Shunji Manabe and Young-Chol Kim

Coefficient Diagram Method for Control System Design

September 28, 2020

Springer Nature

Appendix A
CDM Toolbox User's Guide for Use with Matlab

A.1 Introduction

The coefficient diagram method (CDM) is a new control design technique. In this approach, we need to calculate algebraic equations associated with Diophantine equation and draw polynomial coefficient curves on the coefficient diagram. As a computer aided design tool, the CDM toolbox (CDMTOOL) presented here enclose a set of Matlab® routines. The Matlab toolbox is in the public domain and can be downloaded from the following URL.

https://github.com/ycholkim1366/CDM-Tool
http://www.cityfujisawa.ne.jp/ manabes/CDMCAD.htm

Unzipping the downloaded file, you will have two directories, named by Work and Problem. The CDM M-files are in the "work". You may change the folder name, for example, "CDMTOOL". Hereafter, the name of this directory will be considered as "CDMTOOL". User should copy this folder to
C:\ matlab\ CDMTOOL
and then the path must be added in the Matlab environment. Otherwise, this CDM-TOOL folder may be directly copied in the Matlab toolbox directory
C:\ matlab\ toolbox

Note: It is better for you to open your working directory to create your files.

A.2 The CDM Toolbox

A number of Matlab M-files are available under the directory CDMTOOL. A summary of Matlab routines and their functions are shown below. Symbols and parameter conventions used in the CDM M-files are listed in Table A.1 and the CDM Matlab function files are listed in Table A.2 ~ Table A.4.

Table A.1 CDMTOOL parameter conventions

Parameters	Description
a	Polynomial
aa	Characteristic polynomial (CP) $P(s)$
ac, bc	Denominator/numeraor polynomials of the controller $G_c(s) = B_c(s)/A_c(s)$
ap, bp	Denominator/numeraor polynomials of the plant $G_p(s) = B_p(s)/A_p(s)$
acp, bcp	Denominator/numeraor polynomials of the open-loop function $G(s) = G_c(s)G_p(s)$, where $A_{cp}(s) = A_c(s)A_p(s)$ and $B_{cp}(s) = B_c(s)B_p(s)$
al	Ratio parameter vector in Routh Table. al$(i) := R(i)/R(i+1)$, where $R(i)$ is the element of the i-th row of the 1st column of the Routh Table.
aq	Squared polynomial; $A_q(-s^2) = A(-s)A(s)$
ba	Prefilter polynomial $B_a(s)$ for command input
c	Controller
g	Stability index $\gamma = [\gamma_{n-1} \cdots \gamma_2\, \gamma_1]$
gr	Desired stability index
gs	Stability limit $\gamma^* = [\gamma^*_{n-1} \cdots \gamma^*_2\, \gamma^*_1]$
k	Gain
lp	Loop transfer function
mc, nc	Order of numerator/denominator polynomials of controller
mp, np	Order of numerator/denominator polynomials of plant
rr	Roots of a polynomial
t	Equivalent time constant (ETC) τ
tm	Time scale for time response
unc	indicates whether the coefficient of the controller denominator polynomial to be normalized to 1 is the highest or lowest term
w	Diagonal entry of the weight matrix Q for LQR design

A.3 How to use CDM toolbox?

After the installation as illustrated in A.1, one can use the CDM toolbox function files. In this section, we demonstrate the proper syntax for entering the information to Matlab for CDM design.

Table A.2 CDM Matlab function files: Conversion and Computational Utility

Function	Description
`a2al`	Extract Routh Table and ratio parameters vector from the characteristic polynomial(CP)
`a2aq`	Convert a polynomial to the squared polynomial
`a2g`	Compute γ, τ, γ^*, and roots of a polynomial
`a2w`	Compute the weight matrix for LQR design from `aa` and `ap`
`al2a`	Convert the Routh ratio parameters to the corresponding polynomial
`aq2a`	Convert the squared polynomial to the original polynomial
`bpt`	Compute the break points from the CP
`c2g`	Compute the CP, γ, τ, γ^*, poles, gain/ phase margins of the closed-loop system from the plant (`ap, bp`) with a controller (`ac, bc, ba`)
`c2lp`	Convert the plant and controller polynomials to the numerator/denominator polynomials of the loop transfer function
`convvm`	Convolution of a vector and a matrix
`g2a`	Compute the CP from γ and the highest/ lowest coefficients set
`g2t`	Compute possible ETC τ under the given plant model, desired stability index, and the order of controller
`gk2t`	Compute possible ETC τ under the given plant model, desired stability index, the order of controller, and a fixed steady state gain of controller numerator polynomial
`gt2a`	Compute the CP and squared polynomial from the given γ and τ

Table A.3 CDM Matlab function files: Analysis and Graphics Utility

Function	Description
`bode2`	Bode plot of two transfer functions: $G_1 = N_1/D_1$ and $G_2 = N_2/D_2$
`cdia`	Coefficient diagram of a CP and its γ, τ, γ^*
`fresp`	Frequency responses of loop transfer function $G(s)$, sensitivity function $S(s)$, and complementary sensitivity function $T(s)$ for a plant $G_p(s)$ and a controller $G_c(s)$
`rresp2/` `rresp3`	Step responses of two/ three transfer functions: $G_1 = N_1/D_1$, $G_2 = N_2/D_2$, and $G_3 = N_3/D_3$
`tresp`	Step responses of two transfer functions specified by a common denominator (D) and two numerators (N_1, N_2)
`tresp4`	Step responses of four transfer functions specified by a common denominator (D) and four numerators ($N_1, \cdots, N_4$)

a2al

Purpose:

Extract Routh Table and ratio parameters vector from the characteristic polynomial(CP).

Table A.4 CDM Matlab function files: Design Utility

Function	Description
`a2c`	Find the CDM controller (`ac, bc`) from the plant (`ap, bp`) and a desired CP
`aq2c`	Find the CDM controller (`ac, bc`) from the plant (`ap, bp`) and a squared polynomial
`aqwc`	Provides the weight Q for LQR and a CDM controller from a squared polynomial and plant (`ap, bp`)
`g2c`	Find the CDM controller (`ac, bc`) from the plant (`ap, bp`), desired γ and τ, and the order of controller
`gk2c`	Provides the CDM controller obtained by adding a constraint on the steady state gain of the controller to `g2c`
`cc`	A simplified form of `c2g`
`gc`	A batch file combining `g2c` and `c2g`
`gkc`	A batch file combining `gk2c` and `c2g`

Synopsis and Description:

```
[al, RT]=a2al(aa)
```

Input:

`aa`=input polynomial in descending order.

Outputs:

`al`= Ratio parameters generated in Routh table, defined by `al(i)` $= R(i+1)/R(i)$, where $R(i)$ is the element of the i-th row of the first column of the Routh table.

`RT`= Routh table

Example:

Let us consider the following polynomial,

$$P(s) = 0.25s^5 + s^4 + 2s^3 + 2s^2 + s + 0.2$$

which can be implemented as:

```
>>aa=[0.25  1  2  2  1  0.2];
>>[al,RT]=a2al(aa)
al =
     2.5000e-01    6.6667e-01    1.0976e+00    1.8709e+00    3.6524e+00
```

```
RT =
   2.50e-01          0    2.00e+00          0    1.00e+00          0
          0   1.00e+00           0   2.00e+00          0   2.00e-01
          0          0    1.50e+00          0    9.50e-01          0
          0          0           0 1.3667e+00          0   2.00e-01
          0          0 -2.2204e-16          0  7.3049e-01          0
          0          0           0          0          0   2.00e-01
```

a2aq

Purpose:

Convert a polynomial to the squared polynomial. For a given polynomial $P(s)$, the output is $A_q(-s^2) = P(-s)P(s)$.

