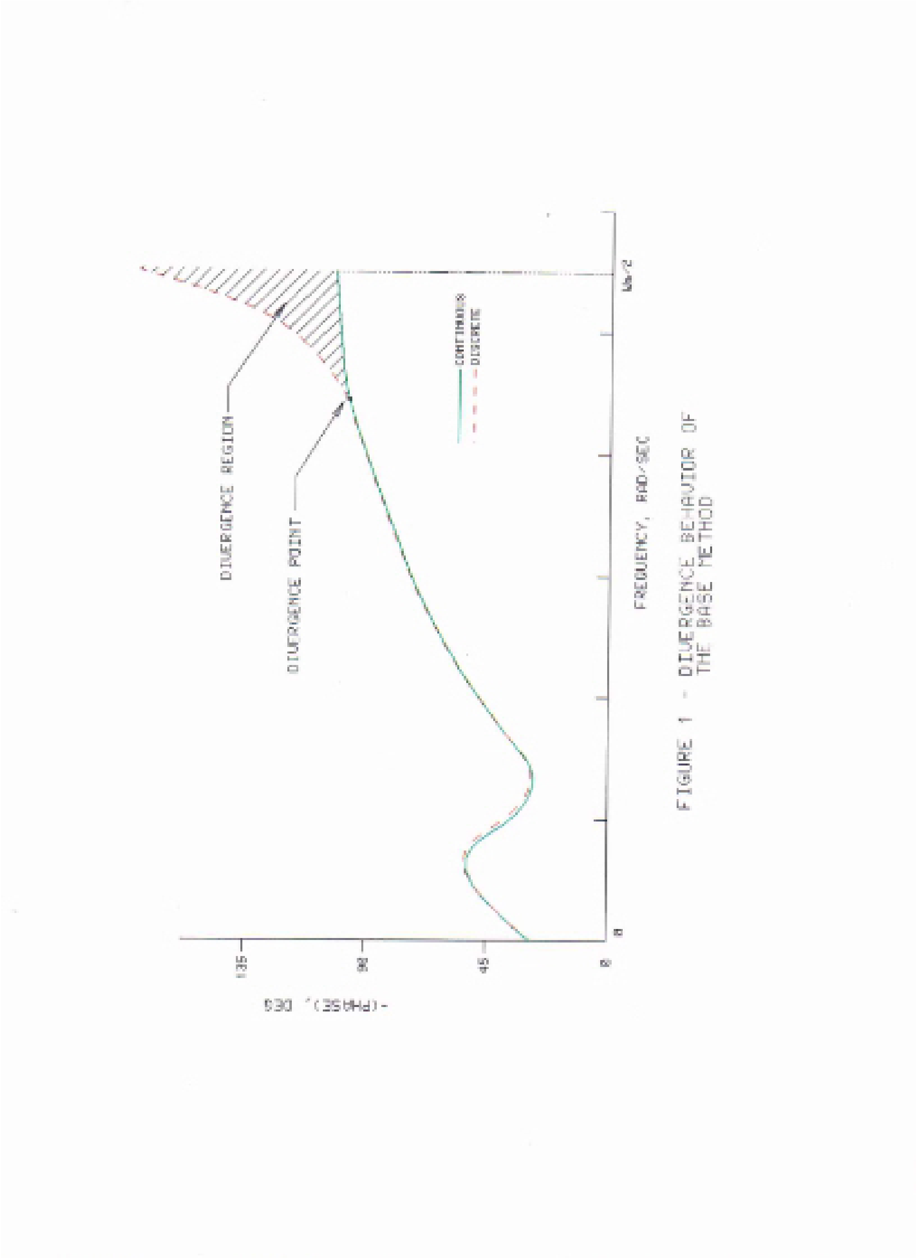

Figure 1: Divergence Behavior of the Base Method

Positive Real:
1 $[(s+1)(s+3)(s+4)]/(s+2)^3$
2 $[(s+2)(s+4)]/[(s+1)(s+3)$
3 $(s^2+s+1)/(s^2+s+4)$
4 $(s^3+2s^2+s+1)/(s^3+s^2+2s+1)$
5 $[(s+2)(s+4)(s+6)(s+8)]/[(s+1)(s+3)(s+5)(s+7)]$
6 $[(s+1.5)(s+3.5)(s+6)]/[(s+1)(s+3)(s+5)(s+7)]$
Non-Positive Real:
7 original problem (ex. 2)
8 ex. 2 less pole @ 0.62
9 ex. 2 less pole @ 5.0
10 ex. 2 less $(s^2+8.4s+36)$,zero @ 1.65
11 ex. 2 less $(s^2+7.25s+81)$,zero @ 5.0

Non-Positive Real (variations of ex.2 from ref(1)) [ex.2: $(s+1.65)(s^2+2.31s+2.72)(s^2+7.25s+81)/(s^2+8.4s+36)(s+5.0)(s+0.62)^2(s^2+11.849s+72)$]

