



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

Факультет: «Специальное машиностроение»

Кафедра: «Робототехнические системы и мехатроника»

Лабораторная работа № 6

по курсу «Теория автоматического управления»

Вариант 13

Выполнил: Садовец Роман
Группа: СМ7-62Б

Проверил(а):

Москва, 2024 г.

1. Структура проекта

Лабораторная работа №6 выполнялась на базе лабораторной работы №7. В рамках лабораторной реализовано:

- Регулятор на нечеткой логике
- LQR-контроллер
- Логическое управление
- Наблюдатель Люенбергера
- Фильтр Калмана
- Тесты к наблюдателю Люенбергера

Внешний вид проекта в среде Matlab представлен на рисунке 1

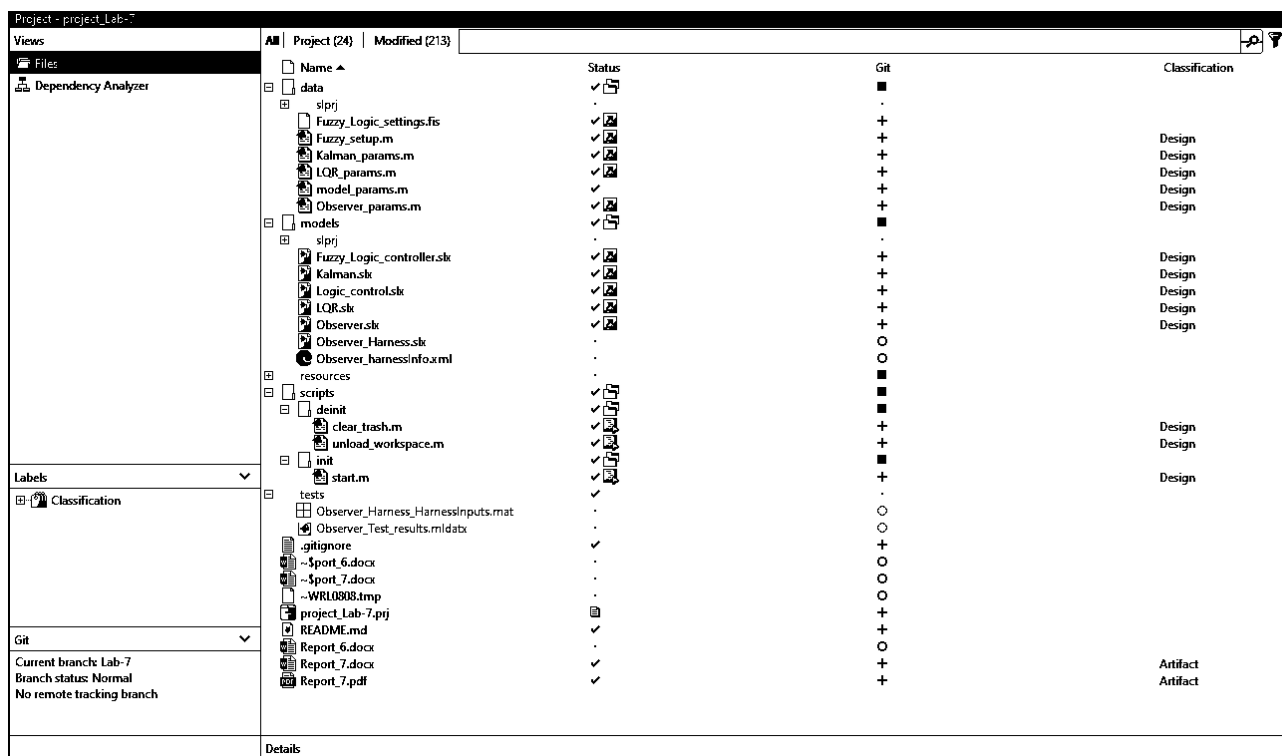


Рис. 1. Структура проекта в среде разработки Matlab

2. Shotrcuts проекта

В системе были также созданы Shortcut-ы (рис. 2), каждый из которых предназначен для облегчения взаимодействия пользователя с системой. Каждый из Shortcut-ов отвечают за каждую из разрабатываемых систем, и включает в себя

