МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФГБОУ ВО «Севастопольский государственный университет»

кафедра Информационные системы

Куркчи Ариф Эрнестович

Институт информационных технологий и управления в технических системах

курс 4 группа ИС/б-41-о

09.03.02 Информационные системы и технологии (уровень бакалавриата)

ОТЧЕТ

по лабораторной работе № 3

по дисциплине «ОСНОВЫ ТЕОРИИ УПРАВЛЕНИЯ»

на тему «ИССЛЕДОВАНИЕ ВРЕМЕННЫХ И ЧАСТОТНЫХ ХАРАКТЕРИСТИК СИСТЕМ УПРАВЛЕНИЯ»

Отметка о зачете \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_

(дата)

Руководитель практикума

  Карлусов В.Ю.

(должность) (подпись) (инициалы, фамилия)

Севастополь 2017

1 ЦЕЛЬ РАБОТЫ

Изучение методов анализа линейных САУ с помощью среды Matlab. Исследование устойчивости и качества САУ.

2 ПОСТАНОВКА ЗАДАЧИ

1. Ввести модель системы в виде передаточной функции.
2. Построить эквивалентные модели в пространстве состояний и в форме «нули-полюса».
3. Определить коэффициент усиления в установившемся режиме и полосу пропускания системы.
4. Научиться строить импульсную и переходную характеристики, карту расположения нулей и полюсов, частотную характеристику.
5. Научиться использовать окно LTIViewer для построения различных характеристик.
6. Научиться строить процессы на выходе линейной системы при произвольном входном сигнале.
7. Оформить отчет по работе.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Вариант | n2 | n1 | n0 | d2 | d1 | d0 |
| 7 | 1.6 | 0.80 | -0.224 | 1.3286 | 0.8959 | 0.4592 |

