

System Identification Toolkit.

ТЕХНОЛОГИИ ИДЕНТИФИКАЦИИ

Идентификация объектов систем управления. System Identification Toolbox.

Цель работы: освоение правил автоматической идентификации объектов в среде МатЛАБ.

Задача работы: построение математической модели динамической системы по измеренным данным входа и выхода реальной системы.

Приборы и принадлежности: Персональный компьютер с интегрированной средой МатЛАБ и пакетом Simulink..

ОБЩИЕ СВЕДЕНИЯ

Идентификация систем как совокупность методов для построения математических моделей динамических систем по данным наблюдений находит широкое применение на практике. Эта область теории управления хорошо изучена и формализована.

Пакет МатЛАБ **System Identification Toolbox** ориентирован на решение задач идентификации. При помощи пакета по временным или частотным входным и выходным экспериментальным данным можно

- оперативно идентифицировать непрерывные и дискретные передаточные функции,
- описывать поведение динамических систем в пространстве состояний,
- определять подходящую структуру и порядок модели,
- выполнять оценку параметров модели,
- проверять достоверность модели (выполнять верификацию).

В пакете реализованы методы

- максимального правдоподобия,
- минимизации ошибки прогноза (РЕМ - prediction-error minimization),
- системной идентификации подмножества и др.

Средствами пакета можно оценивать нелинейные динамические модели

- Хаммерстайна-Винера (Hammerstein-Weiner),
- ARX (AutoRegresive model with eXternal input (АвтоРегрессионная модель с внешним входом))
- с вейвлетной сетью,
- с древовидным разделением
- с нелинейной сигмоидальной сетью.

РАБОТА С ПАКЕТОМ SYSTEM IDENTIFICATION TOOLBOX

Построение модели начинается с формирования входных воздействий которые подаются на объект и измерения реакций (откликов) на входные воздействия. Затем входные и выходные сигналы и выбранная из списка пакета структура используются для оценки значений параметров модели в соответствии с принятым критерием качества. Критерий качества идентификации характеризует степень адекватности модели объекту в рамках согласованных допущений и ограничений. Очень часто используется среднеквадратичный критерий – отклонение отклика модели от отклика прототипа при одном и том же входном воздействии. Определение достоверности модели на этапе верификации также определяется сравнением откликов для независимых воздействий которые не использовались при идентификации.

Ввод экспериментальных данных

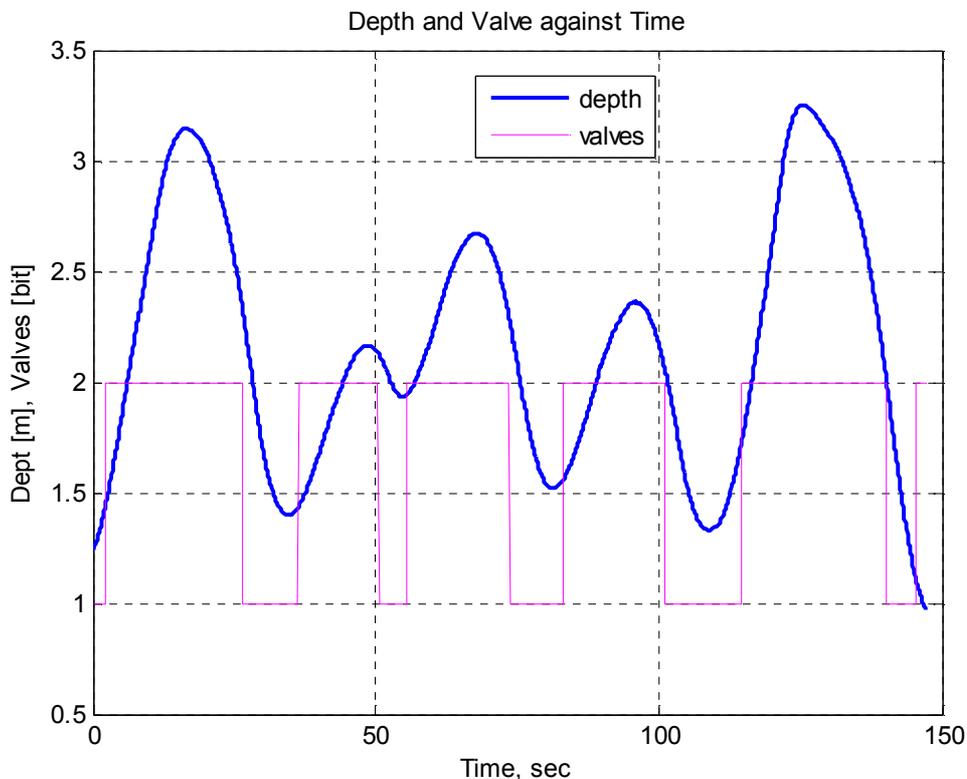


Рис. 1. Пример экспериментальных данных прототипа модели.

Загрузка пакета [System Identification Toolbox](#) выполняется через командную строку МатЛАБ командой
`>> ident`

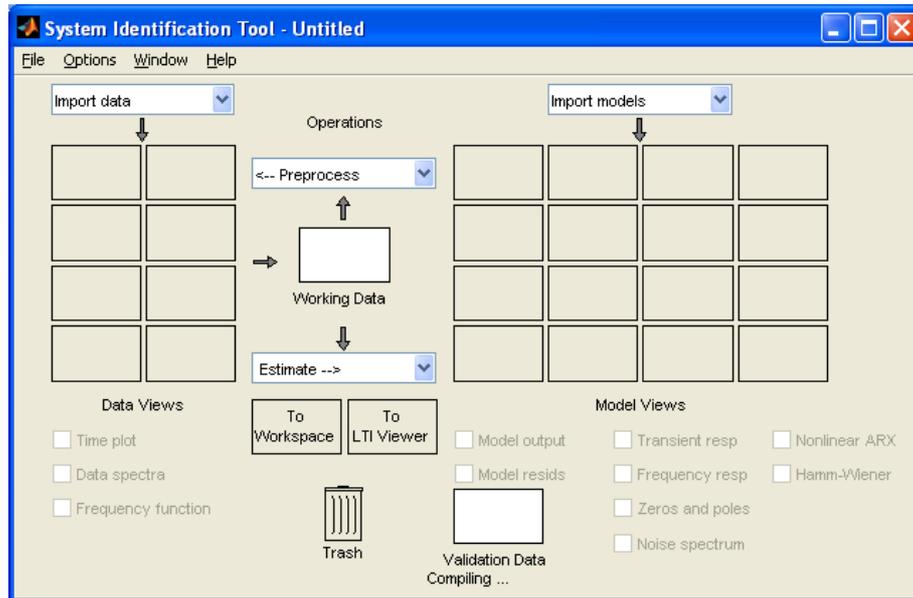


Рис. 2. Диалоговое окно пакета System Identification Toolbox.

Данные временной или частотной областей загружаются из рабочего пространства МатЛАБ. Данные поставляются либо массивами (амплитуд или частот) либо объектами класса IDDATA (Time Domain) или (Freq Domain).

Для загрузки входных и выходных данных необходимо выбрать тип данных (Time domain, Freq domain, Object), ввести имена и параметры данных.

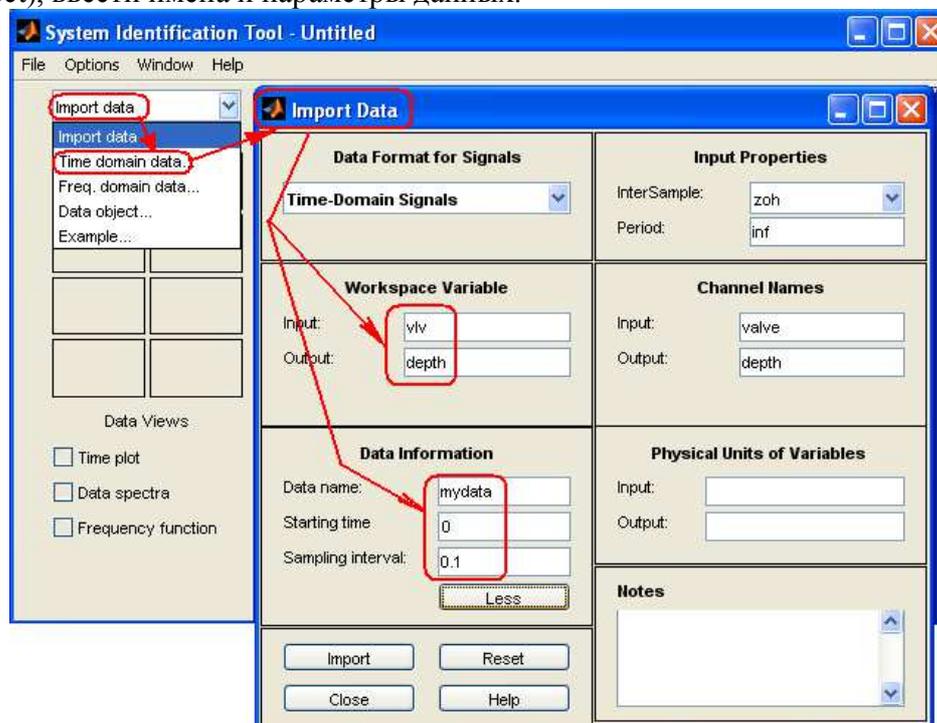


Рис. 3. Импорт экспериментальных данных.

Командой

`>> advice(mydata)`

можно получить информацию о качестве данных.

Можно увидеть временные, спектральные и частотные графики данных если установить соответствующие флажки поля Data View.

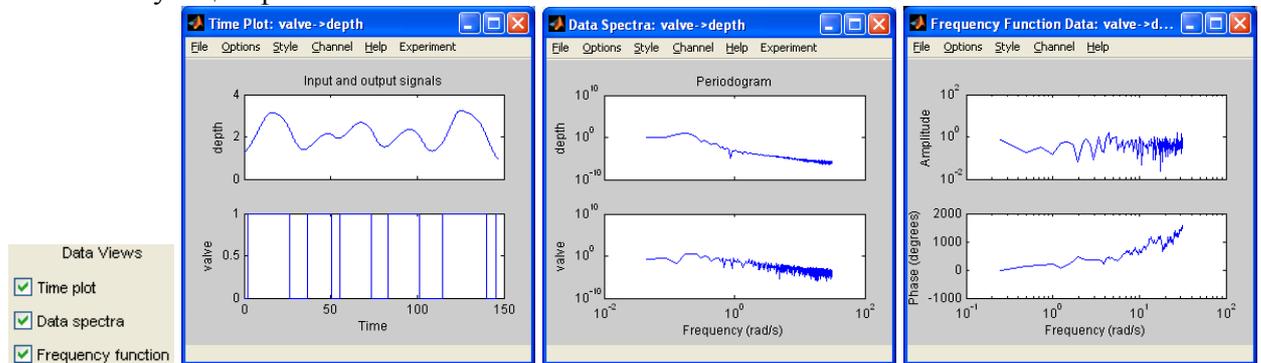


Рис. 4. Форматы отображения экспериментальных данных.

