

## Design of linear control systems with known structure and parameters. Matlab Control Systems Toolbox

---

### ТЕХНОЛОГИИ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ СИСТЕМ УПРАВЛЕНИЯ

**Средства проектирования линейных стационарных систем управления по известной структуре и параметрах системы. МатЛАБ Control Systems Toolbox.**

*Цель работы:* ознакомление с современными инструментальными средствами анализа и синтеза систем автоматического управления.

*Задача работы:* изучение возможностей пакета МатЛАБ - Control System на демонстрационных примерах.

*Приборы и принадлежности:* Персональный компьютер, MatLAB.

#### ОБЩИЕ СВЕДЕНИЯ

Пакет МатЛАБ Control Systems Toolbox предназначен для моделирования, анализа и проектирования линейных непрерывных и дискретных систем автоматического управления. Его можно использовать совместно с пакетом Simulink Control Design.

Control Systems Toolbox имеет следующие основные характеристики.

- Формы представления линейных систем: передаточная функция; система дифференциальных уравнений в форме пространства состояний; в виде наборов полюсов, нулей и коэффициентов усиления; в виде частотной характеристики;
- Образование типовых структур последовательным, параллельным соединениями звеньев или соединениями с обратной связью;
- Построение откликов и других характеристик во временной и частотной областях для анализа стабильности и быстродействия;
- Построение корневых годографов, АФЧ и других классических характеристик.
- Автоматическая настройка ПИД-регуляторов;
- Преобразование форм представления моделей, дискретизация непрерывных моделей, аппроксимация низкого порядка систем высокого порядка;
- Алгоритмы Ляпунова (LAPACK) и Сильвестра (SLICOT) оптимизации систем по точности и быстродействию.

Для описания моделей используется класс объектов **ЛТИ** (линейных стационарных систем) и его дочерние подклассы – формы представления операторов преобразования входных переменных в переменные выхода:

- **tf** (передаточная функция):  $H(s) = \frac{A(s)}{B(s)} = \frac{a_1s^n + a_2s^{n-1} + \dots + a_{n+1}}{b_1s^m + b_2s^{m-1} + \dots + a_{m+1}}$ , полиномиальная форма;
- **filt** (импульсная передаточная функция):  $H(z^{-1}) = \frac{A(z^{-1})}{B(z^{-1})} = \frac{a_0 + a_1z^{-1} + \dots + a_nz^{-n}}{b_0 + b_1z^{-1} + \dots + b_mz^{-m}}$
- **zpk** (набор нулей, полюсов и коэффициент передачи),  $H(s) = k \frac{(s - z_1)\dots(s - z_n)}{(s - p_1)\dots(s - p_m)}$

факторизованная форма;

- **ss** (дифференциальные уравнения для непрерывных объектов и разностные уравнения для дискретных объектов),

$$\left\{ \begin{array}{l} \frac{dx}{dt} = Ax(t) + Bu(t) \\ y(t) = Cx(t) + Du(t) \end{array} \right\}, \quad \left\{ \begin{array}{l} x[n+1] = Ax[n] + Bu[n] \\ y(k) = Cx[n] + Du[n] \end{array} \right\};$$

- **dss** (специальное описание операторов пространства состояния **ss**),

$$\left\{ \begin{array}{l} E \frac{dx}{dt} = Ax(t) + Bu(t) \\ y(t) = Cx(t) + Du(t) \end{array} \right\}, \quad \left\{ \begin{array}{l} Ex[n+1] = Ax[n] + Bu[n] \\ y(k) = Cx[n] + Du[n] \end{array} \right\};$$

- **frd** (отклик в частотной области – зависимость комплексного коэффициента передачи от частоты):

Частота	Реакция
10	0.8142 + 0.5806i
40	-3.1509 + 2.4641i
70	-2.5706 - 6.5109i
90	6.8377 - 5.8520i
100	9.9610 - 0.8828i

## Настройка PID регуляторов SISO систем

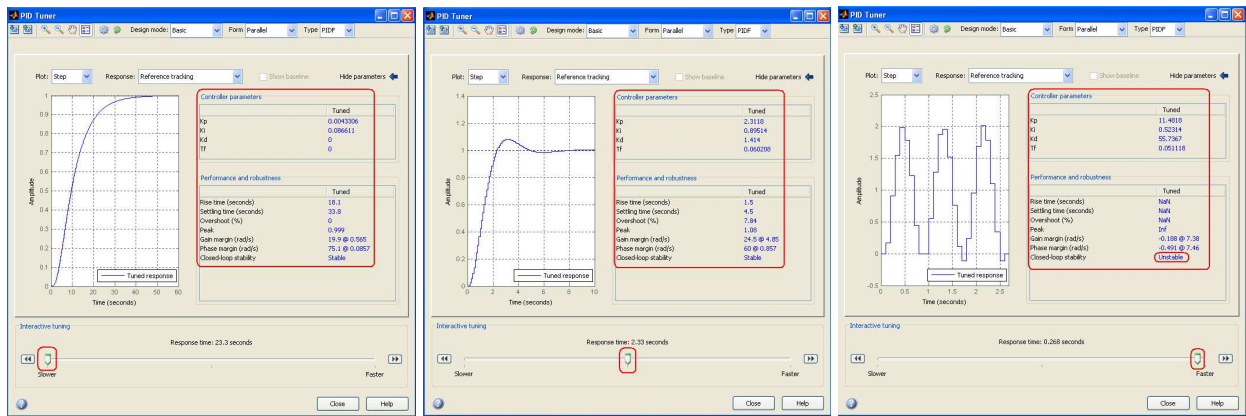
Запуск инструмента для настройки PID регуляторов систем с одним входом и одним выходом (SISO систем) выполняется через командную строку командой

>> **pidtool**

или

>> **pidtool (plant) %**, где **plant** – система, которой необходимо управлять.

Программа строит реакцию на ступенчатое воздействие которое можно изменять движком вручную с целью получения желаемой переходной характеристики по быстродействию или устойчивости при этом параметры регулятора выбранной структуры перерасчитываются автоматически.



**Рис. 1.** Пример интерактивного выбора (движком) переходного процесса с автоматическим пересчетом параметров регулятора и отклика для модели  $G_c = \text{zpk}([], [-1 -1 -1], 1)$ ;  $G_d = \text{c2d}(G_c, 0.1)$ ;  $\text{pidtool}(G_d, 'pidf')$ .

Данные в PID Tuner можно загружать из MAT файла или из рабочей области Base workspace  
MAT file

Import a linear system as: Plant model  
Plant model  
Baseline controller

Plot: Step  
Step  
Bode

в виде

областях для следующих передаточных функций модели

Реакция (Response)	Оператор	Воздействие	Реакция
Reference tracking	$CG/(1+CG)$	r	y
Controller effort	$C/(1+CG)$	r	u
Input disturbance Rejection	$G/(1+CG)$	d1	y
Output disturbance Rejection	$1/(1+CG)$	d2	y
Open-Loop	CG	r	y
Plant	G	u	y

Структура PID регулятора может быть выбрана из следующего списка 
Type PIDF  
P  
I  
PI  
PD  
PDF  
PID  
PIDF
. В режиме проектирования Design mode: Basic  
Basic  
Extended можно подобрать параметры регулятора обеспечивающего требуемые запасы устойчивости по фазе и амплитуде АФЧХ.

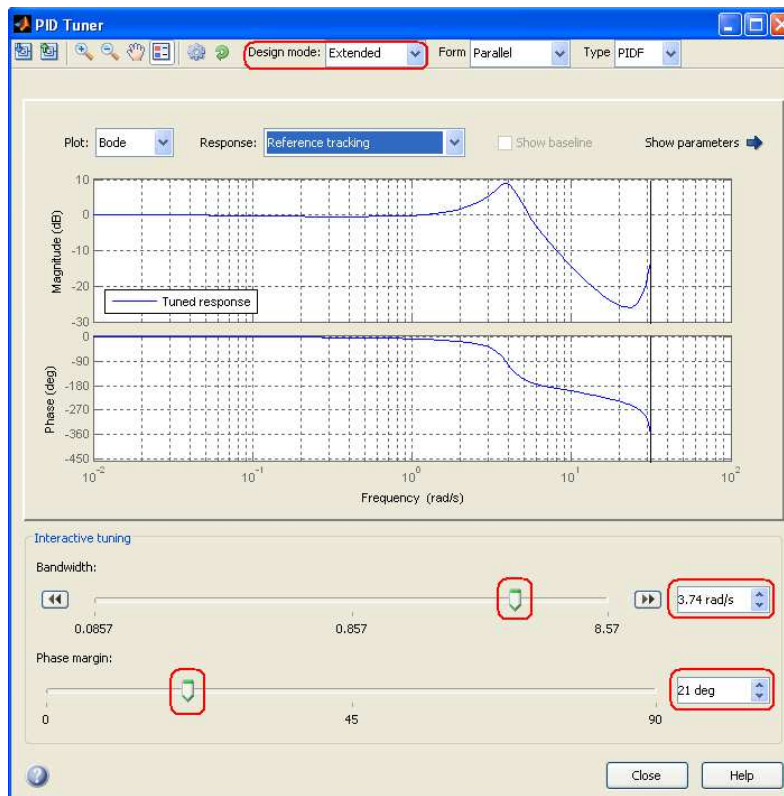


Рис. 2. Формирование требуемого запаса устойчивости по фазе и амплитуде.

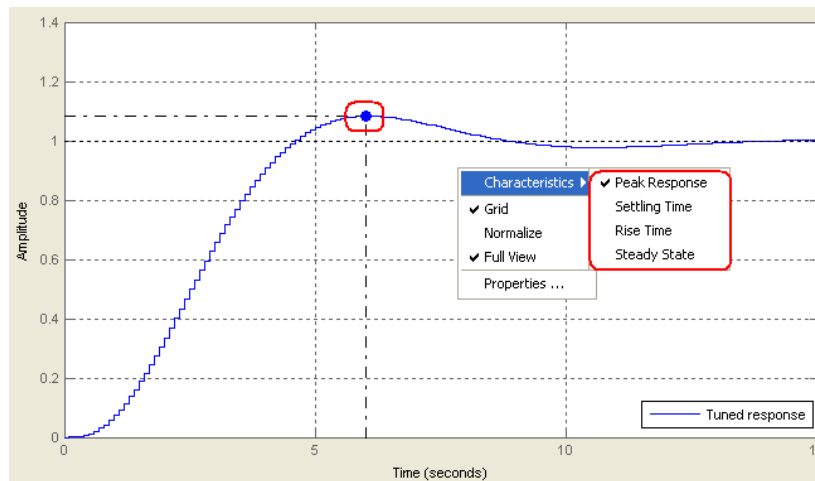


Рис. 3. Выделение параметров переходного процесса.

