

СЪДЪРЖАНИЕ

СЪДЪРЖАНИЕ.....	1
ВЪВЕДЕНИЕ.....	3
ПЪРВА ГЛАВА	5
1. РАЗРАБОТВАНЕ НА ГРАФИЧНИ ПОТРЕБИТЕЛСКИ	
ИНТЕРФЕЙСИ В СРЕДА НА MATLAB.....	5
1.1. Въведение в GUIDE	5
1.1.1 Стартиране на GUIDE.....	5
1.1.2 Палети с елементи.....	7
1.1.3 Инструменти.....	9
1.1.4 Стандартни менюта.....	9
1.1.5 Работно поле	10
1.2. Създаване на интерфейс с GUIDE.....	10
1.2.1 Дизайн на интерфейса	10
1.2.2 Настройка параметрите на елементите от интерфейса	11
1.2.3 Програмиране на функциите на интерфейса.....	15
1.2.4 Базови функции на интерфейса	15
1.2.5 Добавяне на код в М – File –а на интерфейса	16
ВТОРА ГЛАВА.....	18
2. ГРАФИЧЕН ИНТЕРФЕЙС ЗА АНАЛИЗ НА ДИНАМИЧНИ	
СИСТЕМИ.....	18
2.1. Въведение.....	18
2.2. Основен графичен интерфейс	24
2.3. Прозорец „Типови динамични звена във времевата област”.....	26

2.4.	Прозорец „Типови динамични звена от втори ред във времевата област”	33
2.5.	Прозорец „Типови динамични звена в честотната област” ..	36
2.6.	Прозорец „Типови динамични звена от втори ред в честотната област”	39
2.7.	Прозорец „Типови динамични звена с чисто закъснение” ...	41

ТРЕТА ГЛАВА.....45

3. ГРАФИЧЕН ИНТЕРФЕЙС ЗА ИЗСЛЕДВАНЕ НА ТИПОВИ ЗАКОНИ ЗА РЕГУЛИРАНЕ45

3.1.	Въведение.....	45
3.1.1	П регулатор	48
3.1.2	ПИ регулатор	49
3.1.3	ПИД регулатор	50
3.2.	Графичен интерфейс в средата на Matlab за настройка и анализ на системи с П, ПИ и ПИД регулатори.	51

ЗАКЛЮЧЕНИЕ62

ЛИТЕРАТУРА.....64

ПРИЛОЖЕНИЕ 165

ПРИЛОЖЕНИЕ 2.....66

ВЪВЕДЕНИЕ

Програмната среда *Matlab*, заедно с допълнителните библиотеки от качествено разработени функции за анализ и синтез на системи за управление, се е утвърдила през последните години като учебен софтуер за проектиране и изследване на системи за автоматично управление.

Програмния продукт *Matlab* предлага два известни подхода за работа. При първия се активират конкретни функции според дефинирания им синтаксис и се получават резултати, които потребителят преобразува в подходяща по формат информация за използване от следващите функции и т.н., докато се реши определена задача. Когато трябва да се прилагат известни функции в добре познати алгоритми, голяма част от времето на потребителя се губи за обработване и прехвърляне на информация между използваните програми единици.

Другият подход представлява предлагане на завършен програмен продукт – потребителски интерфейс, който автоматизира решаването на типови проектантски задачи и спестява времето на потребителя за сметка на вложения труд от системния програмист при изготвянето на програмната система. В този случай ползвателят получава сравнително ограничени възможности в рамките на заложените опции на системата, но може да се концентрира върху същинската работа. Разработването на потребителски интерфейси за решаване на конкретни задачи е удобно средство за обучение.

Целта на дипломната работа е чрез разработка на система от потребителски графични интерфейси:

- да се улесни и подобри работата на студентите, обучаващи се по дисциплината Теория на Управлението;
- да се осигури бързо запознаване с материала;
- да се акцентира върху разбирането на материала;

- да се предоставят възможности за лесно съпоставяне на различни типови системи, както и влиянието на параметрите върху характеристиките на динамичните системи.

ПЪРВА ГЛАВА

1. РАЗРАБОТВАНЕ НА ГРАФИЧНИ ПОТРЕБИТЕЛСКИ ИНТЕРФЕЙСИ В СРЕДА НА MATLAB

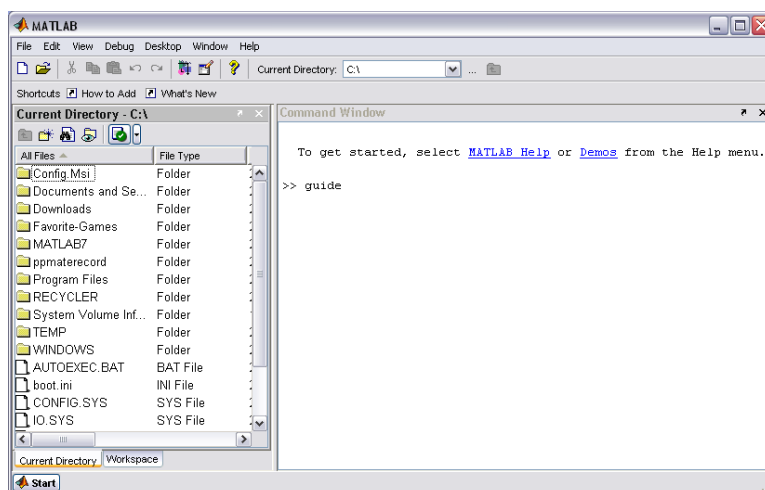
1.1. Въведение в GUIDE

GUIDE (*Graphical User Interface Development Environment*) е модул на **Matlab**, представляващ среда за разработка на графични потребителски интерфейси [1].

1.1.1 Стартиране на GUIDE

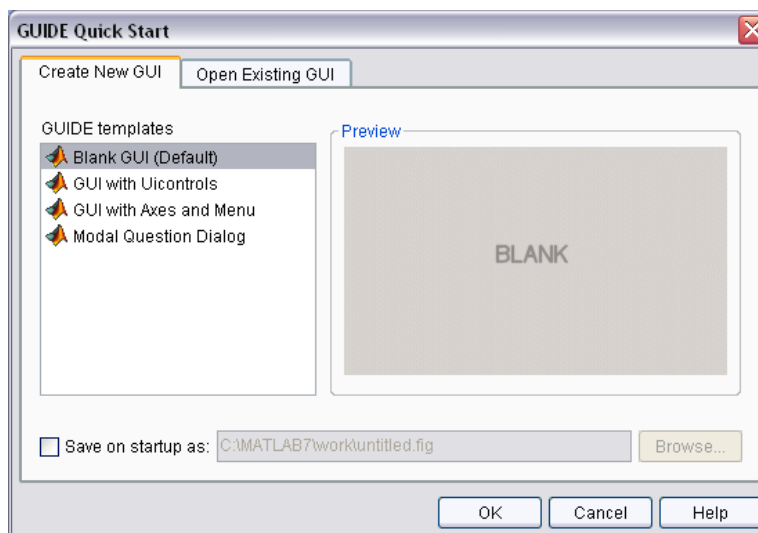
Съществуват няколко начина за стартиране:

- от командния ред, като се напише **guide** - фиг. 1.1
- от **Start** меню, като се избере **Matlab > Guide (GUI Builder)**
- от меню **File**, като се избере **New > GUI**



Фиг. 1.1. Главен прозорец на **Matlab**

Независимо от начина на извикване на модула, при стартирането му се появява меню **GUIDE Quick Start**. Това меню служи за избор на вида стартова бланка за нов интерфейс: **Create New GUI** или избор на вече създаден такъв: **Open Existing GUI** – фиг. 1.2.

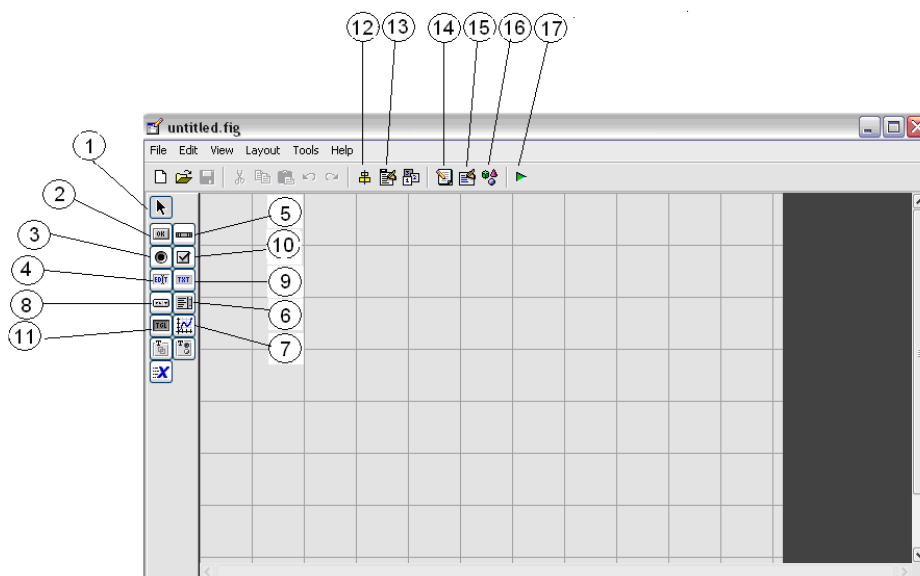


Фиг. 1.2. Меню **GUIDE Quick Start**

В меню **Create New GUI** потребителят има възможност да избира между четири възможни шаблона:

- **Blank GUI (Default)** – отваря празно поле за създаване на интерфейс без предварително заложен елемент
- **GUI with Uicontrols** – отваря шаблон с предварително заложен елемент подходящ за изчислителен интерфейс
- **GUI with Axes and Menu** – отваря шаблон с предварително заложен елемент, между които **Axes** подходяща за интерфейс, изобразяващ графична информация
- **Modal Question Dialog** – отваря шаблон за разработка на диалогов прозорец за съобщение или въпрос към потребителя

След направения избор се отваря главния прозорец на **GUIDE**, който е и самата среда за създаване на потребителските интерфейси – фиг. 1.3.



Фиг. 1.3. Главен прозорец на *GUIDE*

1.1.2 Палети с елементи

Маркери от 1 до 11 върху фиг. 1.3 показват полетата с елементи, които могат да се използват при създаването на нов потребителски интерфейс.

1. **Select Tool** – Осигурява избор на елемент от интерфейса.

2. **Push Button** – Бутон, който се възвръща след натискането му. Този елемент извиква *Callback* функцията си при натискане с мишката от страна на потребителя. По този начин елементът предизвиква изпълнението на операция от страна на интерфейса.

3. **Radio Button** – Бутон, който остава в едно от двете си крайни състояния след въздействие с мишката върху него. Елементът има дискретно действие. Логическите му стойности за *value* са нула (изключен) или единица (включен). Обикновено се използва за избор на опция за работа на интерфейса.

4. **Edit Text** – Поле, в което текстът може да се редактира от потребителят. Използва се за въвеждане на текст или стойност. Останалите елементи на интерфейса имат достъп до съдържанието на полето

посредством променливата ***handles.editxx, string. Callback*** функцията на обекта се активира при потвърждаването на въведения текст с натискане на клавиша ***Enter***.

5. ***Slider*** – Позволява въвеждане на цифрова стойност в определен интервал, чрез преместване на слайда с мишката. Текущата му стойност се пази в променливата ***handles.sliderxx, value***. Обновяването на променливата се извършва в момента на изпускане на слайда от мишката.

6. ***List Box*** – Изобразява лист с опции, които се обслужват от оператор ***Case***. Потребителят избира опцията и тя се активира в момента на изпускане на елемента от мишката, като се позволява избор на повече от една опция в листата.

7. ***Axes*** – Модул, в който се извеждат графики или изображения. Извикването му се осъществява, чрез присвояване на ***handles*** структурата на избрания модул в променливата ***Axes = handles.axesxx***.

8. ***Pop – Up Menu*** – Падащо меню, в което потребителят може да избира едно от предварително зададените полета.