Можно импортировать до 8 блоков данных. Ненужные импортированные данные можно удалить перенеся их в корзину `trash`.

Предварительная обработка экспериментальных данных

При необходимости можно выполнить предварительную обработку данных (фильтрацию в выбранном диапазоне, удаление смещения, изменения диапазона данных, и др.) командами раздела `Preprocess` (Рис. 5).

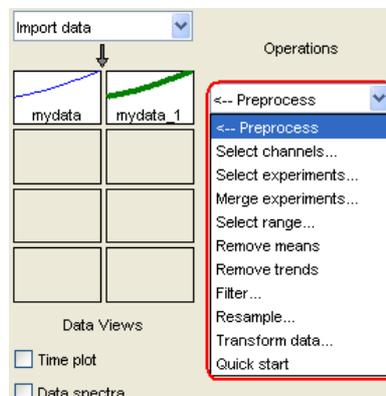
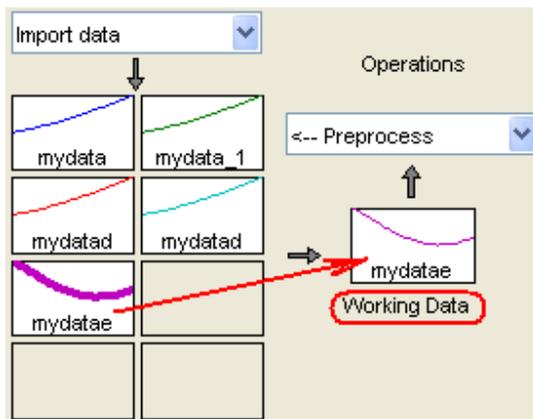


Рис. 5. Список режимов предварительной обработки экспериментальных данных.

Перед предварительной обработки данные раздела `import data` необходимо переместить в зону `Working Data`.



Результаты предварительной обработки (Рис. 6) попадают в окна раздела **import data** автоматически или после нажатия на клавишу **Insert**.

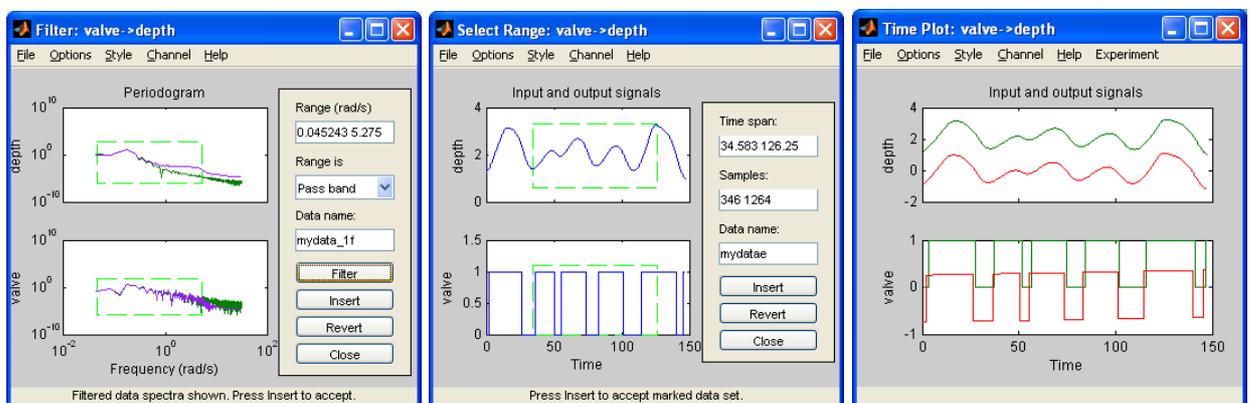


Рис. 6. Варианты предварительной обработки экспериментальных данных.

Нахождение подходящей структуры модели и ее порядка

Алгоритмы раздела **Estimate** позволяют найти связь между оригинальными или обработанными выходными и входными данными (области **Working Data**).

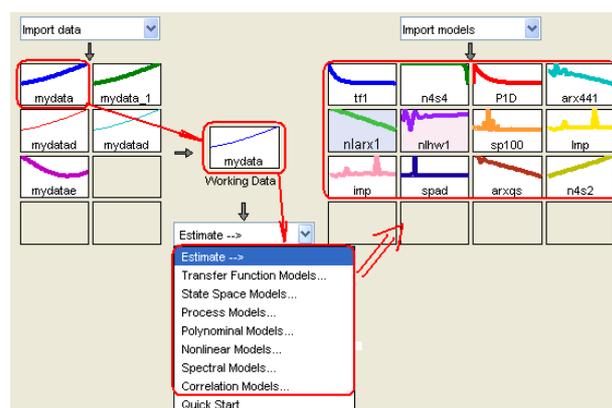


Рис. 7. Выбор структуры модели для идентификации.

Пакет **System Identification Toolbox** работает со следующими структурами моделей (см. Рис. 7):

- Transfer Functions $\frac{B(s)}{A(s)} = \frac{b_0s^m + b_1s^{m-1} + \dots + b_m}{a_0s^n + a_1s^{n-1} + \dots + a_n}$ (непрерывная)

$$\frac{B(z)}{A(z)} = \frac{b_mz^m + b_{m-1}z^{m-1} + \dots + b_0}{a_nz^n + a_{n-1}z^{n-1} + \dots + a_0} \text{ (дискретная)}$$

- State Space

$$x_{new} = Ax + Bu + Ke;$$

$$y = Cx + Du + e$$

- Process Models

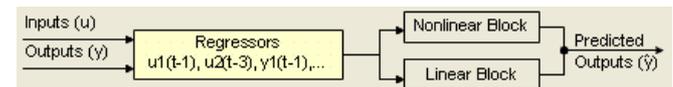
$$\frac{K(1+Tz s)}{(1+Tp1s)}, \frac{K \exp(-Td s)}{(1+Tp1s)}, \frac{K}{s(1+Tp1s)}$$

- Polynomial and State Space Models

$$Ay = Bu + e; Ay = Bu + Ce; y = [B/F]\mu + e;$$

$$y = [B/F]\mu + [C/D]e; \quad \begin{matrix} x_{new} = Ax + Bu + Ke; \\ y = Cx + Du + e \end{matrix}$$

- Nonlinear ARX Models



- Nonlinear Hammerstein-Wiener Models:



- Spectral Models: SPA(Blackman-Tukey); SPAFDR(Freq.dep.res.); ETFE (Smoothed Fourier Trf)

- Correlation Models:

Time span (seconds):

Order of whitening filter:

Параметры получаемой модели (порядок модели, количество полюсов и нулей, наличие задержки, и др.) задаются в соответствующих диалоговых окнах при выборе режима Estimation (см. Рис. 7). Можно запросить модели, например, с разным числом полюсов и нулей, и выбрать из них модель с наибольшей достоверностью (см. раздел “Проверка достоверности модели” ниже).

Параметры полученной модели можно запросить после идентификации дважды щелкнув на блоке зоны **import models** (Рис. 2), например, вот так выглядит окно описания передаточной функции модели:

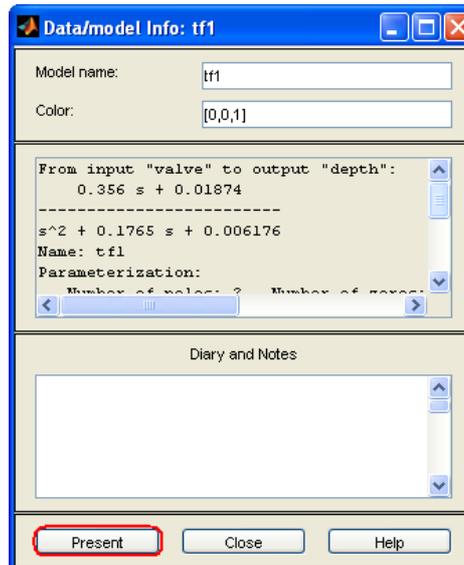


Рис. 8. Пример результата идентификации: модели непрерывной системы в виде передаточной функции.

Команда **Present** окна модели (Рис. 8) распечатывает описание модели в окне команд МатЛАБ:

tf1 =

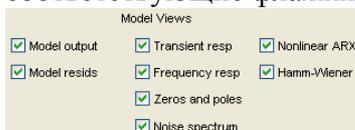
```
From input "valve" to output "depth":
(0.356 +/- 0.1265) s + (0.01874 +/- 0.7712)
-----
s^2 + (0.1765 +/- 2.226) s + (0.006176 +/- 0.2537)
```

```
Name: tf1
Parameterization:
  Number of poles: 2  Number of zeros: 1
  Number of free coefficients: 4
  Use "tfdata", "getpvec", "getcov" for parameters and their uncertainties.
```

```
Status:
Termination condition: Maximum number of iterations reached.
Number of iterations: 20, Number of function evaluations: 204
```

```
Estimated using TFEST on time domain data "mydata".
Fit to estimation data: 23.28% (simulation focus)
FPE: 0.1984, MSE: 0.191
More information in model's "Report" property.
```

Модели, найденные в разделе **Estimate**, можно сравнить по их реакции, расположению нулей и полюсов и др. параметрам. Для этого в секции Model Views необходимо установить соответствующие флажки.



Запуск поиска моделей и построение сравнительных графиков можно получить и по команде **Estimate > Quick Start**.