### Синтез SISO систем управления

Использование SISO Design Tool (проектирование системы с одним входом и одним выходом) позволяет интерактивно изменять форму АФЧХ или расположение корней системы управления наблюдая за соответствующими изменениями реакции и параметров корректирующего устройства.

Анализ устойчивости и быстродействия системы управления можно выполнить и в следующем порядке. Ввести передаточную функцию системы (Рис. 5, Рис. 6) и получить параметры регулятора заданной (Рис. 7) структуры, обеспечивающего требуемые характеристики переходного процесса (Рис. 8).

Вызов средств проектирования систем SISO (один вход – один выход) и графического интерфейса реализующего метод КГ (Рис. 4) выполняется командой

>> **rltool**

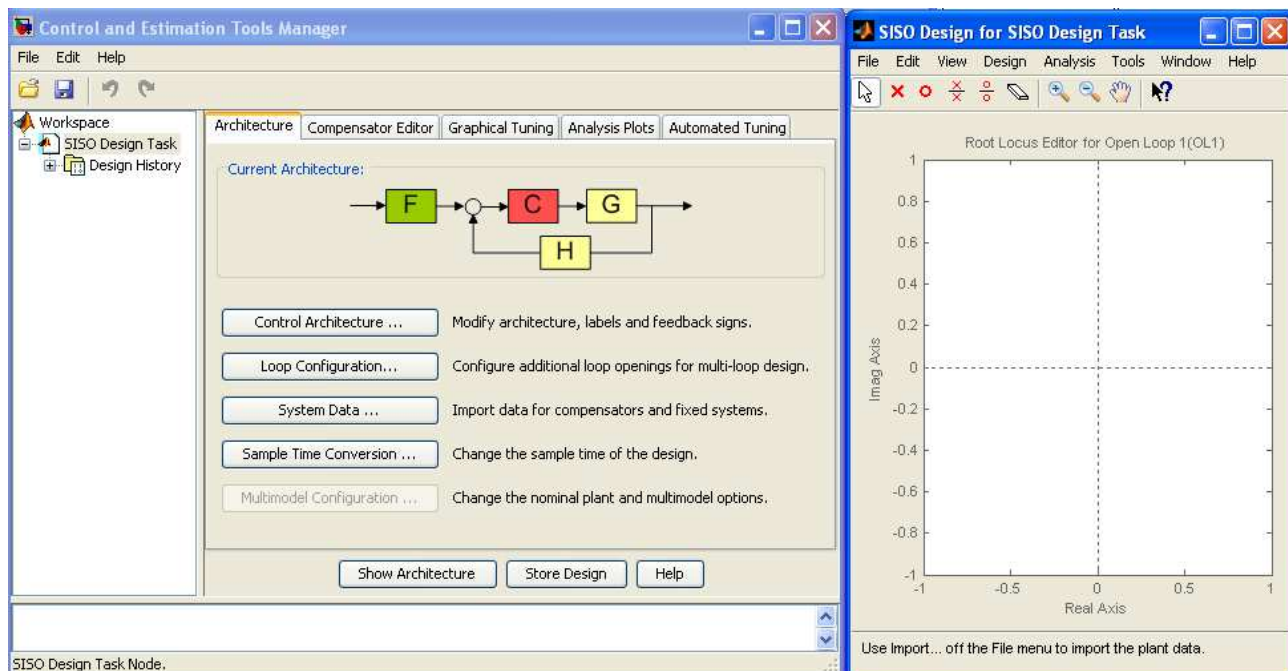


Рис. 4. Менеджер Control and Estimation Tools и SISO Design Tool которые вызываются командой **rltool**.

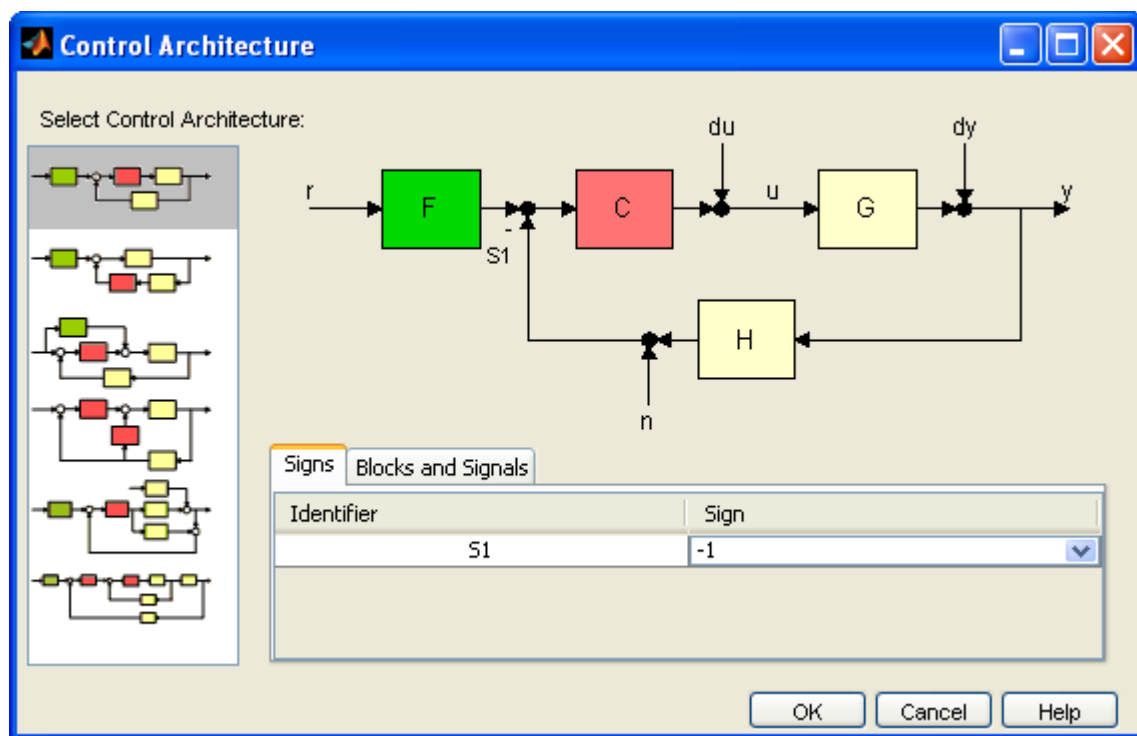


Рис. 5. Выбор структуры системы и обратных связей.

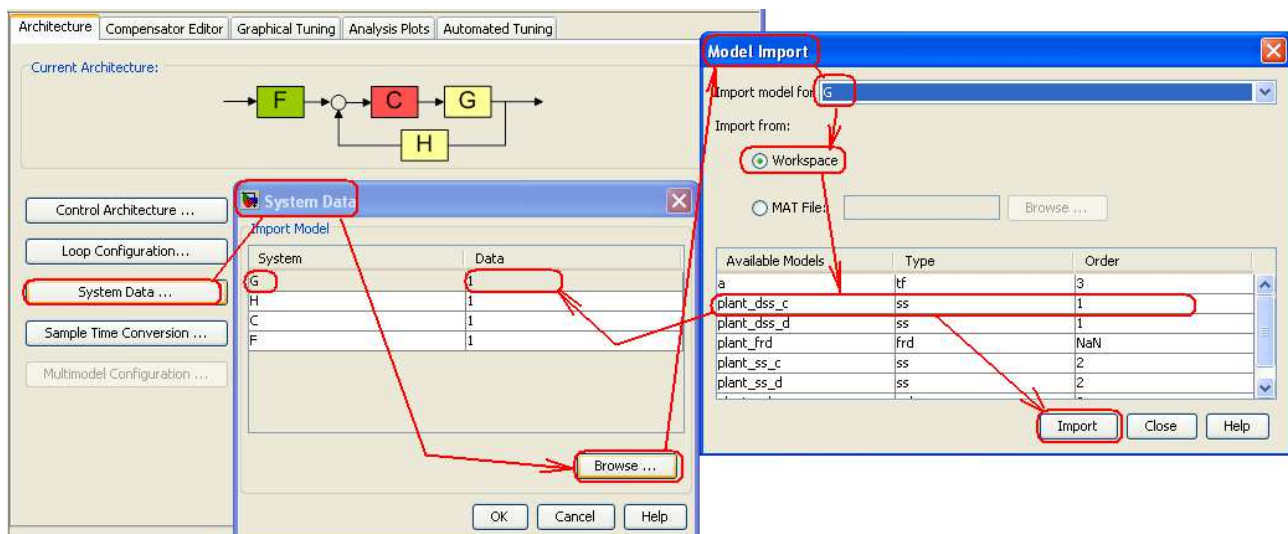


Рис. 6. Заполнение блоков структуры системы операторами из рабочего пространства (workspace) или mat файлов..