9. ***Static Text*** – Текстово поле, което не позволява да бъде променяно от потребителя по време на работа на интерфейса. Текстът в него се задава предварително. При работа на интерфейса, промени на този текст могат да правят функции или елементи от интерфейса. Това се осъществява чрез създаване на ***Callback*** функция на елемента ***staticxx_Callback*** и присвояване на стрингова променлива на ***handles.staticxx, string***.

10. ***Check Box*** – Този елемент е подобен на елемента ***Radio Button***. Отново имаме дискретно действие и логическата му стойност ***handles.checkboxxx, value*** е нула (избран) или единица (свободен).

11. ***Toggle Button*** – Бутон оставащ в едно от двете крайни положения до повторно избиране с мишката от страна на потребителя. Използва се за активиране на функция на интерфейса, работеща докато елемента е във включено състояние. Този елемент също има дискретно действие и

логическата му стойност *handles.togglebuttonxx, value* е нула (включен) или единица (изключен).

1.1.3 Инструменти

Маркер 1 маркери от 12 до 17 (фиг. 1.3.) показват инструментите, осигуряващи настройката на работата и функциите на елементите и интерфейса като цяло:

1. *Select Tool* – осигурява избор на елемент от интерфейса.
12. *Alignment Tool* – подравняване елементите на интерфейса.
13. *Menu Editor* – създаване и редакция на менютата в интерфейса.
14. *M – File Editor* – създаване и редакция на *M–File* с функции на интерфейса.
15. *Property Inspector* – редактиране на опциите на елементите на интерфейса.
16. *Object Browser* – осигурява навигацията между елементите на интерфейса чрез търсене по име.
17. *Run Button* – Стартира изпълнението на интерфейса.

1.1.4 Стандартни менюта

Посредством стандартните менюта от главния прозорец на *GUIDE*, разработващият интерфейс има достъп до следните функции:

- *File* – Осигурява запазване на файла на интерфейса, редакция на настройките на *GUIDE*, както и изход от приложението.

- **Edit** – Осигурява редакция на интерфейса със стандартни за **Windows** функции, като **Cut, Copy, Paste, Undo, Redo** и др.
- **View** – Осигурява избор на видимите в главния прозорец на **GUIDE** модули и инструменти, извикване на **Callback** функциите на елементите съставлящи интерфейса, както и тези, които се отнасят за него.
- **Layout** – Осигурява редакция на графичното поведение на елементите в създавания интерфейс.
- **Tools** – Функцията му е да осигурява дублиране на инструментите от лентата с инструменти **Tool Bar** на **GUIDE**.
- **Help** – Осигурява помощ при работата с **GUIDE**.

1.1.5 Работно поле

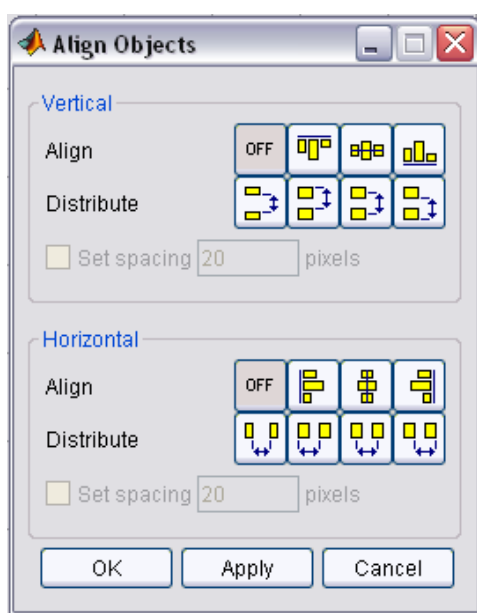
Работното поле на **GUIDE** (фиг. 1.3) всъщност представлява основата на интерфейса, който се разработва в момента. Размерът на това поле дефинира размера на самия интерфейс. Работното поле на **GUIDE** разполага с **Rulers** – линейки, които показват размера на бъдещият интерфейс и **Grid** – решетка от квадратни модули със задаваща се стъпка на квадратите, по която се подреждат елементите с цел по-лесна работа с тях.

1.2. Създаване на интерфейс с GUIDE

1.2.1 Дизайн на интерфейса

След предварително определяне на външния вид и необходимите функционални възможности на интерфейса, се пристъпва към неговото създаване. Елементите се поставят на местата си, след което ако е

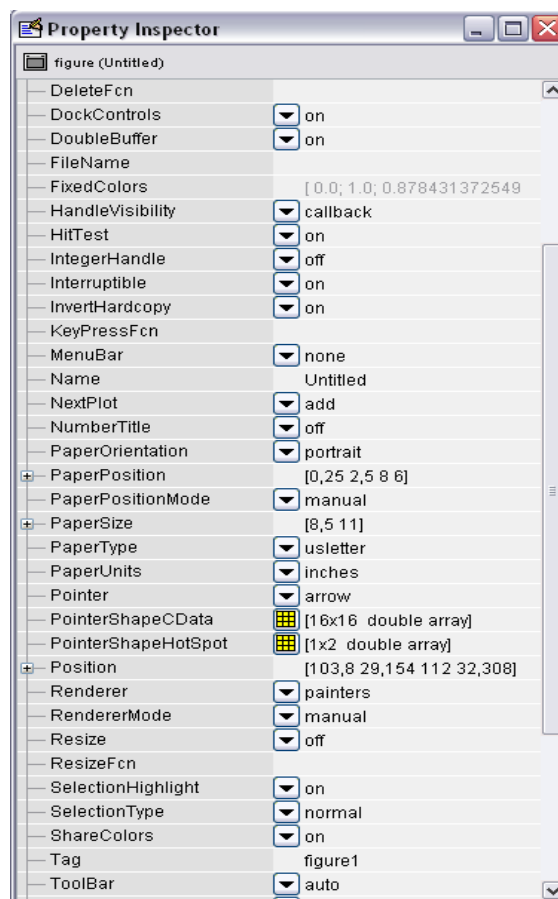
необходимо се подравняват чрез *Align Objects* (фиг. 1.4). Тази опция позволява вертикално и хоризонтално подравняване на обектите, както и задаване на константно разстояние между тях, фиксиращо се от различни техни реперни точки. Разстоянието между обектите може да е произволно, отговарящо на изискванията на разработчика на интерфейса и се задава в пиксели.



Фиг. 1.4. Прозорец за подравняване на елементите

1.2.2 Настройка параметрите на елементите от интерфейса

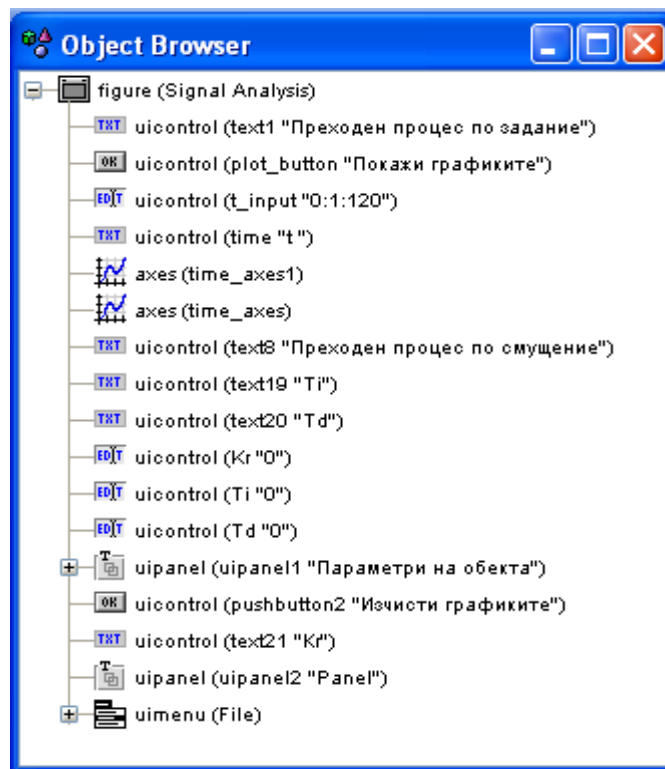
За този вид настройка се използва инструмента *Property Inspector*. На фиг. 1.5 е показан диалоговият прозорец на този инструмент.



Фиг. 1.5. Диалогов прозорец на *Property Inspector*

Този инструмент **позволява** настройката на всички типови характеристики на използвания елемент. Дефинира името на *Callback*, *Create* и *Delete* функциите на елемента, името на самия елемент, потребителските данни, *Tooltip String*, интервалът на стойности, които може да придобива обектът и др. С него се задава стойността и статуса, които елементът има по подразбиране .

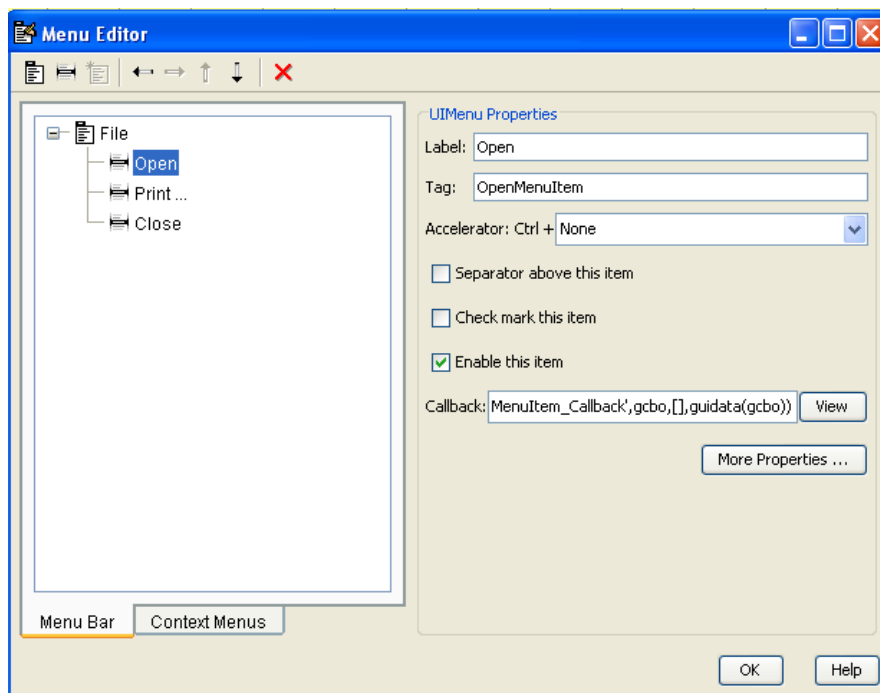
За по-лесна навигация между различните елементи в интерфейса се използва *Object Browser*. Той позволява извикването на *Property Inspector* да се извършва само с избиране на името на елемент от списъка на *Object Browser*. Диалоговия прозорец на този инструмент е показан на фиг. 1.6.



Фиг. 1.6. Диалогов прозорец на *Object Browser*

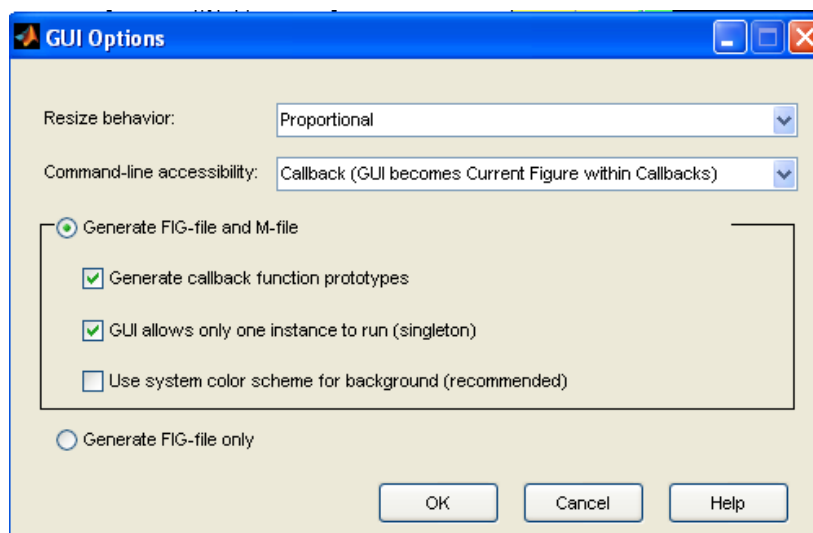
Tab Order Editor е инструмент позволяващ настройка на реда, по който елементите се извикват при навигация, извършвана с *Tab* бутона от клавиатурата, в интерфейса. Тази възможност допълнително подобрява скоростта на работа на потребителя с интерфейса и го прави по-удобен за употреба.