Synopsis and Description:

```
aq=a2aq(aa)
```

Input:

aa=input polynomial in descending order.

Output:

aq= squared polynomial.

Example:

Let us consider the following polynomial,

$$P(s) = s^4 + 2s^3 + 2s^2 + s + 0.2$$

which can be implemented as:

```
>> aa=[1 2 2 1 0.2];
>> aq=a2aq(aa)
aq =
    1.0000e+00    0    4.0000e-01    2.0000e-01    4.0000e-02
```

The above vector aq can be represented in the form, with $x = -s^2$,

$$A_q(x) = x^4 + 0.4x^2 + 0.2x + 0.04 \quad \Rightarrow \quad A_q(-s^2) = s^8 + 0.4s^4 - 0.2s^2 + 0.04.$$

a2c

Purpose:

Find the CDM controller (`ac,bc`) so that a desired CP is assigned for the given plant (`ap,bp`).

Synopsis and Description:

```
[bc,ac,aa,g,tau,gs,rr]=a2c(ap,bp,ar,unc)
```

Inputs:

`bp,ap`= numerator and denominator polynomials of the plant.
`ar`= a reference polynomial for the desired closed-loop system.
`unc`= indicates whether the coefficient of the controller denominator polynomial to be normalized to 1 is the highest or lowest term, which is normally 0 or `nc`.

Outputs:

`bc,ac`= numerator and denominator polynomials of the controller.
`aa`= characteristic polynomial of the closed loop system with a designed controller.
`g`= stability index of `aa`.
`tau`= equivalent time constant of `aa`.
`gs`= stability limit of `aa`.
`rr`= roots of characteristic polynomial `aa`.

Example:

Consider the following plant and the reference characteristic polynomial,

$$G_p(s) = \frac{1}{s^4+s^2}$$

$$P_r(s) = 2^{-9}s^7 + 2^{-5}s^6 + 2^{-2}s^5 + s^4 + 2s^3 + 2s^2 + s + 0.2$$

that will be formed as,

>> `ap=[1 0 1 0 0]; bp=[1]; ar=[`2^{-9} 2^{-5} 2^{-2} `1 2 2 1 0.2];`
>> `[bc,ac,aa,g,tau,gs,rr]=a2c(ap,bp,ar,0)`

```
bc =
      1.8085e+00     1.0645e+00     1.0323e+00     2.0645e-01
ac =
      2.0161e-03     3.2258e-02     2.5605e-01     1.0000e+00
aa =
```

```
   2.016e-3   3.226e-2   2.581e-1   1.032   2.065   2.065   1.032   2.065e-1
g =
     2.00e+00   2.00e+00   2.00e+00   2.00e+00   2.00e+00   2.50e+00
tau = 5
gs =
     5.00e-01   1.00e+00   1.00e+00   1.00e+00   9.00e-01   5.00e-01
rr =
     -4.4502e+00 ± 5.2133e+00i
     -2.9270e+00 + 0.0000e+00i
     -2.2981e+00 + 0.0000e+00i
      -6.4216e-01 ± 3.6956e-01i
      -5.9026e-01 + 0.0000e+00i
```

a2g

Purpose:

Compute γ, τ, γ^*, and roots of the characteristic polynomial $P(s)$.

Synopsis and Description:

```
[g,tau,gs,rr]=a2g(aa)
```

Input:

aa= coefficients of the characteristic polynomial $P(s)$.

Outputs:

g= stability index of aa.
tau= equivalent time constant of aa.
gs= stability limit of aa.
rr= roots of characteristic polynomial aa.

Example:

Consider the following characteristic polynomial,

$$P(s) = s^4 + 2s^3 + 2s^2 + s + 0.2$$

will be formed as,

```
>> aa=[1 2 2 1 0.2];
>> [g,tau,gs,rr]=a2g(aa)
g =
    2.0000   2.0000   2.5000
tau = 5
gs =
    0.5000   0.9000   0.5000
rr =
    -0.5000 ±0.6882i
    -0.5000 ±0.1625i
```

a2w

Purpose:

Compute the weight matrix for LQR design from the characteristic polynomial and plant denominator polynomial.

Synopsis and Description:

```
[aq,apq,qq]=a2w(aa,ap)
```

Inputs:

`ap`= denominator polynomial $A_p(s)$ of the plant .
`aa`= characteristic polynomial $P(s)$.

Outputs:

`aq`= squared polynomial of `aa`, defined as $A_q(-s^2) = P(-s)P(s)$.
`apq`= squared polynomial of `ap`, defined as $A_{pq}(-s^2) = A_p(-s)A_p(s)$.
`qq`= `[q11 q22 ··· qnn]`, where q_{ii} is the diagonal element of the weight matrix of Q.

Example:

Consider the following plant denominator and characteristic polynomials:

$$A_p(s) = s^4 + s^2$$
$$P(s) = s^4 + 2s^3 + 2s^2 + s + 0.2$$

They will be formed as,

```
>> ap=[1 0 1 0 0];   aa=[1 2 2 1 0.2];
>> [aq,apq,qq]=a2w(aa,ap)
aq =
    1.0000    0    0.4000    0.2000    0.0400
apq =
    1    -2    1    0    0
qq =
    2.0000    -0.6000    0.2000    0.0400
```

The above vector `aq` can be represented in the form, with $x = -s^2$,

$$A_q(x) = x^4 + 0.4x^2 + 0.2x + 0.04 \quad \Rightarrow \quad A_q(-s^2) = s^8 + 0.4s^4 - 0.2s^2 + 0.04.$$

al2a

Purpose:
Convert the Routh ratio parameters to the corresponding characteristic polynomial.

Synopsis and Description:

```
[aa,RT]=al2a(al)
```

Input:

`al=` ratio parameter vector in Routh Table. `al`$(i) = R(i)/R(i+1)$, where $R(i)$ is the i-th element of the first column of the Routh Table.

Outputs:

`aa=` characteristic polynomial corresponding to the given `al`.
`RT=` Routh table of `aa`.

Example:
Consider the following characteristic polynomial,

$$P(s) = 0.25s^5 + s^4 + 2s^3 + 2s^2 + s + 0.2$$

It is easy to know by using M-function `a2al` that Routh ratio parameter vector of $P(s)$ above is given as
`al`=[0.25 0.66667 1.0976 1.8709 3.6524]

Then

```
>>[aa,RT]=al2a(al)
aa =
    1    4    8    8    4    0.8

RT =
    1.0000         0    8.0000         0    4.0000         0
         0    4.0000         0    8.0000         0    0.8000
         0         0    6.0000         0    3.8000         0
         0         0         0    5.4667         0    0.8000
         0         0         0         0    2.9220         0
         0         0         0         0         0    0.8000
```

aq2a

Purpose:

Convert the squared polynomial $A_q(-s^2)$ to the original polynomial $P(s)$ and its γ, τ, γ^*.

Synopsis and Description:

```
[aa,g,tau,gs,rr]=aq2a(aq)
```

Input:

aq= squared polynomial $A_q(-s^2)$.

