Лекція 15-16. Моделювання динамічних властивостей вимірювального каналу КІВС в програмі MATLAB

1. Об’єкти (класи) пакету Control System Toolbox

Для аналізу і синтезу систем автоматичного керування доцільно використовувати пакет прикладних програм Control System Toolbox, який входить до складу системи MATLAB. Основні обчислювальні об’єкти цього ППП:

1. “Батьківський” об’єкт (клас) LTI (Linear Time-Invariant System – лінійні інваріантні за часом системи);

2. “Дочірні” об’єкти (класи), тобто підкласи класу LTI, що відповідають трьом варіантам представлення математичних моделей систем керування, а саме:

* TF - об′єкти (Transfer Fanction – передаточна функція);
* ZPK - об′єкти (Zero-Poie-Gain – нулі- полюси-коефіцієнт передачі);
* SS - об′єкти (State Space – простір стану).

Об′єкт LTI є найбільш загальним і охоплює інформацію, яка не залежить від конкретного виду системи (безперервна або дискретна система), а також від імен входів та виходів. Дочірні об′єкти визначаються форматом представлення математичної моделі системи керування (розглядаються математичні моделі в просторі “вхід-вихід” та моделі в просторі стану). Формат математичних моделей в просторі “вхід-вихід” охоплює класи TF та ZPK. Об′єкт класу TF характеризується векторами коефіцієнтів чисельника та знаменника раціональної передаточної функції. Об′єкт класу ZPK характеризується векторами, які мають значення нулів, полюсів передаточної функції системи та коефіцієнта передачі системи. Об′єкт класу SS визначається чотирма матрицями, які представляють динамічну систему або об′єкт керування в просторі стану.

1.6.1. Параметри (поля) LTI-об′єктів

Вхідні сигнали U можуть бути прикладені до кількості входів, які позначаються через NU, вектор виходу Y може вимірюватися на кількості NY виходів системи, а вектор стану системи Х може мати NX компонентів. В даному випадку використовуються такі позначення: SISO (OM) – одновимірна система, тобто система з одним входом та одним виходом; MIMO (БМ) – багатовимірна система, тобто система з багатьма входами та багатьма виходами. Інші параметри LTI-об’єкта:

1. Num – чисельник передаточної функції: вектор-рядок для одновимірних систем; масив комірок із векторів-рядків розміром (NY×NU) для багатовимірних систем, наприклад, {[1 0 3]; [1 2 3 ]}.

2. Den – знаменник передаточної функції: вектор-рядок для одновимірних систем; масив комірок із векторів-рядків розміром (NY×NU) для багатовимірних систем. Наприклад, tf({-5; [1 –5 6]}, {[1 –1]; [1 1 0]}) визначає систему з одним входом та двома виходами

[-5/(s-1)]

[(s^2-5s+6)/(s^2+s)]

3. Variable – найменування (тип) змінної (із переліку). Можливі варіанти: s, p, z, z^(-1) або q. За замовчуванням приймається s (для безперервних змінних) та z (для дискретних). Ім′я змінної впливає на відображення результатів та створює дискретну передаточну функцію для дискретних сигналів.

4. Z – нулі: вектор-рядок для одновимірних систем; масив комірок із векторів-рядків розміром (NY×NU) для багатовимірних систем.

5. P – полюси: вектор-рядок для одновимірних систем; масив комірок із векторів-рядків розміром (NY×NU) для багатовимірних систем.

6. K – коефіцієнти передачі: число для одновимірних систем; матриця (NY×NU) для багатовимірних систем.

7. A, B, C, D – матриці із рівнянь математичної моделі в формі простора стану



8. E – квадратна матриця, за замовчуванням E=eye(size(A)).

9. State Name – ім′я змінної стану (не обов’язкове). Можна використовувати масив комірок розміром (NX×1). Наприклад: {‘положення’; ‘швидкість’}.

10. Ts – період дискретизації в секундах, Ts=-1 для дискретних систем з невизначеною частотою дискретизації, Ts=0 для безперервних систем.

11. Td – затримка вхідних сигналів в секундах, використовується лише для безперервних систем. В дискретних системах для виставлення затримок використовується оператор d2d.

12. Input Name – імена входів. Для систем з одним входом використовується рядок символів. Для систем з декількома входами використовується масив комірок (NU×1) із рядків символів. Наприклад: ‘момент’.

