

ГУАП

КАФЕДРА № 31

ОТЧЕТ  
ЗАЩИЩЕН С ОЦЕНКОЙ  
ПРЕПОДАВАТЕЛЬ

старший преподаватель  
\_\_\_\_\_  
должность, уч. степень, звание

\_\_\_\_\_  
подпись, дата

Н.Л. Гречкин  
\_\_\_\_\_  
инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №3

ИДЕНТИФИКАЦИЯ ОБЪЕКТОВ СИСТЕМ УПРАВЛЕНИЯ

по курсу: НЕЛИНЕЙНЫЕ И АДАПТИВНЫЕ СИСТЕМЫ УПРАВЛЕНИЯ

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. №

\_\_\_\_\_  
подпись, дата

Д. В. Самарин  
\_\_\_\_\_  
инициалы, фамилия

Санкт-Петербург 2025

## ЦЕЛЬ РАБОТЫ

Освоение правил автоматической идентификации объектов в среде matlab. Построение математической модели динамической системы по измеренным данным входа и выхода реальной системы.

## ХОД РАБОТЫ

1) Построим в Simulink виртуальную систему протекания жидкости (рис.1) через два резервуара (Tank 1 и Tank 2). Входом системы является сечение крана (Valve\_pos) – координата  $u(t)$ , а выходом – уровень нижнего резервуара,  $y(t)$ . Выходной поток из резервуара пропорционален корню квадратному от уровня жидкости в резервуаре.

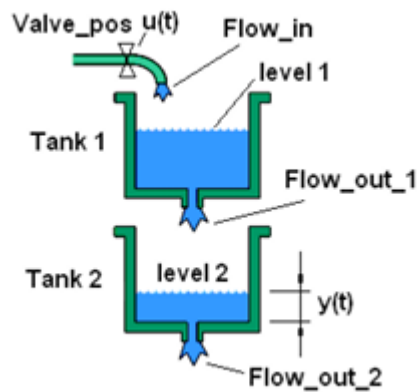


Рисунок 1 – Прототип системы для идентификации

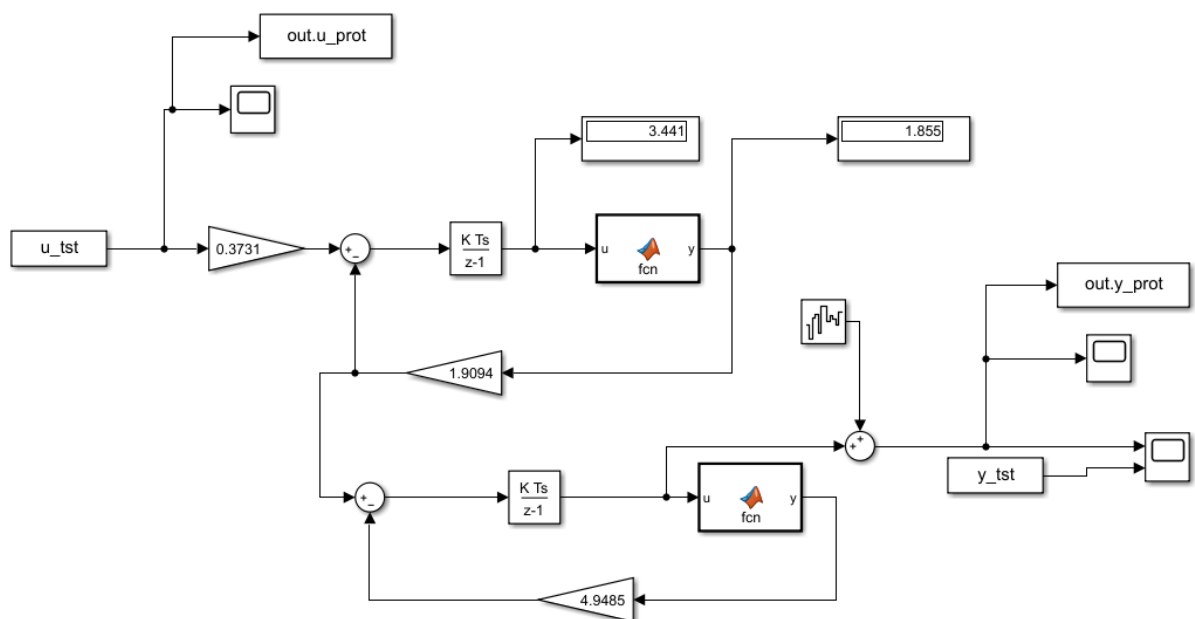


Рисунок 2 – Виртуальная модель прототипа (рис.1)

```
function y = fcn(u)
x = u(1);
y = sqrt(x);
y = u;
```

Ниже будет приведен код для извлечения данных из структурной схемы.