параметры, необходимые для запуска системы, и саму модель (регулятор на нечеткой логике также дополнительно включает настройки в формате.fis)

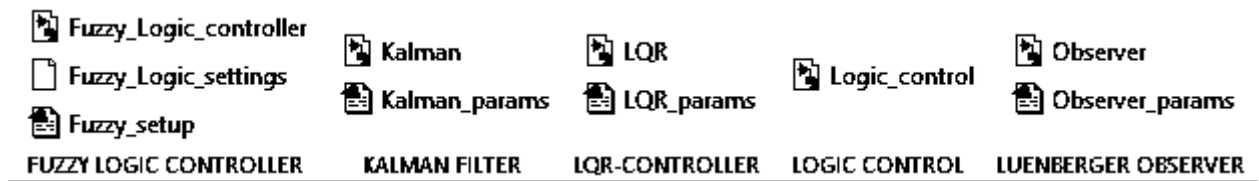


Рис. 2. Shortcuts в проекте

3. Интеграция с системой контроля версий git

Аналогично всем предыдущим лабораторным работам, в системе используется система контроля версий git (см. рис. 1). Для ознакомления с работой можно обратиться по соответствующей ссылке (см. прил. 1).

4. Автоматическое прописывание констант при открытии (инициализация)

Внутри директории ./scripts/init расположен файл start.m с инициализацией констант, которые применяются во всех имеющихся моделях. Структура файла имеет следующий вид:

```
clear all;  
close all;  
clc;  
  
model_params
```

5. Деинициализация

Также имеются и два файла (clear_trash.m и unload_workspace.m), предназначенные для очистки всего проекта от кэша и очищению workspace в Matlab.

```
% Remove project Workspace  
  
clear project_Lab-7;  
clear out;
```

```

% Remove all trash from folder

clearTrash();

function clearTrash()
    all_models = ls('./models/');
    warning ('off','all');

    for i = 1:size(all_models, 1)
        model_name = strip(all_models(i,:));
        model_name = model_name(1:length(model_name)-4);

        delete(strcat(model_name, '.elf'));
        delete(strcat(model_name, '.slxc'));
        delete(strcat(model_name, '.zip'));

        if isfolder('./slprj')
            rmdir('./slprj', 's');
        end
    end
end
end

```

Для автоматического запуска файлов инициализации и деинициализации, выберем эти файлы в меню проекта Startup and Shutdown (рис. 3).