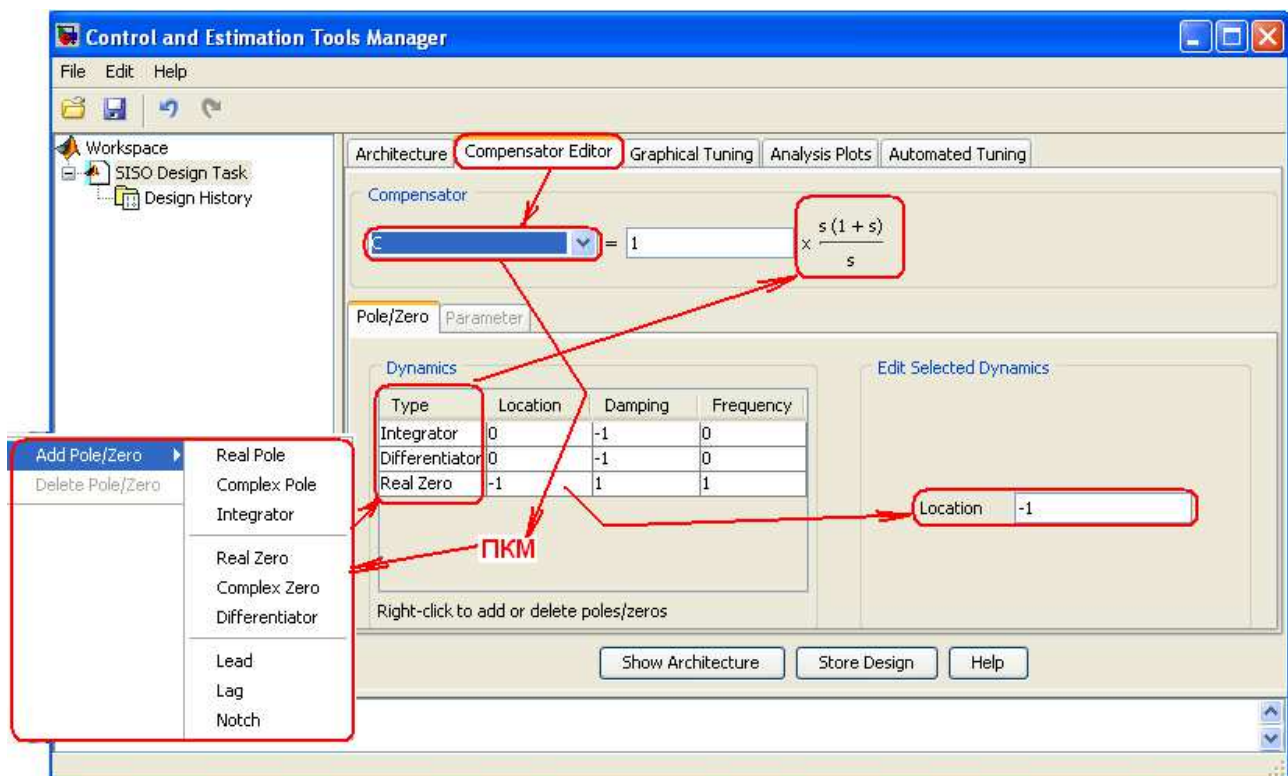
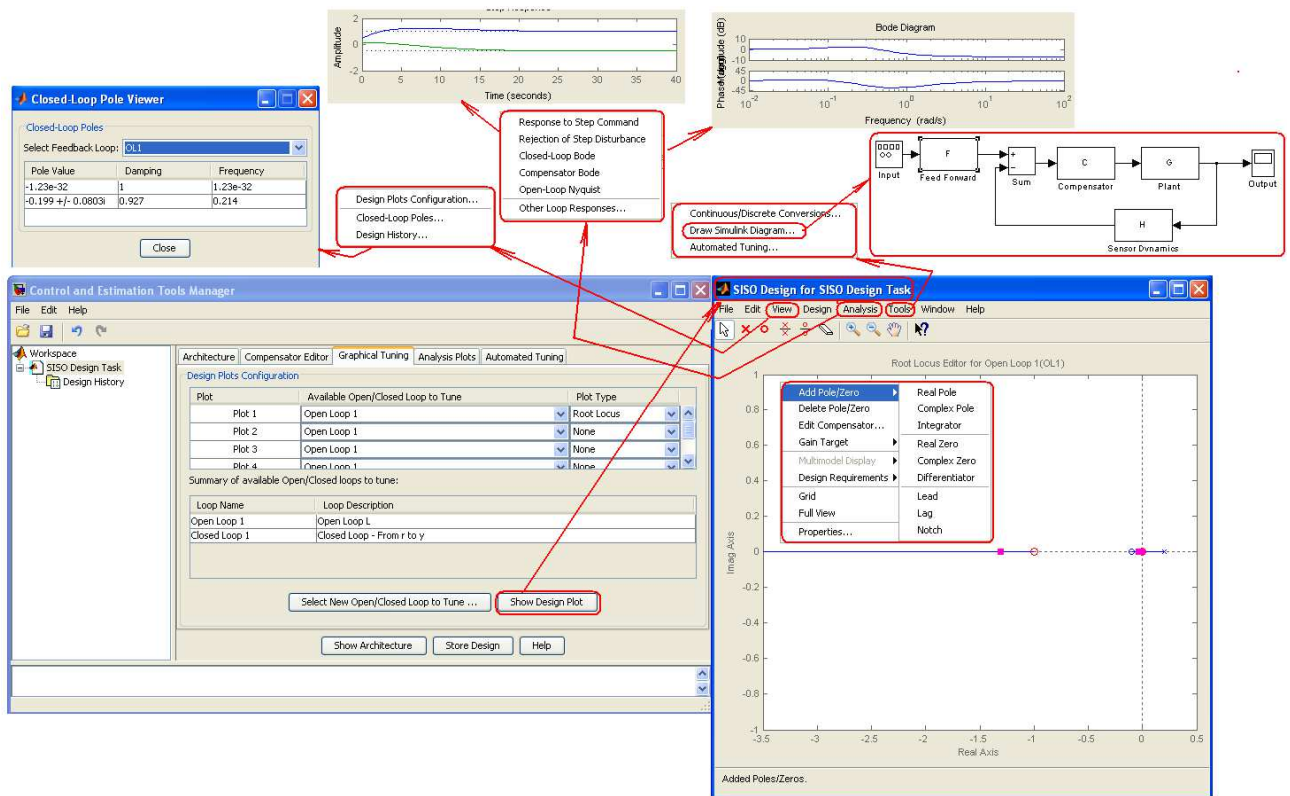


Рис. 7. Формирование структуры компенсатора (корректирующего устройства, регулятора).





**Рис. 8.** Подстройка нулей и полюсов, построение откликов во временной и частотных областях, перевод структуры в Simulink. SISO design Tool позволяет смещать, добавлять, удалять полюса и нули, находить параметры соответствующих корректирующих устройств, строить переходные процессы во временной и частотной областях а также получать структуру в формате Simulink модели.

**Табл. 1.** Функции пакета Control System [2]

Функция	Назначение
	<i>Формирование LTI (линейных стационарных) объектов</i>
ss	Создание модели пространства состояния
zpk	Создание модели нули-полюсы-коэффициенты передачи
tf	Создание модели в виде передаточной функции
dss	Специальное описание модели пространства состояния
filt	Создание модели цифрового фильтра
set	Установка (модификация) атрибутов LTI - модели
ltiprops	Получение детальной справки об атрибутах LTI – моделей
	<i>Извлечение данных</i>
ssdata	Извлечение матриц A,B,C,D пространства состояния
zpkdata	Извлечение данных о нулях, полюсах, коэффициенте передачи
tfdata	Извлечение числителя и знаменателя передаточной функции

dssdata	Получение информации о версии описателя dss
get	Получение информации о значениях свойств LTI_ модели
	<i>Получение информации об отдельных характеристиках модели</i>
class	Получение данных о типе модели
size	Получение данных о размерах матриц входа и выхода
isempty	Проверка, является ли LTI-модель пустой
isct	Проверка, является ли модель непрерывной
isdt	Проверка, является ли модель дискретной
isproper	Проверка, является ли модель правильной
issiso	Проверка, имеет ли модель один вход и один выход
isa	Проверка, является ли LTI-объект моделью заданного типа
	<i>Преобразование одного вида модели в другой</i>
ss	В модель пространства состояния
zpk	В модель нули-полюса-коэффициент передачи
tf	В модель передаточной функции
c2d	Переход от непрерывной модели к дискретной модели
d2c	Переход от дискретной модели к непрерывной модели
d2d	Переопределение дискретной системы или добавление задержек
	<i>Арифметические операции над моделями</i>
+, -	Сложение и вычитание LTI-систем (параллельное соединение)
*	Умножение LTI-систем (последовательное соединение)
inv	Обращение LTI-системы
\	Левое деление $sys1 \setminus sys2$ , равносильно $inv(sys1) * sys2$
/	Правое деление $sys1 / sys2$ , равносильно $sys1 * inv(sys2)$
'	Перетранспонирование
.'	Транспонирование карты входа-выхода
[..., ..., ...]	Горизонтальное объединение LTI-систем
[...; ...; ...]	Вертикальное объединение LTI-систем
	<i>Характеристики динамической системы</i>



pole, eig	Определение полюсов системы
tzero	Определение нулей системы
pzmap	Составление карты нулей-полюсов, вызов сетки $\omega$ , $\zeta$ – sgrid
dcgain	Нахождение коэффициента передачи при нулевой частоте
norm	Определение нормы LTI-системы
covar	Вычисление ковариации отклика на белый шум
damp	Определение частоты собственных колебаний и демпфирования
esort	Сортировка полюсов непрерывной системы по их величине $\text{Re}(p_i)$
dsort	Сортировка полюсов дискретной системы по их величине $\text{Re}(p_i)$
pade	Аппроксимация Паде задержек во времени
	<i>Модели пространства состояний</i>
rss, drss	Генерирование случайных моделей пространства состояний
ss2ss	Преобразование переменных состояния
canon	Переход к каноническим формам пространства состояния
ctrb, obsv	Матрицы управляемости и наблюдаемости системы
gram	Определители Грамма ( управляемости и наблюдаемости)
ssbal	Диагональная балансировка матриц пространства состояний
balreal	Балансировка входа-выхода на основе определителя Грамма
modred	Редукция состояния модели
minreal	Минимальная реализация и сокращение нулей и полюсов
augstate	Вывод на выход системы всех переменных состояния
	<i>Отклик (реакция системы) во времени</i>
step	Определение отклика $h(t)$ на единичный скачок $1(t)$
impulse	Определение отклика $\omega(t)$ на единичный импульс $\delta(t)$
initial	Определение отклика на заданные начальные условия состояния
lsim	Определение отклика при произвольных входных воздействиях
ltiview	Анализ откликов с помощью графического интерфейса
gensig	Генерирование периодических сигналов $u(t)$ для функции lsim
stepfun	Генерирование единичного скачка $1(t)$