Table 1: Representative Problems

Positive Real:
1 $[(z-0.72654)(z-0.32492)(z-0.15838)]/(z-0.50953)^3(Ws=20)$
2 $[(z-0.50953)(z-0.15838)]/(z-0.72654)(z-0.32492)(Ws=20)$
3 $(z^2-1.64755z+0.73234)/(z^2-1.41073z+0.74376)(Ws=20)$
4 $[(z-0.55902)(z^2-1.62850z+0.67539)]/[(z^2-1.71662z+0.87681)(z-0.83528)](Ws=20)$
5 $[(z-0.77568)(z-0.59140)(z-0.43274)(z-0.29053)]/$
$[(z-0.88162)(z-0.67960)(z-0.50953)(z-0.36002)](Ws=50)$
6 $[(z-0.90993)(z-0.80115)(z-0.67960)(z+1)]/$
$[(z-0.93906)(z-0.82727)(z-0.72654)(z-0.63462)](Ws=100)$

Table 2: Some of the Discrete Equivalents

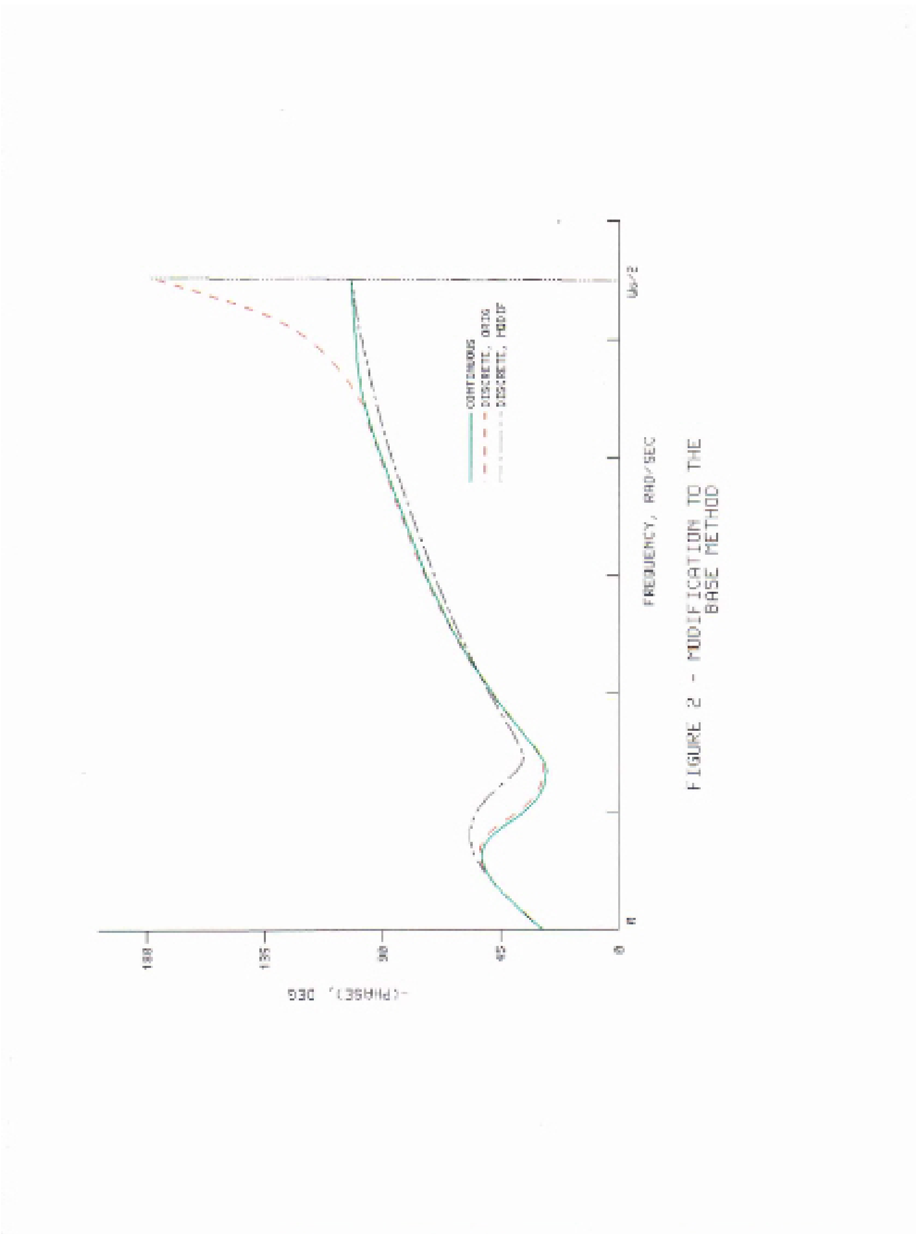

Figure 2: Modification to the Base Method

6 Error Analysis

Simple tracking of mean absolute errors and maximum errors was used as per ref(1) as well as the percentage of data falling below user set target levels. The percent tracking is useful in indicating a divergence problem which might then be circumvented - for example, an error analysis on 100 data comparisons might see very good phase matchup until the last five data (just after a divergence point). This extremely skewed data artificially raises the mean absolute errors and J index. The eventual approach after a number of preliminary analyses was to make use of a performance index J (one of many possible) which is a function of the four quantities (two mean errors and two maximum errors), with magnitude error being more heavily weighted. The weightings were as deemed appropriateby this author. The index helped quicken evaluation of error analysis runs, as often slight changes to a problem would degrade magnitude match while improving phase match or vice versa. A brief note on initial and fnal function values. In this research, none of the positive real examples examined had a constant multiplier, as the computer code was set up to solve for any required scaling while maintaining the DC gain. With non-zero DC gain this meant that the continuous and discrete magnitude curves were shifted to zero DC gain in a buffer, the optimization conducted, and the curves translated back to the original set point. It appeared that just as the matchup for positive real H(s) improves with sample rate, any required scaling for the discrete equivalent approaches unity in the limit i.e. the transformation strictly maintains the initial value (as might be expected). However, this did not appear to be the case for the respective final values. In particular, with an H(s) with one surplus pole (non-zero), the corresponding zero resulting from optimization of the response match does not match that zero which would equalize the limiting final value (final value theorem).

7 Results

Unexpected success was found upon application of the method of ref(1) to a few trial positive real (pr) functions. Note that much of the work listed in the supplemental reference list was done subsequent to the work of ref(1). Positive real functions will not be completely discussed here though some principal, easily discernible properties are:

- the highest (and lowest) powers of the numerator and denominator polynomials must be within 1 of each other,
- the coefficient signs must all be >= 0,
- the coefficients between the highest and lowest power terms must be ordered (exclusive),
- using the form $P(s) = CN * s^N + ...C1 * s + C0$, for $N > 2, C0$ must be <= the product of the next three (nonzero) highest coefficients, and
- no multiple roots are present at either the origin or infinity