Menu Editor представлява редактор за създаване на менютата в диалоговия прозорец на интерфейса. Позволява избор на броя, имената и опциите на гореспоменатите менюта. Видът на диалоговия прозорец на този инструмент е показан на фиг. 1.7.



Фиг. 1.7. Редактор за създаване на менюта

GUI Options – този прозорец се използва за допълнителни настройка на конструктора за графични интерфейси – фиг. 1.8.



Фиг. 1.8. Настройка параметрите на редактора *GUIDE*

Първият разкриващ се списък е ***Resize behavior*** – позволява да се избере вариант на поведение на приложението, при опит потребителят да измени размера на прозореца. Възможните варианти са три:

- ***No-resizable*** - прозорецът не допуска изменение на своите размери.
- ***Proportional*** - изменение на размерите на прозореца се допуска, като оразмеряването става автоматично и пропорционално
- ***User-specified*** - допуска се изменение на размера на прозореца, като оразмеряването се възлага на програмата на потребителя, реагираща на събитието ***Resize*** (свойството ***ResizeFcn*** се явява указател на тази функция).

1.2.3 Програмиране на функциите на интерфейса

За да се покрият функционалните изисквания на интерфейса е необходимо да се зададат функциите на неговите елементи. Това се извършва посредством инструмента ***M-File Editor*** на ***Matlab***. Той осигурява удобство при въвеждането на кода на функциите. При първото стартиране на редактора, ***Matlab*** генерира служебен код. Той съдържа само прототипите на ***Callback*** и ***Create*** функциите на елементите. За облекчава работата при програмирането на интерфейса прототипите съдържат коментари с някои от променливите на съответната функция.

1.2.4 Базови функции на интерфейса

Opening Function - тази функция се генерира автоматично от ***GUIDE***. Тя отговаря за зареждането на интерфейса и за задачите, които

трябва да се изпълнят преди контролът да премине в ръцете на потребителя. В този тип функция не могат да се декларират глобални променливи.

Create Function – този тип функция, която се генерира автоматично само за някои от елементите на интерфейса включени в палетата на **GUIDE**. Тя отговаря за тяхното създаване и начин, по който изглеждат – цвят на фона на интерфейса, цвят на фона и шрифт на текста и др. Тази функция за разлика от **Opening Function** не е само една - тя се появява в него толкова пъти, колкото отделни елементи имат нужда от нея. Може да се създаде такава функция на елемент, за която **GUIDE** не създава такава по подразбиране.

Callback Function – тази функция **Matlab** генерира за всички активни елементи на интерфейса (слайдери, текстови кутии, бутони и др.) За неактивните елементи (статични текстове, рамки) при нужда може да се добави такава. Тази функция отговаря за извършването на операциите на всеки от елементите при въздействие върху тях.

1.2.5 Добавяне на код в М – File –а на интерфейса

Възможно е добавянето на код във вид на собствени функции, които по тип са различни от по-горе дефинираните. Необходимо е съобразяване със синтаксиса на **Matlab** и с някои характерни за файловете на интерфейса особености. Такива особености са:

- Не може да се добавя код, който да не е част от функция.
- Краят на всяка функция е начало на следващата или край на кода като цяло.

- Функциите не могат да използват стойностите на глобални променливи, въпреки че променливите могат да се декларират като такива.
- Не е нужно да се дефинира дадена променлива преди да се използва, но задължително трябва да е инициализирана.

ВТОРА ГЛАВА

2. ГРАФИЧЕН ИНТЕРФЕЙС ЗА АНАЛИЗ НА ДИНАМИЧНИ СИСТЕМИ.

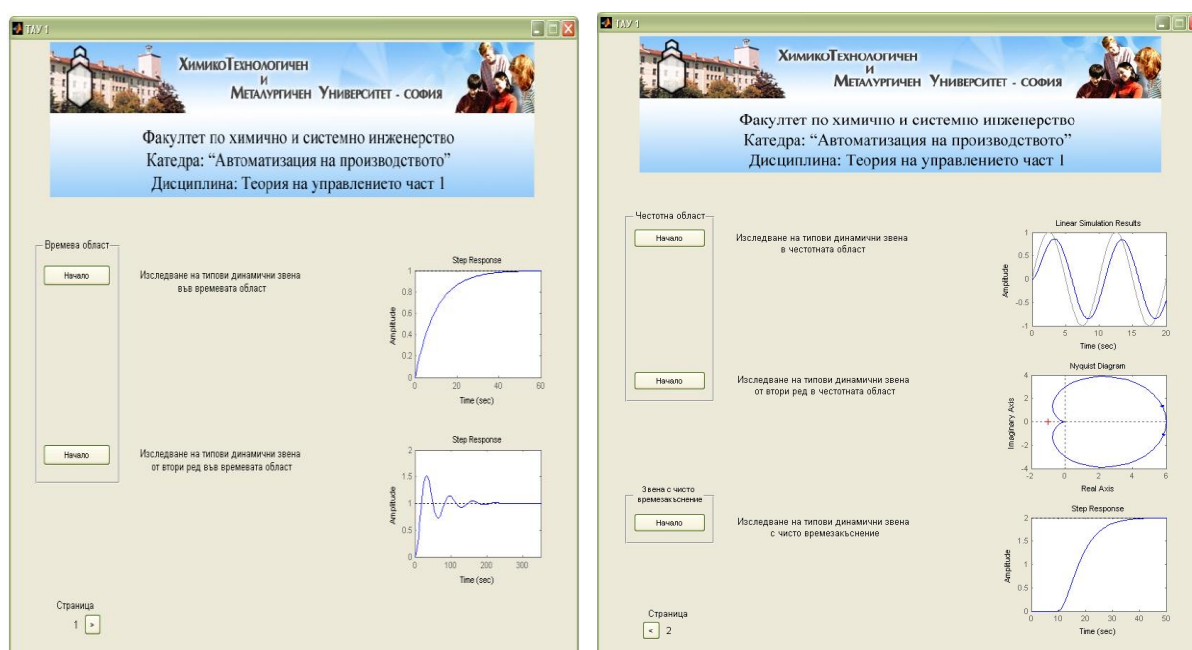
2.1. Въведение

При разработването на графичните интерфейси (GUI) за обучение по дисциплината Теория на Управлението, са използвани основно литературните източници [3-9], като излагането на материала е структурирано основно в два основни дяла: изследване на динамични системи във времевата област и изследване в честотната област. Последователно се разглеждат системи от първи ред, системи от втори ред и системи с чисто времезакъснение. Междувременно се представя и класификацията на типовете динамични звена като позиционни, интегриращи и диференциращи звена, а звената с чисто закъснение са представени отделно, като те могат да се изследват в комбинация и самостоятелно.

Предвидено е първоначално на студентите да се демонстрира пример с основните характеристики на системите, като те се показват при променящи се стойности на параметрите – предавателни коефициенти, времеконстанти и др. В последствие се предоставя на студентите да задават самостоятелно стойности на параметрите и да получават характеристиките на отделните типови системи с наслагване или отделно, като е предвидено да се сравняват и еднотипни характеристики на различни системи и еднотипни системи при вариране на параметрите им. Предвидени са възможности за разглеждане на детайли от графиките като

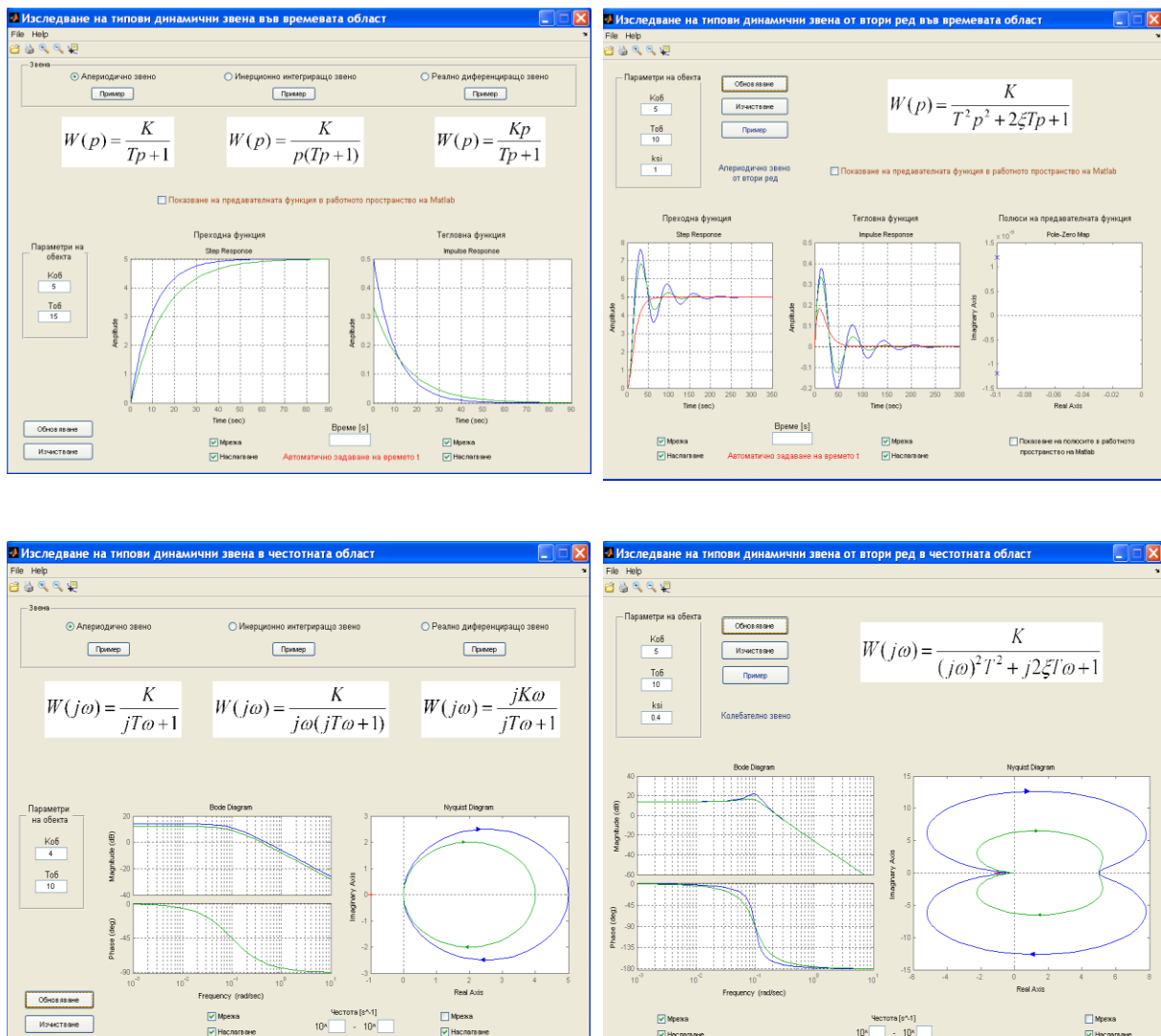
ZoomIn и **ZoomOut**, а също и автоматично генериране на параметрите на графиката или задаване на стойности от потребителя. Числени стойности от всички характеристики могат да се показват в отделен прозорец чрез посочване с мишката върху графиката. Предавателните функции, зададени от студентите могат да се показват и в работното пространство на **Matlab**, както и някои други основни характеристики като нули и полюси и др.

За създаването на системата са използвани девет графични файла и шестнайсет отделни графични интерфейса, които работят автономно, но са свързани помежду си в **основен графичен интерфейс**, състоящ се от два прозореца, показани на фиг. 2.1. Стартирането на системата се извършва като в работното пространство на **Matlab** се изпише **tau1** последвано от натискане на клавиш **Enter** от клавиатурата.