Outputs:

aa= characteristic polynomial $P(s)$.
g= stability index of aa.
tau= equivalent time constant of aa.
gs= stability limit of aa.
rr= roots of characteristic polynomial $P(s)$.

Example:

Consider the following squares polynomial,

$$A_q(-s^2) = s^8+0.4s^4-0.2s^2+0.04, \quad \text{or } A_q(x) = x^4+0.4x^2+0.2x+0.04, \ (x=-s^2)$$

that will be formed as,

```
>> aq=[1 0 0.4 0.2 0.04];
>> [aa,g,tau,gs,rr]=aq2a(aq)
aa =
    1.0   2.0   2.0   1.0   0.2
g =
    2.00   2.00   2.5
tau = 5.0
gs =
    0.50   0.90   0.50
rr =
    -0.50 ± 0.6882i
    -0.50 ± 0.1625
```

aq2c

Purpose:
Find the CDM controller (ac,bc) from the given plant (ap,bp) and a squared polynomial (aq).

Synopsis and Description:

```
[bc,ac,aa,g,tau,gs,rr]=aq2c(ap,bp,aq,unc)
```

Inputs:

bp,ap= numerator and denominator polynomials of the plant.
aq= a squared polynomial of the characteristic polynomial.
unc= indicates whether the coefficient of the controller denominator polynomial to be normalized to 1 is the highest or lowest term, which is normally 0 or nc.

Outputs:

bc,ac= numerator and denominator polynomials of the controller.
aa= characteristic polynomial of the closed loop system with a designed controller.
g= stability index of aa.
tau= equivalent time constant of aa.
gs= stability limit of aa.
rr= roots of characteristic polynomial aa.

Example:
Consider the following plant and a squared polynomial of the characteristic polynomial,

$$G_p(s) = \frac{1}{s^4+s^2}$$

$$A_q(-s^2) = -s^{14} + 1, \quad \text{or } A_q(x) = x^7 + 1, \text{(where } x = -s^2\text{)}$$

that will be formed as,

```
>> ap=[1 0 1 0 0];   bp=[1];   aq=[1 0 0 0 0 0 0 1];
>> [bc,ac,aa,g,tau,gs,rr]=aq2c(ap,bp,aq,0)
bc =
     5.4407e-01   -8.7957e-15   4.4504e-01   9.9031e-02
ac =
     9.9031e-02   4.4504e-01   9.0097e-01   1.0000e+00
```

```
aa =
      9.903e-02   4.450e-01   1.000   1.445   1.445   1.000   4.450e-10   9.903e-02
g =
      2.000     1.555     1.445     1.445     1.555     2.000
tau = 4.494
gs =
      0.6431     1.1920     1.3351     1.3351     1.1920     0.6431
rr =
      -2.2252e-01 ± 9.7493e-01i
      -6.2349e-01 ± 7.8183e-01i
     -1.0000e+00 + 0.0000e+00i
      -9.0097e-01 ± 4.3388e-01i
```

aqwc

Purpose:

From the given plant (ap,bp) and a squared polynomial (aq), this M-file provides the weights for LQR and a CDM controller (ac,bc,ba) by using a2c and aq2a. Also, it shows the frequency responses as well as the step responses of the overall system with the resulting controller by using c2g.

Synopsis and Description:

```
aqwc
```

Initial information:

bp,ap= numerator and denominator polynomials of the plant.
aq= a squared polynomial of the characteristic polynomial.

Outputs:

apq= squared polynomial of (ap), $A_{pq}(-s^2) = A_p(-s)A_p(s)$.
bpq= squared polynomial of (bp), $B_{pq}(-s^2) = B_p(-s)B_p(s)$.
qu= weight polynomial for the control input u in LQR design.
qy= weight polynomial for the system output y in LQR design.
ba= feedforward polynomial of the CDM controller.
bc, ac= numerator and denominator polynomials of the CDM controller.
aa= characteristic polynomial of the closed loop system
g= stability index of aa.
tau= equivalent time constant of aa.
gs= stability limit of aa.
rr= roots of characteristic polynomial aa.
pmgm= phase and gain margins of the loop transfer function
wpmgm= phase and gain crossover frequencies of the loop transfer function

In LQR design, we may use $R = \mathtt{qu}(1)$ and $Q = diag([\mathtt{qu}(2) \;\cdots\; \mathtt{qu}(nc+1)\; \mathtt{qy}(1) \;\cdots\; \mathtt{qy}(np)])$.

Example:

Consider the following plant and a squared polynomial of the characteristic polynomial in Matlab scripts.

```
>> ap=[0.25   1.25   1   0];   bp=[0.1   1];
>> aq=[0.13598   13.772   35.757   221.25   470.38   400];
>> aqwc

apq =
```

```
     6.2500e-02    1.0625e+00    1.0000e+00    0
bpq =
     1.0000e-02    1.0000e+00
qu =
     2.1757e+00    1.8337e+02    -3.1087e+03
qy =
     3.3052e+03    3.5751e+03    4.0000e+02
ba =2.0000e+01

bc =
     2.6491e+01    4.5499e+01    2.0000e+01
ac =
     1.4750e+00    1.4750e+01    9.9722e-01
aa =
     0.3688   5.5313   22.811   47.037   48.496   20.00
g =
     3.6372    2.00    2.00    2.50
tau = 2.4248
pmgm =
     45.761    1.0179e+04
wpmgm =
     1.7715e+00    2.7051e+02
```

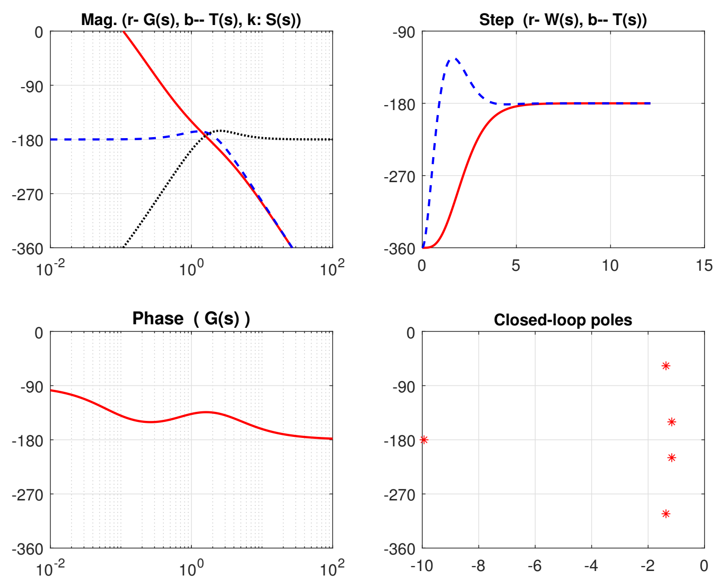

Fig. A.1 Poles, frequency and step responses of the resulting systems obtained by `aqwc`.

bode2

Purpose:

This M-file provides a Bode plot for two transfer functions specified by $G_1(s) = N_1(s)/D_1(s)$ and $G_2(s) = N_2(s)/D_2(s)$.

Synopsis and Description:

```
bode2(num1,den1,num2,den2)
```

Inputs:

`num1, den1`= numerator and denominator polynomials of $G_1(s)$.
`num2, den2`= numerator and denominator polynomials of $G_2(s)$.