	<i>Частотные характеристики САУ</i>
bode	Построение диаграммы Боде – частотной характеристики САУ
sigma	Построение частотного графика сингулярных значений
nyquist	Построение диаграммы Найквиста – ЧХ в полярных координатах
nichols	Построение диаграммы Николса, вызов сетки $L_3, \varphi_3$ – ngrid
ltiview	Анализ характеристик САУ с помощью графического интерфейса
evalfr	Расчет ЧХ САУ на заданной частоте (точки ЧХ)
freqresp	Расчет ЧХ САУ на заданном диапазоне частот
margin	Определение запасов по фазе и амплитуде $L_3, \varphi_3$
	<i>Объединение (коммутация) систем</i>
append	Объединение LTI-системы путем объединения входов и выходов
parallel	Обобщенное параллельное соединение (см. также +)
series	Обобщенное последовательное соединение (см. также *)
feedback	Соединение двух систем в виде петли обратной связи
star	Соединение звездой Редхеффера
connect	Получение ss-модели из описания блок схемы
	<i>Синтез методами корневого годографа и размещением полюсов</i>
rlocus	Построение диаграммы Эванса – корневого годографа (КГ) САУ
rlocfind	Выбор коэффициента передачи САУ по ее полюсам на КГ
rltool	Вызов графического интерфейса реализующего метод КГ
acker	Синтез регулятора К одномерной системы (или наблюдателя L)
place	Синтез регулятора К многомерной системы (или наблюдателя L)
estim	Синтез наблюдателя по заданным коэффициентам $L=[l_1; \dots; l_n]$
reg	Синтез регулятора по заданным матрицам обратной связи К и L
	<i>Синтез оптимальных линейно-квадратичных регуляторов (LQR)</i>
lqr, dlqr	Синтез регуляторов (для непрерывной и дискретной САУ)
lqry	Синтез регуляторов LQR со взвешиванием выхода $y$ и входа $u$
lqrd	Синтез дискретного регулятора LQR для непрерывной системы
kalman	Синтез непрерывного фильтра Калмана

kalmd	Синтез дискретного фильтра Калмана для непрерывной системы
lqgreg	Синтез LQR регулятора по коэффициентам К и фильтру Калмана
<i>Матричные уравнения</i>	
lyap	Решение непрерывных уравнений Ляпунова
dlyap	Решение дискретных уравнений Ляпунова
care	Решение непрерывных алгебраических уравнений Риккати
dare	Решение дискретных алгебраических уравнений Риккати
<i>Демонстрационные примеры в Control System Toolbox</i>	
jetdemo	Классическое проектирование САУ с углом рыскания
diskdemo	Цифровое проектирование контроллера привода жесткого диска
milldemo	Система автоматического управления прокатным станом
kalmdemo	Проектирование и моделирование фильтра Калмана

В пакет Control System входит функция **view** которая открывает графический интерфейс просмотра временных и частотных характеристик модели.

## ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

**Задание 1.** Освоение МатЛАБ форматов описания объектов систем управления (см. функции Табл. 1).

1. Введите в рабочее пространство МатЛАБ (workspace) объекты

1.1. в форме передаточной функции  $H(s) = \frac{A(s)}{B(s)} = \frac{a_1 s^n + a_2 s^{n-1} + \dots + a_{n+1}}{b_1 s^m + b_2 s^{m-1} + \dots + a_{m+1}}$ .

Пример, ввод передаточных функций  $plant\_tf = \begin{bmatrix} \frac{s+1}{s^2+2s+2} \\ \frac{1}{s} \end{bmatrix}$

выполняется с использованием следующих команд

```
num = {[1 1] ; 1};
den = {[1 2 2] ; [1 0]};
plant_tf1 = tf(num,den)
```

В результате, передаточные функции  $plant\_tf =$

$$1: \frac{s+1}{s^2+2s+2} \quad 2: \frac{1}{s}$$

Отдельные функции можно вызвать как  $plant\_tf(1)$  или  $plant\_tf(2)$

1.2. в форме импульсной передаточной функции  $H(z^{-1}) = \frac{A(z^{-1})}{B(z^{-1})} = \frac{a_0 + a_1 z^{-1} + \dots + a_n z^{-n}}{b_0 + b_1 z^{-1} + \dots + b_m z^{-m}}$ .

$$plant\_filt = \left[ \begin{array}{c} 1 \\ \frac{1+z^{-1}+z^{-2}}{1+0.3z^{-1}} \\ \frac{5+2z^{-1}}{5+2z^{-1}} \end{array} \right]$$

num = {1 , [1 0.3]};

den = {[1 1 2] , [5 2]};

plant\_filt = filt(num,den,'inputname',{ 'channel1' 'channel2'})

1.3. в форме набор нулей, полюсов и коэффициент передачи  $H(s) = k \frac{(s-z_1)\dots(s-z_n)}{(s-p_1)\dots(s-p_m)}$ .

Непрерывная функция:

$$plant\_zpk\_c = \frac{-2s}{(s-1+j)(s-1-j)(s-2)}$$

plant\_zpk\_c = zpk(0, [1-i 1+i 2], -2);

Дискретная функция:

$$plant\_zpk\_d = \left[ \begin{array}{c} \frac{1}{z-0.3} \\ \frac{2(z+0.5)}{(z-0.1+j)(z-0.1-j)} \end{array} \right]$$

z = {[ ] ; -0.5};

p = {0.3 ; [0.1+i 0.1-i]};

k = [1 ; 2];

plant\_zpk\_d = zpk(z,p,k,-1); % unspecified sample time

1.4. объект в форме набора дифференциальных уравнений  $\begin{cases} \frac{dx}{dt} = Ax(t) + Bu(t) \\ y(t) = Cx(t) + Du(t) \end{cases}$

$$A = \begin{bmatrix} 0 & 1 \\ -5 & -2 \end{bmatrix}; B = \begin{bmatrix} 0 \\ 3 \end{bmatrix}; C = [0 \ 1]; D = [0];$$

A = [0 1;-5 -2];

B = [0;3];

C = [0 1];

D = 0;

plant\_ss\_c = ss(A,B,C,D);

1.5. объект в форме разностных уравнений  $\begin{cases} x[n+1] = Ax[n] + Bu[n] \\ y(k) = Cx[n] + Du[n] \end{cases}$

$$A = \begin{bmatrix} 3 & 1 \\ -5 & -2 \end{bmatrix}; B = \begin{bmatrix} 0 \\ 3 \end{bmatrix}; C = [4 \ 1]; D = [0];$$

A = [3 1;-5 -2];

B = [0;3];

C = [4 1];

D = 0;

plant\_ss\_d = ss(A,B,C,D,0.25);

1.6.в специальной форме операторов пространства состояния  $\left\{ \begin{array}{l} E \frac{dx}{dt} = Ax(t) + Bu(t) \\ y(t) = Cx(t) + Du(t) \end{array} \right\}$

$$\left\{ \begin{array}{l} Ex[n+1] = Ax[n] + Bu[n] \\ y(k) = Cx[n] + Du[n] \end{array} \right\}$$

Непрерывная функция:

$$\left\{ \begin{array}{l} 5 \frac{dx}{dt} = x(t) + 2u(t) \\ y(t) = 3x(t) + 4u(t) \end{array} \right\}$$

`plant_dss_c = dss(1,2,3,4,5) % dss(A,B,C,D,E)`

Дискретная функция:

$$\left\{ \begin{array}{l} 10x[n+1] = 6x[n] + 7u[n] \\ y(k) = 8x[n] + 9u[n] \end{array} \right\}$$

`plant_dss_d = dss(6,7,8,9,10,0.1) % dss(A,B,C,D,E,Ts)`

1.7.в форме частотных характеристик - набора комплексных коэффициентов передачи и соответствующих частот

```
freq = logspace(1,2);
resp = .05*(freq).*exp(i*2*freq);
plant_frd = frd(resp,freq);
```

2. Извлеките данные различных форм представления из введенных объектов

`[a,b,c,d] = ssdata(sys); [a,b,c,d,Ts] = ssdata(sys)`, например,

```
[a,b,c,d] = ssdata(plant_tf1)
[a,b,c,d] = ssdata(plant_filt)
[a,b,c,d] = ssdata(plant_zpk_c)
[a,b,c,d] = ssdata(plant_zpk_d)
[a,b,c,d] = ssdata(plant_ss_c)
[a,b,c,d] = ssdata(plant_ss_d)
[a,b,c,d] = ssdata(plant_dss_c)
[a,b,c,d] = ssdata(plant_dss_d)
```

`[z,p,k] = zpkdata(sys); [z,p,k,Ts,Td] = zpkdata(sys)`

`[num,den] = tfdata(sys); [num,den,Ts] = tfdata(sys); [num,den,Ts,sdnum,sdden]=tfdata(sys)`

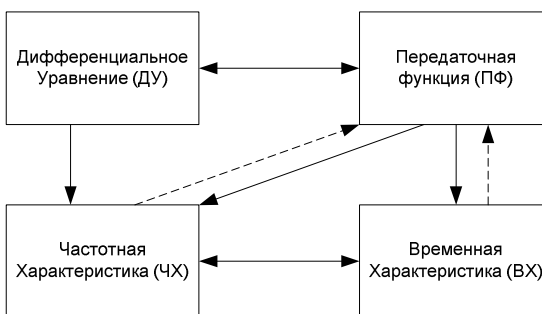
`[A,B,C,D,E] = dssdata(sys); [A,B,C,D,E,Ts] = dssdata(sys)`

`get(plant_zpk_c, 'k')`

3. Получите информацию об отдельных характеристиках модели

```
class(plant_tf)
size(plant_tf)
isempty(plant_tf)
isct(plant_tf)
isdt(plant_tf)
isproper(plant_tf)
issiso(plant_tf)
isa(plant_tf, 'tf')
```

4. Переведите форму представления оператора из одного вида в другой (“...наиболее рациональный для конкретных исследований” [1]).

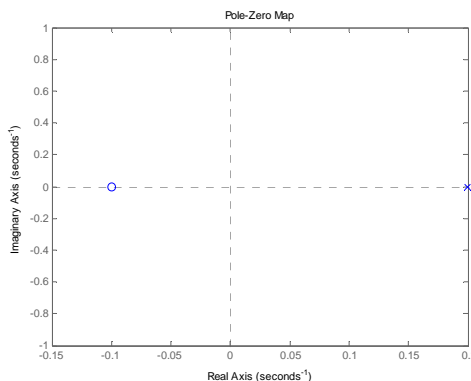


**Рис. 9.** Орграф взаимосвязи форм представления операторов [1, стр.47]. Переходы обозначенные штриховыми линиями могут дать неоднозначный результат.

```

ss(plant_tf)
zpk(plant_tf)
tf(plant_zpk_c)
c2d(plant_tf,0.1)
d2c(plant_ss_d)
d2d(plant_ss_d,0.2)
  
```

5. Выполните арифметические операции над моделями: сложение, вычитание, умножение, обобщение, транспонирование, объединение.
6. Определите характеристики динамической системы: полюса, нули, коэффициент передачи,



```

pzmap(plant_dss_c) :
p = covar(sys,5)
damp(plant_tf)
  
```

7. Выполните команды раздела “Модели пространства состояний” (Табл. 1): получите матрицы управляемости и наблюдаемости; найдите определители Гамма, и др.
8. Получите новые структуры последовательным, параллельным соединениями звеньев соединениями с обратной связью и другими операциями над моделями: сложение, вычитание, обращение, деление, транспонирование, и др. (см. Табл. 1).