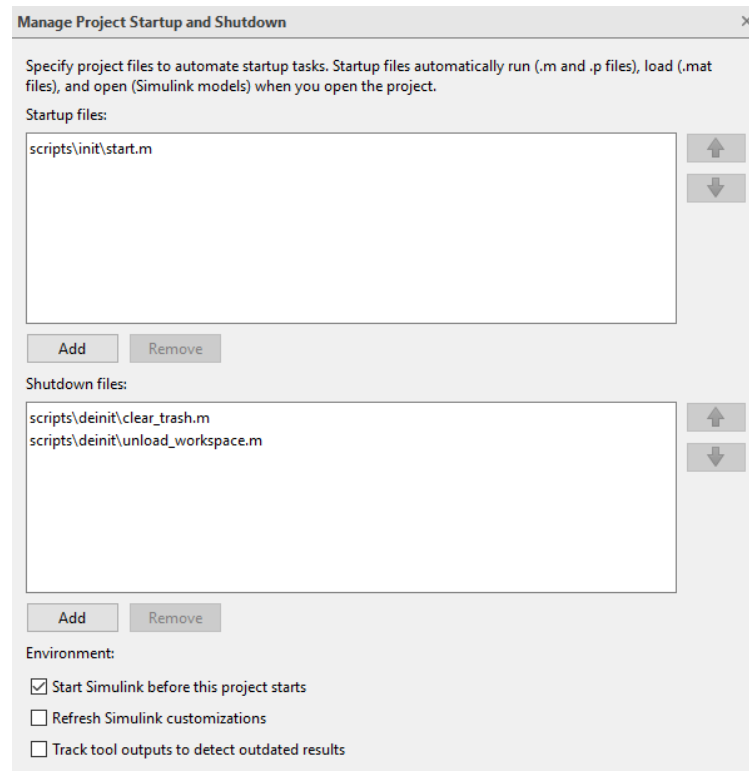


Рис. 3. Запуск файлов инициализации и деинициализации

6. Тесты системы

Проведем тестирования для Наблюдателя Люенбергера (рис. 4). Перед тем как сделать обвязку, определим необходимые для нас тесты и проверки. Пусть на вход $u(t)$ приходит значение в виде:

- Горизонтальная линия
- Треугольник
- Трапеция

Проверка определяется идеальными выходами с модели, полученными при моделировании всей системы (т.е. имеется готовый файл `x_ObserverIdealOut.mat`). Сделаем обвязку для этой модели (рис. 5) с помощью команды Test Harness (появляется при нажатии ПКМ).