Note that the above are necessary but not sufficient conditions. At first it was not clear whether the behavior was due to the pr property or simply due to the composite polynomials being Hurwitz but it is believed the former is the reason. Unfortunately in one respect, the runs for the pr cases were so good that no APF term was applied, which means that no truly successful runs with an APF term applied have been obtained. Readers should not mistakenly conclude from the pr runs that the coefficients must be integer; it is the sign and relative position between the zeros and poles which are the key. Integer problems were run because they were the ones found readily in print. The reference listed for pr functions also details how reactance functions are a class of pr functions, which is significant, as it means such functions are realizable in hardware. Also note that one of the properties of the bilinear transformation is that the transformation of a pr function is guaranteed to be bounded real, which allows generalization about preservation of stability in going from the s- to z-domain through the procedure of ref(1). Isolated results have been summarized for both pr and non-pr H(s), Figures 3 through 7. The positive real functions have the "PR" prefix. Note that a few of the transfer functions are not completely stable and are used for analysis purposes only. Also note that the modified method was not run on the pr problems, as the original method of ref(1) proved completely adequate. It appeared that any desired accuracy could be achieved by increasing the sampling rate (assuming the allowable sampling rate is that flexible).

8 Applications

If one were fortunate to be working with an H(s) either positive real or with enough flexibility to be made positive real, the transformation of ref(1) may be applied directly, with the accuracy limited only be restrictions on the sampling rate. As examples, problems PR1 through PR5 with no surplus poles yield excellent results for moderate sampling rates while problem PR6 with the surplus pole would require a sampling rate significantly higher than the Nyquist rate to achieve the same results. Most of the positive real examples are taken from ref(4). If one is working with non-positive real H(s) with an odd excess of poles, the modified version of the procedure of ref(1) would rarely yield accuracy acceptable to the user. The unmodified method of ref(1) could not yield adequate results due to its asymptotic phase characteristics. Problems 8 through 10 are examples of this scenario. If one is working with non-positive real H(s) with an even excess of poles, the modified version of the procedure of ref(1) would often yield accuracy acceptable to the user. The unmodified method of ref(1) could also yield adequate results, though far less often. The original autopilot example of ref(1) is of this type (problem 7). If one is working with an H(s) which has terms not factorable down to either 1st or 2nd order terms, the transformation of ref(1) is not so trivial. No problems of this type were analyzed. In the unlikely event that one does not need to assess the frequency response of a function up to its Nyquist frequency but to some lower frequency Wx, ref(1) could be quite adequate if Wx occurs before the divergence. Such a scenario would result from analysis of a function where only lower frequency contributions are being examined.