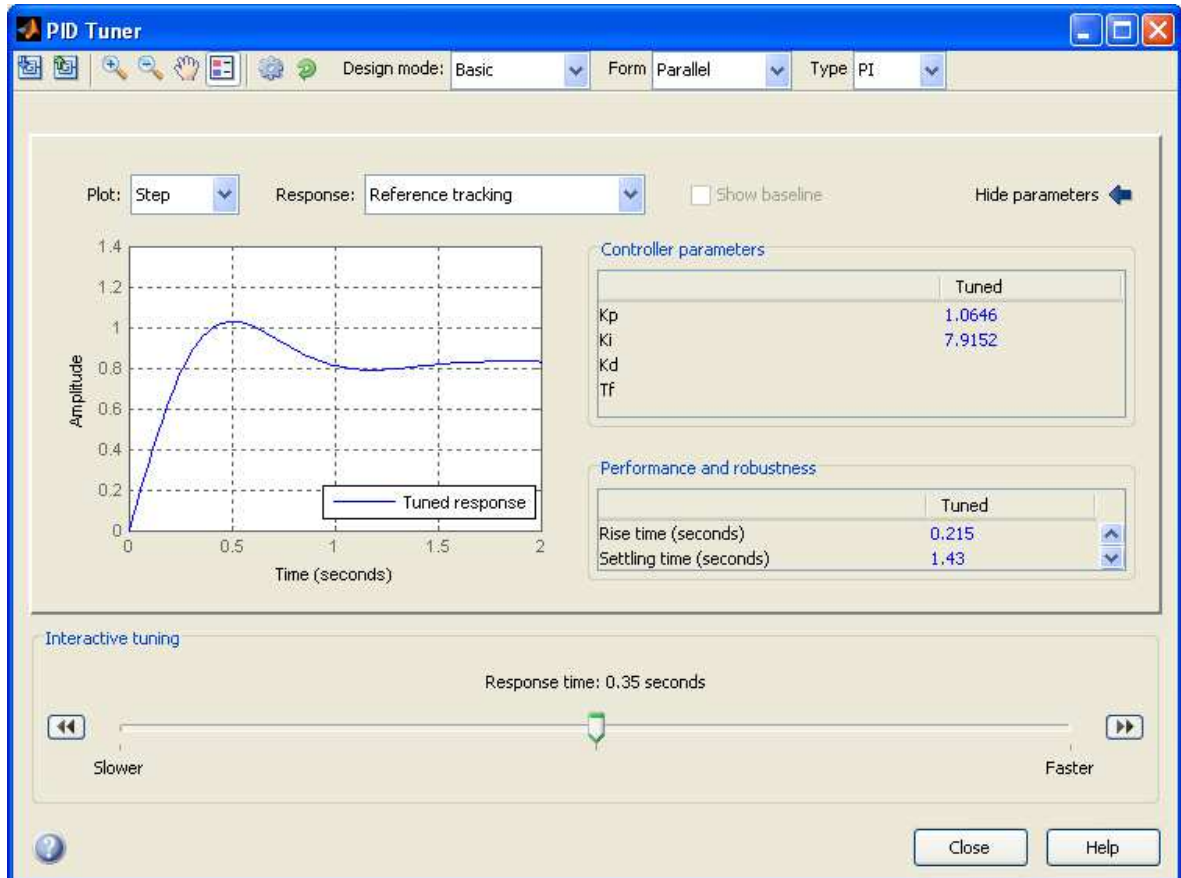
**Задание 2.** Синтез SISO системы методами корневого годографа.

1. Постройте диаграмму корневого годографа объекта, например, plant\_zpk\_c.  
`rlocus(tf(plant_zpk_c))`
2. Загрузите инструмент для синтеза системы управления (Рис. 4).
3. Выберите структуру объекта и введите его параметры (Рис. 5, Рис. 6).
4. Задайте структуру компенсатора (Рис. 7).

5. Расположите нули и полюса которые отвечают требуемым характеристикам системы (Рис. 8).
6. Постройте переходные процессы и найдите параметры корректирующего устройства обеспечивающего требуемое поведение системы (Рис. 8).

**Задание 3.** Автоматическая настройка PID регуляторов SISO системы.

1. Загрузите инструмент для настройки PID регулятора объекта, например, [plant\\_ss\\_c](#) задания 1.



2. Выбором структуры регулятора и смещением движка настройте переходный процесс на желаемый (Рис. 1).
3. Постройте отклики системы управления (Рис. 2, Рис. 3).
4. Запишите характеристики регулятора (Рис. 1).

**Задание 4.** Рассмотрение последовательности проектирования системы автоматического управления средствами пакета Control System на демонстрационном примере МатЛАБ [jetdemo](#) (классическое проектирование САУ с углом рыскания)

1. Настройте МатЛАБ на рабочий каталог.
2. Введите команду [jetdemo](#) через окно команд МатЛАБ.
3. Найдите в окне команд ссылку “View the published version of this demo”
4. Перейдите по ссылке в соответствующий раздел Help МатЛАБ.



5. Создайте новый m-файл.
6. Скопируйте в него код примера, показанного в Help. Задайте модель системы с помощью матриц A, B, C, D и, используя функцию ss.

```

% jetdemo -- from MatLAB HELP

A=[-.0558 -.9968 .0802 .0415;
   .598 -.115 -.0318 0;
  -3.05 .388 -.4650 0;
   0 0.0805 1 0];

B=[ .00729 0;
  -0.475 0.00775;
   0.153 0.143;
   0 0];

C=[0 1 0 0;
   0 0 0 1];

D=[0 0;
   0 0];

sys = ss(A,B,C,D); % Создание модели пространства состояния

set(sys, 'inputname', {'rudder' 'aileron'},...
      'outputname', {'yaw rate' 'bank angle'},...
      'statename', {'beta' 'yaw' 'roll' 'phi'});

axis(gca,'normal')
h = pzplot(sys);
setoptions(h,'FreqUnits','rad/s','Grid','off');

impulseplot(sys)
impulseplot(sys,20)
sys11 = sys('yaw','rudder'); % select I/O pair
h = bodeplot(sys11);
setoptions(h, 'FreqUnits','rad/s','MagUnits','dB','PhaseUnits','deg');
h = rlocusplot(sys11);
setoptions(h,'FreqUnits','rad/s')
h = rlocusplot(-sys11);
setoptions(h,'FreqUnits','rad/s')
k = 2.85;
cl11 = feedback(sys11,-k);
impulseplot(sys11,'b--',cl11,'r')
legend('open loop','closed loop',4)
cloop = feedback(sys,-k,1,1);
impulseplot(sys,'b--',cloop,'r',20) % MIMO impulse response
impulseplot(cloop('bank angle','aileron'),'r',18)
H = zpk(0,-0.2,1);
oloop = H * (-sys11); % open loop'
h = rlocusplot(oloop);
setoptions(h, 'FreqUnits','rad/s')
sgrid
k = 2.34;
wof = -k * H; % washout compensator
cloop = feedback(sys,wof,1,1);
impulseplot(sys,'b--',cloop,'r',20)
impulseplot(sys(2,2),'b--',cloop(2,2),'r',20)
legend('open loop','closed loop',4)

```

7. Выполните программу по шагам, сравнивая результаты с описанием программы в Help.

#### ПРИМЕР выполнения Задания 4.

Рассмотрим модель системы самолета во время полета [3]. Модель заданная в форме пространства состояний:

$$\begin{cases} \frac{dx}{dt} = Ax(t) + Bu(t) \\ y(t) = Cx(t) + Du(t) \end{cases}$$

имеет 4 состояния: beta (sideslip angle), phi (bank angle), yaw rate, roll rate и 2 входа: the rudder and aileron deflections.

Задаем модель системы с помощью матриц A, B, C, D и, используя функцию ss.

```
A=[-.0558 -.9968 .0802 .0415;  
    .598 -.115 -.0318 0;  
    -3.05 .388 -.4650 0;  
    0 0.0805 1 0];
```

```
B=[ .00729 0;  
    -0.475 0.00775;  
    0.153 0.143;  
    0 0];
```

```
C=[0 1 0 0;  
    0 0 0 1];
```

```
D=[0 0;  
    0 0];
```

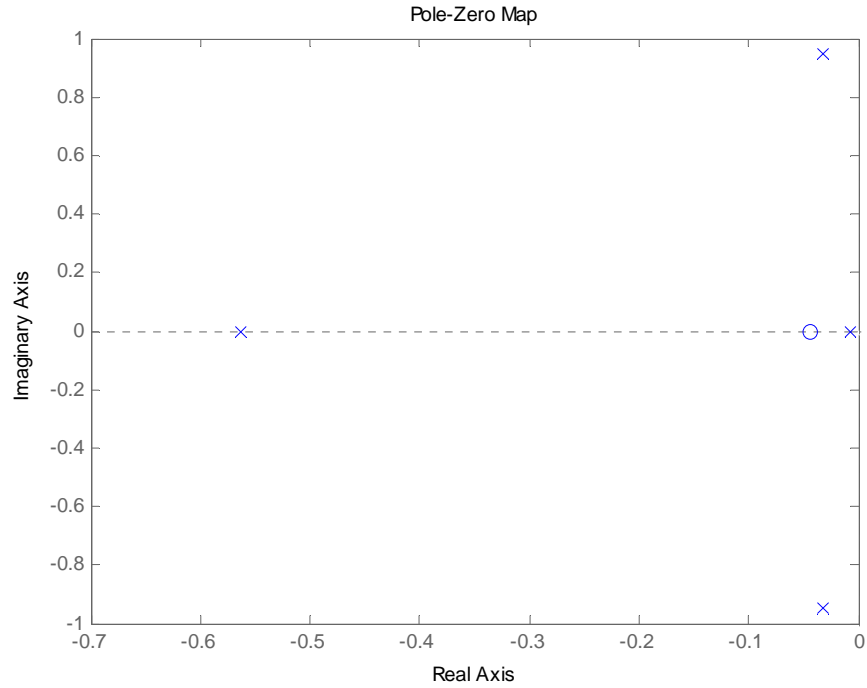
```
sys = ss(A,B,C,D); // Создание модели пространства состояния
```