Example:

Consider the following two transfer functions in Matlab scripts.

```
>> num1=[1.5  1];  den1=[2.5  14.125  18.375  7.75  1];
>> num2=[1];  den2=[1.6667  8.6528  6.25  1];
>> bode2(num1,den1,num2,den2)
```

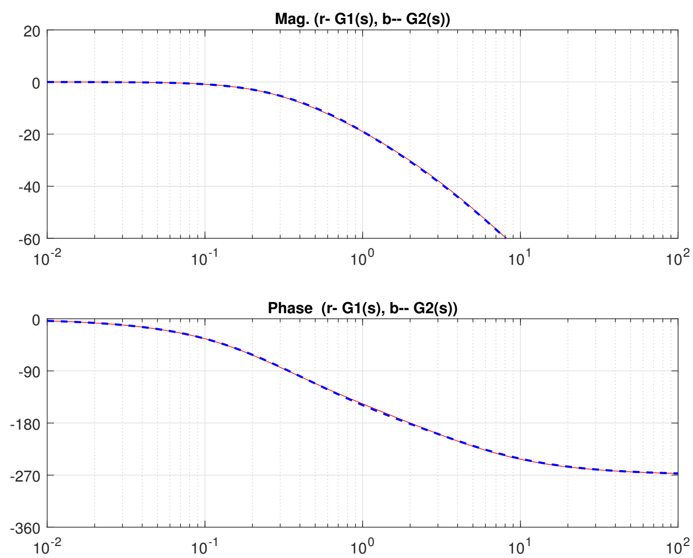

Fig. A.2 Bode plot of two transfer functions $G_1(j\omega)$ and $G_2(j\omega)$.

bpt

Purpose:

Determine the pseudo-break points of a real polynomial.

Synopsis and Description:

```
[bpp]=bpt(aa)
```

Input:

`aa`= a real polynomial, $P(s) = a_n s^n + a_{n-1} s^{n-1} + \cdots + a_0$.

Output:

`bpp`= a vector of pseudo-break points of $P(s)$, defined as $\mathtt{bpp}(i) = a_{n-i}/a_{n+1-i}$.

Example:

Consider the following polynomial,

$$P(s) = s^4 + 2s^3 + 2s^2 + s + 0.2$$

that will be formed as,

```
>> aa=[1 2 2 1 0.2];
>> [bpp]=bpt(aa)

bpp =
    2.0    1.0    0.5    0.2
```

c2g

Purpose:

Compute the CP, γ, τ, γ^*, poles, gain/ phase margins of the closed-loop system from the plant (ap, bp) with a designed controller (ac, bc, ba).

Synopsis and Description:

```
[aa,g,tau,gs,rr,pmgm,wpmgm]=c2g(ap,bp,ac,bc,ba,tm)
```

Inputs:

bp,ap= numerator and denominator polynomials of the plant.
bc,ac= numerator and denominator polynomials of the controller.
ba= numerator polynomial for the command input.
tm= the time scale of the time response figure, which is normally set to 1.0 or 0.5.

Outputs:

aa= characteristic polynomial of the closed loop system with a controller.
g= stability index of aa.
tau= equivalent time constant of aa.
gs= stability limit of aa.
rr= roots of characteristic polynomial aa.
pmgm= phase and gain margins of the loop transfer function
wpmgm= phase and gain crossover frequencies of the loop transfer function

Example:

Consider the following plant and the controller in two-parameter configuration,

$$G_p(s) = \frac{B_p(s)}{A_p(s)} = \frac{1}{s^3+s}$$

$$A_c(s) = 0.071794s^2 + 0.3923s + 1, \quad B_c(s) = 1.0718s^2 + 0.27322,$$
$$B_a(s) = 0.27322,$$

that will be formed as,

```
>> ap=[1 0 1 0];   bp=[1];
>> ac=[0.071794 0.3923 1];   bc=[1.0718 0 0.27322];
>> ba=[0.27322];   tm=1.0
>> [aa,g,tau,gs,rr,pmgm,wpmgm]=c2g(ap,bp,ac,bc,ba,tm)
aa =
```

```
        7.1794e-02   0.3923   1.0718   1.4641   1.0000   2.7322e-01
g =
        2.000    2.000    2.000    2.4999
tau = 3.6601
gs =
        0.500    0.9999    9.0002e-01    0.500
rr =
        -1.5182e+00 ± 1.7480e+00i
         -8.2521e-01 ± 4.8197e-01i
         -7.7733e-01 + 0.0000e+00i
pmgm =
        5.3156e+01    4.8205e+00
wpmgm =
        1.5714e+00    3.7321e+00
```

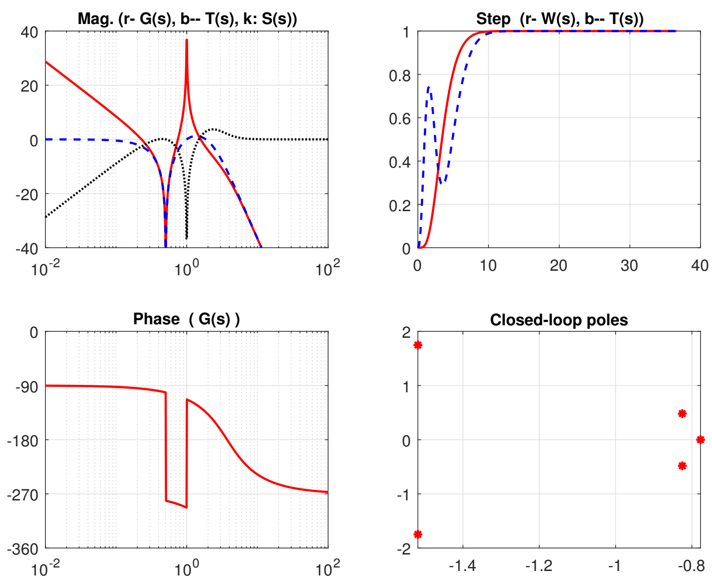

Fig. A.3 Poles, frequency and step responses of the resulting systems provided by `c2g`.

c2lp

Purpose:

Convert the plant $G_p(s) = B_p(s)/A_p(s)$ and controller $G_c(s) = B_c(s)/A_c(s)$ to the numerator and denominator polynomials of the loop transfer function $G(s) = G_c(s)G_p(S)$.

Synopsis and Description:

```
[bcp,acp,aa]=c2lp(ap,bp,ac,bc)
```

Inputs:

`bp,ap`= numerator and denominator polynomials of the plant.
`bc,ac`= numerator and denominator polynomials of the controller.

Outputs:

`bcp,acp`=numerator and denominator polynomials of the loop transfer function, which are $B_{cp}(s) = B_p(s)B_c(s)$ and $A_{cp}(s) = A_p(s)A_c(s)$.
`aa`= characteristic polynomial of the closed loop system.

Example:

Consider the following single input-two output plant and its controller,

$$A_p(s) = s^4 - 3s^2 + 2, \quad B_p(s) = \begin{bmatrix} 2s^3 - 2s \\ s^3 - 2s \end{bmatrix}$$
$$A_c(s) = s^2 + 0.15s + 0.009, \quad B_c(s) = \begin{bmatrix} 8.5s^2 + 12s \\ -12s^2 - 12s \end{bmatrix},$$

that will be formed as,

```
>> ap=[1 0 -3 0 2];   bp=[2 0 -2 0; 1 0 -2 0];
>> ac=[1 0.15 0.009 ];   bc=[8.5 12 0; -12 -12 0];
>>[bcp,acp,aa]=c2lp(ap,bp,ac,bc)
bcp =
     5  12  7  0  0  0
acp =
     1  0.15  -2.991  -0.45  1.973  0.3  0.018
aa =
     1  5.15  9.009  6.55  1.973  0.3  0.018
```

See also: `lp`

cc

Purpose:

This is a simplified form of `c2g`. From the given plant model (`ap,bp`) and a controller (`ac,bc`), it provides the results given by `c2g` and a command prefilter polynomial `ba` for the overall system to be Type 1.