13. Output Name – імена виходів. Для систем з одним виходом використовується рядок символів. Для систем з декількома виходами використовується масив комірок (NU×1) із рядків символів. Наприклад: ‘потужність’ або ‘швидкість’.

14. Notes – нотатки. Це довільний рядок символів або масив комірок із рядків символів.

15. Userdata – додаткова інформація користувача.

Для об'єктів класу LTI використовуються наступні функції (методи): augstate, damp, get, issiso, lticheck, pade, series, trange, balreal, display, gram, kalman, margin, parallel, set, tzero, bode, dssdata, impulse, kalmd, modreg, pzmap, sigma, uplus, canon, eig, inherit, lqgreg, nichols, quickset, ss2ss, zpkdata, connect, estim, initial, lqry, norm, rtg, ssdata, covar, evalfr, isct, lsim, nyquist, rlocfind, step, ctrb, fgrid, isempty, lti, obsv, rlocus, tfdata.

LTI-об′єкти створюються на основі файла lti.m, який знаходиться в папці @LTI. При цьому створюються лише шаблони LTI-об’єктів на основі деяких їх параметрів. Такий м-файл має наступний вигляд [7]:

Function sys = lti(p, m, t)

%LTI – конструктор LTI - об′єкта

%SYS =LTI(P,M) створює LTI - об′єкт

%розміром (P×M)

%SYS = LTI (P,M,T) створює LTI - об′єкт

%розміром (Р×М) з інтервалом дискретизації Т.

%За замовчуванням система вважається

%безперервною, а імена входів і виходів –

%векторами комірок з порожніми рядками

ni = nargin;

error (nargchk(1,3,ni))

if isa (p, ‘lti’)

% Дублювання LTI - об′єкта

sys = p;

return

elseif ni = = 3 & t~ = 0,

sys.Ts = t;

sys.Td = [ ];

else

sys.Ts = 0;

sys.Td = zeros(1, m);

end

estr = {‘ ‘};

sys.Input Name = estr (ones(m,1),1);

sys.Output Name = estr (ones(p,1),1);

sys.Notes = { };

sys.UserData = [ ];

sys.Version = 1.0;

sys = class (sys, ‘lti’); % кінець @lti/lti.m

Із наведеного м-файлу видно, що безпосереднє використання конструктора lti дає можливість задати лише кількість входів та виходів системи та величину кроку дискретизації. Решта параметрів LTI-об′єкта можуть бути визначені лише при використанні інших процедур, наприклад, процедури set. Конкретні чисельні характеристики лінійної системи можна задати лише за допомогою використання одного із конструкторів дочірніх класів.

Розглянемо приклад створення LTI-об′єкта для безперервної одновимірної системи.

>> sys = lti (1, 1)

lti object

Для пересвідчення, що створено LTI-об′єкт саме з вказаними параметрами, використаємо функцію get(sys) для отримання значень його параметрів:

>> get (sys)

Ts = 0

Td = 0

Input Name = {‘ ‘}

Output Name = {‘ ‘}

Notes = { }

UserData = [ ]

Більшість параметрів (полів) створеного LTI-об′єкта порожні, тільки двоє із них дорівнюють нулю. Звернення до конструктора не передбачає можливості задання таких параметрів LTI-об′єкта, як Input Name, Output Name, Notes, UserData. Ці поля можна задати шляхом використанні спеціальної функції set за наступною схемою:

set(<ім′я LTI-об′єкта>,’ <ім′я поля>’, <значення>)

Розглянемо це на прикладі задання в сформованому LTI-об′єкті sys значень деяких параметрів:

>> set (sys, ‘InputName’, ‘Кут’,

’OutputName’, ’Напруга’,

’Notes’, ’Гіротахометр’)

Проконтролюємо записану команду:

>> get (sys)

Ts = 0

Td = 0

InputName = {‘Кут’}

OutputName = {‘Напруга’}

Notes = {‘Гіротахометр’}

UserData = [ ]

1.6.2. Перелік функцій ППП Control System Toolbox

# Створення моделі САК на основі LTI-об’єкта

|  |  |
| --- | --- |
| Функція | Результат виконання функції |
| ss | Створює модель в просторі стану |
| zpk | Створює модель в формі нулі / полюси / коефіцієнт передачі |
| tf | Створює модель в формі передаточної функції |
| dss | Створює дескриптор моделі в просторі стану |
| filt | Створює цифровий фільтр |
| set | Установка / модифікація параметрів LTI-об’єкта |
| ltiprops | Детальна довідка про параметрів LTI-об’єкта |