И отмечаем выходы, входы и состояния.

```
set(sys, 'inputname', {'rudder' 'aileron'},...  
      'outputname', {'yaw rate' 'bank angle'},...  
      'statename', {'beta' 'yaw' 'roll' 'phi'});
```

Для анализа системы рассмотрим расположение нулей и полюсов на комплексной плоскости (Рис. 10) с помощью функции pzplot()

```
h = pzplot(sys);
```

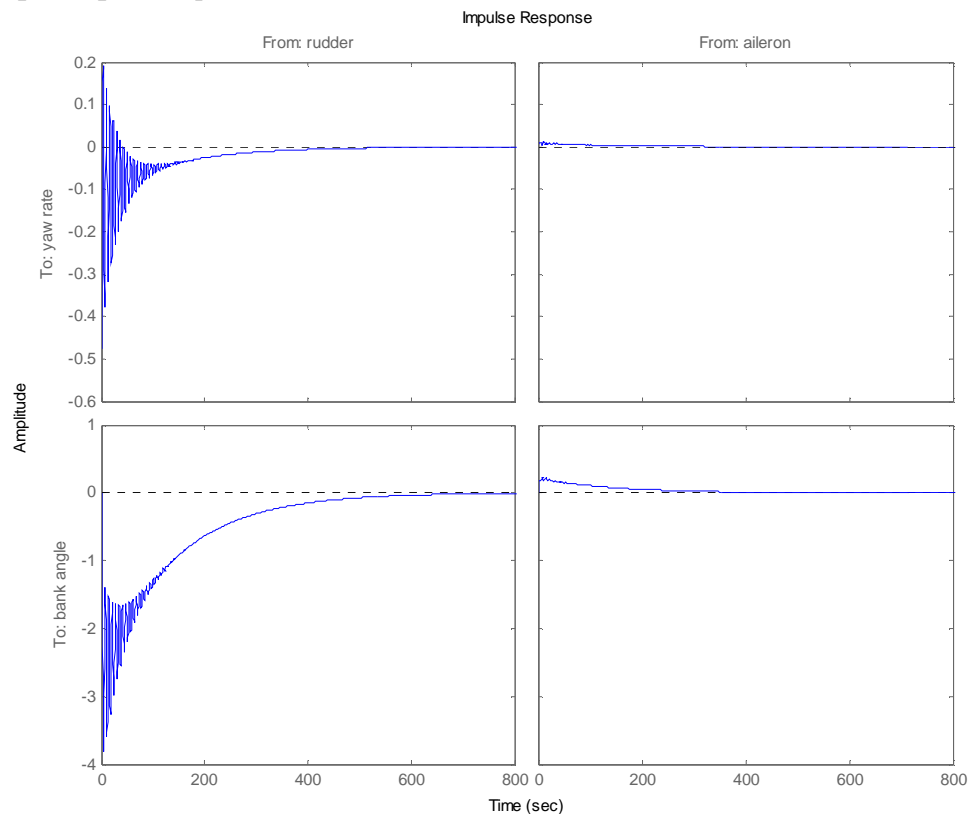


**Рис. 10.** Расположение нулей и полюсов на комплексной плоскости (корневой годограф).

На рисунке 1 видно, что мы имеем 2 медленно затухающих полюса. Следовательно, необходимо синтезировать регулятор, который увеличит затухание этих полюсов.

Для начала рассмотрим разомкнутую систему, для этого построим импульсные характеристики системы (Рис. 11)

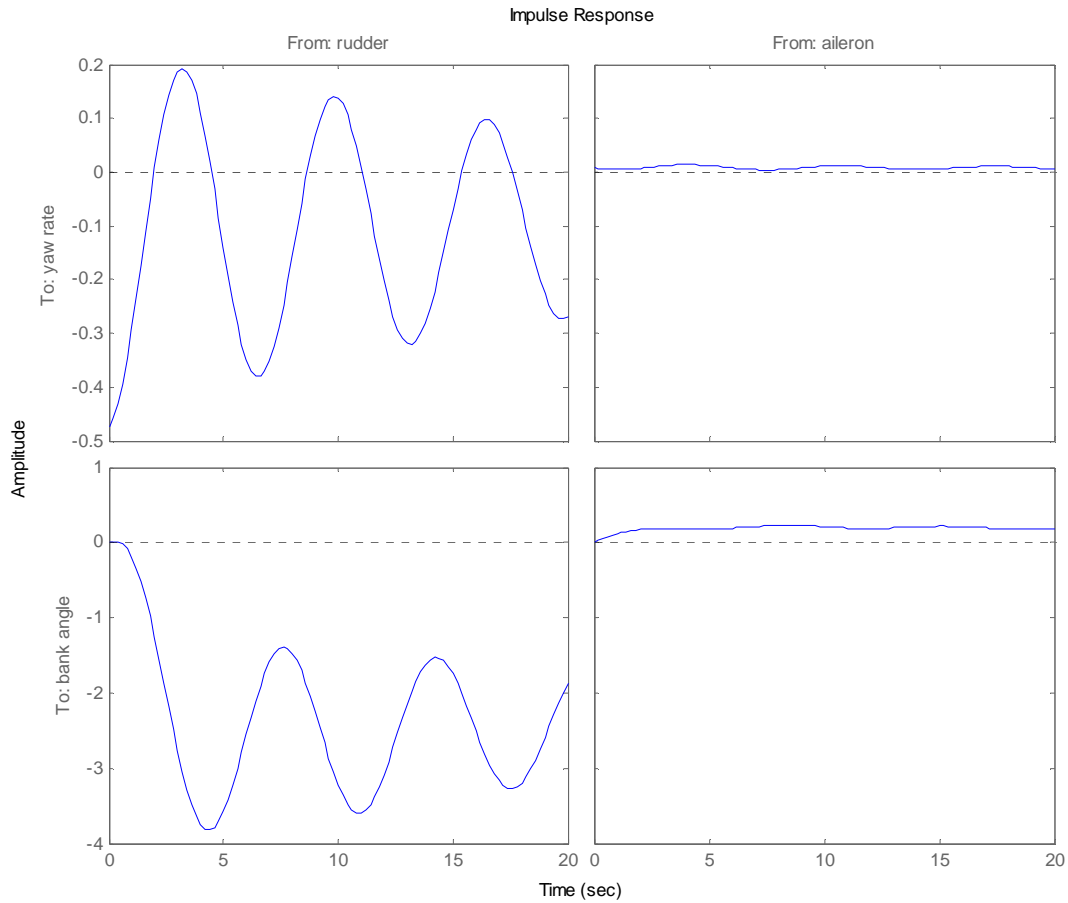
```
impzplot(sys)
```



**Рис. 11.** Импульсная характеристика системы.

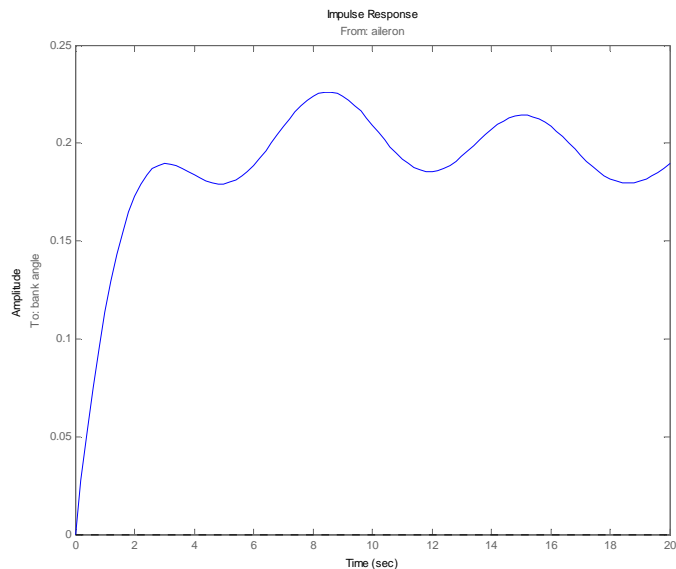
Для того чтобы более детально увидеть процессы в первые 20 секунд (Рис. 12), используем следующую команду:

```
impzplot(sys,20)
```



**Рис. 12.** Переходный процесс.

Для того чтобы более детально посмотреть один из рисунков необходимо правой кнопкой мыши кликнуть на соответствующий рисунок и выбрать нужную комбинацию “I/O Selection”.