Synopsis and Description:

```
cc
```

Initial information:

`bp, ap`= numerator and denominator polynomials of the plant.
`bc, ac`= numerator and denominator polynomials of the controller.

Outputs:

`ba`= feedforward polynomial of the CDM controller.
`aa`= characteristic polynomial of the closed loop system
`g`= stability index of `aa`.
`tau`= equivalent time constant of `aa`.
`gs`= stability limit of `aa`.
`rr`= roots of characteristic polynomial `aa`.
`pmgm`= phase and gain margins of the loop transfer function
`wpmgm`= phase and gain crossover frequencies of the loop transfer function

Example:

Consider a single input double output plant and a controller in Matlab scripts as follows:

```
>> ap=[1 0 2 0]; bp=[0 0 1; 1 0 1];
>> ac=[0.1 1]; bc=[10.5 7.7; 0.0 4.8];
>> cc
ba = 12.5
aa =
     0.1  1  5  12.5  12.5
g =
     2.0  2.0  2.5
tau = 1.0
gs =
     0.5  0.9  0.5
```

```
rr =
      -2.5 ± 3.441i
     -2.5± 0.8123i
pmgm =
      37.083     0.12696
wpmgm =
      4.746     1.8258
```

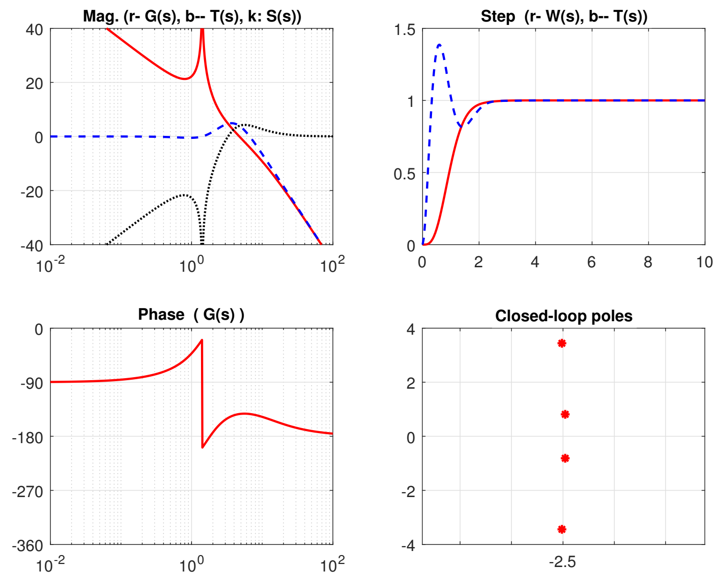

Fig. A.4 Poles, frequency and step responses of the resulting systems provided by `cc`.

cdia

Purpose:

Draw the coefficient diagram for a given CP, $P(s)$ and provides its γ, τ, and γ^*.

Synopsis and Description:

```
[g,tau,gs]=cdia(aa)
```

Input:

`aa`= characteristic polynomial $P(s)$.

Outputs:

`g`= stability index of `aa`.
`tau`= equivalent time constant of `aa`.
`gs`= stability limit of `aa`.

Example:

Consider the following characteristic polynomial,

$$P(s) = s^4 + 2s^3 + 2s^2 + s + 0.2,$$

that will be formed as,

```
>> aa=[1    2    2    1    0.2];
>> [g,tau,gs]=cdia(aa)
g =
     2.0     2.0     2.5
tau =      5
gs =
     0.50     0.90     0.50
```

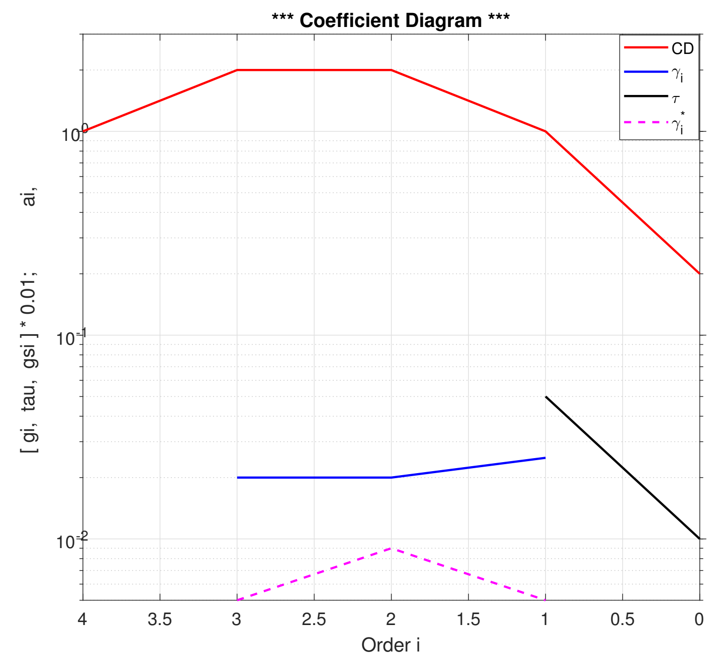

Fig. A.5 Coefficient diagram of the characteristic polynomial in the Example.

convvm

Purpose:

Obtain the convolution of a row vector $V \in \mathcal{R}^{n_v}$ and a matrix $M \in \mathcal{R}^{m_m \mathrm{x} n_m}$, which results in $V_M \in \mathcal{R}^{m_m \mathrm{x}(\mathrm{n_v+n_m}-1)}$.

Synopsis and Description:

```
[vm]=convvm(vv,mm)
```

Inputs:

vv= a row vector representing a polynomial $V(s)$.
mm= a matrix representing a polynomial matrix
$M(s) = [M_{11}(s)\ M_{21}(s)\ \cdots M_{m1}(s)]^T$.

Output:

vm= a matrix representing a polynomial matrix
$V_M(s) = [V(s) * M_{11}(s)\ V(s) * M_{21}(s)\ \cdots V(s) * M_{m1}(s)]^T$.

Example:

Consider the following two polynomials,

$$V(s) = s + 1, \qquad M(s) = \begin{bmatrix} 3s^2 + 2s + 1 \\ s^2 + s \end{bmatrix}.$$

Let us find the convolution of $V(s)$ and $M(s)$.

```
>> vv=[1 1]; mm=[3 2 1; 1 1 0];
>> [vm]=convvm(vv,mm)
vm =
     3   5   3   1
     1   2   1   0
```

The vector vm represents

$$V_M(s) = V(s) * M(s) = \begin{bmatrix} 3s^3 + 5s^2 + 3s + 1 \\ s^3 + 2s^2 + s \end{bmatrix}.$$

fresp

Purpose:
Obtain frequency responses of loop transfer function $G(s)$, sensitivity function $S(s)$, and complementary sensitivity function $T(s)$ for a plant $G_p(s)$ and a controller $G_c(s)$.