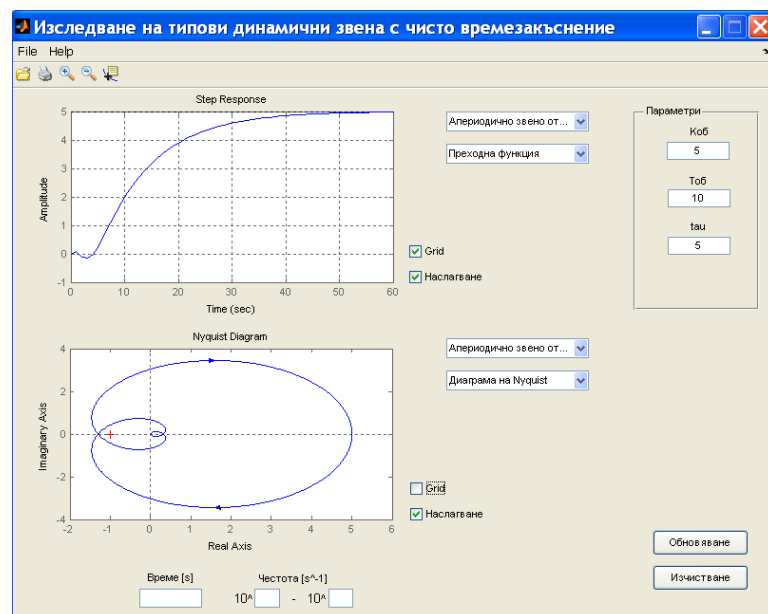


Фиг. 2.1. Основен графичен интерфейс

Основният графичен прозорец на интерфейса се състои от две страници, от които чрез съответни бутони се отварят пет допълнителни прозореца за анализ на конкретни системи във времевата и честотната области - фиг. 2.2 и системи с чисто закъснение – фиг. 2.3.

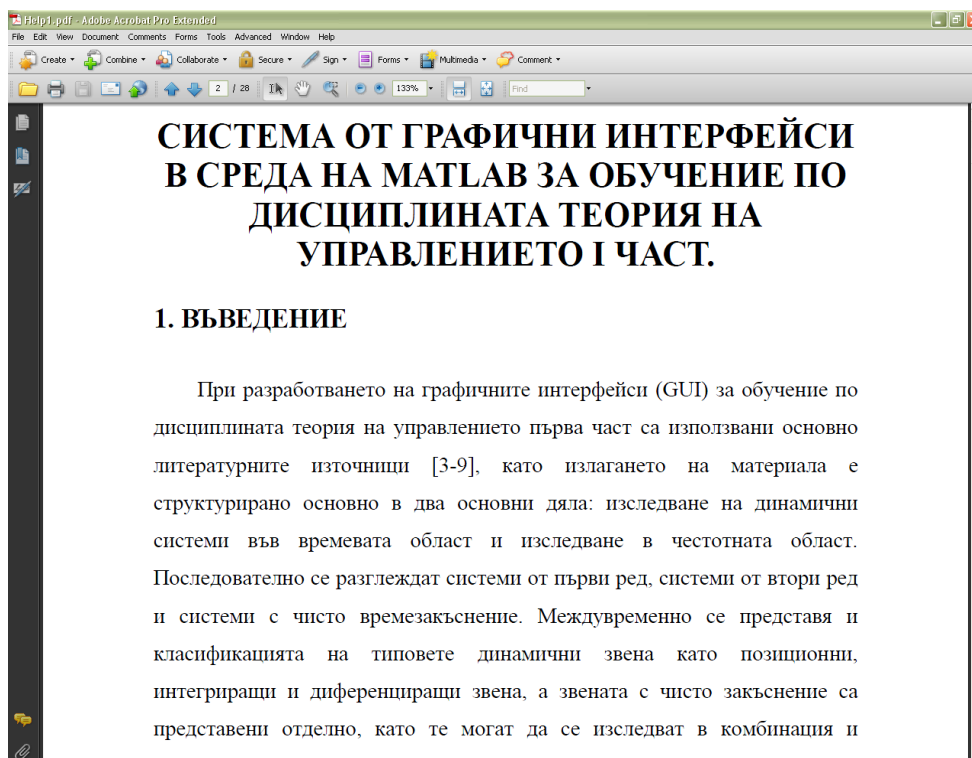


Фиг. 2.2. Прозорци за анализ на системи във времевата и честотната области



Фиг. 2.3. Прозорец за анализ на системи с чисто закъснение

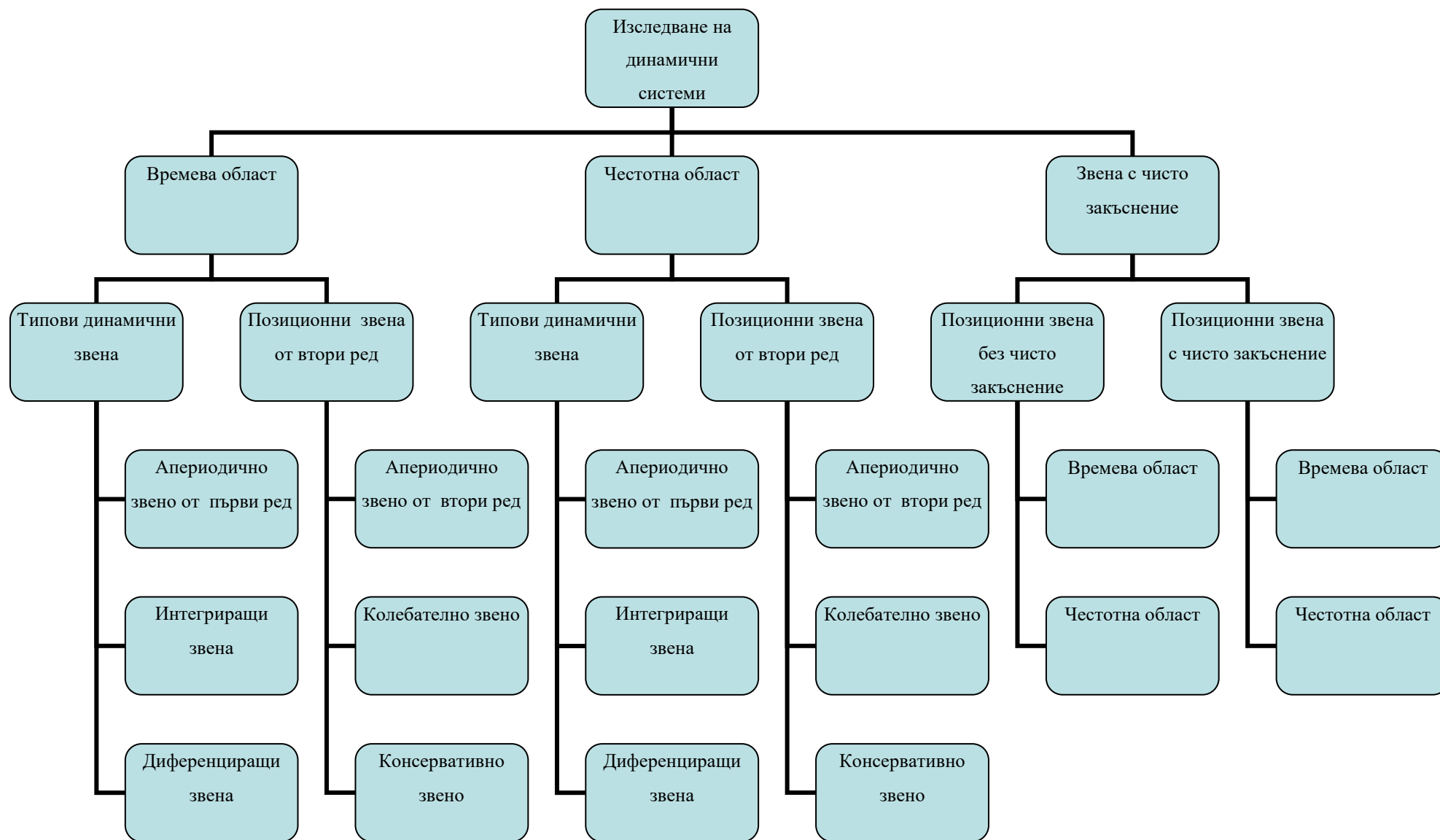
За всички прозорци са създадени еднотипни менюта: стандартните менюта *File* и *Help*, като подменюта на *File* са *Open*, *Print* и *Close*. Подменюта на *Help* са *About*, *GUI Help* и *Matlab Help*. При избор на *GUI Help* се отваря текст с описание на системата на български език – фиг. 2.4. Това до голяма степен ще улесни работата на студентите с графичния интерфейс. Също така са добавени и пет икони за бърз достъп от панела с инструменти. Те са: *Open*, *Print*, *ZoomIn*, *ZoomOut* и *DataCursor*. С *DataCursor* потребителят може да получи информация за числени стойности от графиките в произволна посочена точка.



Фиг. 2.4. Пример за *GUI Help*

Запознаването на студентите с изследването на динамичните системи започва с изследване във времевата област за системи от първи ред и системи от втори ред. Втората страница на основния интерфейс включва изследване на системи в честотната област и системи с времезакъснение. Функционалната схема на разработената система е представена на фиг. 2.5.

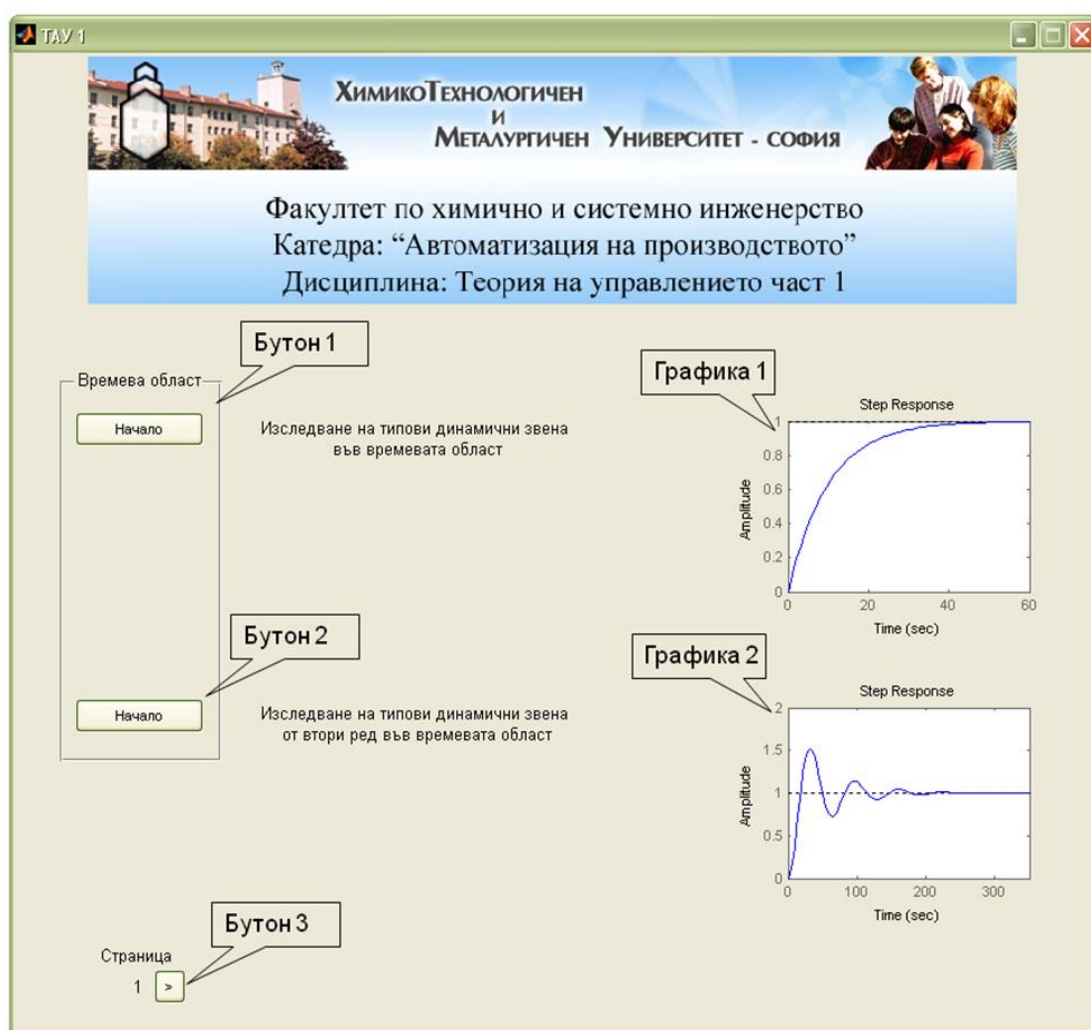
Последователно отделните възможности на системата ще бъдат разгледани детайлно, за да се представи пълната функционалност на системата.