Для загрузки в рабочую зону данных выхода и входа используем команду: *load twotankdata*,

На рисунке 3 представлен выход модели и её отклик.

```
time = ((0:length(u)-1)*0.2)';
t = time;
yout = y;
u_tst = [time, u];
y_tst = [time, y];
```

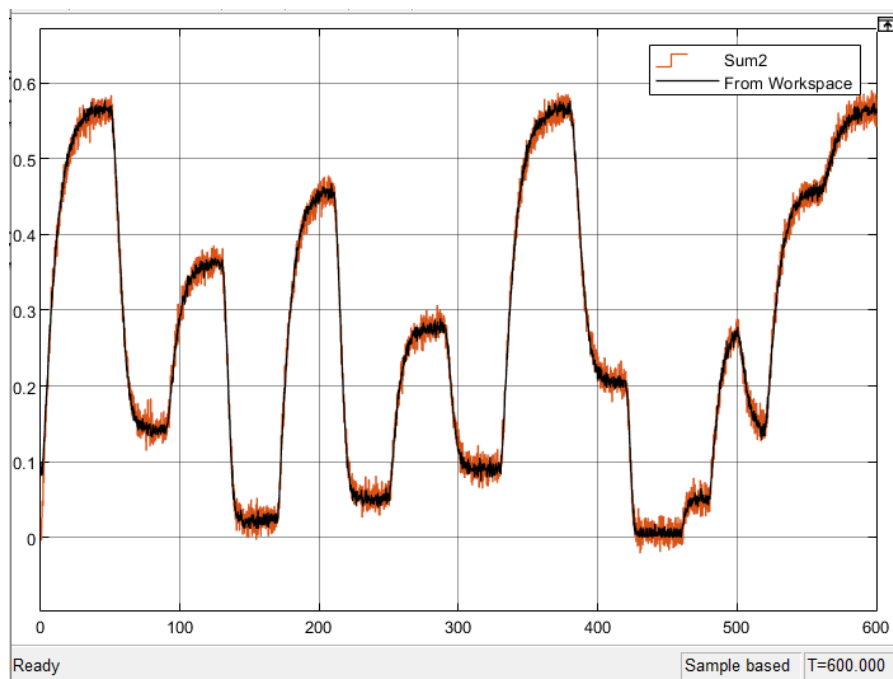


Рисунок 3 – Графики модели и её отклика

2) Предварительная обработка данных и идентификация структуры модели и ее параметров.

Используем воздействие и отклик виртуального прототипа полученные в предыдущем задании как экспериментальные данные реальной системы для ее идентификации.

Зайдем в System Identification Toolbox используя команду *ident*. Открывшееся окно представлено на рисунке 4.

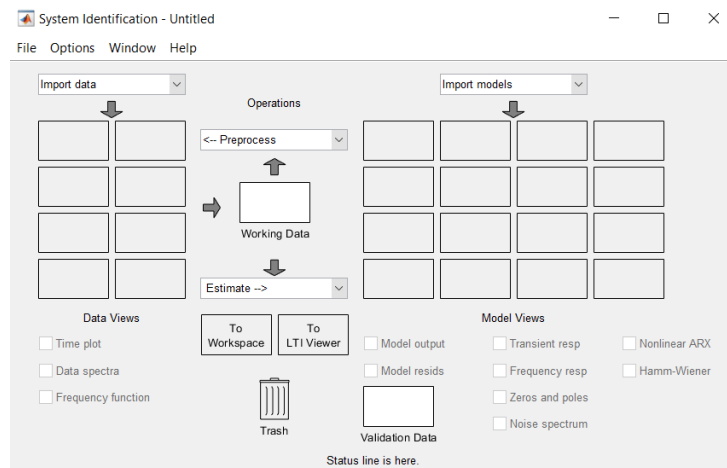


Рисунок 4 – Окно System Identification Toolbox

Добавляем наши сигналы  $u$  и  $y$  (рис. 5). Указываем начальное время 0 интервал 0.2. Выведем временной, частотный и спектральный графики (рис. 6-8).

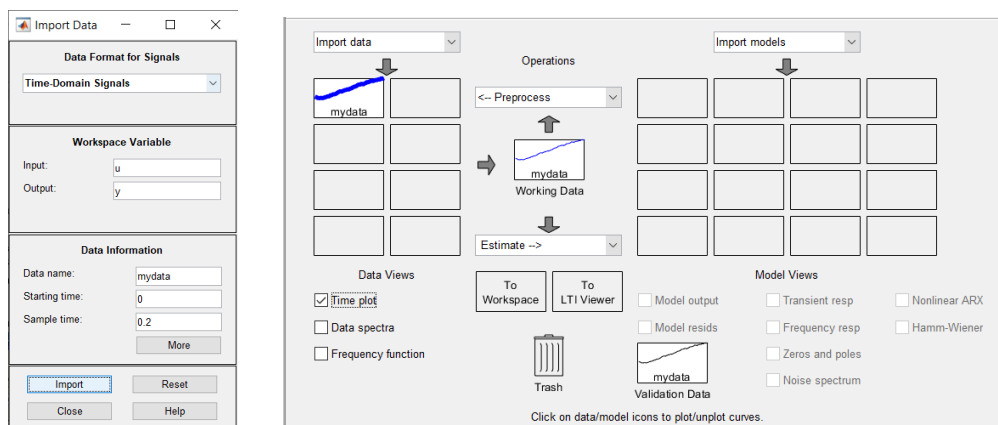


Рисунок 5 – Окна ввода экспериментальных данных ( $u$  и  $y$ )