Synopsis and Description:

```
fresp
```

Initial information:

`bp,ap`= numerator and denominator polynomials of the plant $G_p(s)$.
`bc,ac`= numerator and denominator polynomials of the controller $G_c(s)$.
`w`= a vector of frequencies in rad/s at which G, S, and T are evaluated.

Outputs:

a plot representing frequency responses of $G(j\omega)$, $S(j\omega)$, and $T(j\omega)$.
`G`= magnitude values of $G(j\omega)$ at frequencies w.
`Gphase`= phase values of $S(j\omega)$ at frequencies w.
`T`= magnitude values of $T(j\omega)$ at frequencies w.
`S`= magnitude values of $S(j\omega)$ at frequencies w.

Example:
Consider a single input double output plant and a 2x1 controller in Matlab scripts as follows:

```
>> ap=[1   0   2   0];   bp=[0   0   1; 1   0   1];
>> ac=[0.1   1];   bc=[10.5   7.7; 0.0   4.8];
>> w= [0.5   1   2];
>> fresp
w =
    0.5   1   2
G =
    1.4222e+01    1.2956e+01    5.4037e+0
Gphase =
    -4.2794e+02    -4.0196e+02    -1.7361e+02
T =
    9.7232e-01    9.4460e-01    1.2250e+00
S =
```

6.8366e-02 7.2908e-02 2.2669e-01

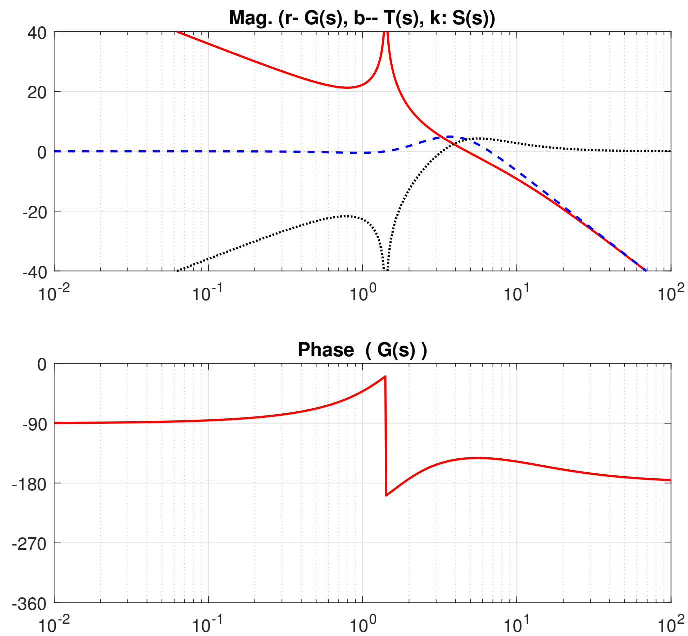

Fig. A.6 Frequency responses of $G(j\omega)$, $S(j\omega)$, and $T(j\omega)$ using `fresp`.

g2a

Purpose:
Obtain the polynomial and the squared polynomial corresponding to the specified stability index, in which the coefficients of the highest and lowest orders are set to the given values.

Synopsis and Description:

```
[aa,aq]=g2a(g,an,a0)
```

Inputs:

`g`= stability index
`an,a0`= coefficients of the highest/ lowest terms of polynomial.

Outputs:

`aa`= polynomial $P(s)$ corresponding to the specified (`g, an, a0`).
`aq`= squared polynomial of `aa`, which is $A_q = P(-s)P(s)$.

Example:
When the stability index $\gamma = [2\ \ 2\ \ 2\ \ 2.5]$ is given, find the polynomial normalized at the both highest and lowest orders.

```
>> g=[2 2 2 2.5]; an=1; a0=1;
>> [aa,aq]=g2a(g,an,a0)

aa =
    1.0000    4.1826    8.7469    9.1461    4.7818    1.0000
aq =
    1.0000    1.0658e-14    9.5635    8.3651    4.5731    1.0000
```

The above coefficient vectors `aa` and `aq` represent

$$P(s) = s^5 + 4.1826s^4 + 8.7469s^3 + 9.1461s^2 + 4.7818s + 1,$$
$$A_q(x) = x^5 + 1.0658\text{x}10^{-14}x^4 + 9.5635x^3 + 8.3651x^2 + 4.5731x + 1, \quad (x = -s^2).$$

g2c

Purpose:

Find a CDM controller (ac,bc) for a given plant (ap,bp) under the design parameters: desired stability index γ, equivalent time constant τ, and the order of controller.

Synopsis and Description:

```
[bc,ac,aa,g,tau,gs,rr]=g2c(ap,bp,nc,mc,gr,t,unc)
```

Inputs:

bp,ap= numerator and denominator polynomials of the plant.
mc,nc= numerator and denominator orders of the controller.
gr= a vector of the desired stability index.
t= equivalent time constant τ. If t is a vector, the left end value specifies a higher-order equivalent time constant.
unc= indicates whether the coefficient of the controller denominator polynomial to be normalized to 1 is the highest or lowest term, which is normally 0 or nc.

Outputs:

bc,ac= numerator and denominator polynomials of the CDM controller.
aa= characteristic polynomial of the closed loop system
g= stability index of the resulting aa.
tau= equivalent time constant of aa.
gs= stability limit of aa.
rr= roots of characteristic polynomial aa.

Example:

Consider the following third order plant.

$$G_p(s) = \frac{s + 0.1}{s^3}.$$

Suppose that we want to find a first-order controller $G_c(s) = (k_1(s) + k_0)/(l_1 s + 1)$ with design parameters $\gamma = [2\ \ 2\ \ 2.5]$ and $\tau = 2.0$.

```
>> ap=[1 0 0 0];   bp=[1 0.1];
>> gr=[2 2 2.5];   nc=1;   mc=1;   unc=0;
>> [bc,ac,aa,g,tau,gs,rr]=g2c(ap,bp,nc,mc,gr,t,unc)
```

```
bc =
    1.0000e+00    4.0000e-01
ac =
    5.0000e-01    1.0000e+00
aa =
    5.0000e-01   1.0000e+00   1.0000e+00   5.0000e-01   4.0000e-02
g =
    2.00    2.00    6.25
tau =
    2.0000e+00    1.2500e+01
gs =
    5.0000e-01    6.6000e-01    5.0000e-01
rr =
    -5.0000e-01 ± 8.1382e-01i
   -9.0288e-01 + 0.0000e+00i
   -9.7122e-02 + 0.0000e+00i
```

g2t

Purpose:

Find possible equivalent time constants τ for a given plant (ap,bp) under the conditions of desired stability index γ and the order of controller.

Synopsis and Description:

```
tau=g2t(ap,bp,nc,mc,gr)
```

Inputs:

bp,ap= numerator and denominator polynomials of the plant.
mc,nc= numerator and denominator orders of the controller.
gr= a vector of the desired stability index.

Outputs:

tau= possible equivalent time constants.

Example:

Consider the following third order plant.

$$G_p(s) = \frac{1}{0.25s^3 + 1.25s^2 + s}.$$

Let us find possible equivalent time constants under the condition of the desired stability index of $\gamma = [4\ 2\ 2.5]$ when a first-order controller $G_c(s) = (k_1s+k_0)/(l_1s+1)$ is considered.

```
>> ap=[0.25  1.25   1   0];    bp=[1];
>> gr=[4   2   2.5];    nc=1;    mc=1;
>> tau=g2t(ap,bp,nc,mc,gr)

tau =
                 0
                 0
        3.3333e+00
        1.4286e+00
```

gc

Purpose:
A batch file combining `g2c` and `c2g`. It provides various design results including CDM controller and their time/frequency responses.