Фиг. 2.5. Функционална схема на разработения графичен интерфейс

2.2. Основен графичен интерфейс

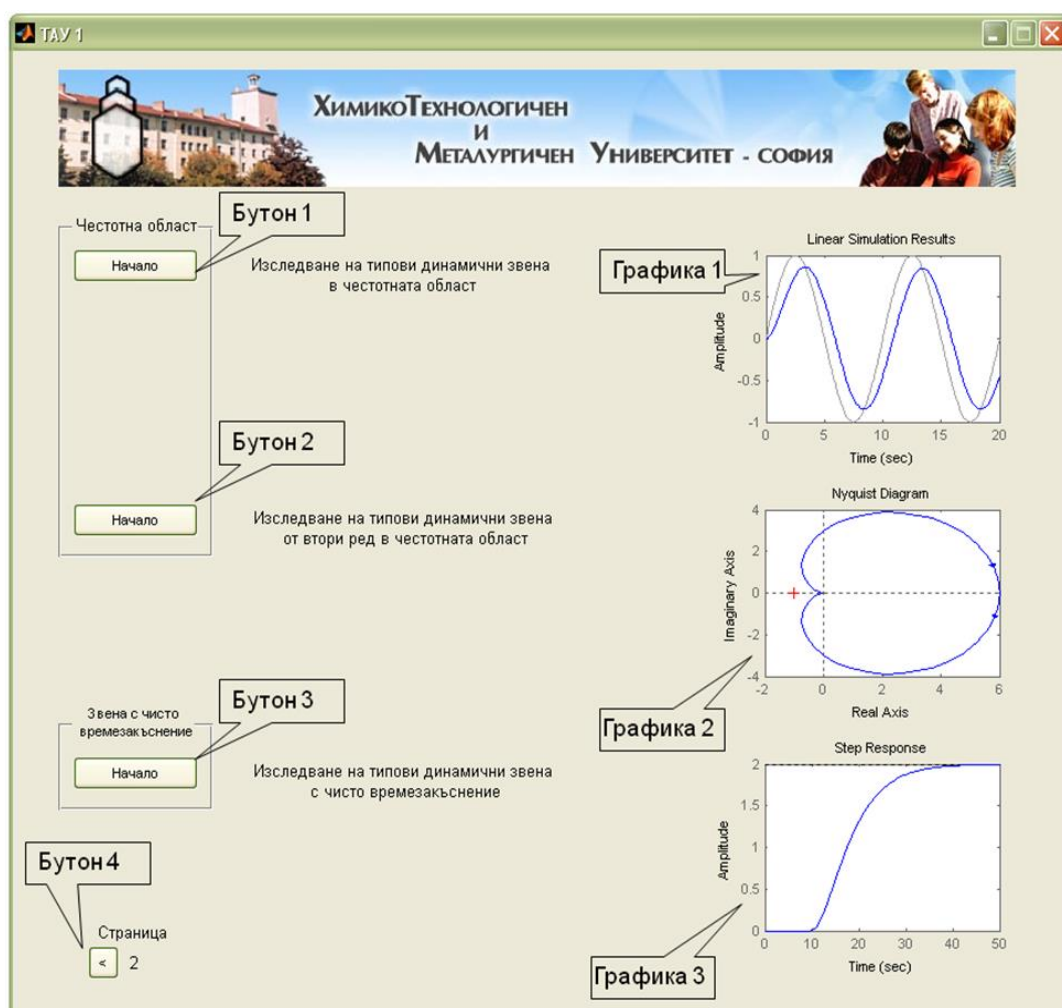
След стартирането на основния графичен интерфейс, в работното пространство на *Matlab* се показва главния прозорец - фиг. 2.6. Той е разделен на две страници, разграничавайки основните методи за изследване на динамичните системи времева и честотна област. В горната част е поместено логото на университета.



Фиг. 2.6. Главен прозорец – страница 1

Първоначалното запознаване на студентите с графичния интерфейс започва с изследване на системи от първи и втори ред във времевата

област. Чрез натискане на **Бутон 1** и **Бутон 2** се отварят съответните прозорци за изследване на стационарни системи във времевата област, като бутоните за стартиране са оградени в панел със заглавие **Времева област**. С цел насочване на студентите са изобразени преходните характеристики на типови динамични звена от първи (**Графика 1**) и втори ред (**Графика 2**). С натискане на **Бутон 1** в нов прозорец се стартира интерфейс за изследване на типови динамични звена във времевата област. С натискане на **Бутон 2** се стартира интерфейс за изследване на звена от втори ред във времевата област. С натискане на **Бутон 3** се преминава на втора страница на главния интерфейс, където е изследването на звената в честотната област и изследването на звена с чисто закъснение (фиг. 2.7).



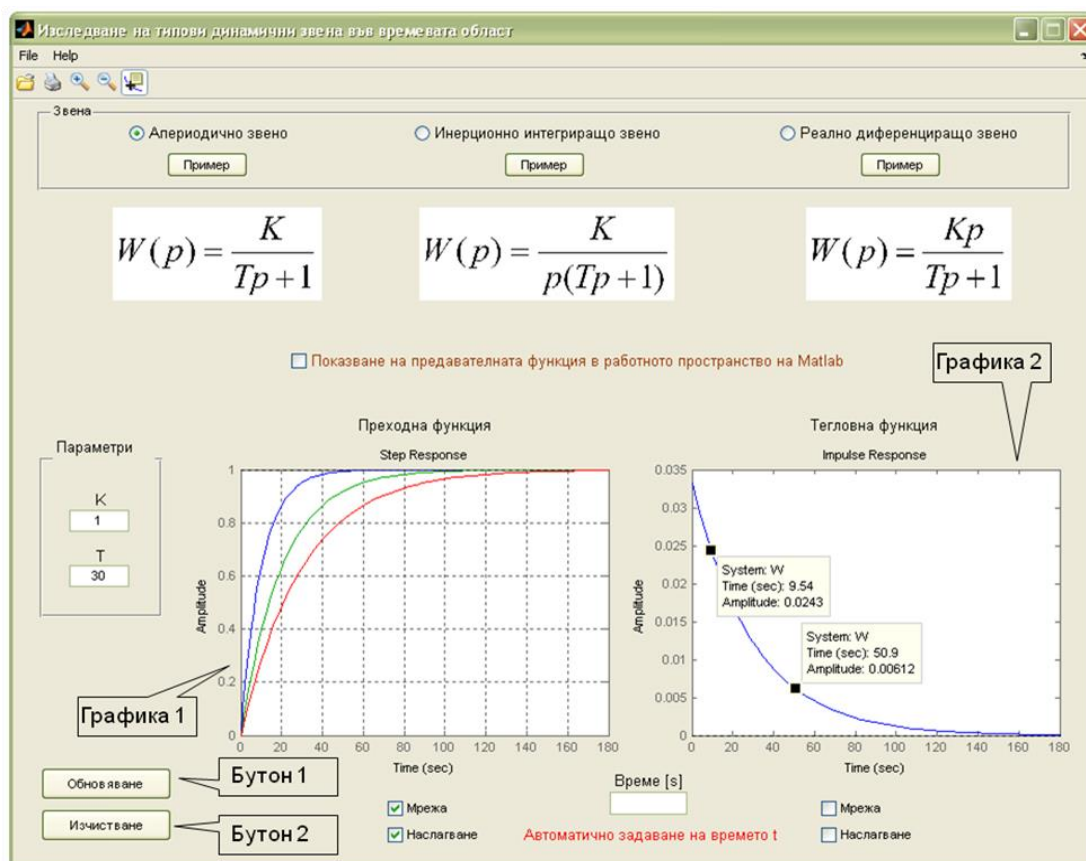
Фиг. 2.7. Главен прозорец – страница 2

Чрез натискане на **Бутон 1** и **Бутон 2** се отварят съответните прозорци за изследване на стационарни системи в честотната област. С натискане на **Бутон 3** се стартира интерфейс за изследване на типови динамични звена с чисто закъснение. С натискане на **Бутон 4** се преминава на първа страница на главния прозорец.

Чрез групирането на бутоните в панели се цели по-доброто структуриране на поднасяния материал, а графичните изображения целят по-лесното ориентиране, разграничаване и по-бързо възприемане от потребителя на конкретните части от материала. С натискане на **Бутон 1** в нов прозорец се стартира интерфейс за изследване на типови динамични звена във времевата област. С натискане на **Бутон 2** се стартира интерфейс за изследване на звена от втори ред във времевата област. С натискане на **Бутон 3** се преминава на втора страница на главния интерфейс, където е изследването на звената в честотната област и изследването на звена с чисто закъснение. Програмният код на първата страница от основния графичен интерфейс е представен в *Приложение 1*.

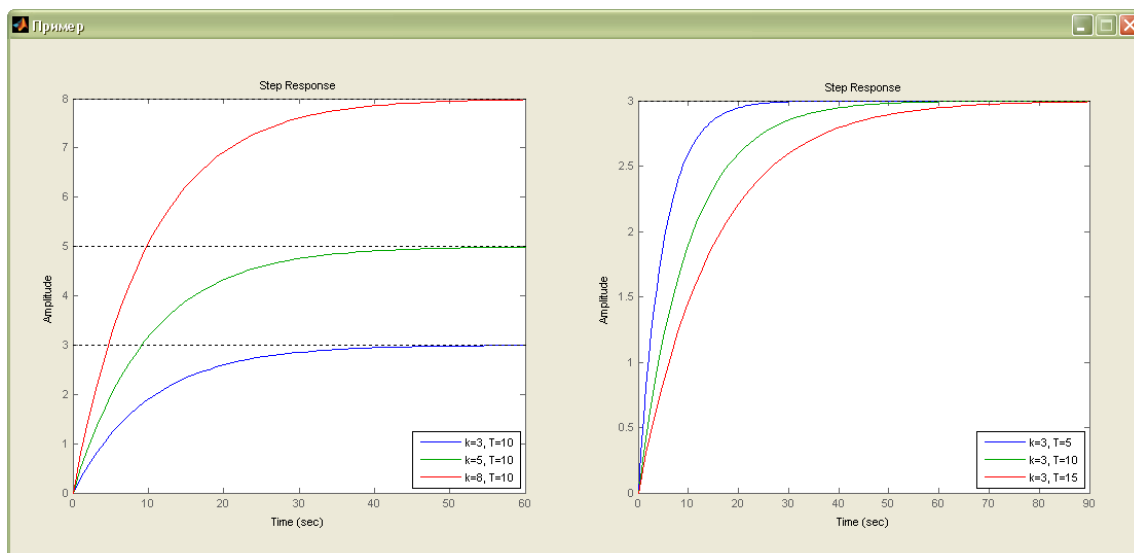
2.3. Прозорец „Типови динамични звена във времевата област”

Потребителят има възможност да избира между три типови динамични звена от радио бутоните поставени в панел със заглавие **Звена:** Апериодично звено от първи ред, Инерционно интегриращо и Реално диференциращо звено (фиг. 2.8).