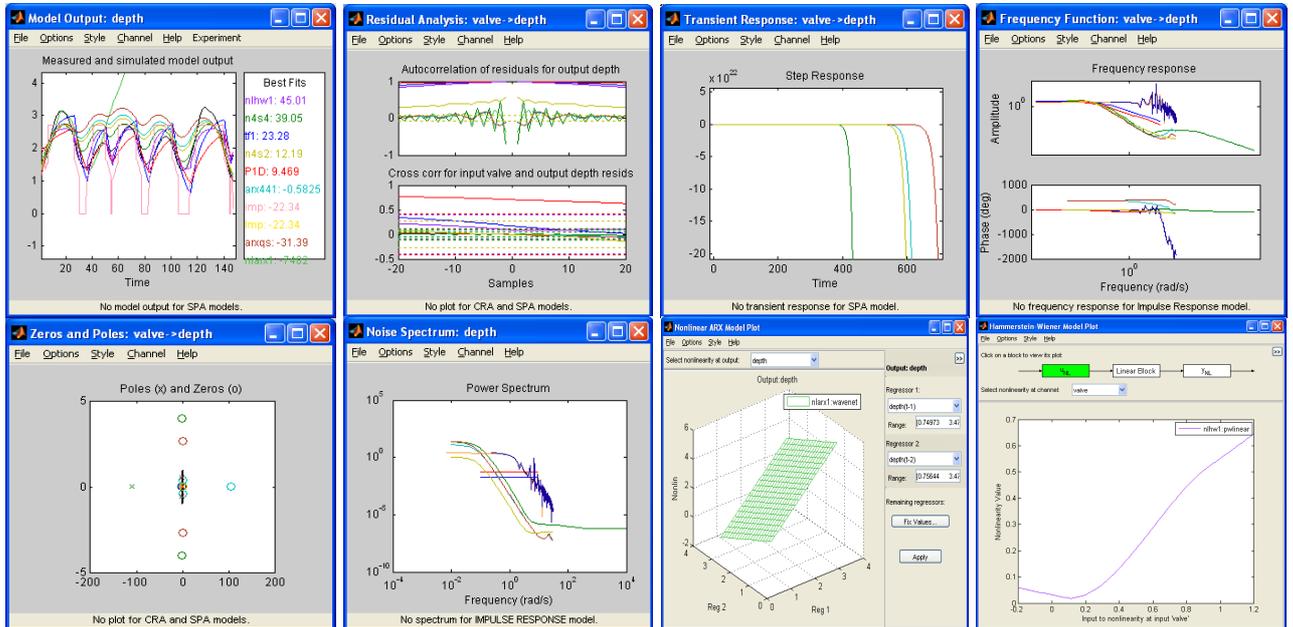


Рис. 9. Варианты (Model Views) отображения характеристик моделей.

При помощи **LTI Viewer** (просмотрщика) окна **System Identification Tool** можно увидеть характеристики линейной модели:

- реакции на ступенчатое и импульсное воздействия,
- амплитудно-фазовые частотные характеристики,
- распределение полюсов и нулей,
- и др.

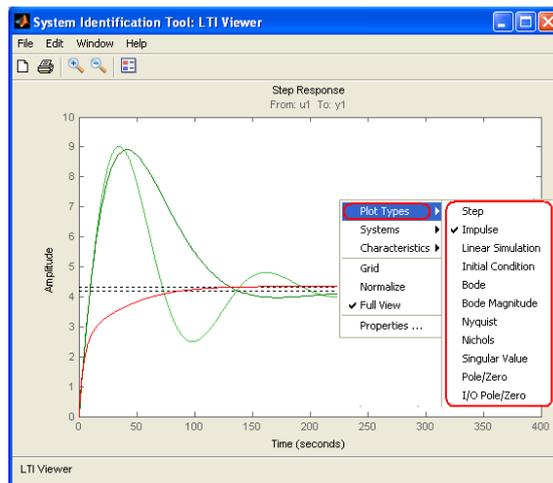


Рис. 10. Отклики разных моделей полученные переносом данных из зоны **import models** в окно **To LTI Viewer**.

Данные просмотрщику передаются из зоны **import models** через окно **To LTI Viewer**.

Проверка достоверности модели

System Identification Toolbox помогает проверить достоверность полученных моделей путем сравнения отклика модели и реальной системы на одно и то же экспериментальное воздействие. Также можно построить график рассогласования между откликами, рассчитать ошибку, изменить границы доверительного интервала, и т.д.

Для определения достоверности необходимо перенести независимый набор данных из зоны **import data** в зону **Validation data**, выполнить **Estimation**, и построить выюером график отклика модели (**Model Views**). На окне графика отклика модели будет присутствовать и отклик прототипа.

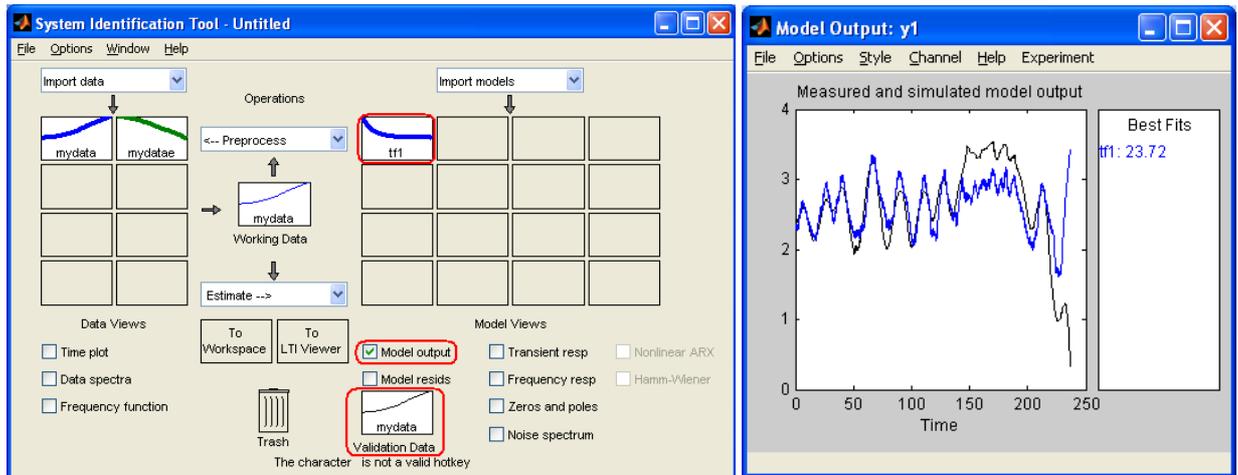


Рис. 11. Схема и результат валидации (Validation).

Отображение других сравнительных характеристик можно выполнить как показано на Рис. 9.

Отображение модели средствами МатЛАБ и Simulink

Данные зон **import data** и **model data** можно в виде объекта передать через окно  в рабочую зону МатЛАБ. Команда `> get()` раскрывает структуру объекта. Входные и выходные сигналы объекта зоны **import data** выделяются командами

```
> ampIn = get(mydatae, 'InputData'),  
> ampOut = get(mydatae, 'OutputData'), где mydatae – имя объекта, переданного в  
Workspace. Временной интервал данных  
> Ts = get(mydatae, 'Ts');
```

Параметры объекта зоны **model data**, например, передаточной функции `tf1` (числитель и знаменатель) можно получить командами

```
> num = get(tf1, 'num');  
> den = get(tf1, 'den');
```

Модель результата идентификации можно включить в окно модели Simulink через блок idmodel раздела **Simulink Library Browser > Libraries > System Identification Toolbox**, например

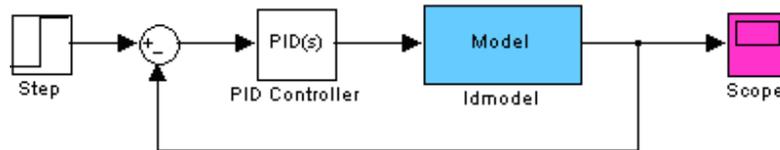


Рис. 12. Модель управления с моделью (idmodel) реальной системы.

Параметры модели вводятся в блок idmodel следующими форматами: idpoly, idss, idarx, idgrey, idproc, idft. Например, непрерывная передаточная функция $\frac{B(s)}{A(s)} = \frac{b_0s^m + b_1s^{m-1} + \dots + b_m}{a_0s^n + a_1s^{n-1} + \dots + a_n}$ задается как **idtf**(коэф. числителя, коэф. знаменателя), а дискретная передаточная функция: **idtf**(коэф. числителя, коэф. знаменателя, интервал времени)

Сравнить реакцию полученной модели с экспериментальными значениями в Simulink можно, например, средствами следующей модели

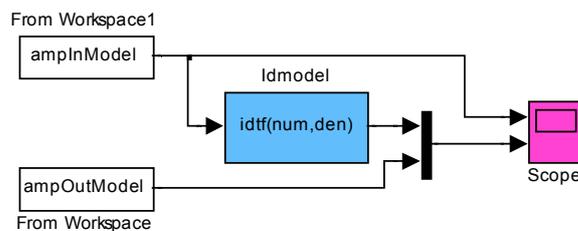


Рис. 13. Схема сравнения откликов прототипа и его модели.

в которой формат экспериментальных данных зоны **import data** окна идентификации преобразован для ввода в модель следующим кодом.

```
ampIn = get(mydatae, 'InputData');
ampOut = get(mydatae, 'OutputData');
Ts = get(mydatae, 'Ts');
time = ((0:length(ampIn)-1)*0.1)';
ampInModel = [time, ampIn];
ampOutModel = [time, ampOut];
```

ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

Для выполнения этапов идентификации при помощи пакета **System Identification Toolbox** необходимо создать в Simulink виртуальный объект который будет рассматриваться как реальный объект идентификации с неизвестными параметрами.

Задание 1. Построение виртуального объекта.

1. Постройте в Simulink виртуальную систему протекания жидкости (см. Рис. 14) 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.

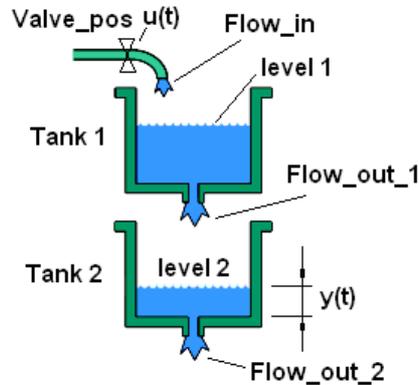


Рис. 14. Прототип системы для идентификации.