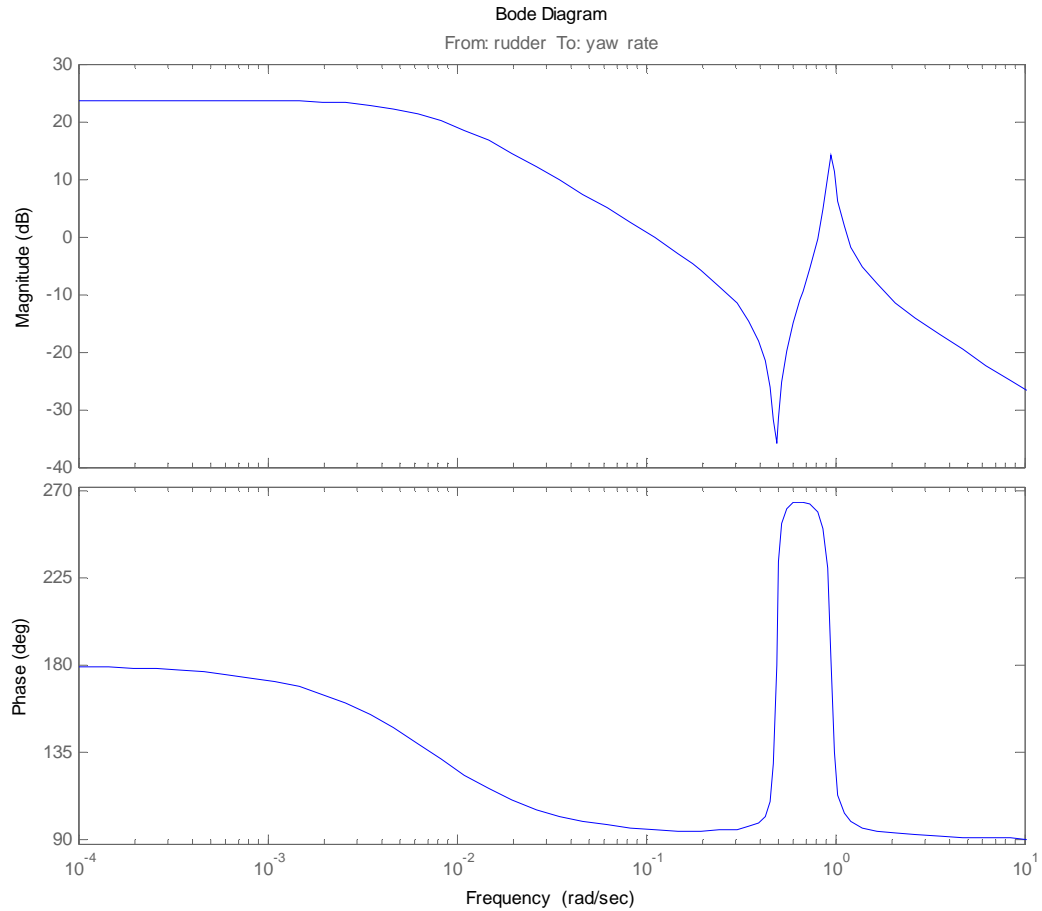


**Рис. 13.** Зависимости элерона от угла крена.

Обычно отклонение заслонок создается с помощью скорости отклонения как выхода и руля как входа.

Рассмотрим частотную характеристику данного процесса (Рис. 14).

```
sys11 = sys('yaw', 'rudder');    % select I/O pair  
h = bodeplot(sys11);
```

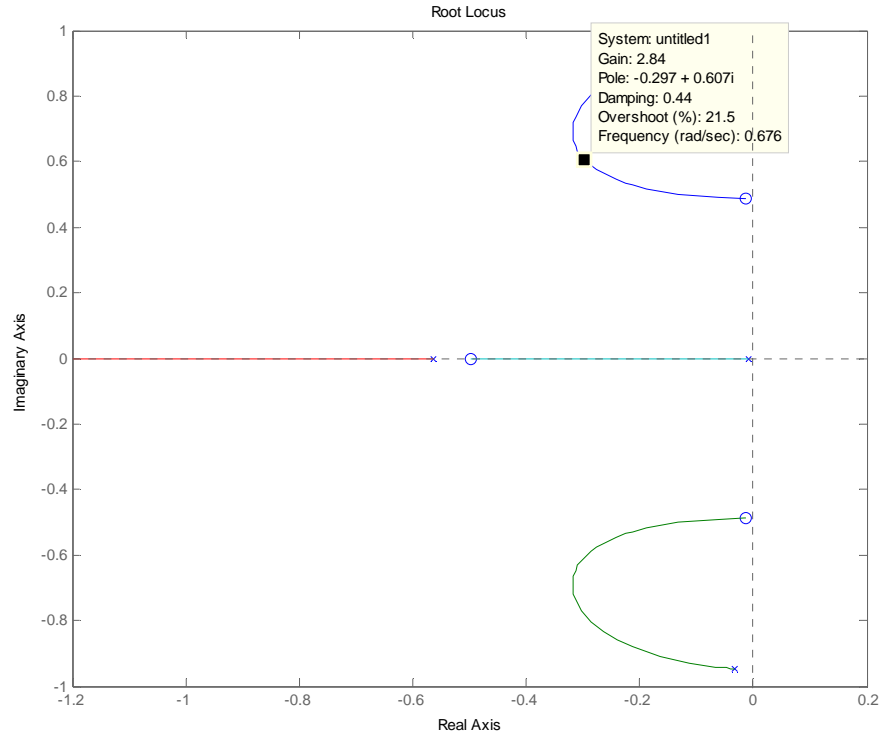


**Рис. 14.** Частотная характеристика.

На графике видно, что медленно-затухающий полюс сказывается на характеристиках.

Для обеспечения коэффициента демпфирования  $< 0,35$  используем самый простой регулятор – усиление. Для выбора наилучшего коэффициента усиления обратной связи воспользуемся корневым годографом (Рис. 15).

```
h = rlocusplot(-sys11);
```

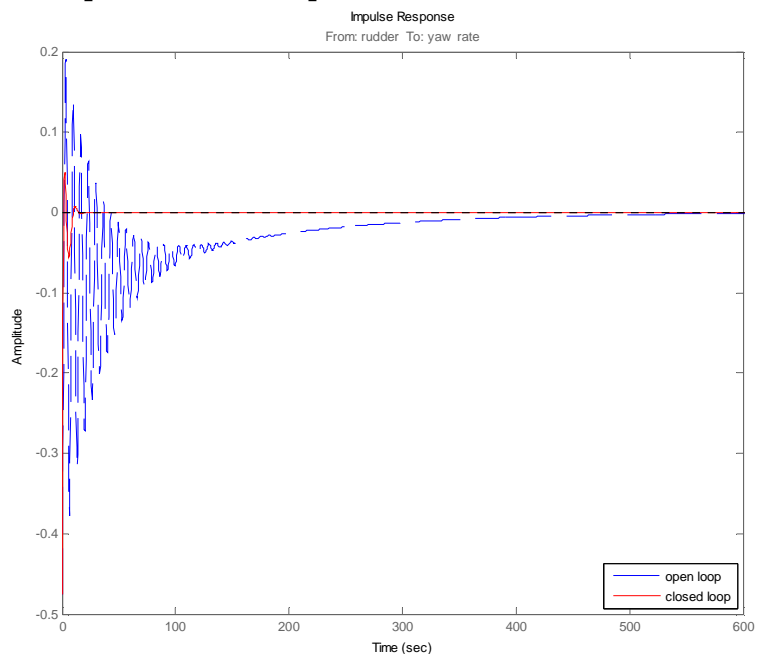


**Рис. 15.** Корневой годограф.

Для выбора подходящего коэффициента усиления необходимо нажать на синюю кривую и двигать квадратик до получения необходимого значения. Наилучшее затухание – 0.44, следовательно  $K=2.84$ .

Далее рассмотрим импульсную характеристику с регулятором (Рис. 16).

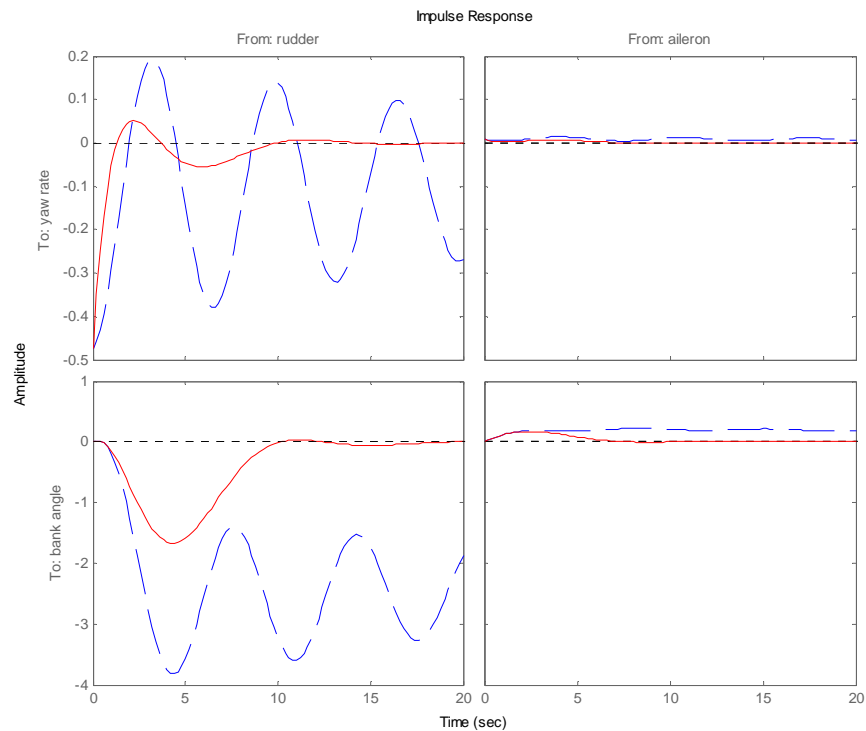
```
k = 2.84;
cl11 = feedback(sys11,-k);
impzplot(sys11,'b--',cl11,'r')
legend('open loop','closed loop',4)
```



**Рис. 16.** Импульсная характеристика системы с регулятором.

Образует обратную связь для всей модели и рассмотрим импульсную характеристику (Рис. 17).