# Отримання інформації про модель САК

|  |  |
| --- | --- |
| Функція | Результат виконання функції |
| ssdata | Отримання інформації про матриці простору стану |
| zpkdata | Отримання інформації про нулі / полюси / коефіцієнт передачі |
| tfdata | Отримання чисельника та знаменника передаточної функції |
| dssdata | Отримання інформації про дескриптор Ssdata |
| get | Отримання інформації про параметри LTI-об’єкта |

*Отримання інформації про окремі характеристики моделі*

|  |  |
| --- | --- |
| Функція | Результат виконання функції |
| class | Інформація про тип моделі (SS, ZPK, TF) |
| size | Інформація про розміри матриць входу та виходу |
| isempty | Перевірка, чи є LTI-об’єкт порожнім |
| iset | Перевірка, чи є модель безперервною |
| isdt | Перевірка, чи є модель дискретною |
| isproper | Перевірка, чи є модель безпомилковою |
| issiso | Перевірка, чи має модель один вхід та один вихід |
| isa | Перевірка, чи є LTI-об’єкт моделлю заданого типу |

# Перетворення моделі САК

|  |  |
| --- | --- |
| Функція | Результат виконання функції |
| ss | Перетворення в форму простору стану |
| zpk | Перетворення в форму нулів / полюсів / коефіцієнта передачі |
| tf | Перетворення в форму передаточної функції |
| c2d | Перетворення безперервного часу на дискретний |
| d2c | Перетворення дискретного часу на безперервний |
| d2d | Перевизначення дискретної системи або додавання затримок вхідних впливів |

# “Арифметичні” операції

|  |  |
| --- | --- |
| Функція | Результат виконання функції |
| + або - | Додавання або віднімання LTI-систем (паралельне з′єднання) |
| \* | Добуток LTI-систем (послідовне з′єднання) |
| \ | Лівостороннє ділення – sys1\sys2 рівнозначно inv(sys1)\*sys2 |
| / | Правостороннє ділення – sys1/sys2, що рівнозначно sys1\*inv(sys2) |
| ‘ | Перетранспонування |
| .’ | Транспонування карти входу / виходу |
| [..] | Горизонтальне / вертикальне з′єднання LTI-систем |
| inv | Інверсія LTI- системи |

##### Моделі динаміки

|  |  |
| --- | --- |
| Функція | Результат виконання функції |
| pole,eig | Полюси системи |
| tzero | Нулі системи |
| pzmap | Карта нулів-полюсів |
| dcgain | Коефіцієнт передачі по постійній напрузі |
| norm | Норма LTI-системи |
| covar | Коваріація відгуку на білий шум |
| damp | Частота власних коливань та демпфування на основі полюсами системи |
| esort | Сортування полюсів безперервної системи за їх дійсними частинами |
| dsort | Сортування полюсів дискретної системи за їх модулями |
| pade | Апроксимація затримок за часом методом Паде |

# Моделі простору стану

|  |  |
| --- | --- |
| Функція | Результат виконання функції |
| rss, drss | Генерування випадкових моделей простору стану |
| ss2ss | Перетворення змінних стану |
| canon | Канонічна форма моделі простору стану |
| ctrb, obsv | Матриці керованості та спостережуваності |
| gram | Детермінанти Грама для керованості та спостережуваності системи |
| ssbal | Діагональне балансування матриць простору стану |
| balreal | Балансування входу-виходу на основі детермінантів Грама |
| modreg | Редукція стану моделі |
| minreal | Мінімальна реалізація та скорочення нулів і полюсів |
| augstate | Збільшення виходу за рахунок приєднання станів |

# Відгук за часом

|  |  |
| --- | --- |
| Функція | Результат виконання функції |
| step | Відгук на ступеневу функцію |
| impulse | Відгук на одиничний імпульс |
| initial | Відгук на задані початкові умови стану |
| lsim | Відгук на довільні входи |
| ltiview | Аналіз відгуків за допомогою графічного інтерфейсу |
| gensig | Генерує періодичні сигнали |
| stepfun | Генерує одиничний стрибок |