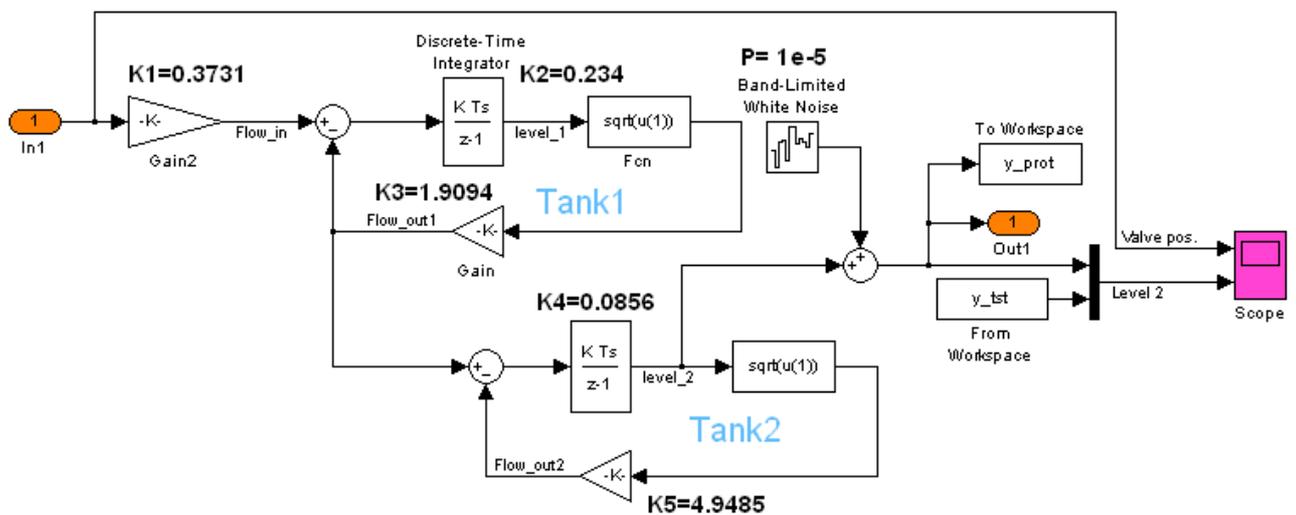


Рис. 15. Виртуальная модель прототипа (Рис. 14).

- Загрузите в рабочую зону данные выхода y и входа u модели МатЛАБ
`> load twotankdata`
- Подключите загруженные данные к модели (Рис. 15). Временной интервал данных: 0.2 сек.

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

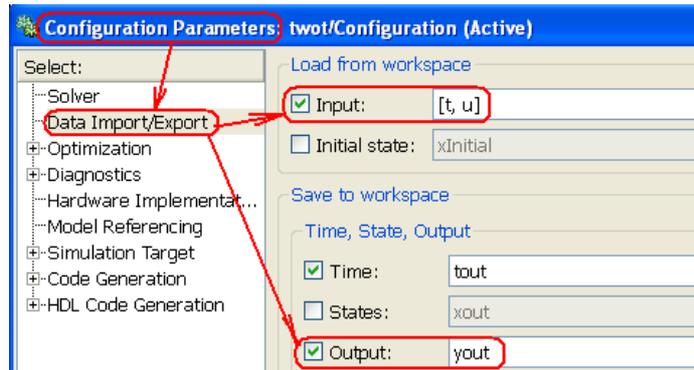


Рис. 16. Настройка конфигурации модели: Подключение портов In1 и Out1 для выполнения (при необходимости) поиска параметров модели с откликом $y(t)$ на входное воздействие $u(t)$.

4. Запустите модель и сравните отклик модели с $y(t)$.

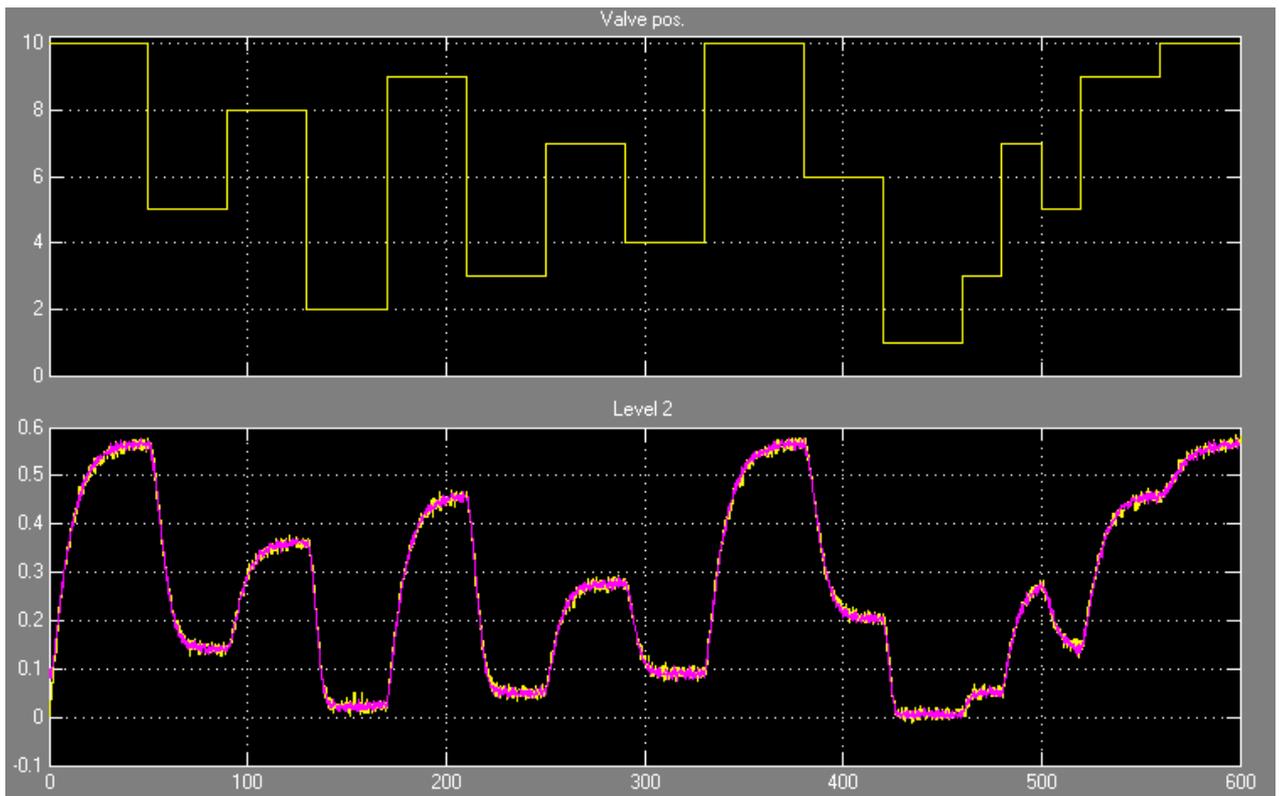


Рис. 17. Вход модели, $u(t)$ – верхний график. Отклик модели (желтый график) и $y(t)$ – нижние графики.

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

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

1. Через командную строку загрузите **System Identification Toolbox** (Рис. 2)

> load ident

2. Введите сигналы **u** и **y** (см. Рис. 3) Укажите начальное время **0** и интервал **0.2**.

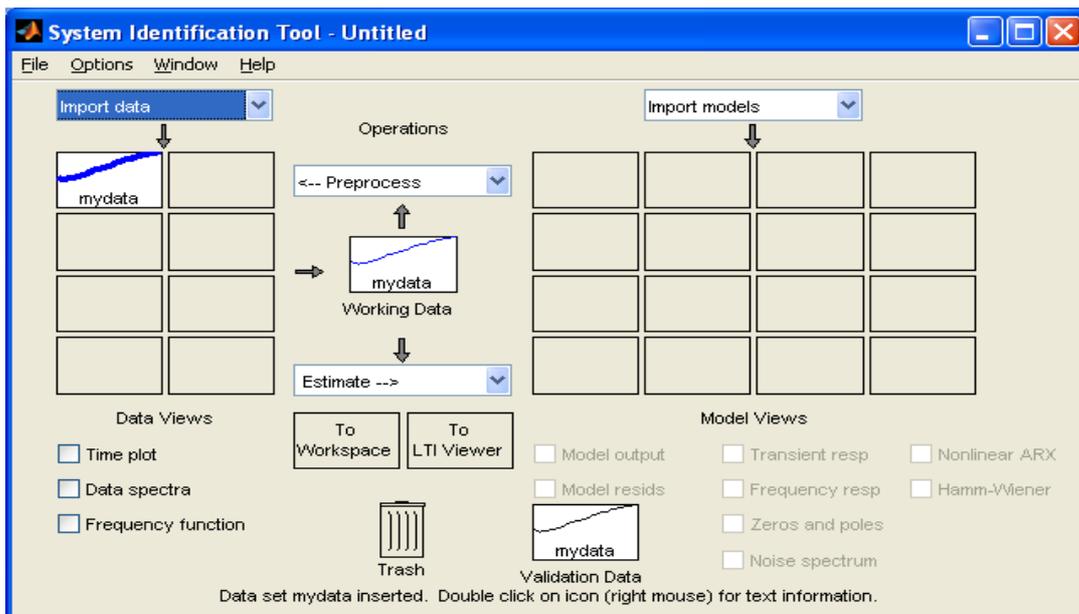
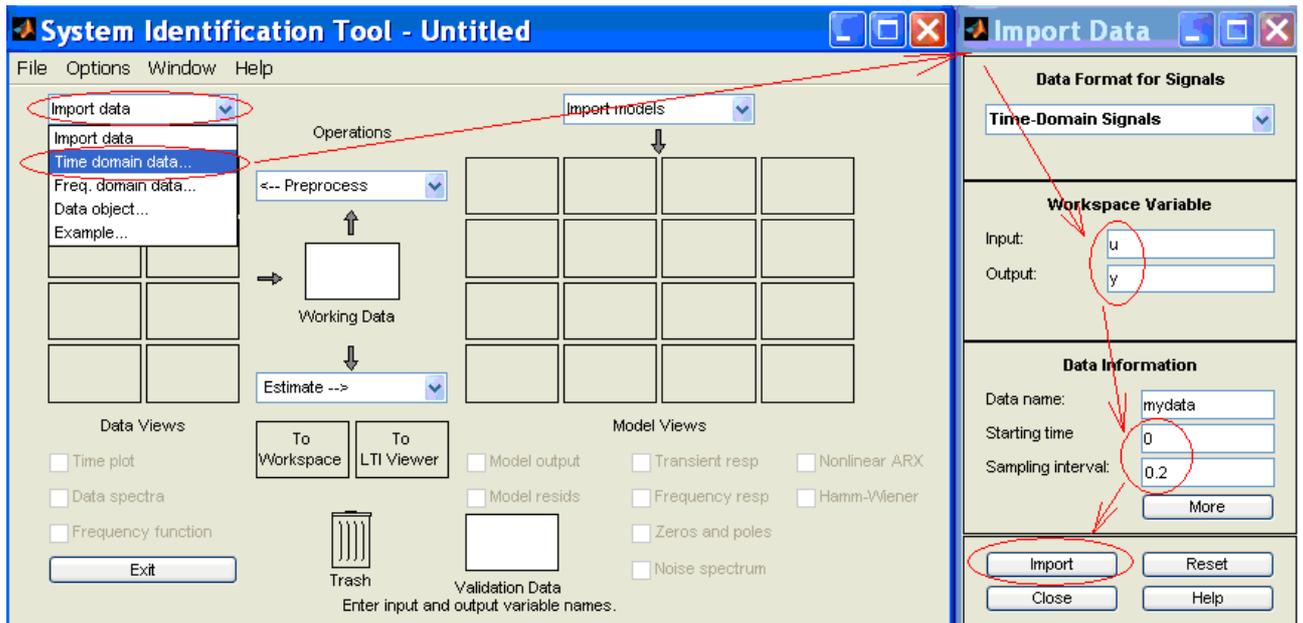


Рис. 18. Окно ввода экспериментальных данных (**u** и **y**).