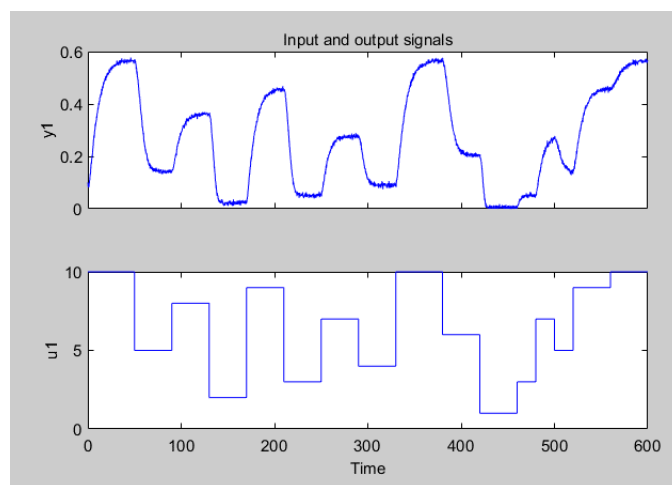


Рисунок 6 – Временной график

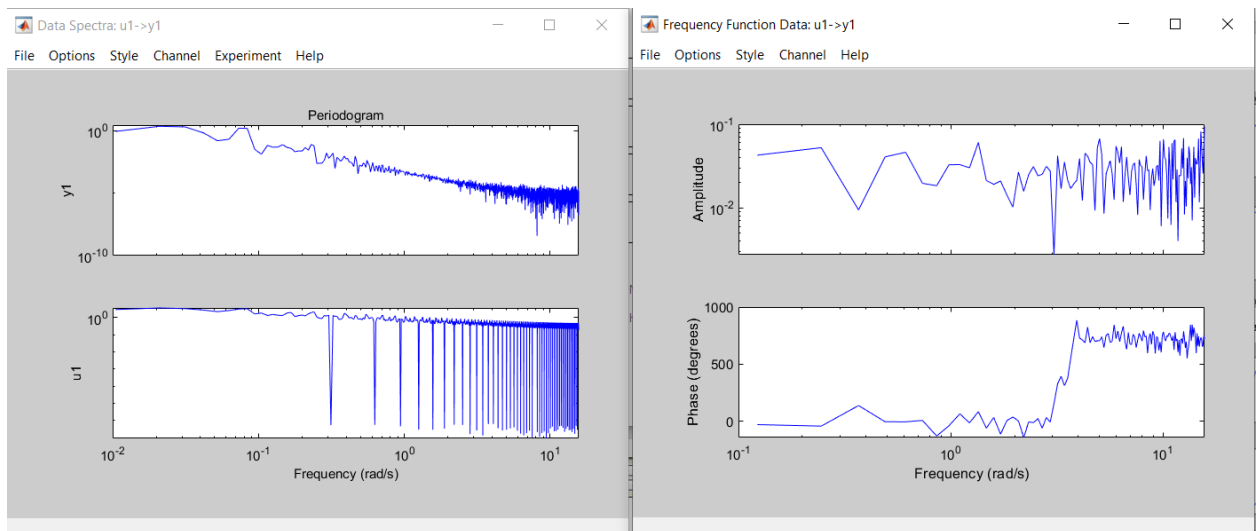


Рисунок 7 – Спектральный и частотный графики

Выбираем метод *Polynomial Models*. Используем структуру модели идентификации, например, ARMAX (рис. 8) и введём ее параметры [na nb nc nk].

Рисунок 8 – Окно ввода параметров модели ARMAX

Нажатием на клавишу *Estimate* запустим процесс идентификации параметров выбранной структуры модели по экспериментальным значениям реакции и воздействия прототипа системы. Результаты идентификации в виде параметров модели приведены на рисунке 9-10. Вид результата в рабочей области приведен на рисунке 11.

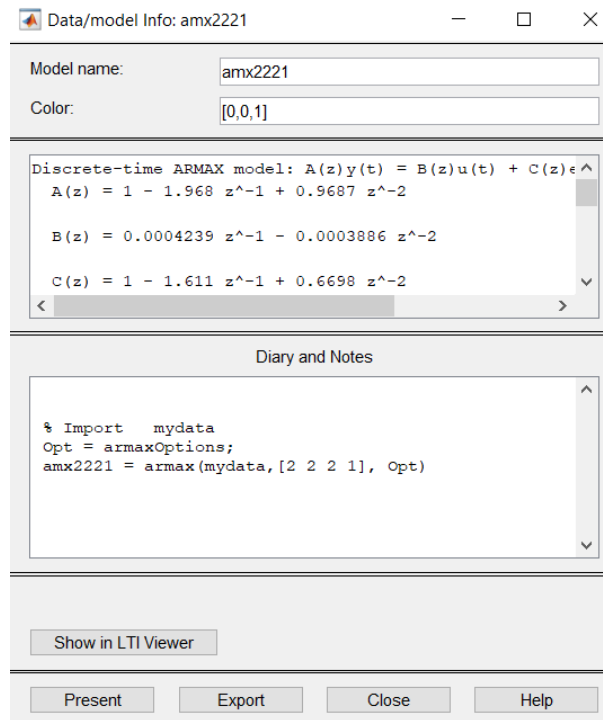


Рисунок 9 – Окно получения информации

```
Discrete-time ARMAX model: A(z)y(t) = B(z)u(t) + C(z)e(t)
A(z) = 1 - 1.968 z^-1 + 0.9687 z^-2

B(z) = 0.0004239 z^-1 - 0.0003886 z^-2

C(z) = 1 - 1.611 z^-1 + 0.6698 z^-2

Name: amx2221
Sample time: 0.2 seconds

Parameterization:
Polynomial orders:  na=2  nb=2
nc=2  nk=1
Number of free coefficients: 6
Use "polydata", "getpvec", "getcov" for parameters and their uncertainties.

Status:
Estimated using PEM on time domain data "mydata".
Fit to estimation data: 96.65% (prediction focus)
FPE: 3.858e-05, MSE: 3.843e-05
```

Рисунок 10 – Вывод в окне команд

Перенесем полученные данные в виде структуры в рабочую область (кнопкой *To Workspace*).