# Частотний відгук

|  |  |
| --- | --- |
| Функція | Результат виконання функції |
| bode | Діаграма Боде для частотного відгуку (ЛАХ, ЛФЧХ) |
| sigma | Частотний графік сингулярних значень |
| nyquist | Діаграма Найквіста |
| nichols | Діаграма Ніколса |
| evalfr | Обчислення частотного відгуку на заданій частоті |
| freqresp | Частотний відгук для набору частот |
| margin | Запаси по фазі та по амплітуді (підсиленню) |

# З′єднання систем

|  |  |
| --- | --- |
| Функція | Результат виконання функції |
| append | Об′єднання LTI-систем шляхом з′єднання входів та виходів |
| parallel | Узагальнене паралельне з′єднання (процедура +) |
| series | Узагальнене послідовне з′єднання (процедура \*) |
| feedback | З′єднання двох систем зворотнім зв′язком |
| star | З′єднання зіркою Редхеффера |
| connect | Отримання SS-моделі на основі опису структурної схеми |

Оператори синтезу регуляторів методами класичної теорії керування

|  |  |
| --- | --- |
| Функція | Результат виконання функції |
| rlocus | Діаграма Еванса розташування коренів (нулів та полюсів) |
| rlocfind | Інтерактивне визначення коефіцієнта підсилення каналу керування на основі цілеспрямованого розташування полюсів |
| acker | Розташування полюсів одновимірної системи |
| place | Розташування полюсів багатовимірної системи |
| estim | Інтерактивне визначення підсилення кола похибки спостерігаючого пристрою (естіматора) |
| reg | Створює регулятор (матриці підсилення каналу керування та каналу спостереження) |

# Оператори інтерактивного проектування регуляторів LQG

|  |  |
| --- | --- |
| Функція | Результат виконання функції |
| lqr,dlqr | Лінійно-квадратичний (LQ) регулятор (матриця підсилення кола зворотного зв′язу) |
| lqry | LQ-регулятор з виважуванням виходу |
| lqrd | Дискретний LQ-регулятор для безперервних систем |
| kalman | Фільтр Калмана (спостерігаючий пристрій) |
| kalmd | Дискретний фільтр Калмана для безперервних систем |
| lqgreg | Формувач LQG-регулятора на основі LQ-коефіцієнтів та фільтру Калмана |

### *Вирішувачі матричних рівнянь*

|  |  |
| --- | --- |
| Функція | Результат виконання функції |
| lyap | Рішення безперервних рівнянь Ляпунова |
| dlyap | Рішення дискретних рівнянь Ляпунова |
| care | Рішення безперервних алгебраїчних рівнянь Ріккаті |
| dare | Рішення дискретних алгебраїчних рівнянь Ріккаті |

### *Демонстраційні програми*

|  |  |
| --- | --- |
| Функція | Результат виконання функції |
| ctrldemo | Введення до Control System Toolbox |
| jetdemo | Класичне проектування САК |
| diskdemo | Цифрове проектування контролера приводу жорсткого диску |
| milldemo | Одновимірні та багатовимірні моделі LQG-керування прокатним станом |
| kalmdemo | Проектування та моделювання фільтра Калмана. |

1.6.3. Приклад створення моделі об’єкта керування

### Синтез математичної моделі розглянемо на прикладі триступеневого астатичного гіроскопу. Рівняння рівноваги такого гіроскопу мають вигляд [7]:

**,**

**,**

### де n(t) i l(t) - моменти сил, що діють на гіроскоп за вісями підвісу;

### ε та β - кути обертання гіроскопу в просторі;

### λ - частота власних коливань гіроскопу.

### SS- модель, тобто модель в просторі стану, визначається, як відомо [8], двома рівняннями:

,

,

де U – вектор вхідних змінних;

Y – вектор вихідних змінних;

Х – вектор стану системи або об′єкта керування.

Тому для формування SS-моделі необхідно:

• визначити компоненти вектора U вхідних змінних;

• визначити компоненти вектора Y вихідних змінних, тобто змінних, які вимірюються в даній системі;

• установити, які змінні мають створювати вектор Х стану системи (вони є вихідними змінними інтеграторів, що входять до складу системи);

• за допомогою змінних стану привести задану систему диференційних рівнянь до нормальної форми Коші, тобто до системи диференційних рівнянь першого порядку, записаних відносно похідних.