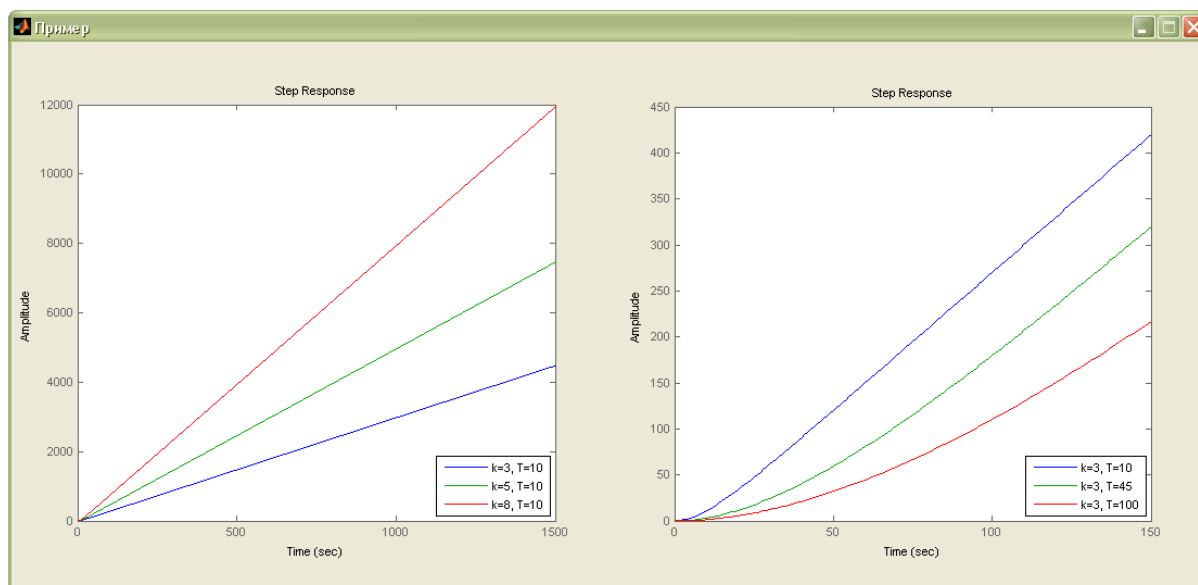
Фиг. 2.8. Прозорец „Изследване на типови динамични звена във времевата област”.

В панела са поместени и три бутона *Пример*. Целта е първоначално студентите да видят времевите характеристики на съответните динамични звена при различни стойности на параметрите им. Примерът за апериодично звено (фиг. 2.9) представлява две преходни характеристики, като при първата се варира предавателният коефициент k , а на втората - времеконстантата T . Целта е да се покаже по какъв начин тези два коефициента влияят на преходната характеристика.



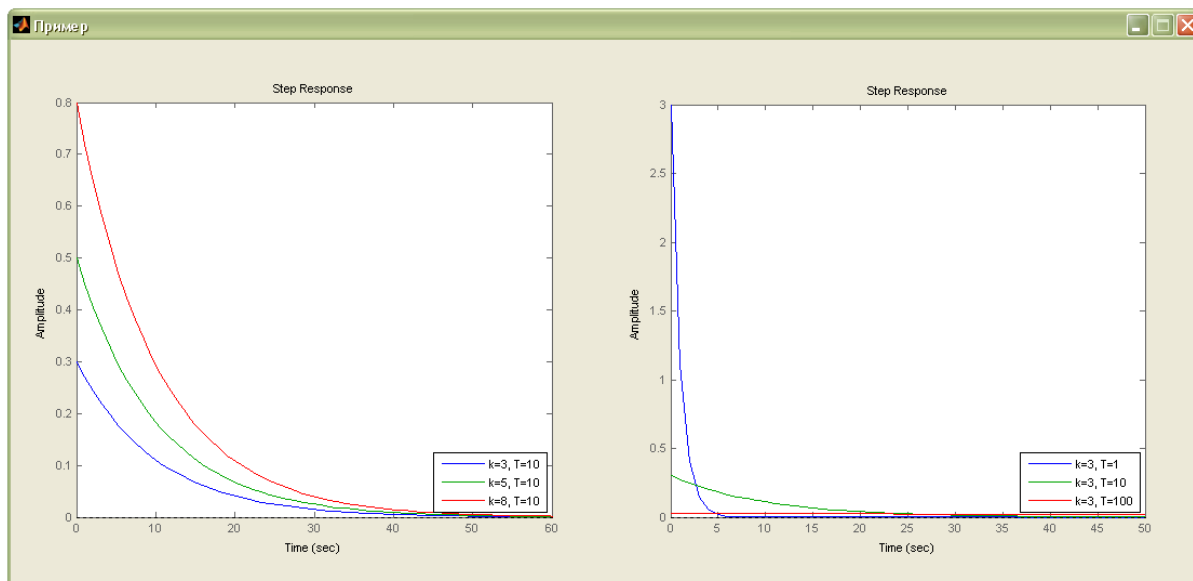
Фиг. 2.9. Пример за вариране на коефициентите на аperiодично звено от първи ред.

На фиг. 2.10 е показан зададения пример за инерционно интегриращо звено. Целта е да се покаже на потребителя по какъв начин влияе предавателният коефициент на наклона на преходната характеристика и как намаляването на времеконстантата доближава характеристиките на инерционно интегриращото звено към идеално интегриращото звено.



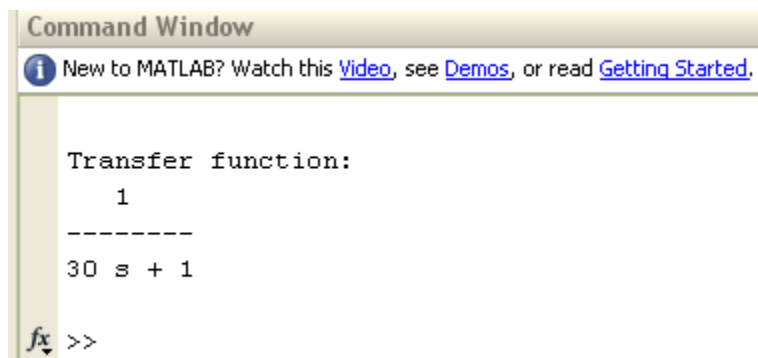
Фиг. 2.10. Пример за вариране на коефициентите на инерционно интегриращо звено.

На фиг. 2.11 е показан пример на реално диференциращо звено.



Фиг. 2.11. Пример за вариране на коефициентите на реално диференциращо звено.

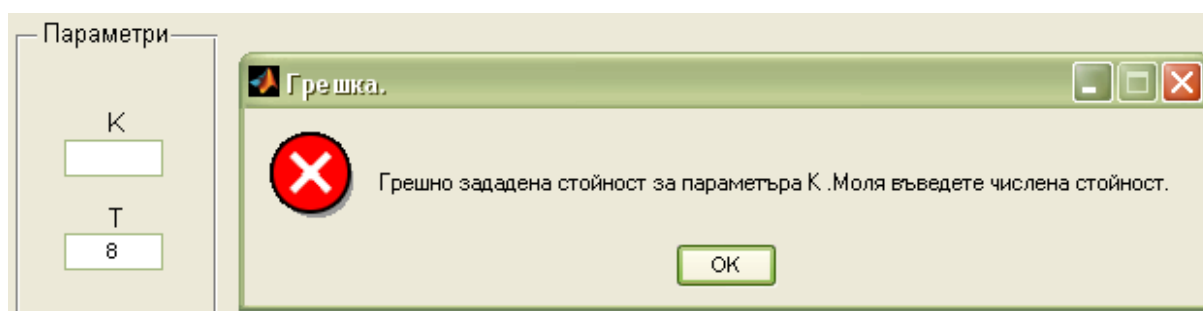
След запознаване с общия вид на характеристиките, чрез показаните примери, на потребителя се предоставя възможност да задава стойности на параметрите на различните звена. Съществуват три полета за въвеждане на данни от потребителя – за коефициента k , коефициента T и за **Време** като параметър за изчертаване на характеристиката. За по-добра нагледност има възможност да се покаже предавателната функция в работното пространство на *Matlab*. Така потребителят, след като е въвел коефициентите, може да види зададената от него предавателна функция - фиг. 2.12.



Фиг. 2.12. Показване на предавателната функция в работното пространство на *Matlab*.

Предвидени са защиты от неправилно въведени стойности:

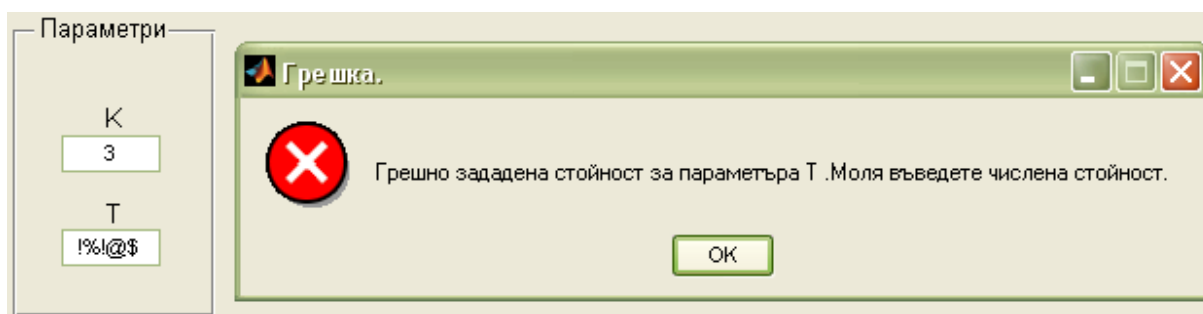
- При отсъствие на въведен символ или при въведен нечислов символ за коефициента k , на екрана се появява съобщението от фиг. 2.13.



Фиг. 2.13. Съобщение за грешно въведен коефициент k .

Важно е да се отбележи, че не може да се продължи работата докато съобщението не бъде опознато от потребителя от бутона **OK**. В този случай не е въведен символ.

- При отсъствие на въведен символ или при въведен нечислов символ за времеконстантата T се появява съобщение за грешка показано на фиг. 2.14. В този случай е въведен нечислов символ.

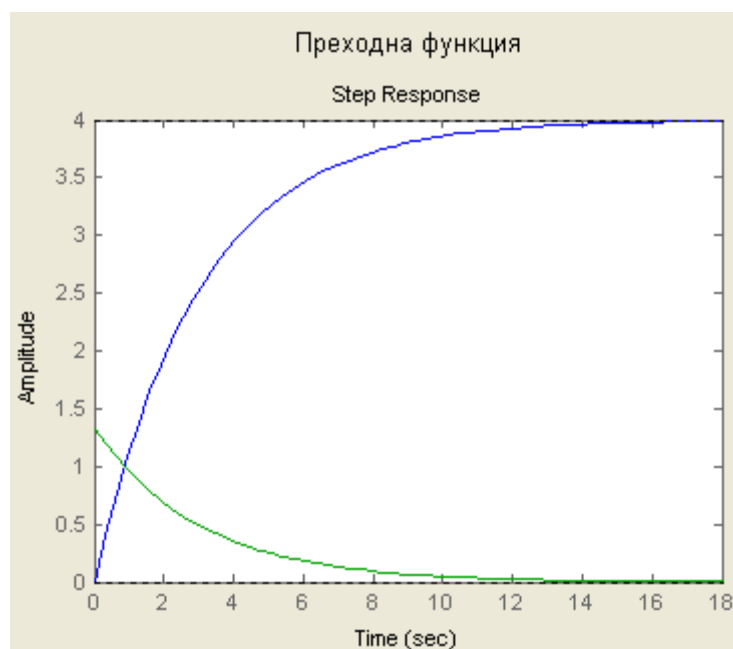


Фиг. 2.14. Съобщение за грешно въведен коефициент T .

- В случай, че в полето **Време** се въведат нечислови символи или отрицателни числови символи, както и в случай, че не се въведе символ, под полето за въвеждането на символи се изписва следното съобщение с червен цвят – *Автоматично задаване на времето t* . Това означава, че **Matlab** автоматично генерира времевият интервал и изобразява преходната и тегловната характеристика. Ако се въведе правилен символ (число различно от нула) в полето, след натискане на **Бутон 1** съобщението се скрива.