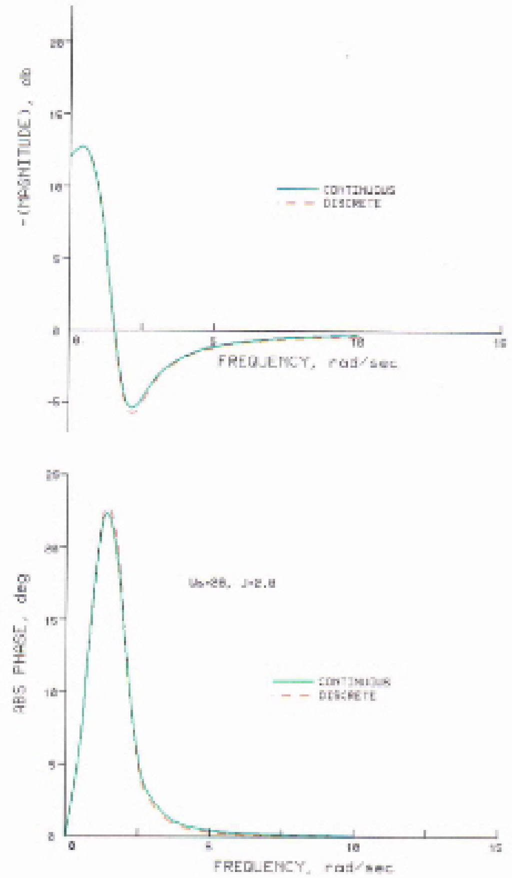

FIGURE 3 - PROBLEM PR3 FREQUENCY RESPONSE

Figure 3: Problem PR3 Frequency Response

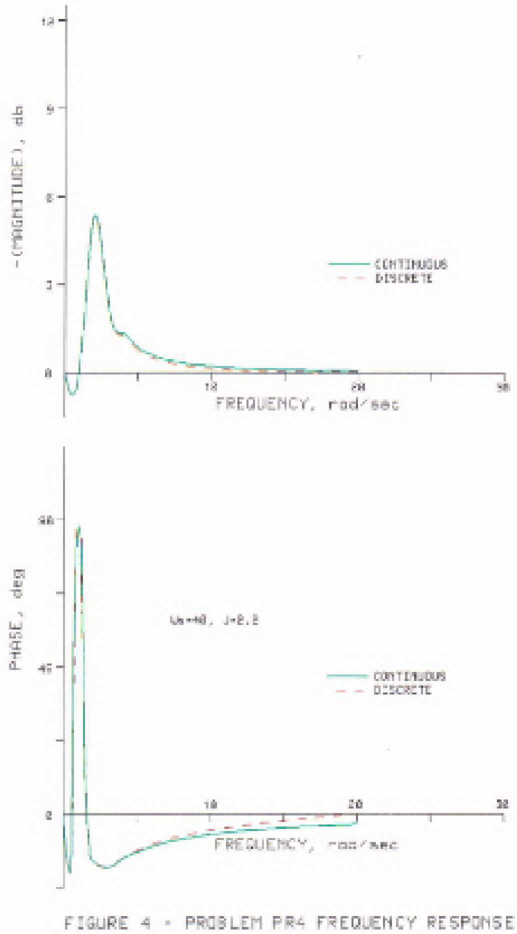

Figure 4: Problem PR4 Frequency Response

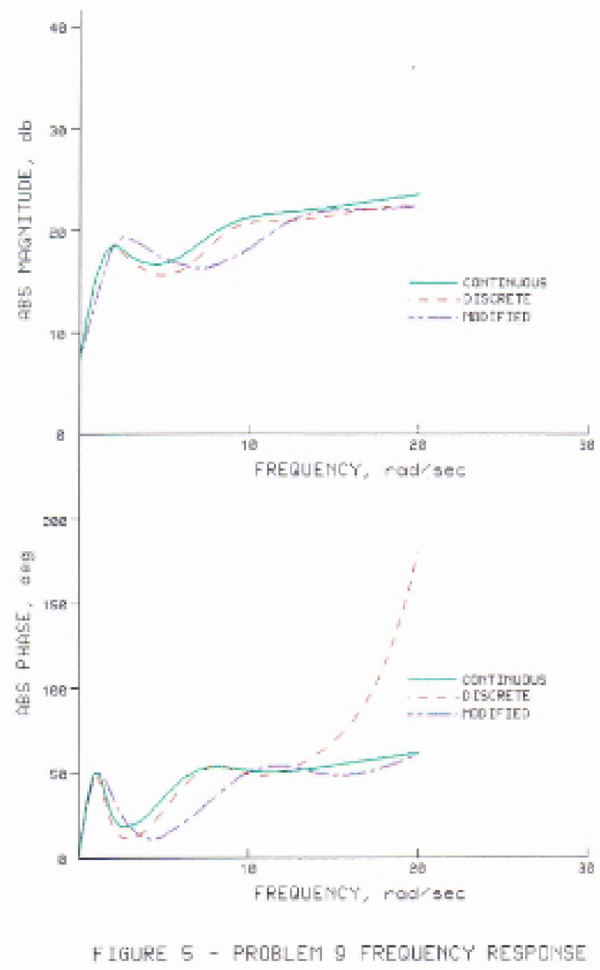

Figure 5: Problem 9 Frequency Response

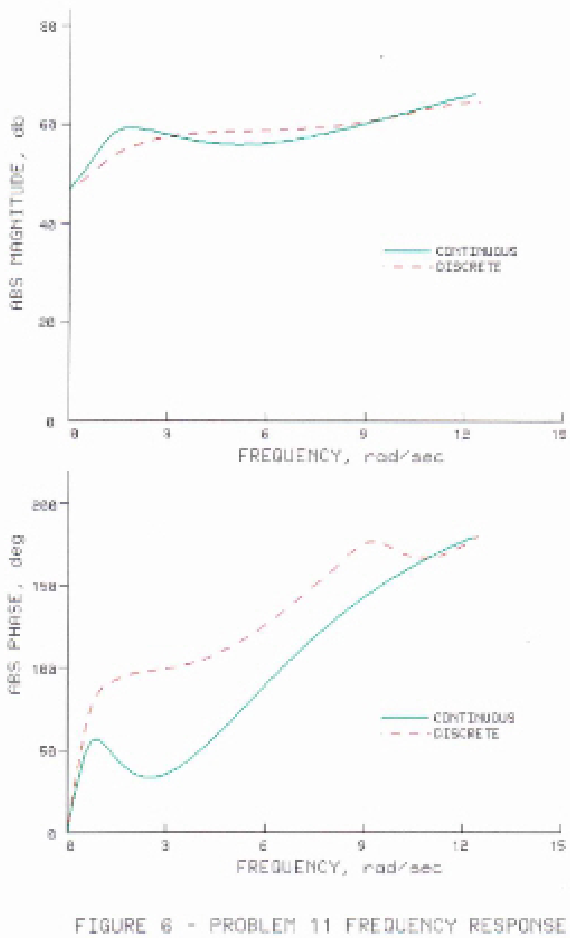

Figure 6: Problem 11 Frequency Response

TABLE 3. REPRESENTATIVE RESULTS

Function	Ws rad/s	J index	MAME db	MAPE deg	Max magn error,db	Max phase error,deg	Added zero(s)	All pass filter
PR1	20 32	5.4 3.4	0.10 0.04	3.4 2.3	0.35 0.16	10.0 6.9	none none	none none
PR2	20 80	5.7 1.6	0.15 0.01	3.1 1.2	0.39 0.03	10.5 2.9	none none	none none
PR3	20	2.0	0.12	0.6	0.49	2.4	none	none
PR4	20	5.1	0.25	2.1	0.33	5.5	none	none
PR5	50	4.8	0.10	2.8	0.31	8.8	none	none
PR6	50 100	34 25	2.57 2.10	3.9 2.1	29.9 12.4	11.2 5.3	1.0 1.0	none none
7 orig modif	40 53	16 24	0.36 1.25	8.3 8.5	1.17 4.15	42 29.5	.23 .29	0.37 0.32
8 orig modif	40 46.5	22 14	1.04 0.70	6.1 4.6	3.8 2.8	51.0 17.9	.82 .82	none none
9 orig modif	40 57	31 26	0.29 1.38	16.2 8.1	0.9 9.2	118 26.0	.27 .27	0.65 0.65
10 orig modif	40 42.2	35 19	1.57 1.33	5.1 4.1	6.1 5.6	132 10.7	.62 .62	0.95 0.95

J here taken as = C1(MAME)+C2*(MAPE)+C3*(max. magn error)+ C4*(max. phase error), magnitude and phase in db and degree respectively. Here the coefficients were chosen at (10,1,0.1,0.1).

Figure 7: Representative Results

Surplus poles H(s)	(1) method applicable	can it be positive real?	comment
0	Yes	Yes	no zeros to set, ideal if positive real
1	No	Yes	(1) inappropropriate unless positive real;use modified version (1)
2	Maybe	No	not positive real,try (1), modify as needed
Higher	Yes for even	No	(1) applies to even surplus

Table 3: Possible Cases

9 Conclusions and Recommendations

1. This paper illustrates the outstanding results of the procedure of ref(1) on positive real continuous transfer functions. To this author's knowledge and limited literature search, no other method is being promoted for such complete transfer function matching.

2. The procedure of ref(1) can be modified to produce better, although not always satisfactory results on non-positive real H(s). Future work should include examining the use of a different transformation other than the bilinear for thr last step of the A/D conversion, especially when numerator and denominator powers are matched and there are no large roots present. It is easy to show that in this instance, the provisions of the bilinear transformation developed for wider applications are such that the mapping is probably not as optimal or as advantageous as other transformations (i.e. variations on the 5,8,-1 rule of integration and similar). It is clear from Fig 6, ref(1) that the transformation before all-pass filtering does not inherently maintain phase, at least for non-positive real functions.

3. Specific situations, especially as regards allowable sampling rate, network implementation flexibility, and specific application (simulation, hardware implementation, etc) will dictate the viability of the procedures of this work and its predecessor ref(1) in individual cases.