Будемо вважати, що моменти сил є “входами” гіроскопу, а кути обертання гіроскопу – його “виходами”. При цьому модель гіроскопу має два входи (n(t),l(t)) та два виходи (ε,β ).

В якості змінних стану приймаємо кути обертання гіроскопу та їх перші похідні за часом:



Тоді рівняння стану системи записуються:



Введемо до розгляду вектори та матриці. Вектор виходу приймає вигляд:

Y=[ε,β]T

Вектор входу дорівнює:

U=[n(t),l(t)]T

Вектор стану системи:

X=[x1, x2, x3, x4]T

При цьому матриці мають наступний вигляд:





Введемо матриці A, B, C, D до командного вікна MATLAB з урахуванням того, що λ=10:

>> lambda=10;

>> A=zeros(4,4);

>> A(1,3)=1; A(2,4)=1; A(3,4)= - lambda;

>> A(4,3)= lambda; A

A=

0 0 1 0

0 0 0 1

0 0 0 –10

0 0 10 0

>> B=zeros(4,2); B(3,1)=1; B(4,2)=1;B

B=

0 0

0 0

1 0

0 1

>> C=zeros(2,4); C(1,1)=1; C(2,2)=1; С

C=

1 0 0 0

0 1 0 0

>> D=zeros(2,2); D

D=

0 0

0 0

Після цього можна розпочати створення LTI-об′єкта з ім′ям GYROss, використовуючи модель в просторі стану:

>> GYROss=ss(A, B, C, D)

A=

x1 x2 x3 x4

x1 0 0 1 0

x2 0 0 0 1

x3 0 0 0 -10

x4 0 0 -10 0

B=

u1 u2

x1 0 0

x2 0 0

x3 1 0

x4 0 1

C=

x1 x2 x3 x4

y1 1 0 0 0

y2 0 1 0 0

D=

x1 x2

y1 0 0

y2 0 0

Continuous-time system

Далі можна виконати деякі перетворення моделі системи.

Знайдемо передаточні функції створеної системи. Їх має бути чотири, так як система має два входи та два виходи. Використаємо процедру перетворення tf:

>> GYROtf=tf(GYROss)

Transfer function from inout 1 to output …

S^2

#1: --------------

s^4+100s^2

10s

#2: ---------------

s^4+100s^2

Transfer function from input 2 to output …

-10s

#1: --------------

s^4+100s^2

s^2

#2: -----------------

s^4+100s^2

Далі виконаємо перетворення введеної SS-моделі в ZPK-модель за допомогою процедури zpk:

>> GYROzpk=zpk(GYROss)

Zero/pole/gain from input 1 to output …

S^2

#1: -------------

s^2(s^2+100)

10s

#2:-----------------

s^2(s^2+100)

Zero/pole/gain from input 2 to output …

-10s

#1:----------------

s^2(s^2+100)

s^2

#2:----------------

s^2(s^2+100)

В зв′язку з тим, що перша модель GYROss була створена безпосередньо процедурою-конструктором на основі заданих чисельних даних, а наступні (GYROtf i GYROzp) – шляхом перетворення уже створеної моделі, то модель, яка створена конструктором вважається основною, а решта – допоміжними.

SS-модель в MATLAB можна створювати на основі системи диференційних рівнянь, не вирішених відносно похідних, тобто коли система описується сукупністю рівнянь:



де Е – довільна квадратна матриця розміром (n×n);

n – порядок заданої системи диференційних рівнянь.

При цьому використовується вже не функція ss, а спеціальна функція dss, яка відрізняється від попередньої функції тим, що необхідно додатково задати п’яту матрицю Е. Для прикладу розглянемо той же самий гіроскоп, що заданий рівняннями:

 (1.1)

При тих же самих змінних U, X і Y, що і в попередньому прикладі, отримуємо векторно-матричні рівняння, де матриці А та Е мають наступний вигляд:



Матриці В,С і D залишаються такими ж, як і в попередньому прикладі.

Введемо в MATLAB матриці A і E при таких значеннях параметрів: H=10, J1=2, J2 =3.