3. Убедитесь, что введены правильные данные - постройте графики данных (см. Рис. 4).
4. При необходимости выполните предварительную обработку данных (см. Рис. 5 и Рис. 6).
5. Выберите структуру модели идентификации, например, ARMAX (Рис. 7, Рис. 19) и введите ее параметры [na nb nc nk].

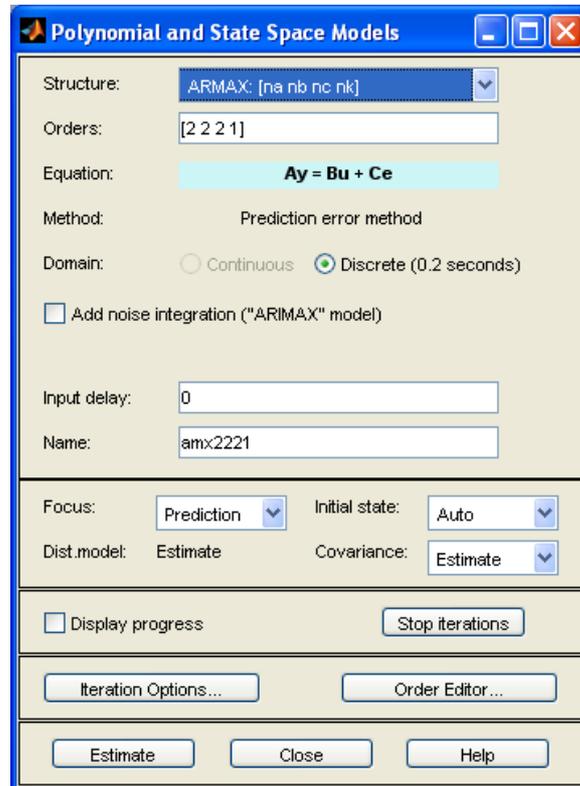


Рис. 19. Окно ввода параметров модели ARMAX. Число полюсов, число нулей плюс 1; число коэффициентов и задержка системы: $n_a = 2$, $n_b = 2$, $n_c = 2$, $n_k = 1$; [2 2 2 1].

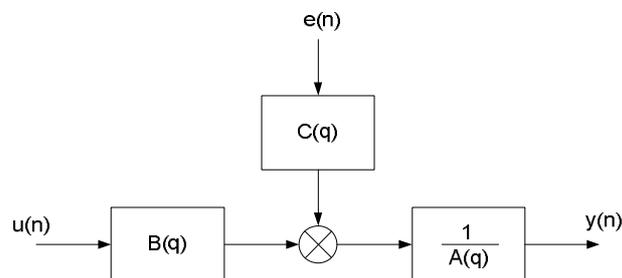


Рис. 20. Полиномиальное представление ARMAX модели

Структура ARMAX модели:

$$y(t) + a_1 y(t-1) + \dots + a_{n_a} y(t-n_a) = b_1 u(t-n_k) + \dots + b_{n_b} u(t-n_k-n_b+1) + c_1 e(t-1) + \dots + c_{n_c} e(t-n_c) + e(t)$$

В компактной форме:

$$A(q)y(t) = B(q)u(t-n_k) + C(q)e(t),$$

где $y(t)$ - выход в момент времени t ; n_a - число полюсов; n_b - число нулей плюс 1; n_c - число С коэффициентов; n_k - число тактов прохода входного сигнала на выход (задержка системы); $y(t-1) \dots y(t-n_a)$ - предыдущие выходы от которых зависит текущий выход; $u(t-n_k) \dots u(t-n_k-n_b+1)$ - предыдущие входы от которых зависит текущий выход; $e(t-1) + \dots + c_{n_c} e(t-n_c)$ - значение белого шума.

Параметры n_a , n_b и n_c - порядки ARMAX модели, n_k - задержка, q – оператор задержки.

$$A(q) = 1 + a_1 q^{-1} + \dots + a_{n_a} q^{-n_a}$$

$$B(q) = b_1 + b_2 q^{-1} + \dots + b_{n_b} q^{-n_b+1}$$

$$C(q) = 1 + c_1 q^{-1} + \dots + a_{n_c} q^{-n_c}$$

6. Нажатием на клавишу  запустите процесс идентификации параметров выбранной структуры модели по экспериментальным значениям реакции и воздействия прототипа системы.
7. Выведите значения параметров модели - результаты идентификации в окне команд и Workspace.

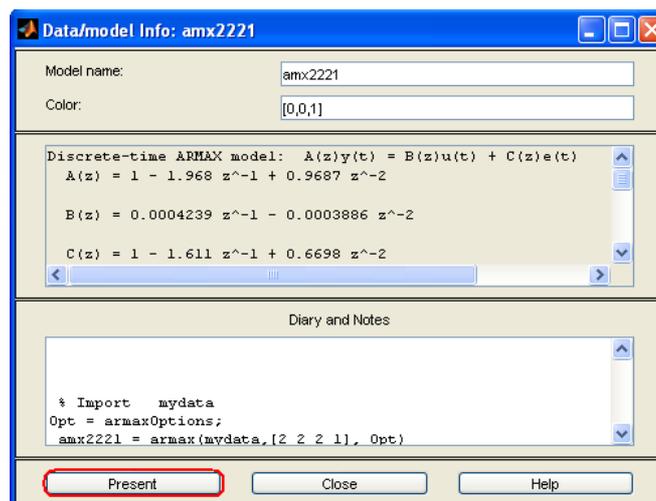


Рис. 21. Получение информации двойным щелчком по соответствующему полю зоны **Import models**.

Вывод в окне команд:

amx2221 =

Discrete-time ARMAX model: $A(z)y(t) = B(z)u(t) + C(z)e(t)$

$A(z) = 1 - 1.968 (+/- 0.003824) z^{-1} + 0.9687 (+/- 0.003765) z^{-2}$

$B(z) = 0.0004239 (+/- 6.504e-05) z^{-1} - 0.0003886 (+/- 6.639e-05) z^{-2}$

$C(z) = 1 - 1.611 (+/- 0.0136) z^{-1} + 0.6698 (+/- 0.01346) z^{-2}$

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:

Termination condition: Near (local) minimum, (norm(g) < tol).

Number of iterations: 8, Number of function evaluations: 20

Estimated using POLYEST on time domain data "mydata".

Fit to estimation data: 96.65% (prediction focus)

FPE: 3.867e-05, MSE: 3.844e-05

More information in model's "Report" property.

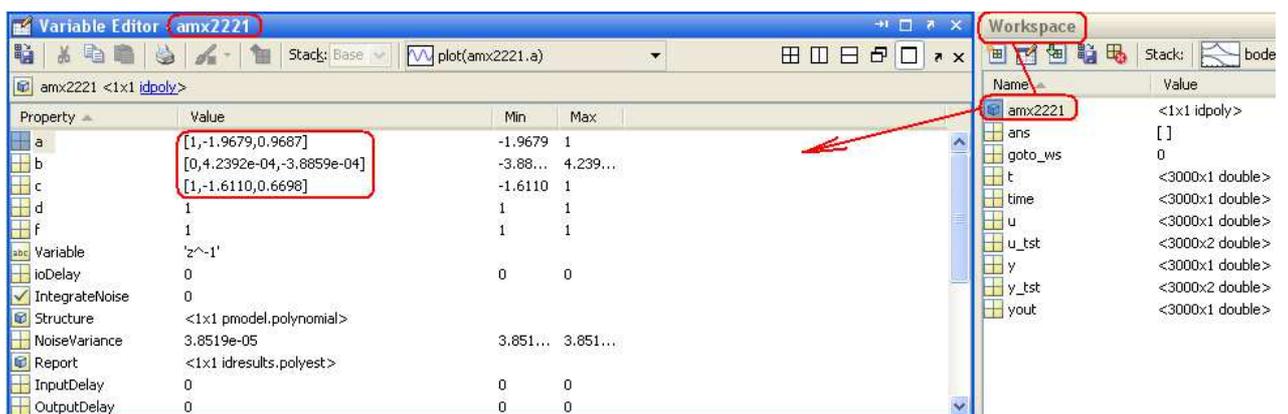


Рис. 22. Раскрытие amx2221 структуры рабочей области.

8. Отобразите графики структуры Workspace **amx2221**.

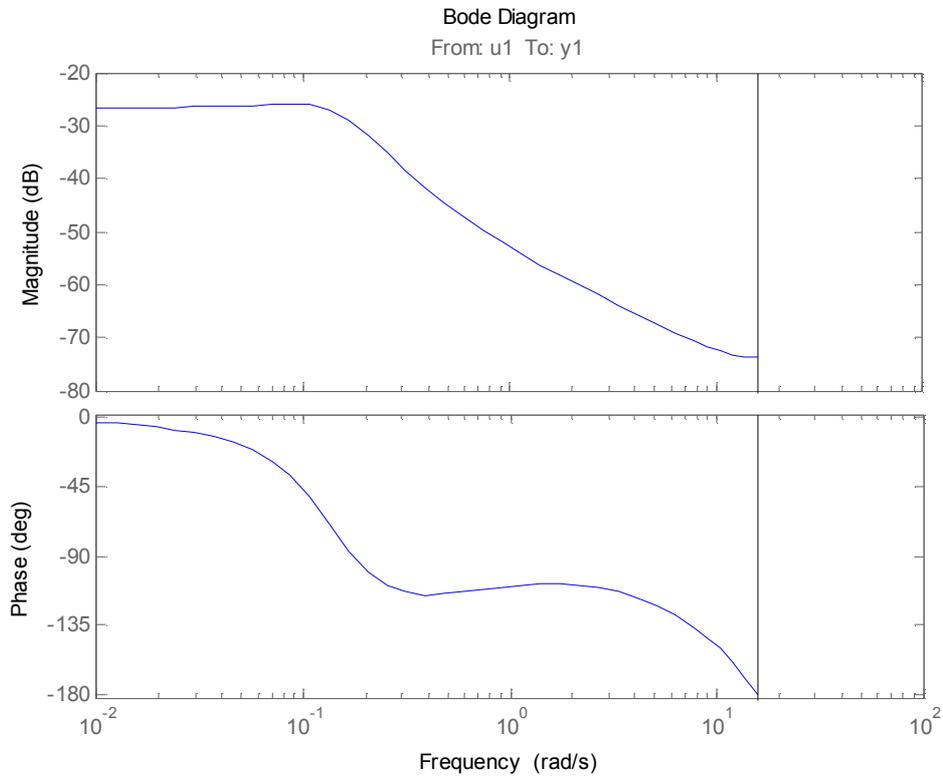


Рис. 23. График **amx2221** структуры рабочей области.