3	1x1 <a href="#">idpoly</a>	
4		
5	Property ▲	Value
6	A	[1,-1.9679,0.9687]
7	B	[0,4.2392e-04,-3.8859e-04]
8	C	[1,-1.6110,0.6698]
9	D	1
0	F	1
1	<input checked="" type="checkbox"/> IntegrateNoise	0
2	Variable	'z^-1'
3	IODelay	0
4	Structure	<i>1x1 polynomial</i>
5	NoiseVariance	3.8506e-05
6	Report	<i>1x1 polyst</i>
7	InputDelay	0
8	OutputDelay	0
9	Ts	0.2000
0	TimeUnit	'seconds'
1	InputName	<i>1x1 cell</i>
2	InputUnit	<i>1x1 cell</i>
3	InputGroup	<i>1x1 struct</i>
4	OutputName	<i>1x1 cell</i>
5	OutputUnit	<i>1x1 cell</i>
6	OutputGroup	<i>1x1 struct</i>
7	Notes	<i>5x1 cell</i>
8	UserData	[ ]
9	Name	'amx2221'
0	SamplingGrid	<i>1x1 struct</i>

Рисунок 11 – Раскрытие amx2221 структуры рабочей области

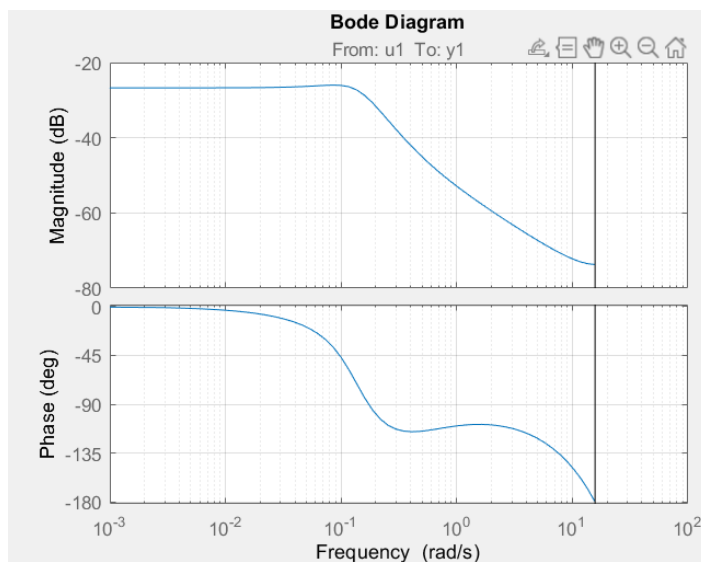


Рисунок 12 – Диаграмма bode модели amx2221

Полученные результаты идентификации в сравнении с исходными данными приведены на рисунке 13.