>> H=10; J1=2; J2=3;

>> A=zeros(4,4); A(1,3)=1; A(2,4)=1;

>> A(3,4)= -H; A(4,3)=H; A

A=

0 0 1 0

0 0 0 1

0 0 0 –10

0 0 10 0

>> E=eye(4,4);E(3,3)=J1; E(4,4)=J2; E

E=

1 0 0 0

0 1 0 0

0 0 2 0

0 0 0 3

Після цього можна задати SS-модель за допомогою процедури dss:

>> GYROss=dss(A, B, C, D, E)

A=

x1 x2 x3 x4

x1 0 0 1 0

x2 0 0 0 1

x3 0 0 0 -10

x4 0 0 10 0

B=

u1 u2

x1 0 0

x2 0 0

x3 1 0

x4 0 1

C=

x1 x2 x3 x4

y1 1 0 0 0

y2 0 1 0 0

D=

u1 u2

y1 0 0

y2 0 0

E=

x1 x2 x3 x4

x1 1 0 0 0

x2 0 1 0 0

x3 0 0 2 0

x4 0 0 0 3

Continuous-time system

На основі створеної моделі отримаємо допоміжні TF- та ZPK- моделі.

>> GYROtf=tf(GYROss)

Transfer function from input 1 to output …

0.5s^2

#1: ------------------

s^4+16.67s^2

1.667s

#2: ---------------------

s^4+16.67s^2

Trasfer function from input 2 to output …

-1.667s

#1: -------------------

s^4+16.67s^2

0.3333s^2

#2: --------------------

s^4+16.67s^2

>> GYROzp=zpk(GYROss)

Zero/pole/gain from input 1 to output …

0.5 s^2

#1: --------------------

s^2 (s^2+16.67)

1.6667s

#2: ---------------------

s^2 (s^2+16.67)

Zero/pole/gain from input 2 output …

-1.6667s

#1: -------------------

s^2 (s^2+16.67)

0.3333 s^2

#2: --------------------

s^2 (s^2+16.67) .

В наведених прикладах за основу приймалася SS-модель. Але за основну можна обрати будь-яку із двох інших моделей. Наприклад, за основну виберемо модель в форматі передаточних функцій.

Система, що задається диференційними рівняннями (1.1), і має ті ж самі вхідні та вихідні змінні, описується за допомогою чотирьох передаточних функцій, які утворюють матрицю передаточних функцій розміром (2х2). Кожний із стовпців цієї матриці включає передаточні функції, які відповідають деякій одній вхідній величині та усім (двом) вихідним величинам. Певний рядок матриці, навпаки, включає передаточні функції довільної вихідної величини відносно усіх входів системи. Матриця передаточних функцій записується таким чином:

 (1.2)

Відповідно до рівнянь (1.1) елементи матриці (1.2) дорівнюють:



Для створення TF-моделі на основі передаточних функцій з урахуванням їх конкретних значень спочатку необхідно створити два масиви комірок:

* масив комірок розміром (2х2) із векторів коефіцієнтів знаменників передаточних функцій;
* масив комірок розміром (2х2) із векторів коефіцієнтів усіх чисельників передаточних функцій.

Спочатку створюється вектор коефіцієнтів загальної частини знаменників:

,

потім – вектор додаткового множника в деяких знаменниках:

.

Далі створюється вектор коефіцієнтів другого знаменника шляхом згортання цих двох векторів (операція відповідає добутку поліномів):

.

На основі цих двох векторів формується масив комірок знаменників передаточних функцій

.

Масив комірок чисельників передаточних функцій дорівнює:

.

На основі цих даних створюється TF-модель. Вказана послідовність дій реалізується таким командами MATLAB:

>> Vzn1=[J1\*J2, 0, H^2]

Vzn1=

6 0 100

>> V=[1, 0]

V=

1 0

>> Vzn2=conv(Vzn1,V)

Vzn2=

6 0 100 0

>> for k1=1:2

for k2=1:2

den(k1, k2)={Vzn1};

end

end

>> den(1, 2)={Vzn2}; den(2, 1)={Vzn2}

den =

[1×3 double] [1×4 double]

[1×4 double] [1×3 double]

>> num ={J2, -H; H, J1}

num =

[3] [-10]

[10] [2]

>> GYROtf=tf(num, den)

Transfer function from input 1 to output …

3

#1: -------------

6 s^2 + 100

10

#2: ---------------

6 s^3 + 100s

Transfer function from input 2 to output …

-10

#1: ----------------

6 s^3 + 100s

2

#2: -----------------

6 s^2 + 100