9. Постройте характеристики модели (Рис. 9, Рис. 10).

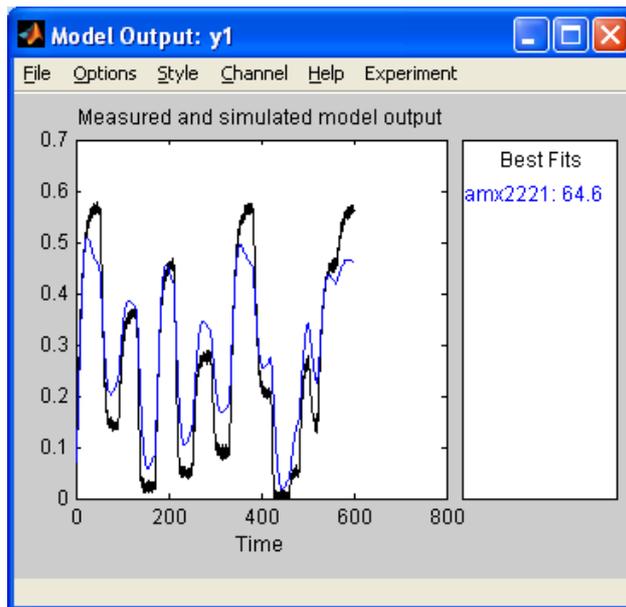


Рис. 24. Отклик модели и прототипа.

10. Для увеличения сходимости откликов модели и ее прототипа повторите пункты 4 .. 10 с другими структурами и параметрами моделей.

Задание 3. Верификация модели.

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.

1. Введите независимые данные - сигналы которые не использовались при идентификации (см. Рис. 3).
2. Перенесите поле данных из зоны **Import data** в поле **Validation Data**.
3. Выделите модель в зоне **Import models** и постройте сравнительные характеристики модели (см. Рис. 9): реакции на единичное ступенчатое воздействие; частотную характеристику; распределение полюсов и нулей и др.

Задание 4. Построение модели прототипа в Simulink.

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** как показано на Рис. 12.

1. Постройте модель прототипа и сравните переходные процессы модели и прототипа.

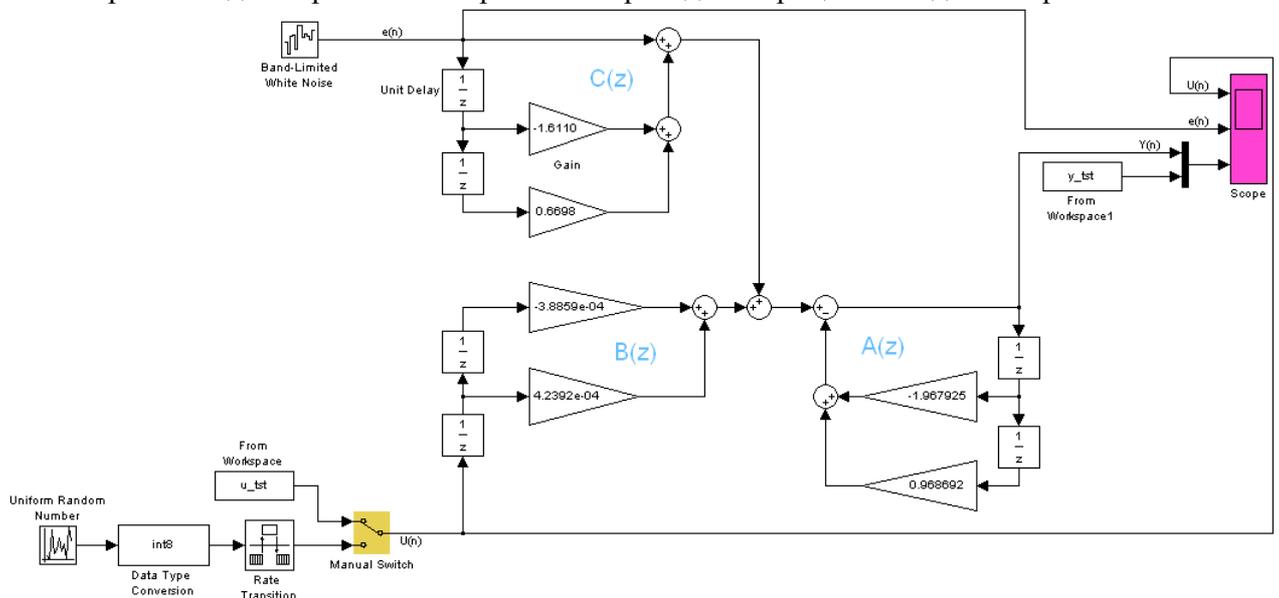
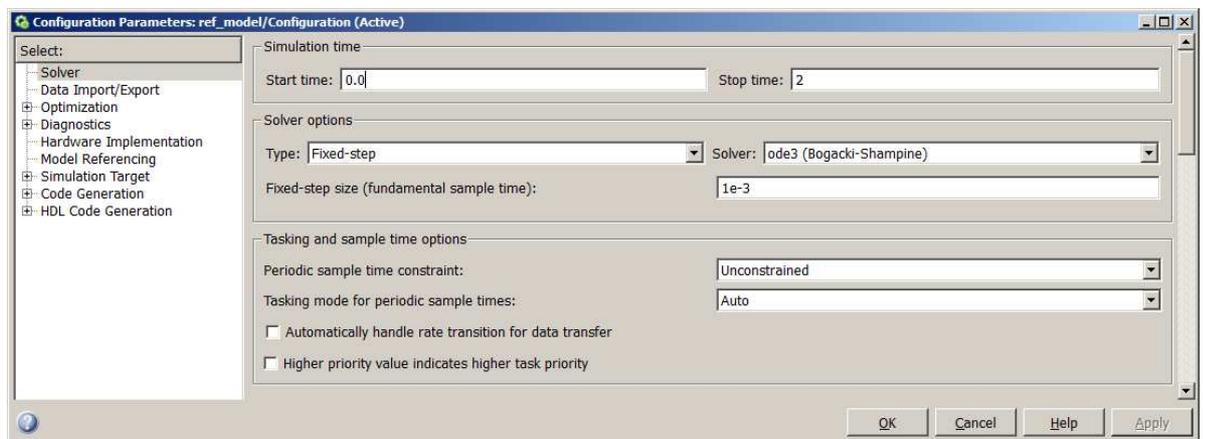
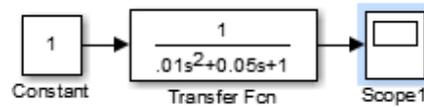


Рис. 25. Модель идентификации прототипа ARMAX [2221]. $A(z) = 1 - 1.967925 z^{-1} + 0.968692 z^{-2}$; $B(z) = 0.00042392 z^{-1} - 0.00038859 z^{-2}$; $C(z) = 1 - 1.6110 z^{-1} + 0.6698 z^{-2}$.

Задание 5. Проверка эффективности идентификации в среде MATLAB R2015a.

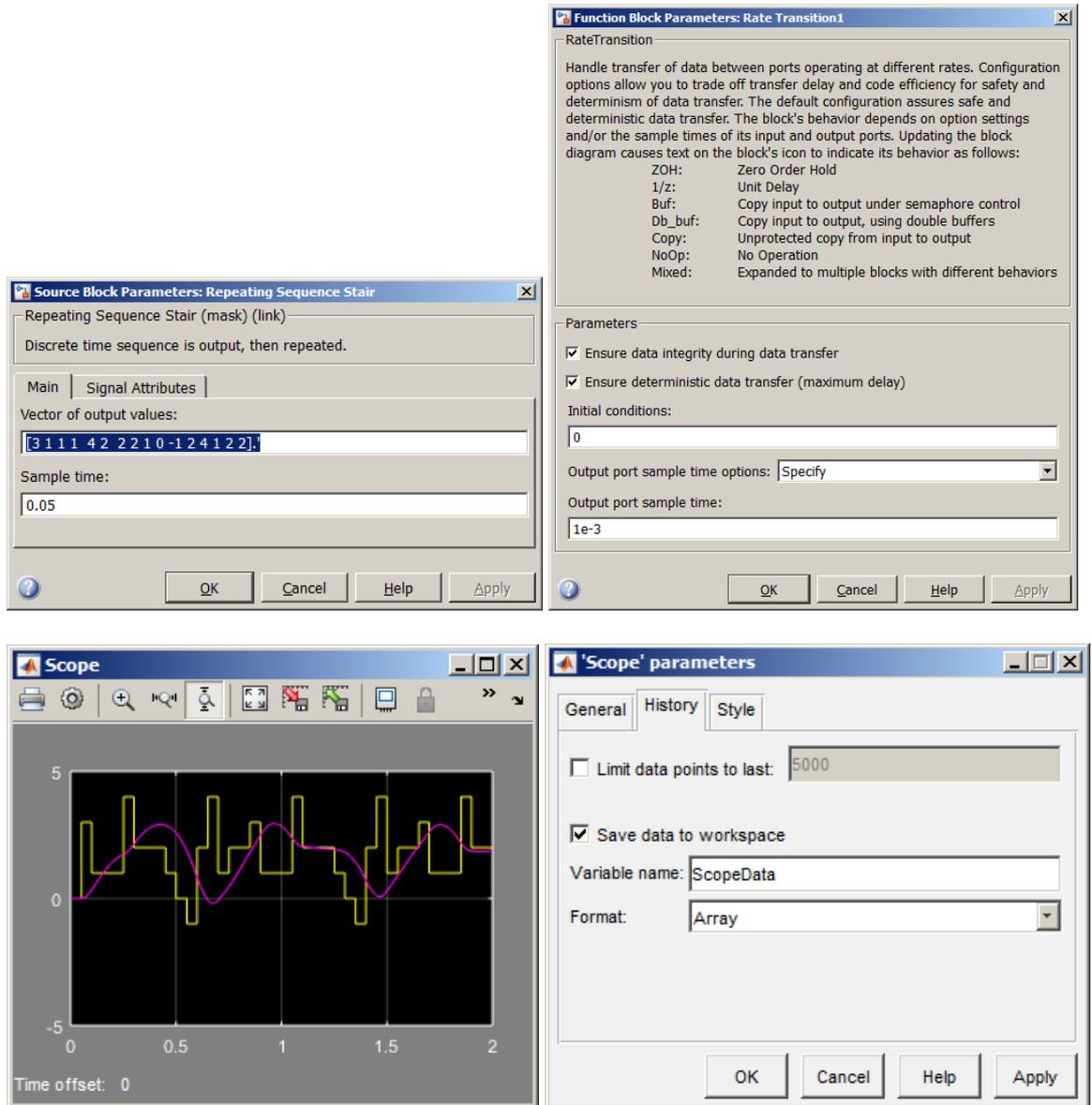
1. Для проверки эффективности идентификации – нахождения передаточной функции по известной реакции и входному воздействию будем использовать заранее вычисленные реакцию и воздействие известной структуры. Для этого