3 ХОД РАБОТЫ

Код программы:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | clear all | | clc | | help tf | | which('tf') | | n2 = 1.6; n1 = 0.80; n0 = -0.224; | | d2 = 1.3286; d1 = 0.8959; d0 = 0.4592; | | n = [n2 n1 n0] | | d = [1 d2 d1 d0] | | f = tf ( n, d ) | | [n1,d1] = tfdata ( f, 'v' ) | | z = zero ( f ) | | p = pole ( f ) | | k = dcgain ( f ) | | b = bandwidth ( f ) | | f\_ss = ss ( f ) | | f\_ss.d = 1 | | k1 = dcgain ( f\_ss ) | | f\_zp = zpk ( f ) | | who | | whos | | pzmap ( f ) | | [wc,ksi,p] = damp ( f ) | | pzmap ( f ) | | ltiview | | w = logspace(-1,2,100); | | r = freqresp(f,w); | | r = r(:); | | figure | | semilogx(w, abs(r)); | | print -dmeta | | [u,t] = gensig('square', 4); | | figure | | lsim(f,u,t); | |  | |  | | TF  Creation of transfer functions or conversion to transfer function. | |  | | Creation: | | SYS = TF(NUM,DEN) creates a continuous-time transfer function SYS with | | numerator(s) NUM and denominator(s) DEN.  The output SYS is a TF object. | |  | | SYS = TF(NUM,DEN,TS) creates a discrete-time transfer function with | | sample time TS (set TS=-1 if the sample time is undetermined). | |  | | S = TF('s') specifies the transfer function H(s) = s (Laplace variable). | | Z = TF('z',TS) specifies H(z) = z with sample time TS. | | You can then specify transfer functions directly as rational expressions | | in S or Z, e.g., | | s = tf('s');  H = (s+1)/(s^2+3\*s+1) | |  | | SYS = TF creates an empty TF object. | | SYS = TF(M) specifies a static gain matrix M. | |  | | In all syntax above, the input list can be followed by pairs | | 'PropertyName1', PropertyValue1, ... | | that set the various properties of TF models (type LTIPROPS for details). | | To make SYS inherit all its LTI properties from an existing LTI model | | REFSYS, use the syntax SYS = TF(NUM,DEN,REFSYS). | |  | | Data format: | | For SISO models, NUM and DEN are row vectors listing the numerator and | | denominator coefficients in | | \* descending powers of s or z by default | | \* ascending powers of q = z^-1 if the 'Variable' property is set to | | 'z^-1' or 'q' (DSP convention). | |  | | For MIMO models with NY outputs and NU inputs, NUM and DEN are NY-by-NU | | cell arrays of row vectors where NUM{i,j} and DEN{i,j} specify the | | transfer function from input j to output i.  For example, | | H = TF( {-5 ; [1 -5 6]} , {[1 -1] ; [1 1 0]}) | | specifies the two-output, one-input transfer function | | [     -5 /(s-1)      ] | | [ (s^2-5s+6)/(s^2+s) ] | |  | | By default, transfer functions are displayed as functions of 's' or 'z'. | | Alternatively, you can set the variable name to 'p' (continuous time) | | and 'z^-1' or 'q' (discrete time) by modifying the 'Variable' property. | |  | | Arrays of transfer functions: | | You can create arrays of transfer functions by using ND cell arrays for | | NUM and DEN above.  