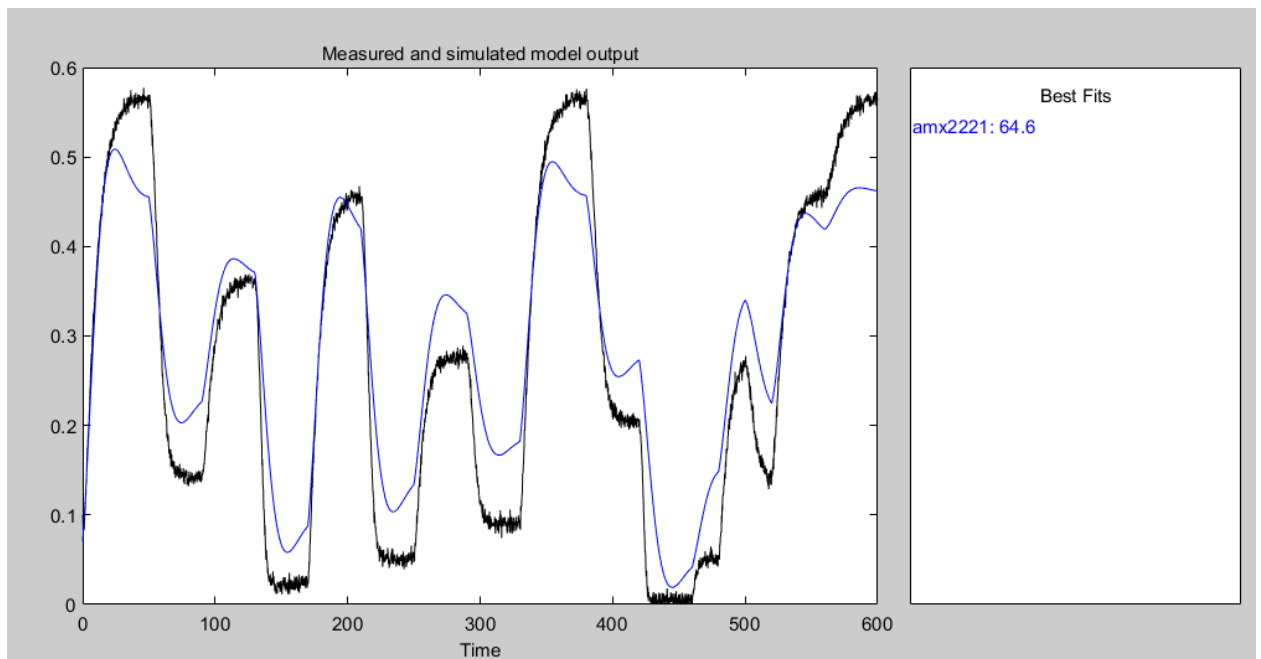


Рисунок 13 – График модели amx2221

3) Для проверки адекватности модели прототипу необходимо выполнить процедуру верификации. Она заключается в подаче на вход объекта и модели данных, которые не использовались для идентификации, и сравнении откликов. Построим сравнительные характеристики модели: реакция, а единичное ступенчатое воздействие; частотную характеристику; распределение полюсов и нулей и др. (рис. 14).

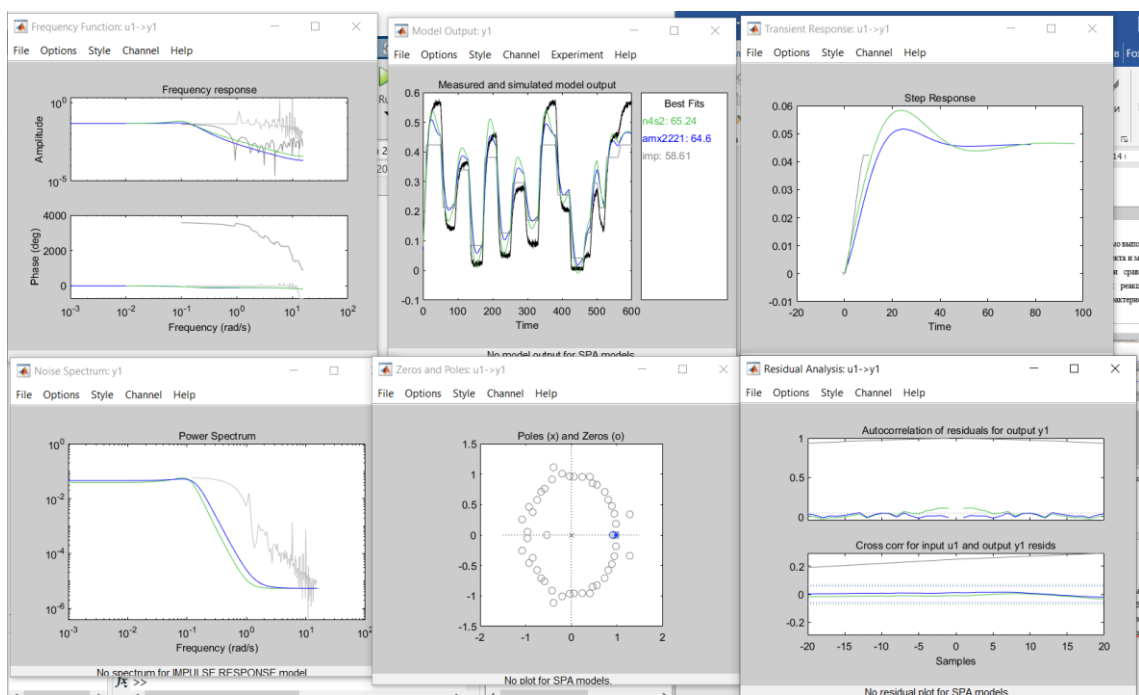


Рисунок 14 – Сравнительных характеристики, полученные с помощью Quick Start



4) Построим модель прототипа и сравним переходные процессы модели и прототипа. Пусть полученная ARMAX модель наилучшим образом отображает поведение прототипа – реальной системы. Эту модель можно интегрировать в Simulink через блоки Simulink Library Browser> Libraries> System Identification Toolbox.

Структурная схема прототипа приведена на рисунке 15. Сравнение прототипа и модели приведено на рисунке 16.