1.1. Постройте следующую модель в виде передаточной функции второго порядка в среде Simulink.



1.2. Подайте следующее воздействие на вход модели и вычислите реакцию на входное воздействие. Сохраните воздействие и реакцию в Workspace.



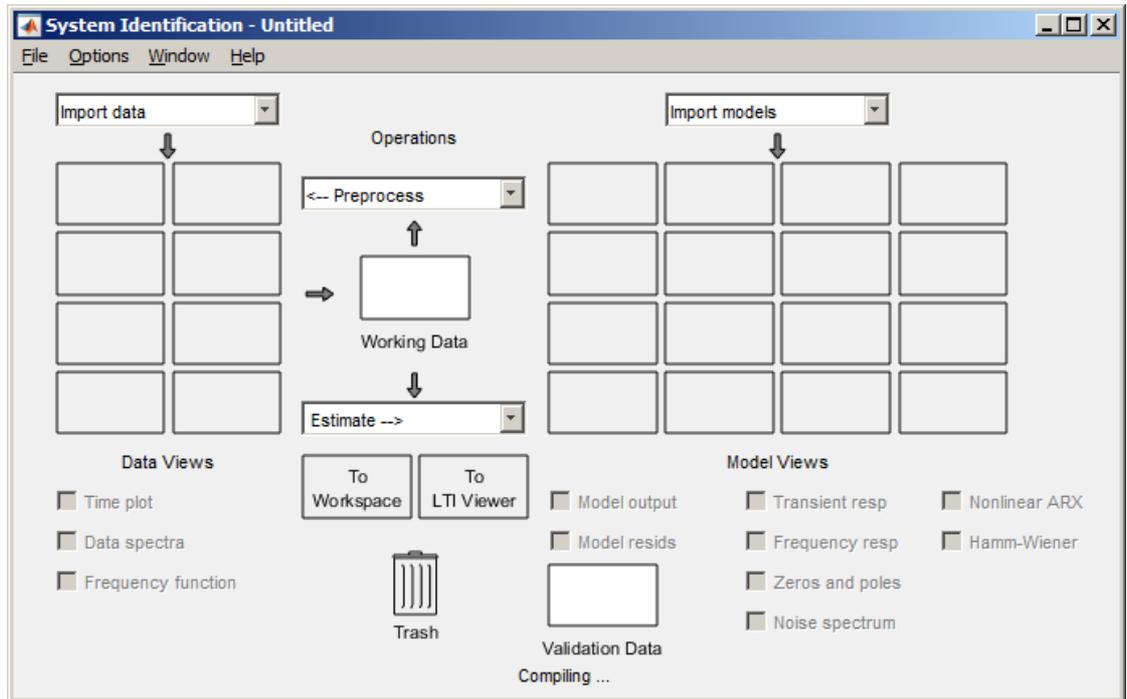


Вычисленную реакцию и входное воздействие будем использовать для сравнения заданной (в п.1) передаточной функции с передаточной функцией, вычисляемой по реакции и входному воздействию далее методом идентификации.

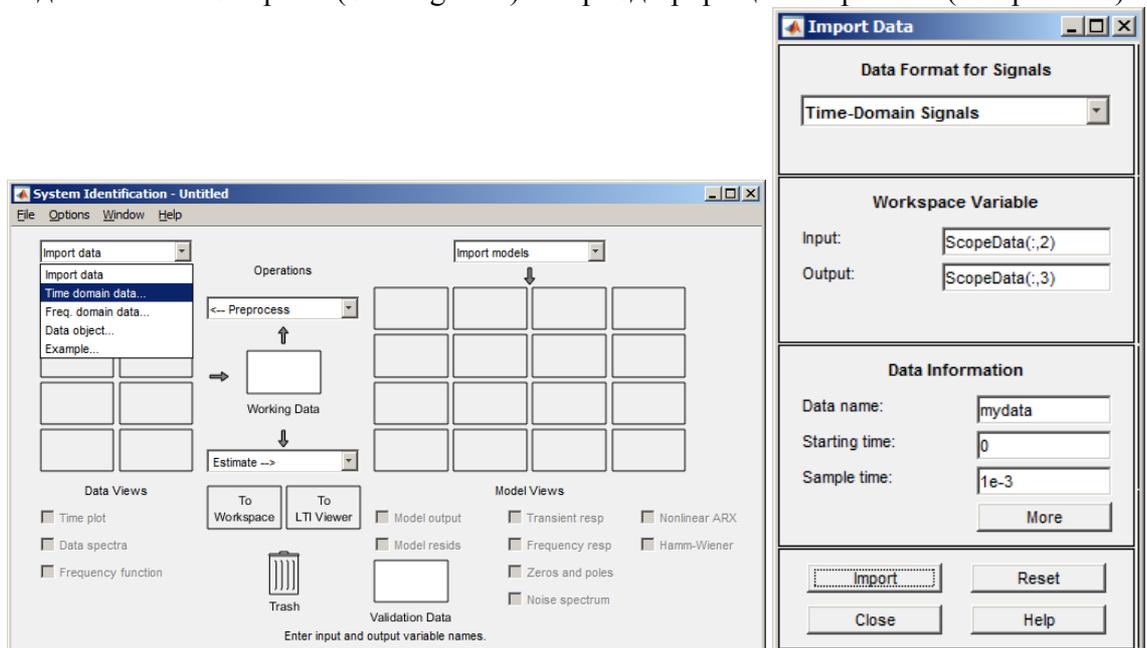
2. Используя полученные (в п.1) воздействие и реакцию найдите методом идентификации передаточную функцию “черного ящика”. Для этого

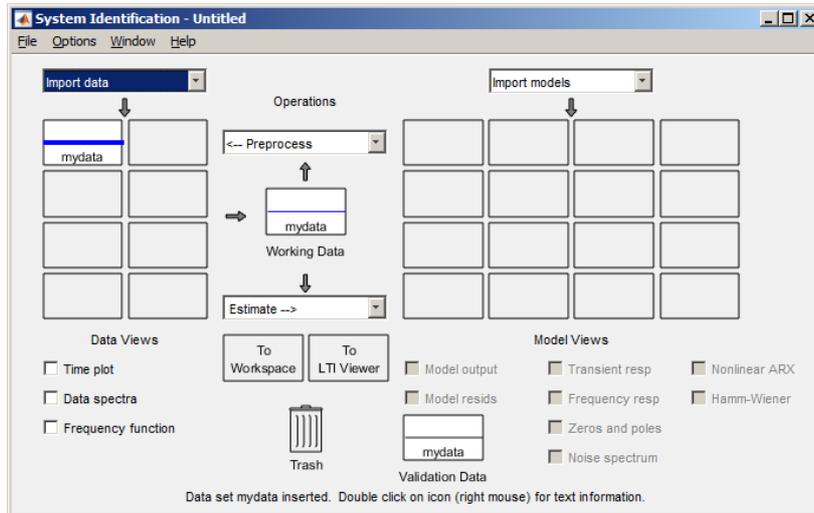
2.1. Загрузите инструмент идентификации через окно Command Window.

> systemIdentification

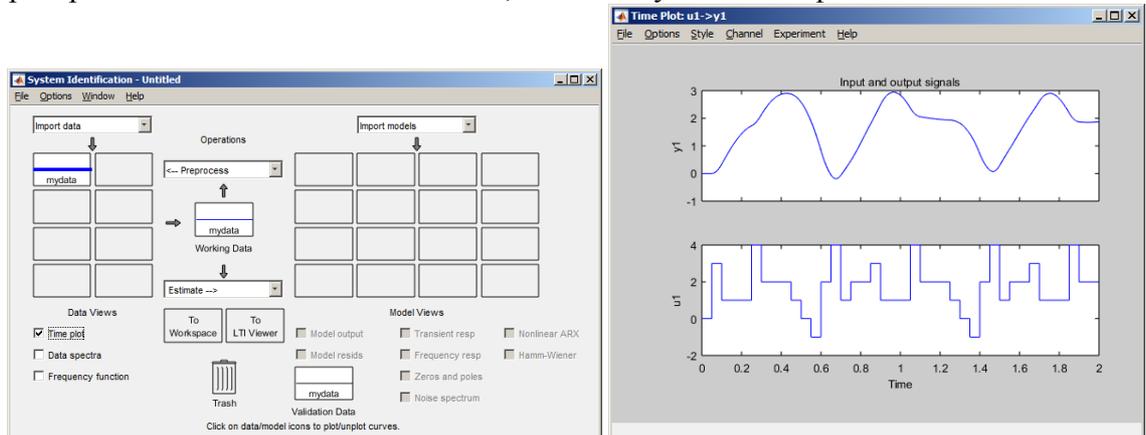


- 2.2. Импортируйте вход и выход “неизвестной модели”: Import data > Time domain data. Введите начальное время (Starting time) и период приращения времени (Sample time).