For example, if NUM and DEN are cell arrays of size | | [NY NU 3 4], then | | SYS = TF(NUM,DEN) | | creates the 3-by-4 array of transfer functions | | SYS(:,:,k,m) = TF(NUM(:,:,k,m),DEN(:,:,k,m)),  k=1:3,  m=1:4. | | Each of these transfer functions has NY outputs and NU inputs. | |  | | To pre-allocate an array of zero transfer functions with NY outputs | | and NU inputs, use the syntax | | SYS = TF(ZEROS([NY NU k1 k2...])) . | |  | | Conversion: | | SYS = TF(SYS) converts an arbitrary LTI model SYS to the transfer | | function representation. The result is a TF object. | |  | | SYS = TF(SYS,'inv') uses a fast algorithm for conversion from state | | space to TF, but is typically less accurate for high-order systems. | |  | | See also LTIMODELS, FILT, SET, GET, TFDATA, SUBSREF, LTIPROPS, ZPK, SS. | |  | | C:\MATLAB6p5\toolbox\control\control\@tf\tf.m  % tf метод | |  | | n = | |  | | 1.6000    0.8000   -0.2240 | |  | |  | | d = | |  | | 1.0000    1.3286    0.8959    0.4592 | |  | |  | | Transfer function: | | 1.6 s^2 + 0.8 s - 0.224 | | ----------------------------------- | | s^3 + 1.329 s^2 + 0.8959 s + 0.4592 | |  | |  | | n1 = | |  | | 0    1.6000    0.8000   -0.2240 | |  | |  | | d1 = | |  | | 1.0000    1.3286    0.8959    0.4592 | |  | |  | | z = | |  | | -0.7000 | | 0.2000 | |  | |  | | p = | |  | | -0.9001 | | -0.2143 + 0.6814i | | -0.2143 - 0.6814i | |  | |  | | k = | |  | | -0.4878 | |  | |  | | b = | |  | | 4.6934 | |  | |  | | a = | | x1       x2       x3 | | x1   -1.329   -0.448  -0.4592 | | x2        2        0        0 | | x3        0      0.5        0 | |  | |  | | b = | | u1 | | x1   1 | | x2   0 | | x3   0 | |  | |  | | c = | | x1      x2      x3 | | y1     1.6     0.4  -0.224 | |  | |  | | d = | | u1 | | y1   0 | |  | | Continuous-time model. | |  | | a = | | x1       x2       x3 | | x1   -1.329   -0.448  -0.4592 | | x2        2        0        0 | | x3        0      0.5        0 | |  | |  | | b = | | u1 | | x1   1 | | x2   0 | | x3   0 | |  | |  | | c = | | x1      x2      x3 | | y1     1.6     0.4  -0.224 | |  | |  | | d = | | u1 | | y1   1 | |  | | Continuous-time model. | |  | | k1 = | |  | | 0.5122 | |  | |  | | Ноль/Полюс/Увеличение: | | 1.6 (s+0.7) (s-0.2) | | ------------------------------------ | | (s+0.9001) (s^2  + 0.4285s + 0.5102) | |  | |  | | Your variables are: | |  | | b     d     d0    d1    d2    f     f\_ss  f\_zp  k     k1    n     n0    n1    n2    p     z | |  | | Name       Size                   Bytes  Class | |  | | b          1x1                        8  double array | | d          1x4                       32  double array | | d0         1x1                        8  double array | | d1         1x4                       32  double array | | d2         1x1                        8  double array | | f          1x1                     2206  tf object | | f\_ss       1x1                     2940  ss object | | f\_zp       1x1                     2472  zpk object | | k          1x1                        8  double array | | k1         1x1                        8  double array | | n          1x3                       24  double array | | n0         1x1                        8  double array | | n1         1x4                       32  double array | | n2         1x1                        8  double array | | p          3x1                       48  double array (complex) | | z          2x1                       16  double array | |  | | Grand total is 146 elements using 7858 bytes | |  | |  | | wc = | |  | | 0.7143 | | 0.7143 | | 0.9001 | |  | |  | | ksi = | |  | | 0.3000 | | 0.3000 | | 1.0000 | |  | |  | | p = | |  | | -0.2143 + 0.6814i | | -0.2143 - 0.6814i | | -0.9001 | |  | |  |