Synopsis and Description:

```
gc
```

Initial information:

`bp,ap`= numerator and denominator polynomials of the plant $G_p(s)$.
`mc,nc`= numerator and denominator orders of the controller $G_c(s)$.
`gr`= a reference stability index.
`t`= a desired equivalent time constant
`tm`= time scale for time response, which is normally set to 1 or 0.5.

Outputs:

a plot representing time and frequency responses of $G(s)$, $T(s)$, and $S(s)$.
`ba,bc,ac`= polynomials of the designed CDM controller.
`aa`= characteristic polynomial of the closed loop system
`g`= stability index of `aa`.
`tau`= equivalent time constant of `aa`.
`gs`= stability limit of `aa`.
`rr`= roots of characteristic polynomial `aa`.
`pmgm`= phase and gain margins of the loop transfer function
`wpmgm`= phase and gain crossover frequencies of the loop transfer function

Example:
Consider a single input double output plant and a 2x1 controller in Matlab scripts as follows:

```
>> ap=[1  0  2  0];  bp=[0  0  1; 1  0  1];
>> gr=[2  2  2.5];  nc=1  mc=[1; 0];  t= 1;  tm= 1
>> gc

ba =1.2500e+01

bc =
     1.0500e+01    7.70
              0    4.80
```

```
ac =
      1.0000e-01    1.0000
aa =
      1.0000e-01    1.00    5.00    1.2500e+01    1.2500e+01
g =
      2.00    2.00    2.50
tau =1

gs =
      0.5    0.9    0.5
rr =
      -2.5000e+00 ± 3.4410e+00i
      -2.5000e+00 ± 8.1230e-01i
pmgm =
      3.7083e+01    1.2696e-01
wpmgm =
      4.7460e+00    1.8258e+00
```

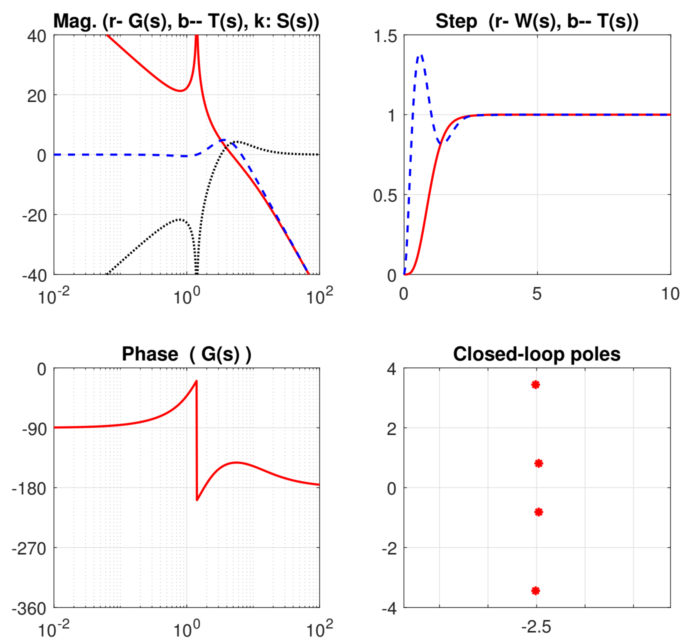

Fig. A.7 Poles, frequency and step responses of resulting systems provided by gc.

gk2c

Purpose:

Find a CDM controller (ac,bc) for a given plant (ap,bp) by adding a constraint on the steady state gain of controller to g2c.

Synopsis and Description:

```
[bc,ac,aa,g,tau,gs,rr]=gk2c(ap,bp,nc,mc,gr,t,k0)
```

Inputs:

bp,ap= numerator and denominator polynomials of the plant.
mc,nc= numerator and denominator orders of the controller.
gr= a vector of the desired stability index.
t= equivalent time constant τ.
k0= a steady state gain of numerator polynomial of the controller.

Outputs:

bc,ac= numerator and denominator polynomials of the CDM controller.
aa= characteristic polynomial of the closed loop system
g= stability index of the resulting aa.
tau= equivalent time constant of aa.
gs= stability limit of aa.
rr= roots of characteristic polynomial aa.

Example:

Consider the following third order plant.

$$G_p(s) = \frac{1}{0.25s^3 + 1.25s^2 + s}.$$

Suppose that we want to find a second-order controller $G_c(s) = (k_2 s^2 + k_1 s + k_0)/(l_2 s^2 + l_1 s + 1)$ with design parameters $\gamma = [2\ \ 2\ \ 2\ \ 2.5]$, $\tau = 0.77314$, and a fixed gain $k_0 = 10$.

```
>> ap=[0.25  1.25  1  0];   bp=[1];
>> gr=[2  2  2  2.5];   nc=2;   mc=2;   t=0.77314;   k0=10;
>> [bc,ac,aa,g,tau,gs,rr]=gk2c(ap,bp,nc,mc,gr,t,k0)
```

```
bc =
     1.0487     6.7314     10.0000
ac =
     0.0044     0.0922     1.0000
aa =
     0.0011     0.0286     0.3697     2.3910     7.7314     10.00
g =
     1.9998     2.0000     2.0000     2.5000
tau =0.77314

gs =
     0.5000     1.0001     0.9000     0.5000
rr =
     -7.1863 ± 8.2769i
     -3.9071 ± 2.2819i
     -3.6788 + 0.0000i
```

gk2t

Purpose:

Find possible equivalent time constants τ for a given plant (`ap,bp`) by adding a constraint on the steady state gain of controller to `g2t`.

Synopsis and Description:

```
tau=gk2t(ap,bp,nc,mc,gr,k0)
```

Inputs:

`bp,ap`= numerator and denominator polynomials of the plant.
`mc,nc`= numerator and denominator orders of the controller.
`gr`= a vector of the desired stability index.
`k0`= a steady state gain of numerator polynomial of the controller.

Outputs:

`tau`= possible equivalent time constants.

Example:

Consider the following third order plant.

$$G_p(s) = \frac{5}{0.05s^3 + 0.6s^2 + s}.$$

Let us find possible equivalent time constants under the condition of the desired stability index of $\gamma = [2\ 2\ 2.5]$ when a first-order controller $G_c(s) = (k_1 s + k_0)/(l_1 s + 1)$ with a fixed gain $k_0 = 1.0$ is considered.

```
>> ap=[0.05  0.6   1   0];    bp=[1];
>> gr=[2   2   2.5];    nc=1;    mc=1;    k0=1.0;
>> tau=gk2t(ap,bp,nc,mc,gr,k0)

tau =
       2.9084
       0.5640
      -0.4724
```

gkc

Purpose:

A batch file combining `gk2c` and `c2g`. It provides various design results including CDM controller and their time/frequency responses.

Synopsis and Description:

```
gkc
```

Initial information:

`bp,ap`= numerator and denominator polynomials of the plant $G_p(s)$.
`mc,nc`= numerator and denominator orders of the controller $G_c(s)$.
`gr`= a reference stability index.
`t`= a desired equivalent time constant
`tm`= time scale for time response, which is normally set to 1 or 0.5.
`k0`= a steady state gain of numerator polynomial of the controller.