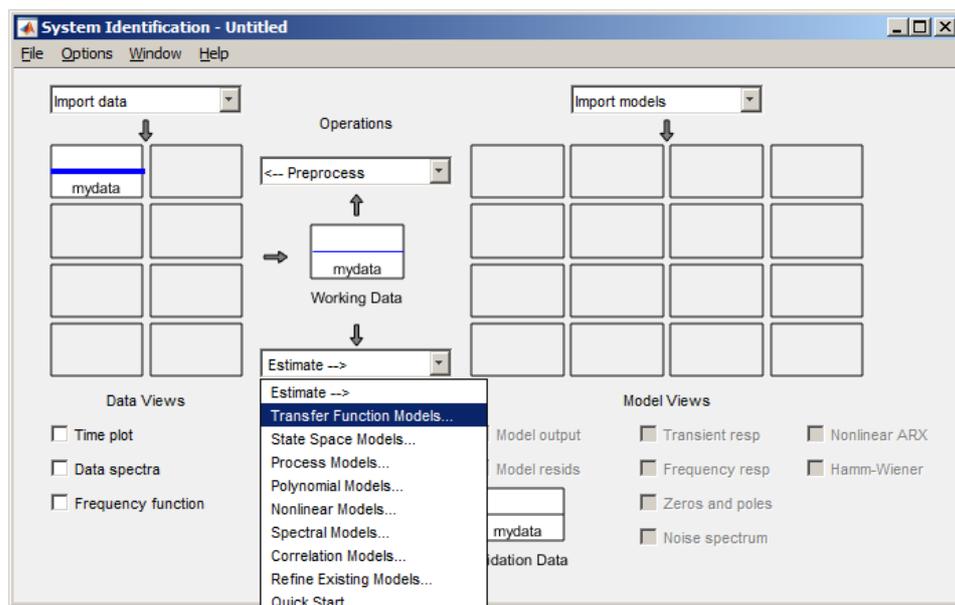




2.3. Проверьте введенные входные данные, для этого установите флажок 



2.4. Выполните идентификацию (Estimate > Transfer function Models).



Transfer Functions

Model name: tf2

Number of poles:

Number of zeros:

Continuous-time
 Discrete-time (Ts = 0.001)
 Feedthrough

I/O Delay
Estimation Options

Plant Identification Progress

Transfer Function Identification
 Estimation data: Time domain data mydata
 Data has 1 outputs, 1 inputs and 2001 samples.
 Number of poles: 2, Number of zeros: 1
 Initialization Method: "iv"

Estimation Progress

Initializing model parameters...
 Initializing using "iv" method...
 done.

Initialization complete.

Nonlinear least squares with automatically chosen line search method

Iteration	Cost	Norm of step	First-order optimality	Improvement (%) Expected	Achieved	Bisections
0	1.12876e-28	-	3.17e+16	6.55e+29	-	-
1	9.66654e-29	1.32e-12	1.34e+18	6.55e+29	14.4	0
2	6.35457e-29	1.16e-13	1.4e+18	7.34e+29	34.3	0
3	6.35126e-29	7.44e-15	1.41e+18	9.86e+29	0.052	3
4	6.35126e-29	2.91e-17	1.41e+18	9.86e+29	3.18e-05	11
5	6.35126e-29	1.82e-18	1.41e+18	9.86e+29	2.53e-06	15
6	6.35126e-29	9.1e-19	1.41e+18	9.86e+29	1.29e-09	16
7	6.35126e-29	1.14e-19	1.41e+18	9.86e+29	1.6e-10	19
8	6.35126e-29	2.84e-20	1.41e+18	9.86e+29	4.01e-11	21
9	6.35126e-29	3.55e-21	1.41e+18	9.86e+29	5.01e-12	24
10	6.33538e-29	1.96e-14	7.5e+17	9.86e+29	0.25	5
11	6.26846e-29	7.56e-15	8.31e+17	9.78e+29	1.06	3
12	6.26846e-29	8.45e-19	8.31e+17	9.83e+29	1.74e-05	16
13	6.26846e-29	2.11e-19	8.31e+17	9.83e+29	4.35e-06	18
14	6.26846e-29	1.06e-19	8.31e+17	9.83e+29	2.17e-06	19
15	6.26846e-29	5.28e-20	8.31e+17	9.83e+29	1.09e-06	20
16	6.26846e-29	2.64e-20	8.31e+17	9.83e+29	5.43e-07	21
17	6.00293e-29	1.38e-14	5.33e+17	9.83e+29	4.24	5
18	5.99721e-29	1.81e-15	5.56e+17	1.04e+30	0.0952	5
19	5.99479e-29	9.03e-16	5.67e+17	1.04e+30	0.0405	6
20	5.99448e-29	1.13e-16	5.68e+17	1.04e+30	0.00503	9

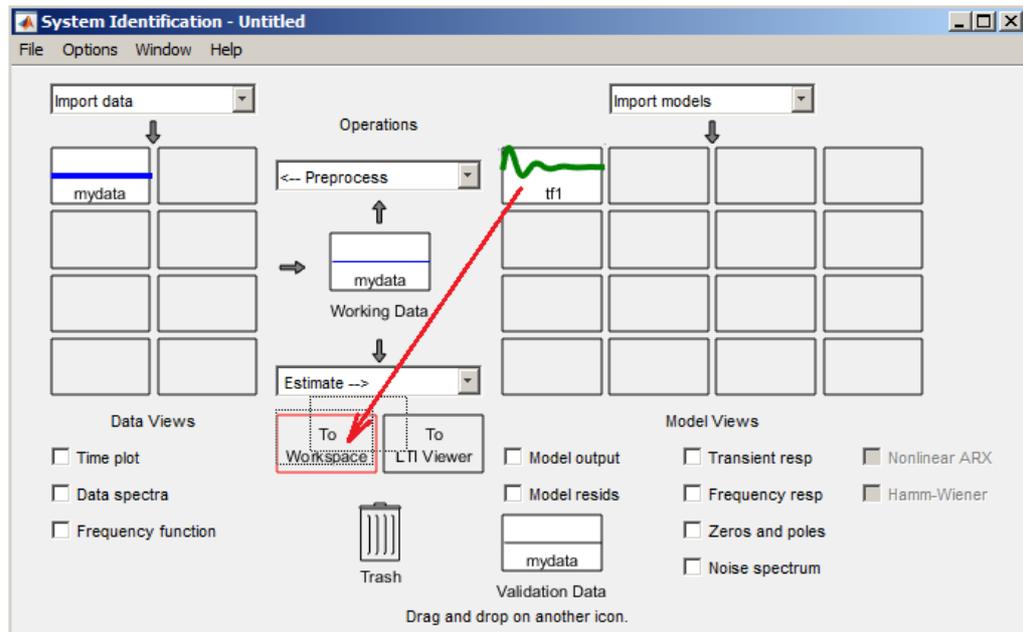
Estimating parameter covariance...
 done.

Result

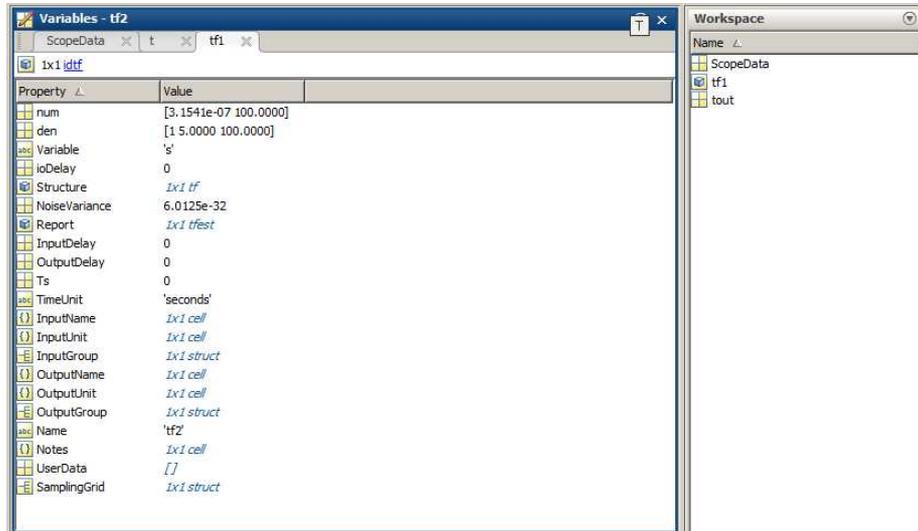
Termination condition: Near (local) minimum, (norm(g) < tol).
 Number of iterations: 20, Number of function evaluations: 329

Status: Estimated using TFEST with Focus = "simulation"
 Fit to estimation data: 100%, FPE: 6.03043e-29

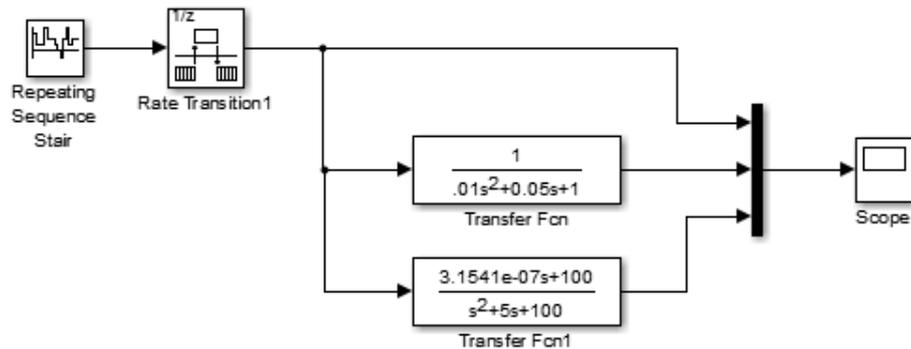
2.5. Переместите параметры найденной передаточной функции в Workspace.



2.6. Раскройте структуру передаточной функции, содержащей параметры числителя (num) и знаменателя (den).



3. Введите полученную передаточную функцию (п. 2.6.) с параметрами числителя (num) и знаменателя (den) в Simulink. Сравните реакции исходной передаточной функции (п.1.) и передаточной функции, полученной методом идентификации (п. 2.6.).



4. Сделайте вывод по результатам сравнения.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Назовите цели и задачи идентификации.
2. Дайте оценку технологии компьютерной идентификации в среде МатЛАБ.
3. Сравните идентификацию с оптимизацией в МатЛАБ.
4. Каким образом задается структура модели идентификации?
5. Как определяется погрешность идентификации?
6. Что такое верификация и чем она отличается от тестирования?

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Help MatLAB
2. Dr. Bob Davidov. Компьютерные технологии управления в технических системах
<http://portalnp.ru/author/bobdavidov>