4. Any extension of the effort discussed here to better define its limitations and applications would be extremely worthwhile, namely how to extend the success with positive real functions to less structured problems more typically confronted in actual design work.

5. When the procedure discussed does work, digital time response to various input signals should inherently improve in terms of agreement to its continuous counterpart, this having been a major objective of this work and of the general A/D process. No time response analysis was performed in the course of this work, though ref(1) does give some mention.

6. A logical extension of this work would be to incorporate the zero-order hold (ZOH) associated with digital implementation, a matter not addressed by either ref(1) or this work.

7. The author wholeheartedly recommends to any individual routinely involved with H(s) to H(z) conversions currently preserving only magnitude to assess the possible benefits of preserving both parts, especially if there will be little or no degradation in magnitude match. The transformation procedure of ref(1) is trivial to perform at least for H(s) factorable down to 1st and 2nd order terms.

8. In order to eliminate any frequency shift machinations for non-positive real treatment, further (un-derived) modifications to Beale's transformation procedure are needed. The intent of such modification would be to allow direct implementation of Beale's procedure on an H(s) with terminal phase outside the set (-90,0,+90) allowed by positive real functions.

10 Acknowledgments

The clarity of ref(1) contributed to the ability for its work to be continued. Thanks to all imposed upon for review of various stages of this write up and to the Department of Transportation librarian, Mr. Frank Riley for his help with the literature search, topic and author searches. Thanks to Professor Beale for the proper degree of oversight and trust of my progress and for having initially suggested the topic for research. Thanks to the U.S. Coast Guard for funding a large part of my graduate studies. Lastly, thanks to Dean Rowley, whose flexibility at the start of my GMU program helped make this possible.

11 References

1. G.O. Beale and Gerald Cook, "Frequency Domain Synthesis of Discrete Representations", IEEE Transactions of Industrial Electronics and Control Instrumentation 1976

2. Gene F. Franklin and J. David Powell, Digital Control of Dynamic Systems, Addison-Wesley 1980

3. Richard C Dorf, Modern Control Systems, Addison-Wesley 1986

4. Norman Balabanian and Theodore A. Bickart, Electrical Network Theory, John Wiley and Sons Inc 1969 (sect 7.4 positive real functions)

5. Kendell E. Atkinson, An Introduction to Numerical Analysis, John Wiley and Sons Inc 1978 (pp. 78-87)

6. Frank L. Lewis, Optimal Control Theory, John Wiley and Sons Inc 1986

7. (Richard J. Karwoski), "New math transform bridges the linear-digital gap", Electronic Design, August 1979

8. R.J.Karwoski, "Introduction to the Z Transform and Its Derivation", TRW publication August 1981

note: some of ref(1) content can be viewed online at
http://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/19760012070.pdf

12 Appendix A

transformation example

Ex. $H(s)$ term $s^2 + 2.31s + 2.72$ (from problem 7)
$Wn^2 = 2.72$ yields $Wn = 1.649$
$2 * \eta * Wn = 2.31$ yields $\eta = 0.7003$

for problem 7, largest $Wn^2 = 81$ yields Nyquist rate=9 rad/s rad/s

let $Ws = 40$ rad/s yields $T = 2\pi/Ws = 0.15708$ sec

form $p^2 + (2vr)p + r^2$

$r^2 = tan(Wi * T/2))^2 = (0.13024)^2$ $v = \eta$

$2vr = 2 * (0.7003) * (0.13024) = 0.18241$

yields

$$p^2 + 0.18241p + 0.16963$$

each p goes to $(z - 1)/(z + 1)$ or

$$[(z - 1)^2 + 0.18241(z - 1)(z + 1) + 0.16963(z + 1)^2]/(z + 1)^2$$

z^2: $1 + 0.18241 + 0.016963 = .1994$

z^1: $-2 + 2(0.016963) = -1.9661$

z^0: $1 - 0.18241 + 0.016963 = 0.83455$

yields term $z^2 - 1.9661z/1.1994 + 0.83455/1.1994$

$z^2 - 1.6392z + 0.69581$

this yields:

H(s) term	$Wn * \eta$	$tan(WiT/2)$	z-term coeff
$s^2 + 2.31s + 2.72$	$0.1302, 0.7003$	0.13024	$(1, -1.6392, 0.69581)$
$s + 1.65$	–	0.13032	$(0, 1, -0.76941)$
$s^2 + 7.25s + 81$	$9.0, 0.4028$	0.85408	$(1, -0.22383, 0.4308)$
$s^2 + 8.4s + 36$	$6.0, 0.7000$	0.50953	$(1, -0.75054, 0.27689)$
$s^2 + 5.62s + 3.1$	$s + 5.0$	0.41421	$(0, 1, -0.41422)$
	$s + 0.62$	0.048733	$(0, 1, -0.90706)$
$s + 0.62$	–	0.048733	$(0, 1, -0.90706)$
$s^2 + 11.849s + 72$	$8.485, 0.6982$	0.78647	$(1, -0.28083, 0.19152)$

for (s+1.65), $-(1 - tan(WiT/2)/(1 + tan(WiT/2)) = -0.76941$

13 Appendix B

Octave coding (local or Google extension)
(note: ":" used in lieu of spaces in tf,num,den for clarity)