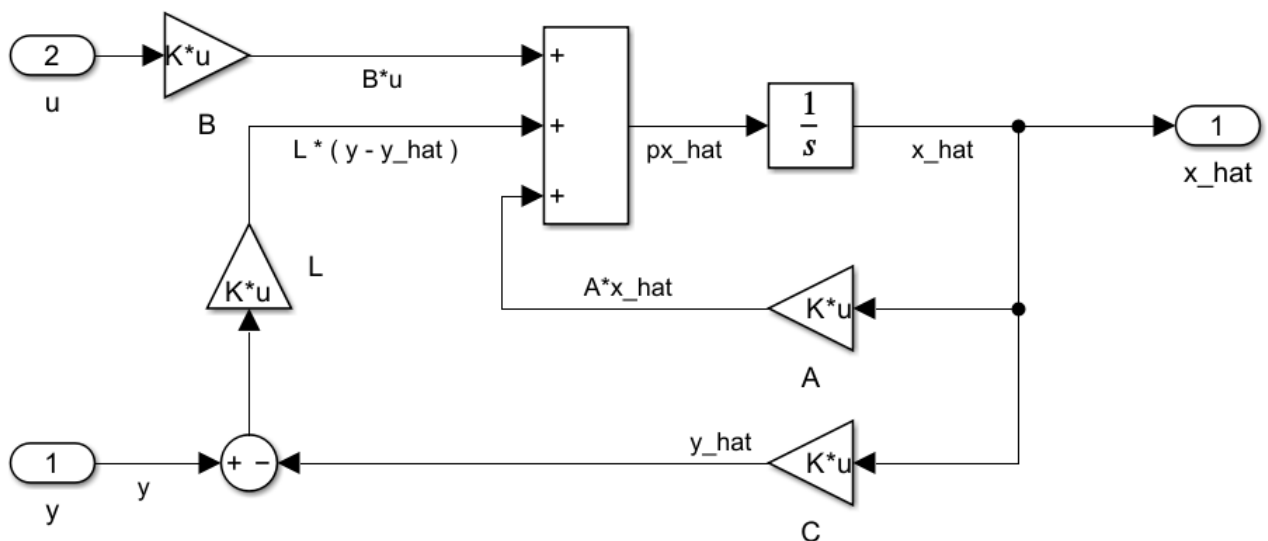


Рис. 4. Модель наблюдателя Люенбергера

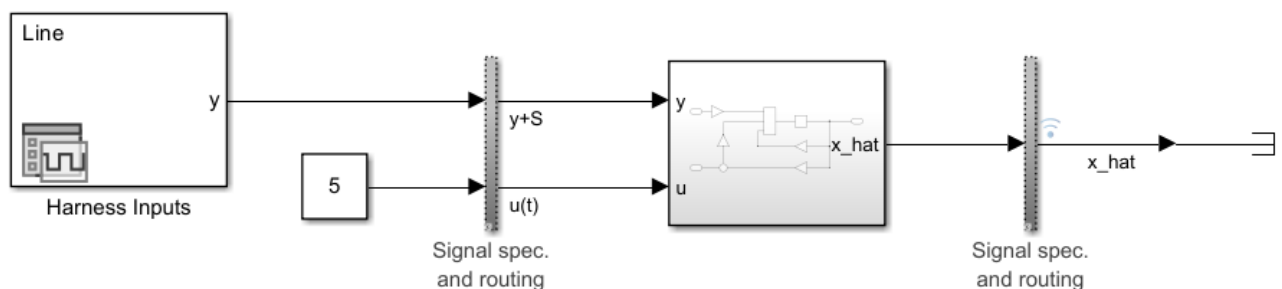


Рис. 5. Обвязка для наблюдателя Люенбергера

Настроим указанные тесты в параметрах входных сигналов в обвязке (блок Harness Inputs) (рис. 6).

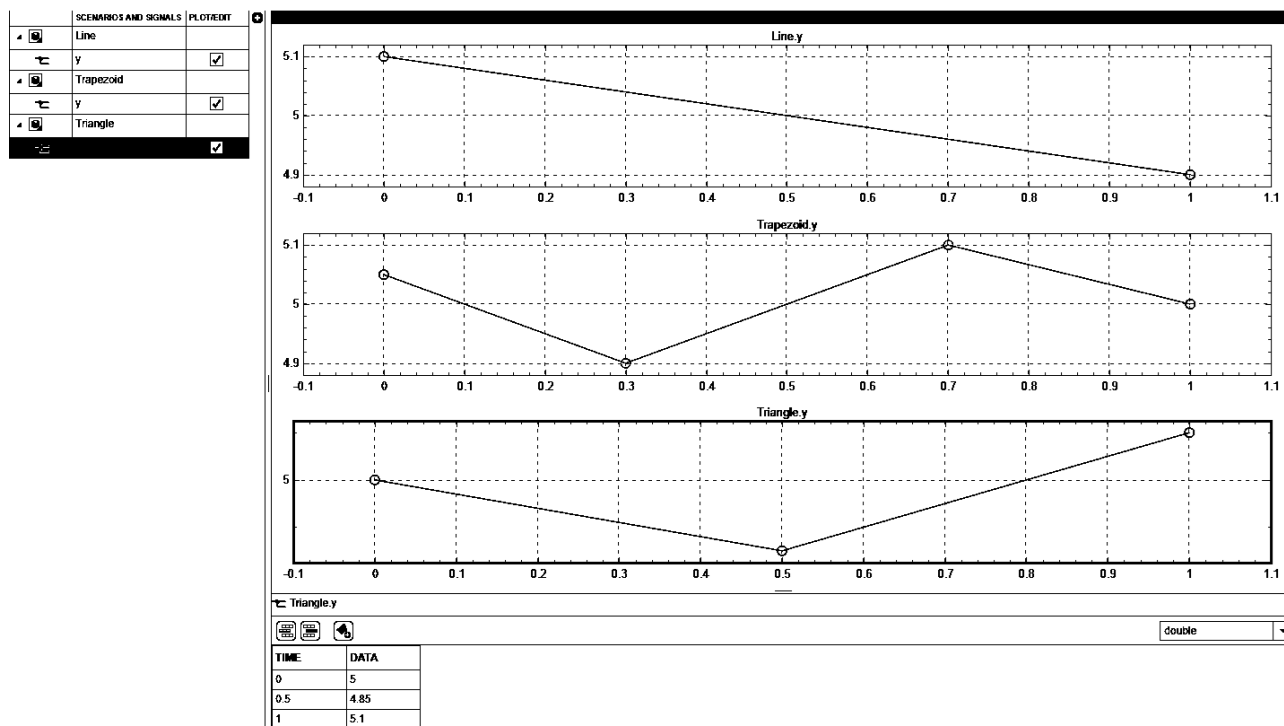


Рис. 6. Тесты к наблюдателю Люенбергера

Сохраняем файл и переходим в раздел Simulink Test Manager. В нём настраиваем все три теста и выбираем файл с сигналом сравнения как базовую линию, назначаем отклонения от этой базовой линии (рис. 7).