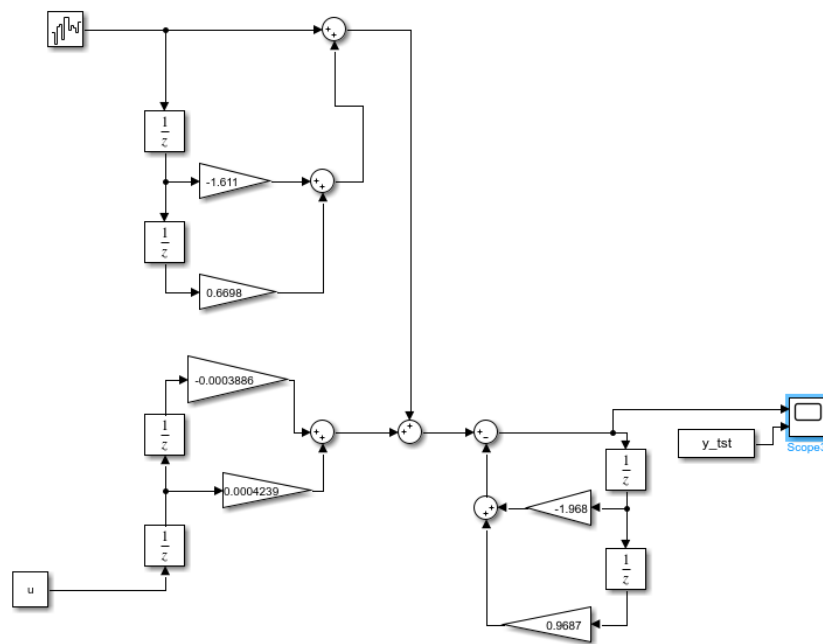


Рисунок 15 – Структурная схема прототипа

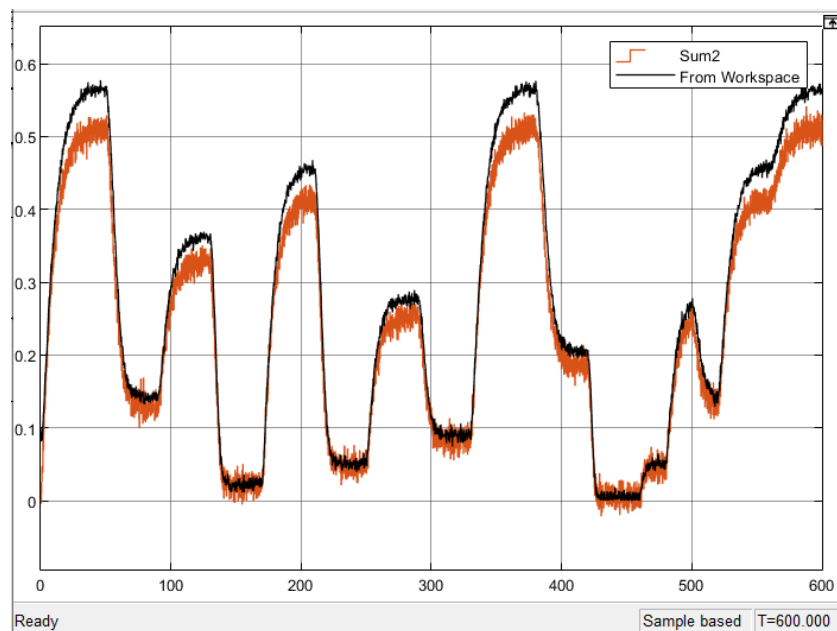


Рисунок 16 – Сравнение прототипа и модели

5) Для проверки эффективности идентификации – нахождения передаточной функции по известной реакции и входному воздействию будем использовать заранее вычисленные реакцию и воздействие известной структуры. Для этого построим следующую модель в виде передаточной функции второго порядка в среде Simulink (рис. 17).

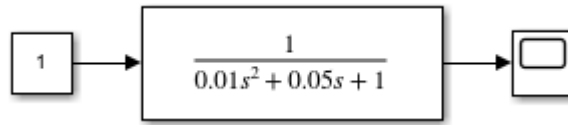


Рисунок 17 – Передаточная функция

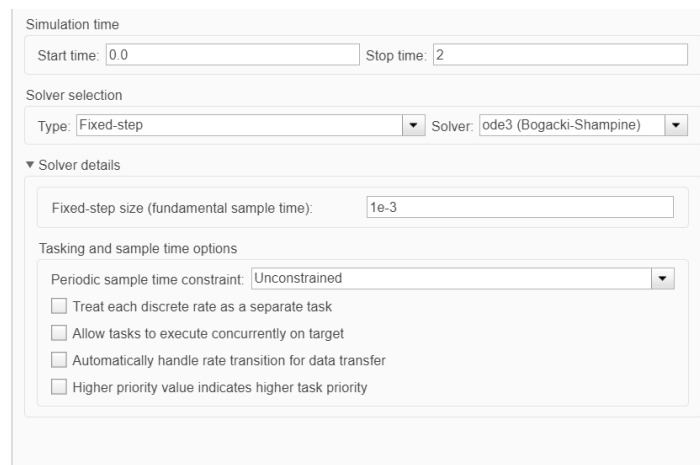


Рисунок 18 – Параметры решения

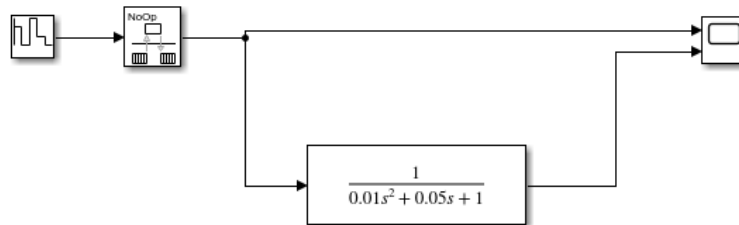


Рисунок 19 – Структурная схема воздействия и реакции

Параметры блоков приведены ниже на рисунке 20.