Трябва да се обърне внимание и на случая, в който след успешно изчертаване на характеристиките, се направи една от горе посочените грешки при въвеждането на коефициентите k и T . В този случай съобщението за автоматично задаване на времето ще се скрие и прозорците за изчертаването на преходната и тегловната характеристика ще се изчистят. Това е направено, за да не остават с грешна представа студентите, че едновременно е въведена грешна стойност и се изчертават характеристиките. Останалите настройки остават непроменени.

Има възможност за използване на т.н. *мрежа* (функцията **grid** в **Matlab**), както и т.н. *наслагване* (функцията **hold on** в **Matlab**) за всяка от двете графики. При наслагването могат да се видят характеристиките с различни коефициенти, както и характеристики на различни звена с еднакви (фиг. 2.15) или различни коефициенти.



Фиг. 2.15. Преходна характеристика на апериодично звено и на реално диференциращо звено с еднакви коефициенти.

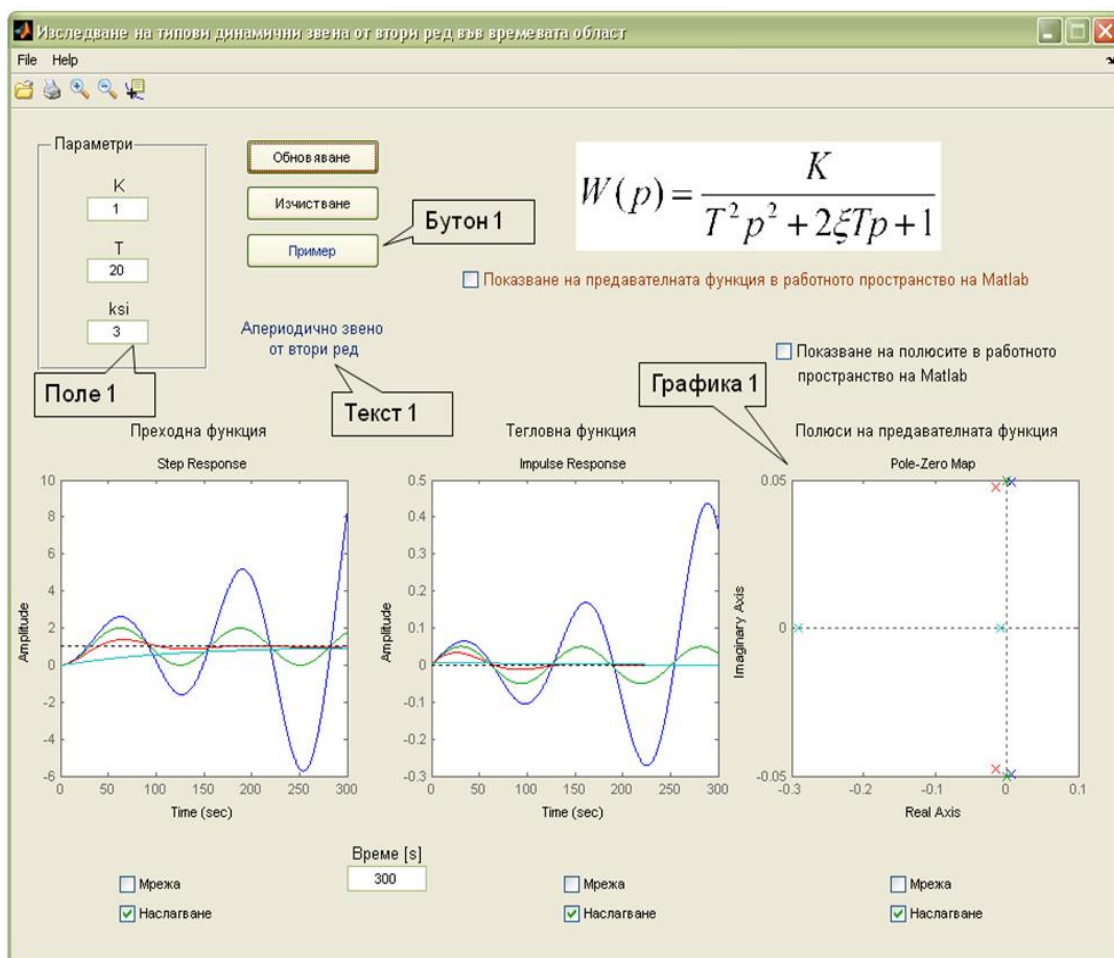
Бутон 2 от фиг. 2.8 служи за изчистване на:

- вече изчертаните графики;
- съобщението за автоматично задаване на времето;
- отметките указващи наслагването и мрежата за двете графики.

Програмният код на графичния интерфейс е предоставен в **Приложение 2.**

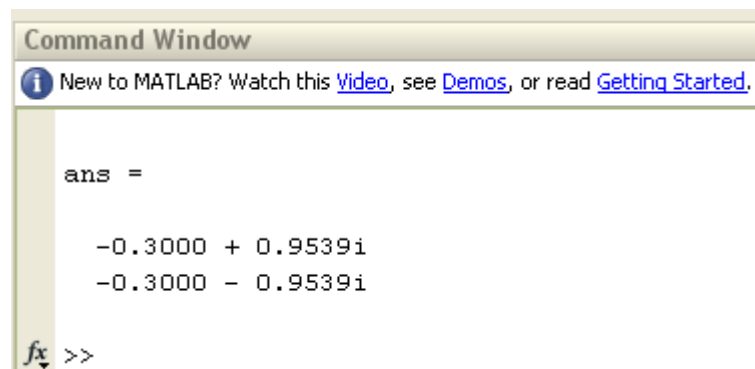
2.4. Прозорец „Типови динамични звена от втори ред във времевата област”

Този прозорец представлява средство за изследване на типови динамични звена от втори ред във времевата област - фиг. 2.16. Различното тук е, наличието на **Поле 1** за въвеждане на данни и **Графика 1**, която изчертава полюсите на предавателната функция.



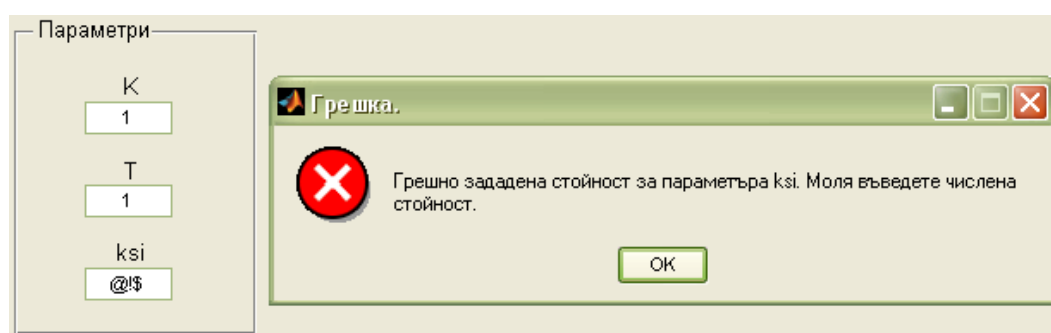
Фиг. 2.16. Прозорец „Исследване на типови динамични звена от втори ред във времевата област”.

Съществува и възможност за показване на изчислените полюси на предавателната функция в работното пространство на **Matlab** – фиг. 2.17.



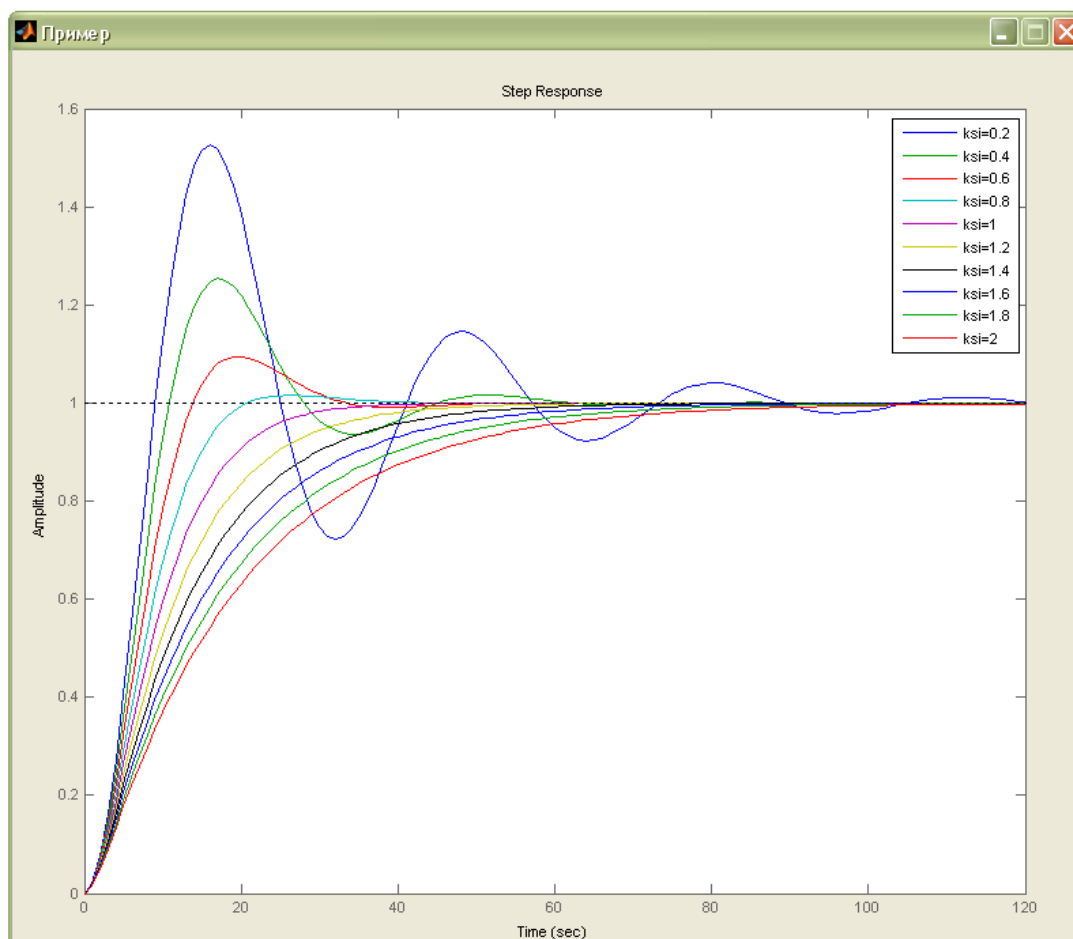
Фиг. 2.17. Показване на полюсите на предавателната функция в работното пространство на *Matlab*

От **Поле 1** се променя фактора на затихване ξ в предавателната функция $W(p) = \frac{K}{T^2 p^2 + 2\xi T p + 1}$. Чрез вариране на стойността на ξ , потребителят може да наблюдава консервативно звено, колебателно звено и апериодично звено от втори ред. В зависимост от избраната стойност се появява **Текст 1**, информирайки студентите за типа на звеното. Отново са предвидени защитите за коефициентите k и T , като освен това са направени защити за фактора на затихване – фиг. 2.18. При грешно задаване на някой от тези параметри освен споменатите защити, тук **Текст 1** (фиг. 2.16) също се скрива.



Фиг. 2.18. Съобщение за грешно въведен коефициент ksi .

Бутон 1 препраща потребителя към пример, на който е показана преходната характеристика при вариране на фактора на затихване от 0.2 до 2 – фиг. 2.19.