После начала тестирования получаем следующую картину (рис. 8).

Filter tests by name or tags, e.g. tags: test

- Observer_Test_results
 - Tests
 - Line test
 - Triangle test
 - Trapezoid test

Line test Enabled

Observer_Test_results > Tests > Line test

Baseline Test

☐ Create Test Case from External File

TAGS

DESCRIPTION

REQUIREMENTS

SYSTEM UNDER TEST*

Model: Observer_Harness

TEST HARNESS

SIMULATION SETTINGS AND RELEASE OVERRIDES

PARAMETER OVERRIDES

CALLBACKS

INPUTS*

SIMULATION OUTPUTS

CONFIGURATION SETTINGS OVERRIDES

BASELINE CRITERIA*

☐ Include baseline data in test result

SIGNAL NAME	ABS TOL	REL TOL	LEADING TOL	LAGGING TOL
<input checked="" type="checkbox"/> x_ObserverIdealOut.mat	5	10.00%	5	5

PROPERTY VALUE

PROPERTY	VALUE
Name	Line test
Type	Baseline Test
Model	Observer_Harness
Simulation Mode	[Model Settings]
Location	C:\Users\Tik\Desktop\Workl...
Enabled	<input checked="" type="checkbox"/>
Hierarchy	Observer_Test_results > Te...
Taos	Type comma or space separat...

Рис. 7. Настройка базовой линии, параметров проверки, кейсов и обвязки

NAME	STATUS
Results: 2024-May-23 04:13:05	3
Tests	3
Line test	
Baseline Criteria Result	
x_hat	
x_hat(1)	
x_hat(2)	
x_hat(3)	
Baseline (x1.mat)	
Sim Output (Observer_Hame	
Triangle test	
Baseline Criteria Result	
x_hat	
x_hat(1)	
x_hat(2)	
x_hat(3)	
Sim Output (Observer_Hame	
Trapezoid test	
Baseline Criteria Result	
x_hat	
x_hat(1)	
x_hat(2)	
x_hat(3)	
Sim Output (Observer_Hame	

Рис. 8. Результаты тестов

Наша программа не прошла проверку по вторым и третьим переменным состояниям ни в одном из тестов, однако отработала значение по положению. Однако это скорее исключение, поскольку было малое время симуляции и ошибка не успела вырасти из области перерегулирования. При том стоит учитывать сложности с расчетом параметров наблюдателя, величину помехи на выходе $y(t)$. Пример вывода сравнения графиков для требования «линия» показан на рисунках 9 и 10 – положение и скорость соответственно.