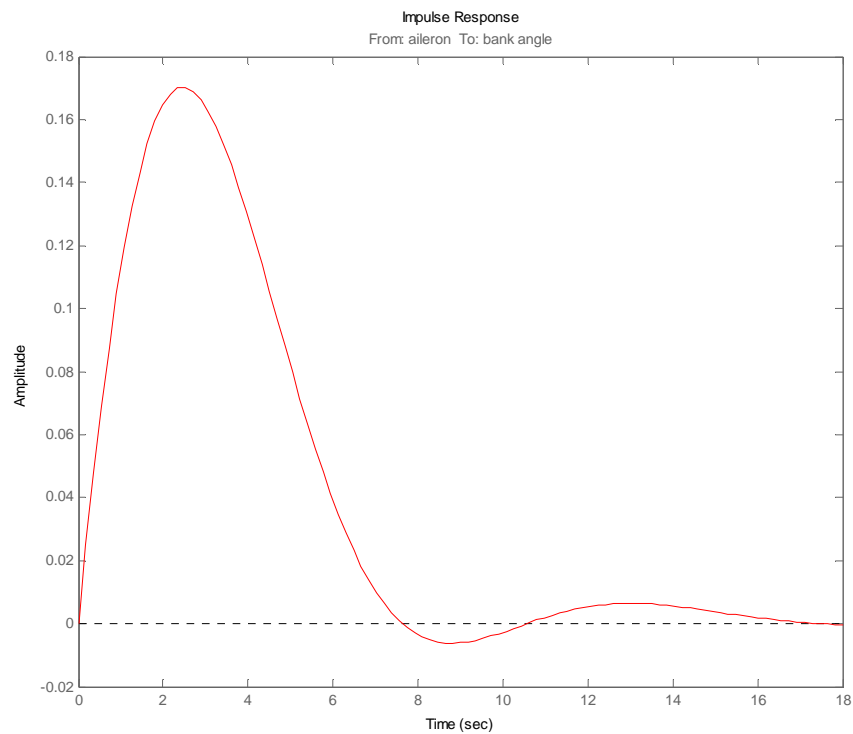
```
cloop = feedback(sys,-k,1,1);
impzplot(sys,'b--',cloop,'r',20)
```



**Рис. 17.** Характеристика системы с обратной связью.

Теперь видно, что скорость отклонения затухает хорошо. Однако, при перемещении элерона система не кренится как обычный самолет, что видно из Рис. 18.

```
impzplot(cloop('bank angle','aileron'),'r',18)
```



**Рис. 18.** Переходный процесс.



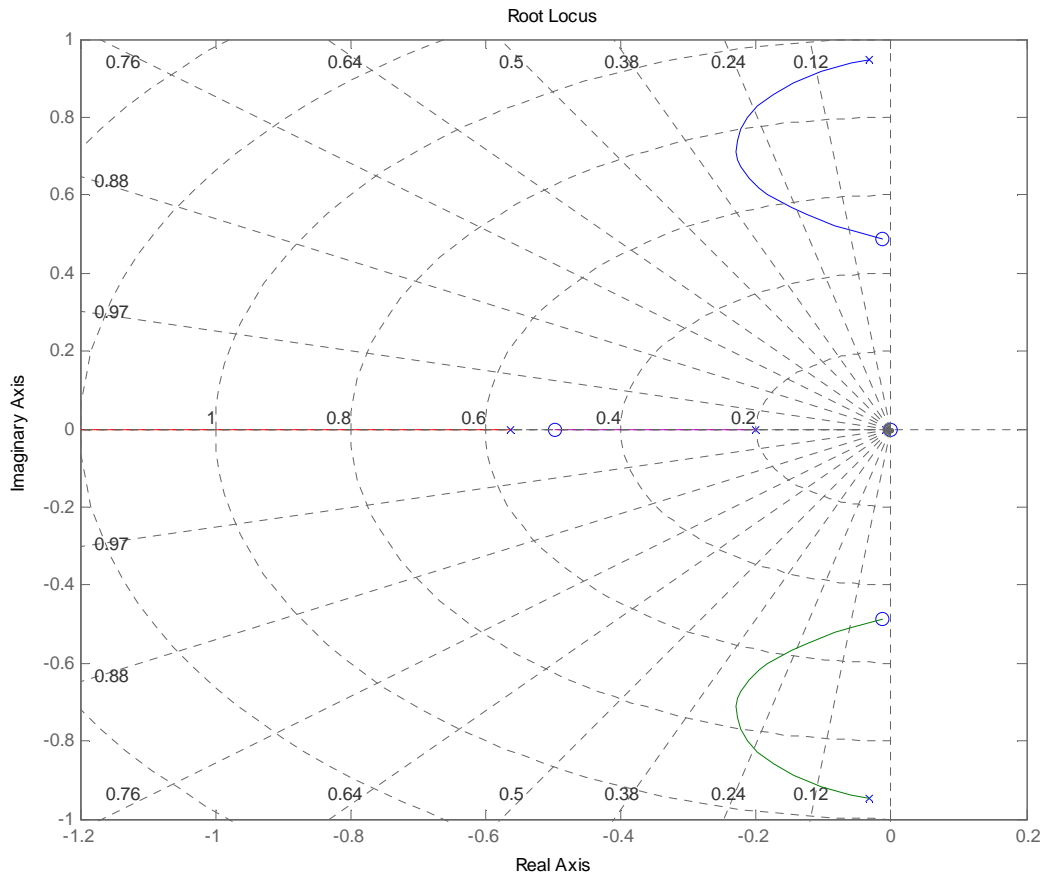
Необходимо убедиться, что данный режим не выходит за пределы левой полуплоскости при замыкании системы обратной связью. Для этого можно использовать фильтр.

$$H(s) = \frac{ks}{s+a}$$

`H = zpk(0,-0.2,1);`

Соединим фильтр последовательно с моделью и используем корневой годограф для определения коэффициента усиления (Рис. 19).

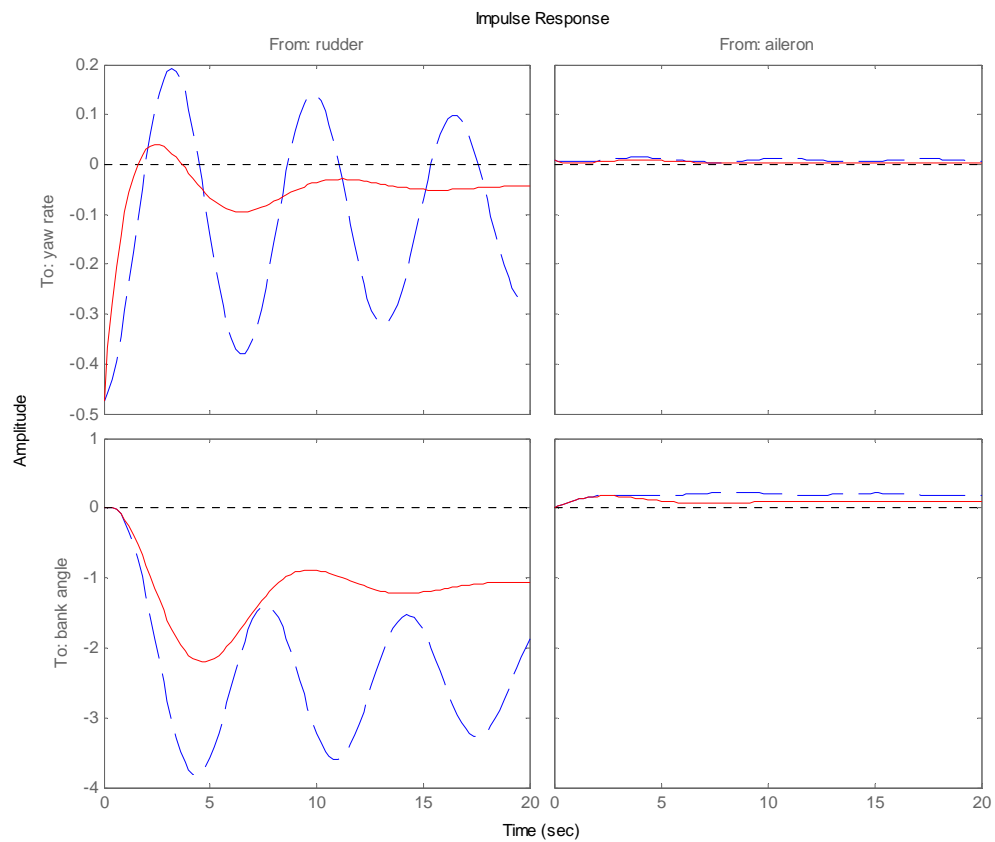
```
oloop = H * (-sys11);           % open loop'
h = rlocusplot(oloop);
```



**Рис. 19.** Корневой годограф.

Наилучший коэффициент затухания – 0,305 для  $K=2,34$ . Охватим систему обратной связью и проверим импульсную характеристику (Рис. 20).

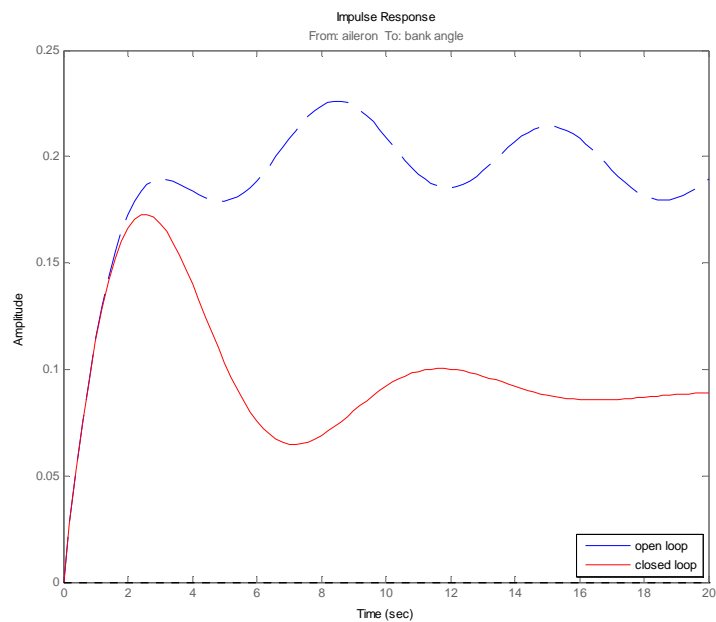
```
k = 2.34;
wof = -k * H;                   % washout compensator
cloop = feedback(sys,wof,1,1);
impzplot(sys,'b--',cloop,'r',20)
```



**Рис. 20.** Импульсная характеристика.

Рассмотрим отдельно характеристику зависимости элерона от угла крена (Рис. 21).

```
impzplot(sys(2,2), 'b--', cloop(2,2), 'r', 20)
legend('open loop', 'closed loop', 4)
```



**Рис. 21.** Зависимость элерона от угла крена.

Данная конструкция существенно увеличивает затухание, что хорошо сказывается на полете в обычном режиме.

## **КОНТРОЛЬНЫЕ ВОПРОСЫ**

1. Для чего предназначен пакет МатЛАБ – Control System Toolbox?
2. Сравните проектирование САУ средствами Control System Toolbox и Simulink.

## **БИБЛИОГРАФИЧЕСКИЙ СПИСОК**

1. А.А. Алексеев, Д.Х. Имаев, Н.Н. Кузьмин, В.Б. Яковлев. Теория управления: Учеб./ СПб.: Изд-во СПбГЭТУ “ЛЭТИ”, 1999.
2. Л.А.Озеров. Автоматизированное проектирование систем. Учебное пособие. Часть 1. Челябинск. Издательский центр ЮУрГУ, 2009.
3. Help MatLAB
4. Dr. Bob Davidov. Компьютерные технологии управления в технических системах <http://portalnp.ru/author/bobdavidov>