Outputs:

a plot representing time and frequency responses of $G(s)$, $T(s)$, and $S(s)$.
`ba,bc,ac`= polynomials of the designed CDM controller.
`aa`= characteristic polynomial of the closed loop system
`g`= stability index of `aa`.
`tau`= equivalent time constant of `aa`.
`gs`= stability limit of `aa`.
`rr`= roots of characteristic polynomial `aa`.
`pmgm`= phase and gain margins of the loop transfer function
`wpmgm`= phase and gain crossover frequencies of the loop transfer function

Example:

Consider the same Example as in the function `gk2c`. The outcomes of `gkc` are the same as those in the previous Example and in addition, `gkc` provides the following figure.

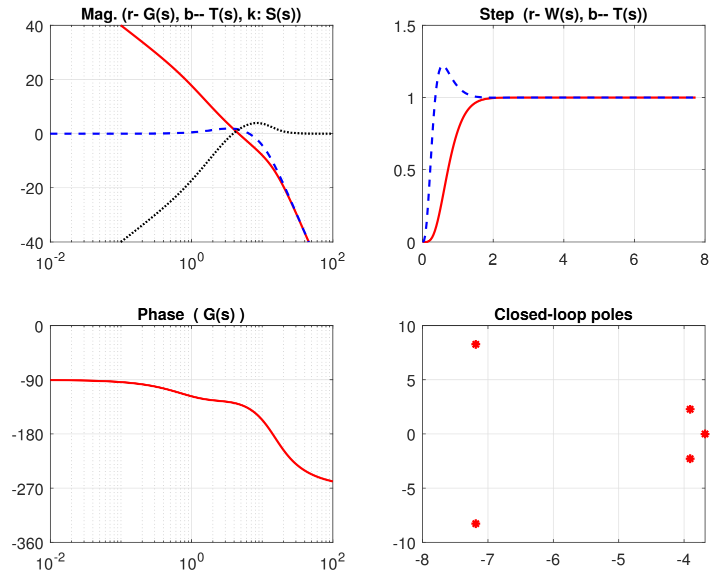

Fig. A.8 Poles, frequency and step responses of resulting systems provided by `gkc`.

gt2a

Purpose:
Obtain the characteristic polynomial and the squared polynomial corresponding to the specified stability index and equivalent time constant.

Synopsis and Description:

```
[aa,aq]=gt2a(g,t)
```

Inputs:

g= reference stability index
t= desired equivalent time constant.

Outputs:

aa= characteristic polynomial $P(s)$ corresponding to the specified (g,t).
aq= squared polynomial of aa, which is $A_q = P(-s)P(s)$.

Example:
Find the polynomial and its squared polynomial corresponding to the stability index $\gamma = [2\ \ 2\ \ 2\ \ 2.5]$ and $\tau = 2.5$.

```
>> g=[2 2 2 2.5]; t=2.5;
>> [aa,aq]=g2a(g,an,a0)

aa =
      1.00   8.00   32.00   64.00   64.00   25.60
aq =
      1.00   0   128.00   409.60   819.20   655.36
```

The above coefficient vectors aa and aq represent

$$P(s) = s^5 + 8s^4 + 32s^3 + 64s^2 + 64s + 25.6,$$
$$A_q(x) = x^5 + 128x^3 + 409.6x^2 + 819.2x + 655.36, \quad (x = -s^2).$$

rresp2/rresp3

Purpose:

`rresp2` provides a step response plot of two transfer functions specified by $G_1(s) = N_1(s)/D_1(s)$ and $G_2(s) = N_2(s)/D_2(s)$. Similarly, `rresp3` is to obtain a step response plot of three transfer functions.

Synopsis and Description:

```
rresp2(num1,den1,num2,den2,tmax),
rresp3(num1,den1,num2,den2,num3,den3,tmax)
```

Inputs:

`num1, den1`= numerator and denominator polynomials of $G_1(s)$.
`num2, den2`= numerator and denominator polynomials of $G_2(s)$.
`num3, den3`= numerator and denominator polynomials of $G_3(s)$.
`tmax`= the maximum time to be simulated.

Example:

Obtain step responses of the following two transfer functions using `rresp2`.

```
>> num1=[0.4];   den1=[1   1   0.4];
>> num2=[0.4];   den2=[1   1.3416   0.4];   tmax=20;
>> rresp2(num1,den1,num2,den2,tmax)
```

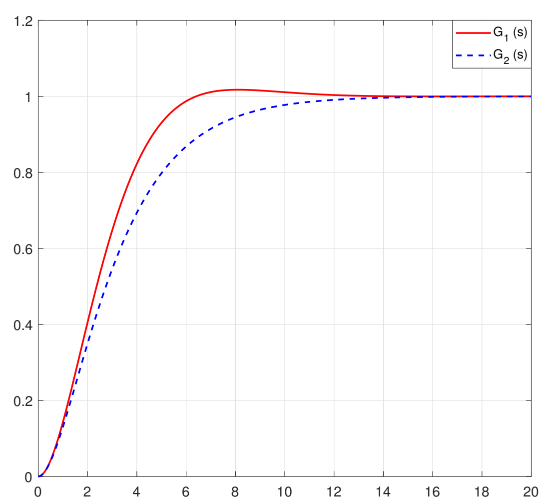

Fig. A.9 Step responses of two transfer functions $G_1(s)$ and $G_2(s)$.

tresp/tresp4

Purpose:

`tresp` obtains step response of two transfer functions specified by a common denominator $D(s)$ and two numerators $N_1(s), N_2(s)$. Similarly, `tresp4` is to obtain a step response plot of four transfer functions specified by four numerators and a common denominator.

Synopsis and Description:

```
tresp(num1,num2,den,tmax),
tresp4(num1,num2,num3,num4,den,tmax, ty1,ty2,ty3,ty4)
```

Inputs:

`num1, num2`= numerator polynomials of $G_1(s)$ and $G_2(s)$ respectively.
`num3, num4`= numerator polynomials of $G_3(s)$ and $G_4(s)$.
`den`= a common denominator polynomials of all transfer functions.
`tmax`= the maximum time to be simulated.
`ty1,` $\cdots$`, ty4`= title of each response.

Example:

Obtain step responses of the following two transfer functions using `tresp`.

```
>> num1=[0  0.2];  num2=[1  0];den=[1  2  2  1  0.2];  tmax=20;
>> tresp(num1,num2,den,tmax)
```

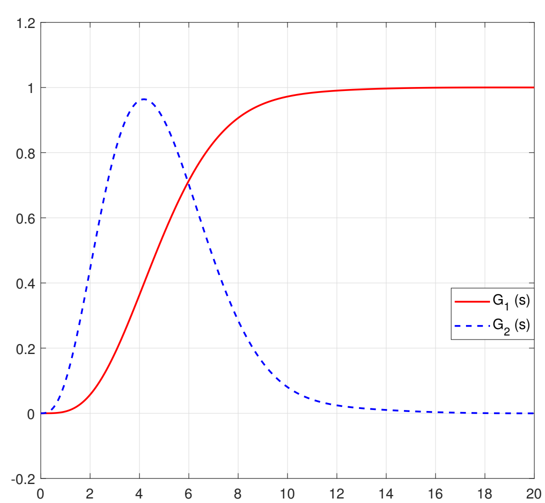

Fig. A.10 Step responses of two transfer functions $G_1(s)$ and $G_2(s)$.