13) Коэффициент k1 больше k на 1, это произошло в следствии того что коэффициент прямой передачи был установлен в 1 (увеличился на 1).

15) who - выводит только названия переменных, whos - более подробную информацию (добавляется тип и вес переменных).

23) Импульсные характеристики систем f и f\_ss получились, одинаковые, потому что импульсная характеристика не зависит от коэффициента усиления.

36) Коэффициент усиления можно определить по переходной графику переходной характеристики как значение, к которому стремится график, или же по АЧХ как значение в нулевой частоте. Полосу пропускания системы равна частоте, после которой значение АЧХ падает ниже -3 дБ (коэффициент усиления меньше, чем 0.708).

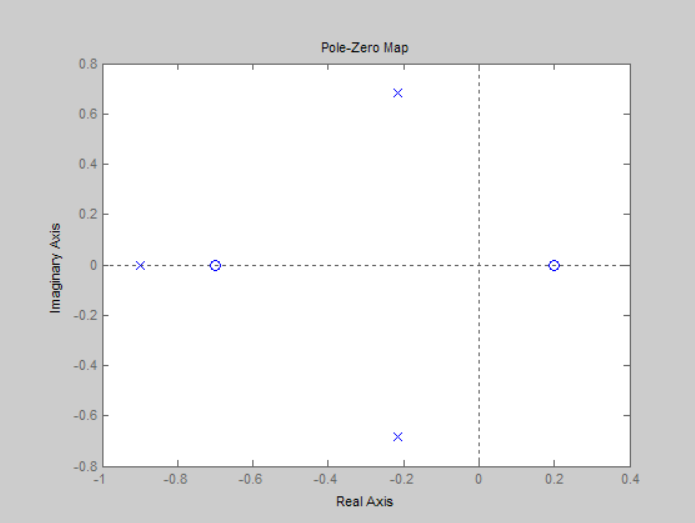


Рисунок 1 - Расположение нулей и полюсов системы

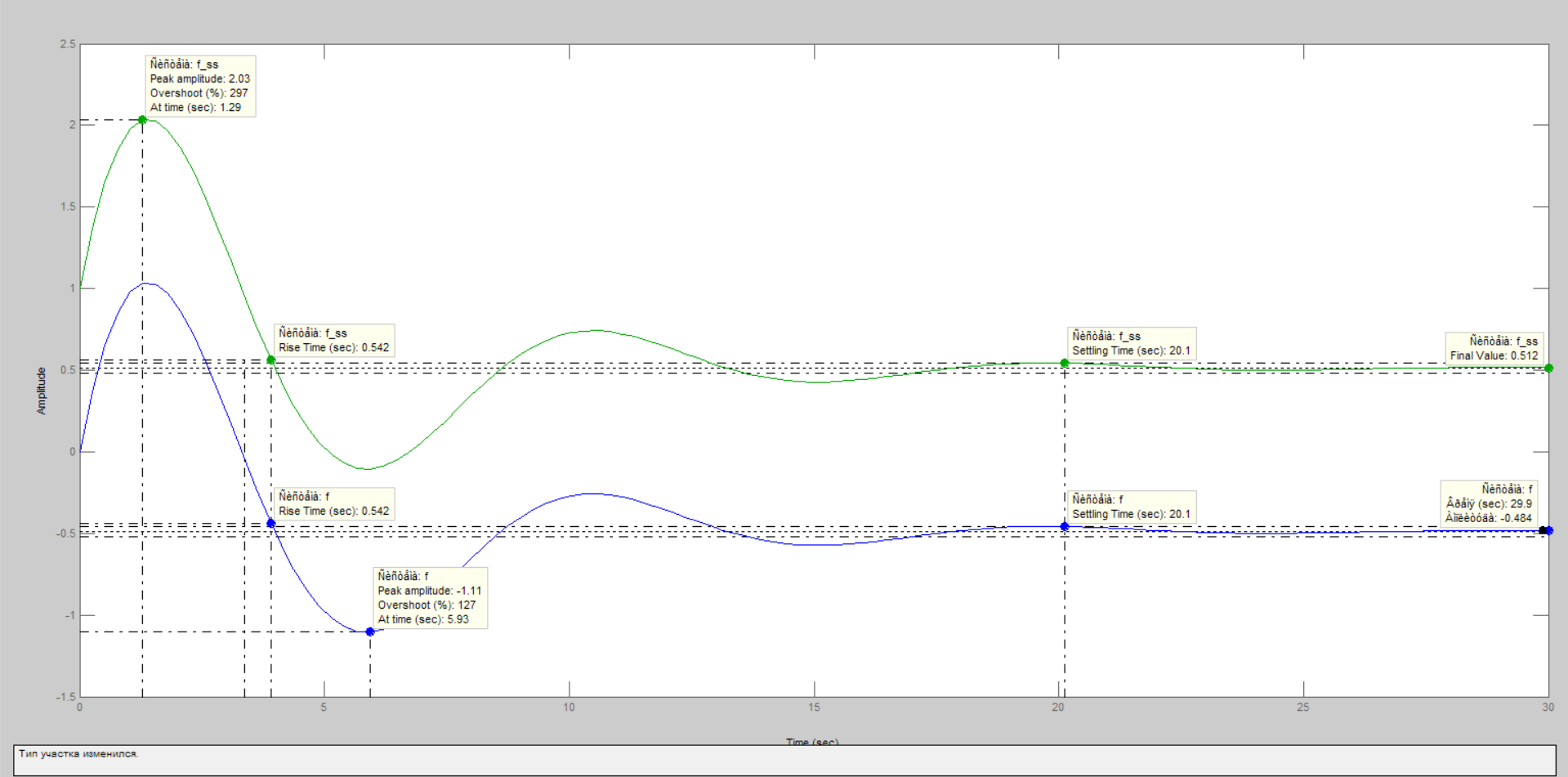


Рисунок 2 - Переходные характеристики систем

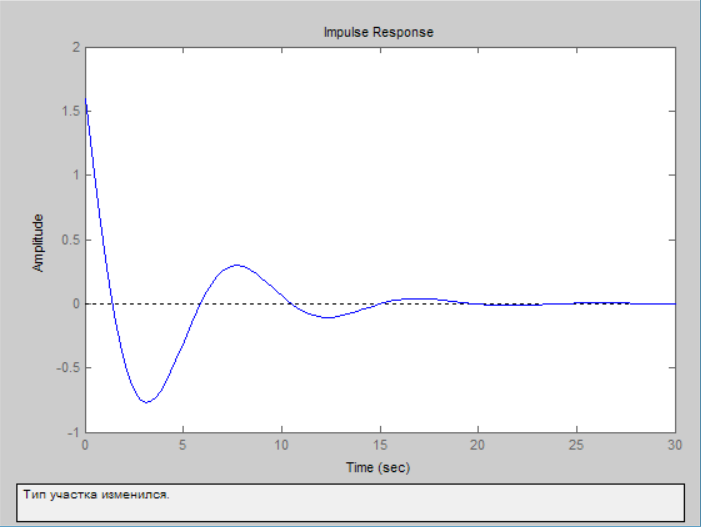


Рисунок 3 - Импульсные характеристики систем (совпадают)

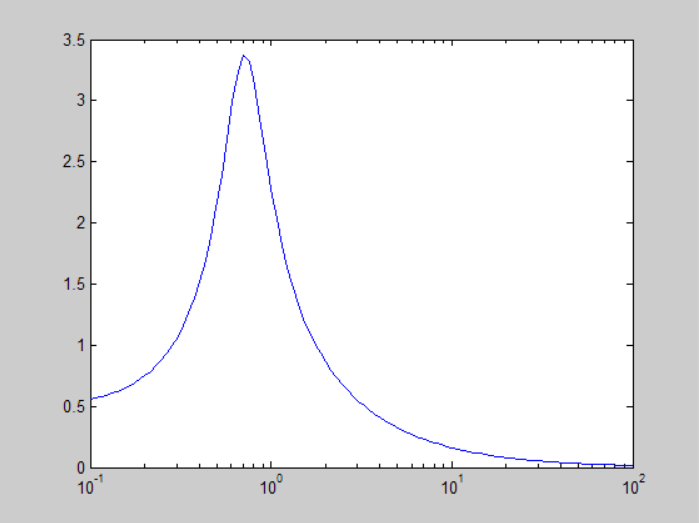


Рисунок 4 - Частотная характеристика (логарифмический масштаб по оси абсцисс)

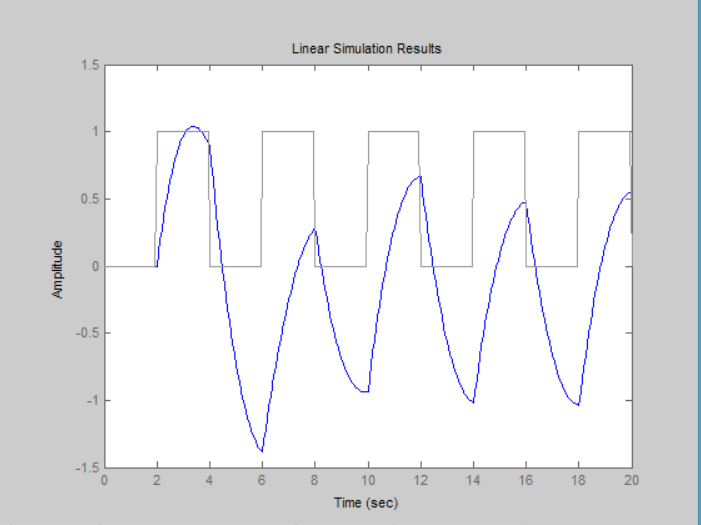


Рисунок 5 - Сигнал имитирующий прямоугольные импульсы

ВЫВОДЫ

В результате выполнения лабораторной работы были изучены методы анализа линейных САУ с помощью среды Matlab. Исследование устойчивости и качества САУ.