PR1:
$C = tf(1.0 * [1 : 4 : 3], [1 : 6 : 12 : 8]);$
$Hd = c2d(C, 2 * pi/20,' bi');$
$num = [1 : -1.20984 : 0.402598 : -0.0373884];$
$den = [1 : -1.52859 : 0.778862 : -0.132285];$
$Hd = tf(num, den, 2 * pi/20);$
$k = dcgain(C)/dcgain(Hd);$
$bode(C,' g', Hd * k,' r - -');$ typical, not shown on rest

PR2:
$C = tf(1.0 * [1 : 6 : 8], [1 : 4 : 3]);$
$Hd = c2d(C, 2 * pi/20,' bi');$
$num = [1 : -0.66791 : 0.0806994];$ $den = [1 : -1.05146 : 0.236067];$
$Hd = tf(num, den, 2 * pi/20);$
$k = dcgain(C)/dcgain(Hd);$

PR3:
$C = tf(1.0 * [1 : 1 : 1], [1 : 1 : 4]);$
$Hd = c2d(C, 2 * pi/20,' bi');$
$num = [1 : -1.64755 : 0.73234];$ $den = [1 : -1.41073 : 0.74376];$
$Hd = tf(num, den, 2 * pi/20);$
$k = dcgain(C)/dcgain(Hd);$

PR4:
$C = tf(1.0 * [1 : 2 : 1 : 1], [1 : 1 : 2 : 1]);$
$Hd = c2d(C, 2 * pi/20,' bi');$
$num = [1 : -2.18752 : 1.58575 : -0.377557];$
$den = [1 : -2.5519 : 2.31067 : -0.732382];$
$Hd = tf(num, den, 2 * pi/20);$
$k = dcgain(C)/dcgain(Hd);$

PR5:
$C = tf(1.0 * [1 : 20 : 140 : 400 : 384], [1 : 16 : 86 : 176 : 105]);$
$Hd = c2d(C, 2 * pi/50,' bi');$
$num = [1 : -2.09035 : 1.57323 : -0.503666 : 0.0576742];$
$den = [1 : -2.43077 : 2.14015 : -0.807382 : 0.109908];$
$Hd = tf(num, den, 2 * pi/50);$
$k = dcgain(C)/dcgain(Hd);$

.

PR6:

$C = tf(1.0 * [1 : 11 : 35.25 : 31.5], [1 : 16 : 86 : 176 : 105]);$

$Hd = c2d(C, 2 * pi/100,' bi');$

$num = [1 : -1.39068 : -0.49884 : 1.39642 : -0.495422];$

$den = [1 : -3.12749 : 3.64219 : -1.87184 : 0.35819];$

$Hd = tf(num, den, 2 * pi/100);$

$k = dcgain(C)/dcgain(Hd);$

P7:

$C = tf(1.0 * [1 : 11.21 : 116.242 : 372.601 : 561.589 : 363.528], [1 : 26.489 : 340.47 : 2461.61 : 10433.1 : 23363.9 : 19049 : 4981.82]);$

$Hd = c2d(C, 2 * pi/40,' bi');$

$$Hd : \frac{0.002524z^7 - 0.00226z^6 - 0.003061z^5 + 0.004265z^4 - 0.001063z^3 - 0.001333z^2 + 0.001663z - 0.0006096}{z^7 - 3.545z^6 + 5.43z^5 - 4.798z^4 + 2.702z^3 - 0.9835z^2 + 0.2186z - 0.02319}$$

note ref(1) used an APF form of $z = (z + z1)/(z + p1)$

$where \quad p1 = 1/z1 \quad ie \quad (z + 2.724)/(z + 0.3671)$

for a specific frequency, analog and digital respectively,

$freqresp(C, w) \quad or [fr, w] = freqz(num, den, N);$

14 Appendix C

For clarification, we illustrate the divergence correction

digital root from TF $(s^2+7.25+81)/(s^2+8.4s+36)$

s-roots: $-3.625+/-8.238i, -4.2+/4.2849i$

using
$Ws = 40rad/s$
$T = 2*PI/Ws$
note the bilinear modified z-transform as follows
$z-roots: exp(-aT)[1-0.85*(4*a/Ws)^(3.17)], where \quad 4a/Ws <= 1$

ORIGINAL: $0.063925+/-0.212033i, 0.42632+/-0.23811i$
yielding
$$(z^2-0.12785z+0.049044)/(z^2-0.85264z+0.23845) \qquad (2a)$$