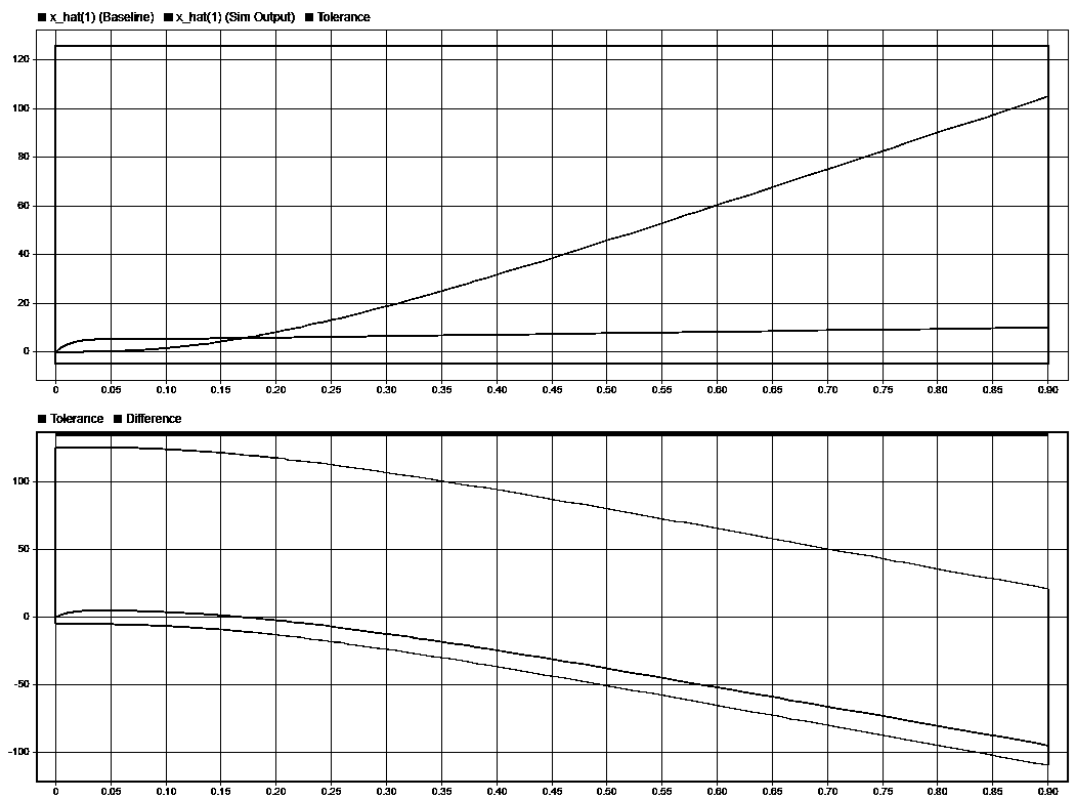


Рис. 9. Сравнение по положению при тесте линия

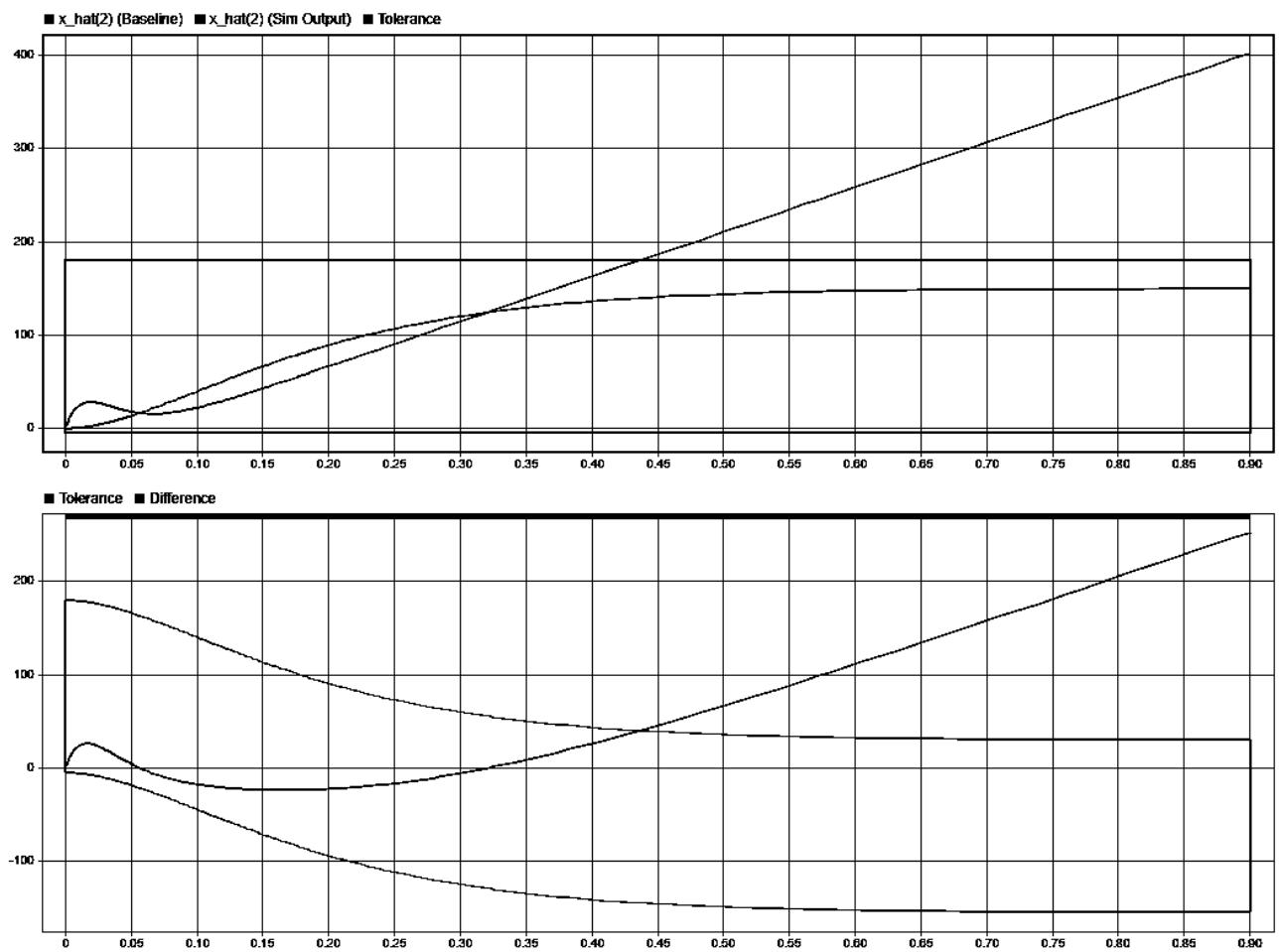


Рис. 10. Сравнение по скорости при тесте линия

7. Настройка файла README

Файл README выполняется на языке разметки Markdown. Далее представлен код, который отражает внутреннюю структуру этого файла.

```
# Theory-of-Automatic-Control

## Лабораторные работы номер 6 и 7 в рамках курса нелинейного ТАУ



---



### В репозитории отражены

<ul>
  <li> Регулятор на нечеткой логике ( Fuzzy logic controller) </li>
  <li> LQR-контроллер ( Logic controller) </li>
  <li> Наблюдатель Люенбергера ( Luenberger Observer) </li>
  <li> Фильтр Калмана ( Kalman Filter) </li>
  <li> Логическое управление ( Logic control )</li>
  <li> Тесты к системе</li>
</ul>



---



### Структура репозитория

<ul>
  <li><b>./data </b> - директория с файлами формата .m, в которых записаны константы и расчётные значения для каждой из моделей </li>
  <li><b>./models </b> - директория с моделями для каждого из реализуемого блока</li>
  <li><b>./scripts </b> - директории со скриптами инициализации и деинициализации при запуске проекта </li>
  <li><b>./tests </b> - директория с файлами для тестирования системы </li>
  <li><b>Report_6 </b> - отчёт по 6-ой лабораторной</li>
  <li><b>Report_7 </b> - отчёт по 7-ой лабораторной (включает подробное описание всех разработанных моделей)</li>
</ul>



---



### Запуск

<h4>Для успешного старта изучения и эксплуатации разработанных программ, необходимо первоочередно запустить проект project_Lab-7.prj в среде разработки Matlab. После запуска проекта произойдет инициализация проектов, после чего Вы можете обращаться к необходимым Вам частям данного репозитория</h4>
```


Приложение

1. Публичный репозиторий для лабораторных по ТАУ // GitHub URL:
<https://github.com/RiXenGC/Theory-of-Automatic-Control>