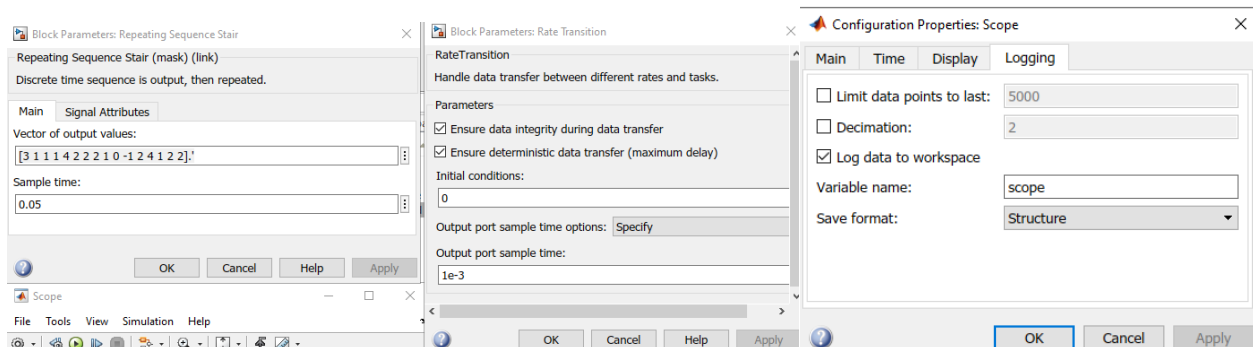


Рисунок 20 – Параметры блоков

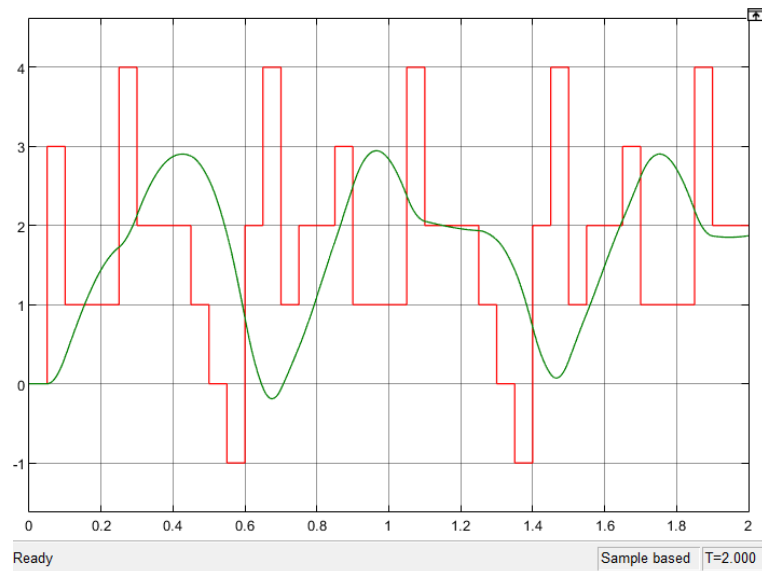


Рисунок 21 – Выход системы

Произведем идентификацию используя уже знакомый инструмент System Identification (рис. 22-26).

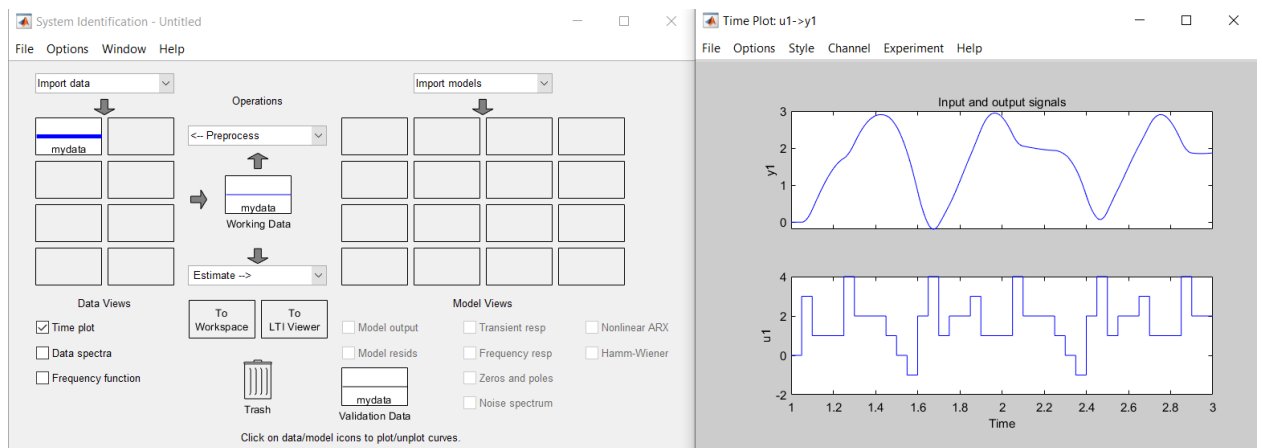


Рисунок 22 – Ввод исходных данных в System Identification

Выбираем метод идентификации Transfer Functions.