MODIFIED:(Wbar=53 rad/s replacing Ws for root conversion ONLY)
$Wbar = Ws*1.33$ the multiplier can be iterated, typ. default 1.15
$Tbar = 2*PI/Wbar$
$exp(-aTbar)[1-0.85*(4*a/Wbar)^(3.17)], \quad where \quad 4a/Ws <= 1$
$0.27479+/-0.40373i, 0.53488+/-0.25389i$
yielding $(z^2-0.54958z+0.23851)/(z^2-1.06976z+0.35056)$ $\qquad (2b)$

we can then compare analog and (2a,2b) via

$Hd1 = c2d(H, 0.15708, 'bi') =$ baseline via Octave bilinear

$1.1z^2-0.5318z+0.4943$
$Hd1: -------------$
$z^2-0.8268z+0.2988$
dcgain(Hd1); 2.25

normalizing,
$z^2-0.48345z+0.44936$
$Hd1: -------------$
$z^2-0.8268z+0.2988$

$k1 = 1.1$
$num2 = [1:-0.12785:0.049044]$ $\quad den2 = [1-0.852640.23845]$
$Hd2 = tf(num2, den2, T)$
dcgain(Hd2); 2.3877 so require k2=2.25/2.3877=0.94233
$Wbar = 1.33*Ws$
$Tbar = 2*PI/Wbar$
$num3 = [1:-0.54958:0.23851]$ $\quad den3 = [1-1.069760.35056]$

$Hd3 = tf(num3, den3, T)$
dcgain(Hd3); 2.4535 so require $k3 = 2.25/2.4535 = 0.91706$

$bode(H,'r', Hd2*k2,'g--', Hd3*k3,'b-.');$

The resulting plots (ignoring beyond Ws/2=20 rad/s):

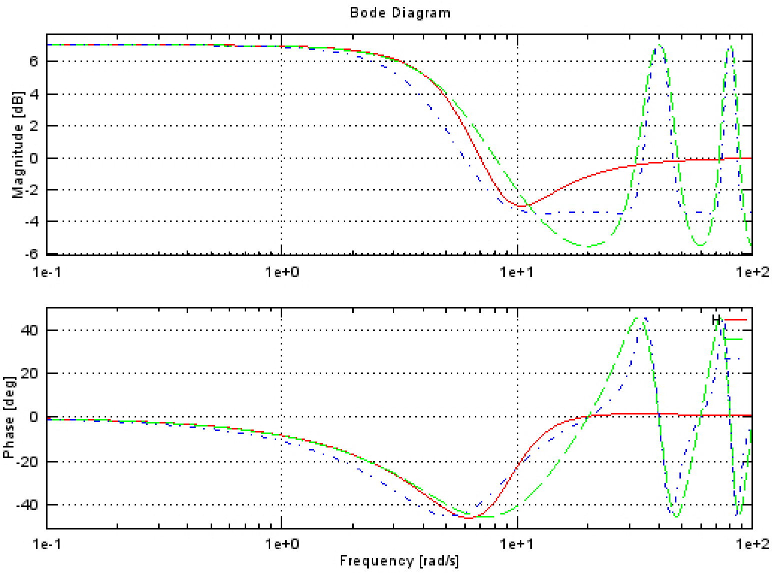

Figure 8: Analog(red) vs Original (blue) vs Modified(green)

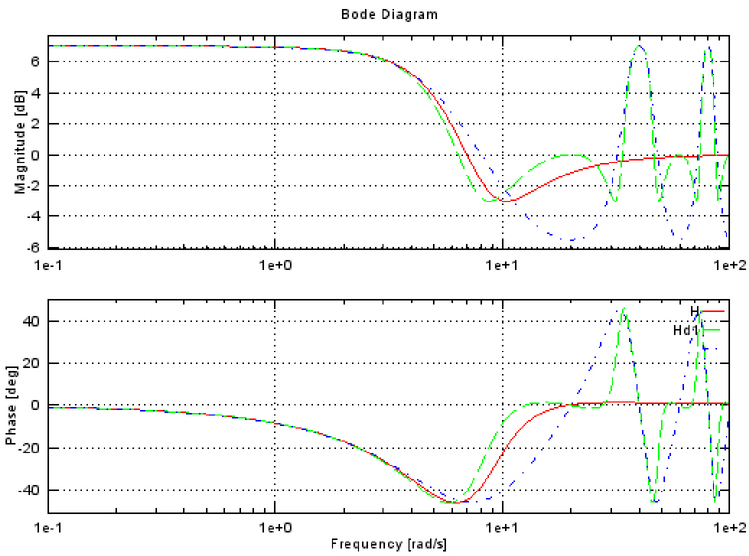

Figure 9: Analog(red) vs Bilinear(blue) vs modified z-transform(green)

15 About the Author

Mr. McEachen did his undergraduate work at Webb Institute, Glen Cove, NY graduating in 1984. He has been working for the U.S. Coast Guard since that time. He is a member of IEEE, ASNE, and SNAME. Comments or questions concerning this work can be addressed to bill DOT mceachen AT Gmail DOT com.