Рисунок 23 – Окно параметров

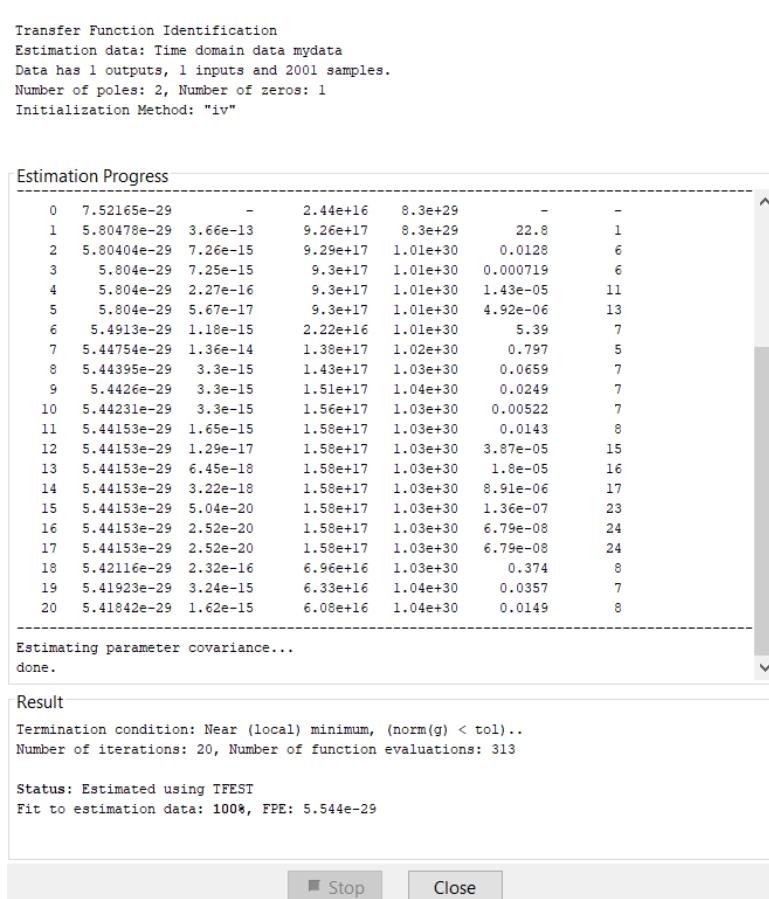


Рисунок 24 – Выполнение идентификации

Переместим параметры найденной передаточной функции в Workspace.

1x1 idtf	
Property ^	Value
Numerator	[3.1541e-07,100.0000]
Denominator	[1,5.0000,100.0000]
Variable	's'
IODelay	0
Structure	1x1 tf
NoiseVariance	5.5274e-32
Report	1x1 tfest
InputDelay	0
OutputDelay	0
Ts	0
TimeUnit	'seconds'
InputName	1x1 cell
InputUnit	1x1 cell
InputGroup	1x1 struct
OutputName	1x1 cell
OutputUnit	1x1 cell
OutputGroup	1x1 struct
Notes	10x1 cell
UserData	[]
Name	'tf1'
SamplingGrid	1x1 struct

Рисунок 25 – Раскрытие идентификации ПФ в workspace

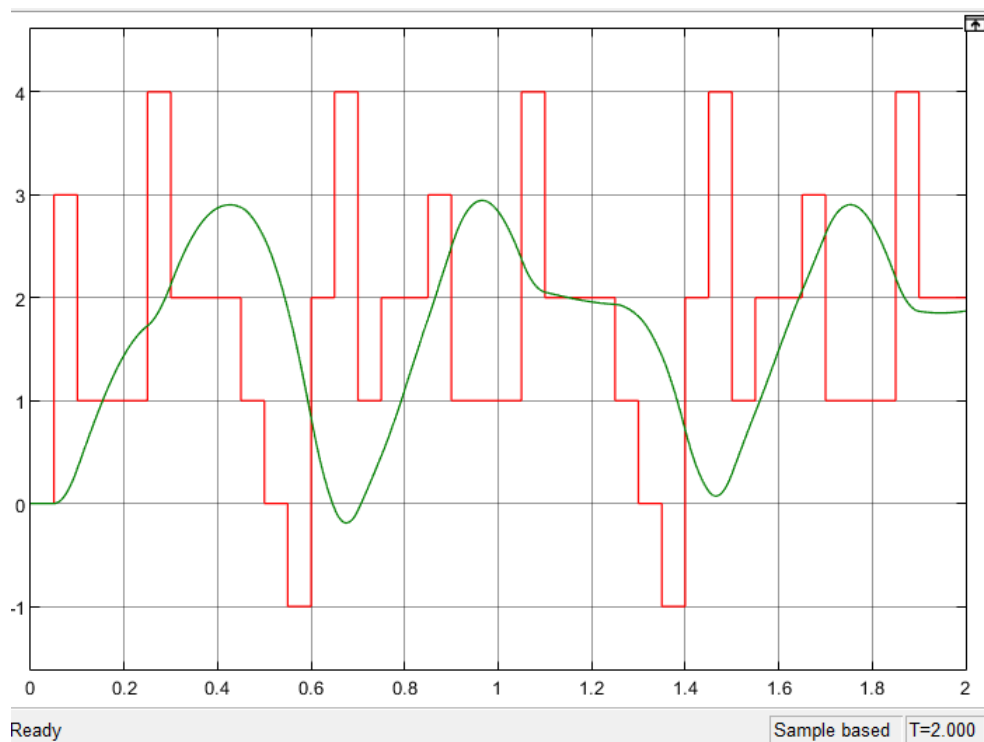


Рисунок 26 – График сравнения результатов

## ВЫВОД

В результате трехэтапного процесса был успешно освоен метод автоматической идентификации. Ключевым шагом стало создание в Simulink нелинейной модели двух резервуаров, которая выступила источником данных для последующего анализа. Опираясь на физические принципы работы системы, в качестве результирующей модели была обоснованно выбрана передаточная функция второго порядка. Её параметры были точно оценены по сгенерированным данным, что подтвердило адекватность модели экспериментальному отклику. Таким образом, методика позволила получить точное математическое описание системы, исключив необходимость в работе с физическим прототипом.