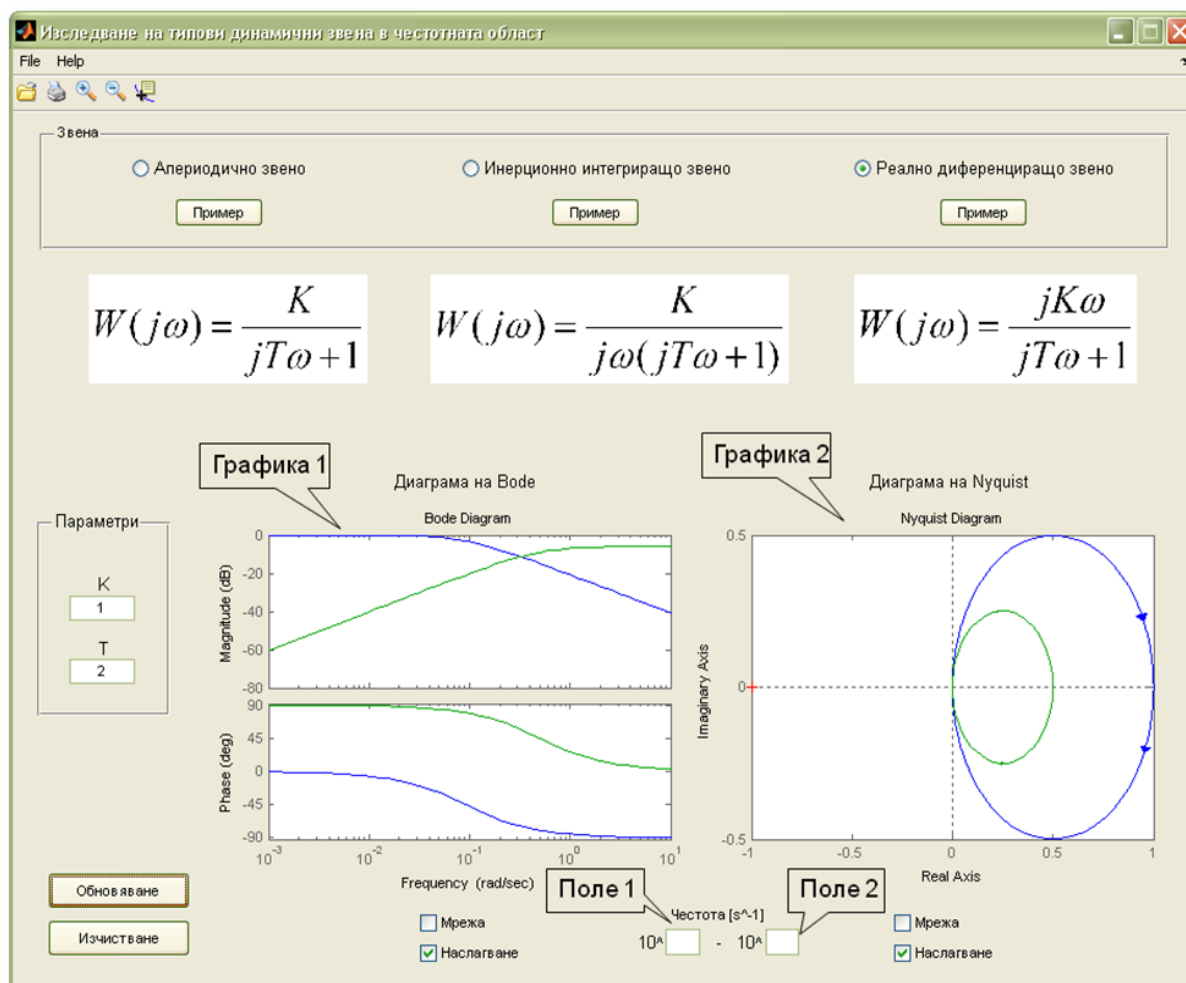
Фиг. 2.19. Пример за вариране на фактора на затихване

Този пример представлява преминаването от колебателно до апериодично звено от втори ред със стъпка на вариране 0.2. Целта е да се направи нагледна аналогия между стойността на фактора на затихване и вида на преходния процес. Умишлено е пропуснато изчертаването на неустойчива система ($\xi < 0$) и консервативно звено ($\xi = 0$) за по-добра нагледност. Студентите лесно могат да пресъздадат този пример заедно с неустойчива система и консервативно звено в основния прозорец благодарение на функцията *наслагване*.

2.5. Прозорец „Типови динамични звена в честотната област”

Тук потребителят има възможност да избира между три типови динамични звена от радио бутоните поставени в панел със заглавие *Звена*: Аперiodично звено от първи ред, Инерционно интегриращо и Реално диференциращо звено. В панела са поместени и три бутона *Пример*. Целта е първоначално студентите да видят честотните характеристики на съответните динамични звена при различни стойности на параметрите им. При задаване на параметрите на конкретното звено, на *Графика 1* се изчертава диаграмата на *Bode*, а на *Графика 2* се изчертава диаграмата на *Nyquist*. – фиг. 2.20.

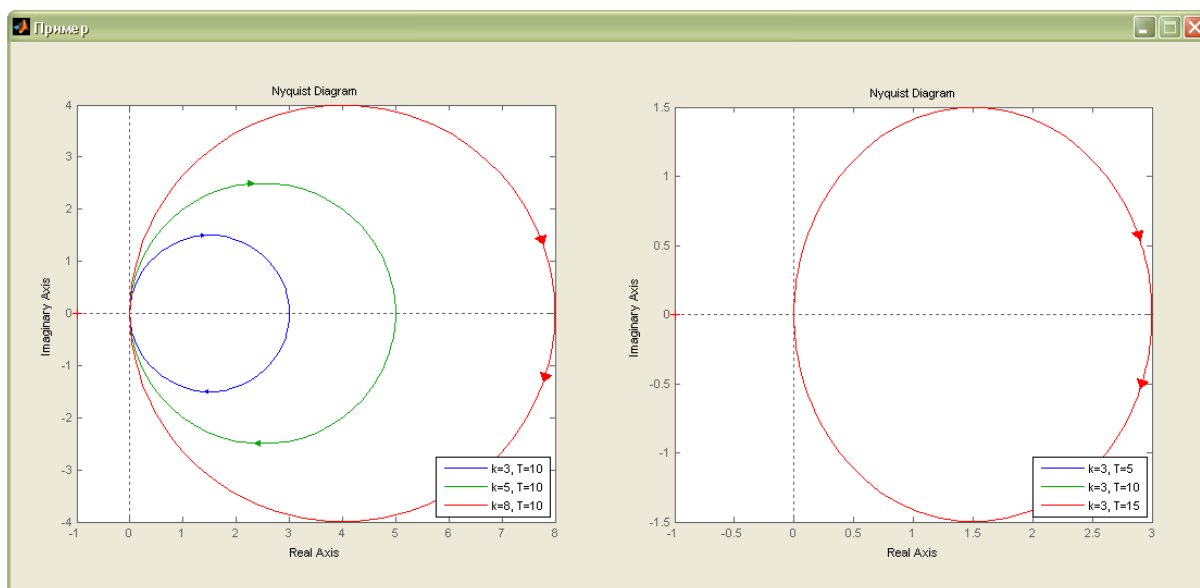
В *Поле 1* се задава долната граница на честотния диапазон, а в *Поле 2* горната граница. Ако не са зададени стойности, то диапазонът се избира в достатъчно широки граници. Отново под всяко звено е дадена как изглежда предавателната функция в явен вид. Има наличие на защиты срещу неправилно въвеждане на данни както при интерфейса за изследване на типови динамични звена във времевата област.



Фиг. 2.20. Прозорец „Изследване на типови динамични звена в честотната област”.

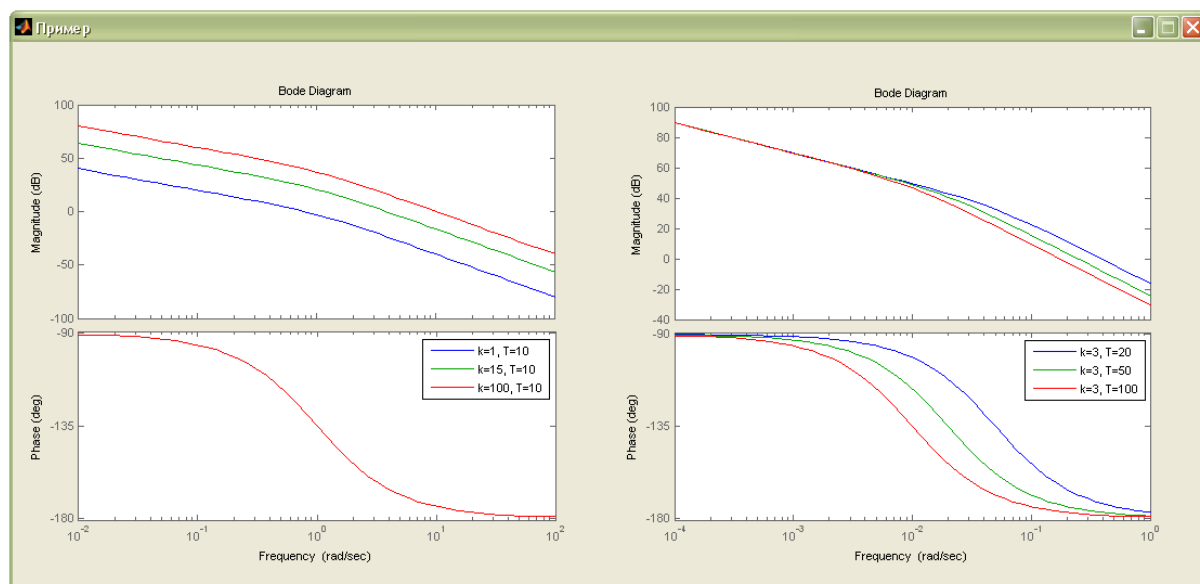
На фиг. 2.20 е показан пример с изчертаването на диаграмите на *Bode* и *Nyquist* на апериодично звено и реално диференциращо звено с различен параметър T .

Чрез натискане на бутона за пример на апериодичното звено се отваря графичен интерфейс. Примерът представлява две диаграми на *Nyquist*. Целта му е да покаже на студентите как коефициента k влияе на диаграмата за разлика от коефициента T – фиг. 2.21.



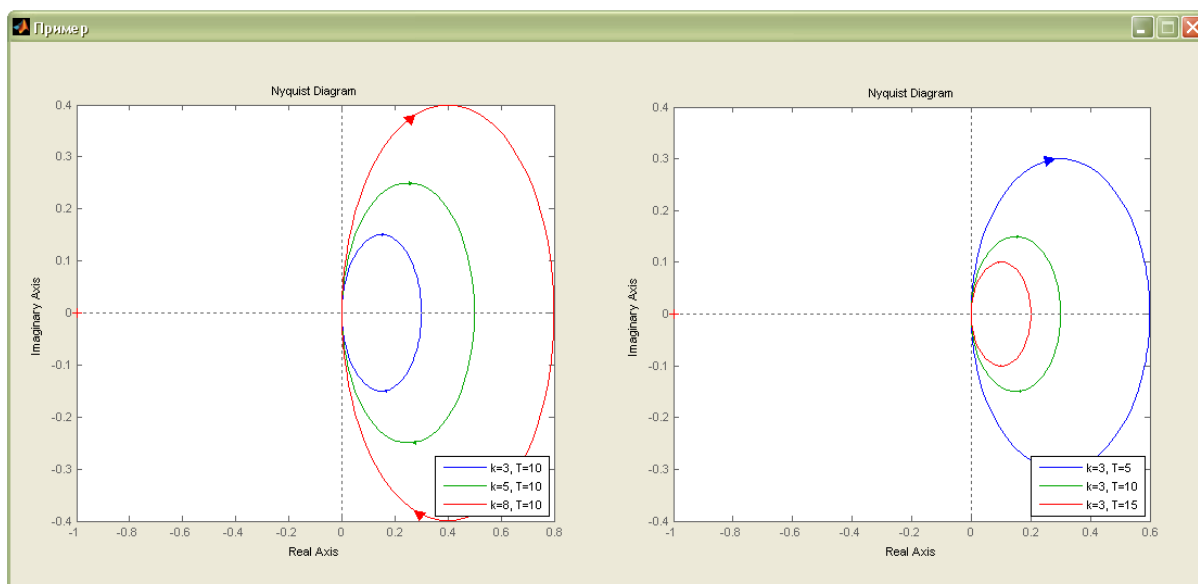
Фиг. 2.21. Пример за Апериодично звено

Примерът за инерционно интегриращото звено представлява две диаграми на *Bode*. Целта му е да покаже на потребителя как коефициентите k и T влияят на диаграмата – фиг. 2.22



Фиг. 2.22. Пример за инерционно интегриращо звено

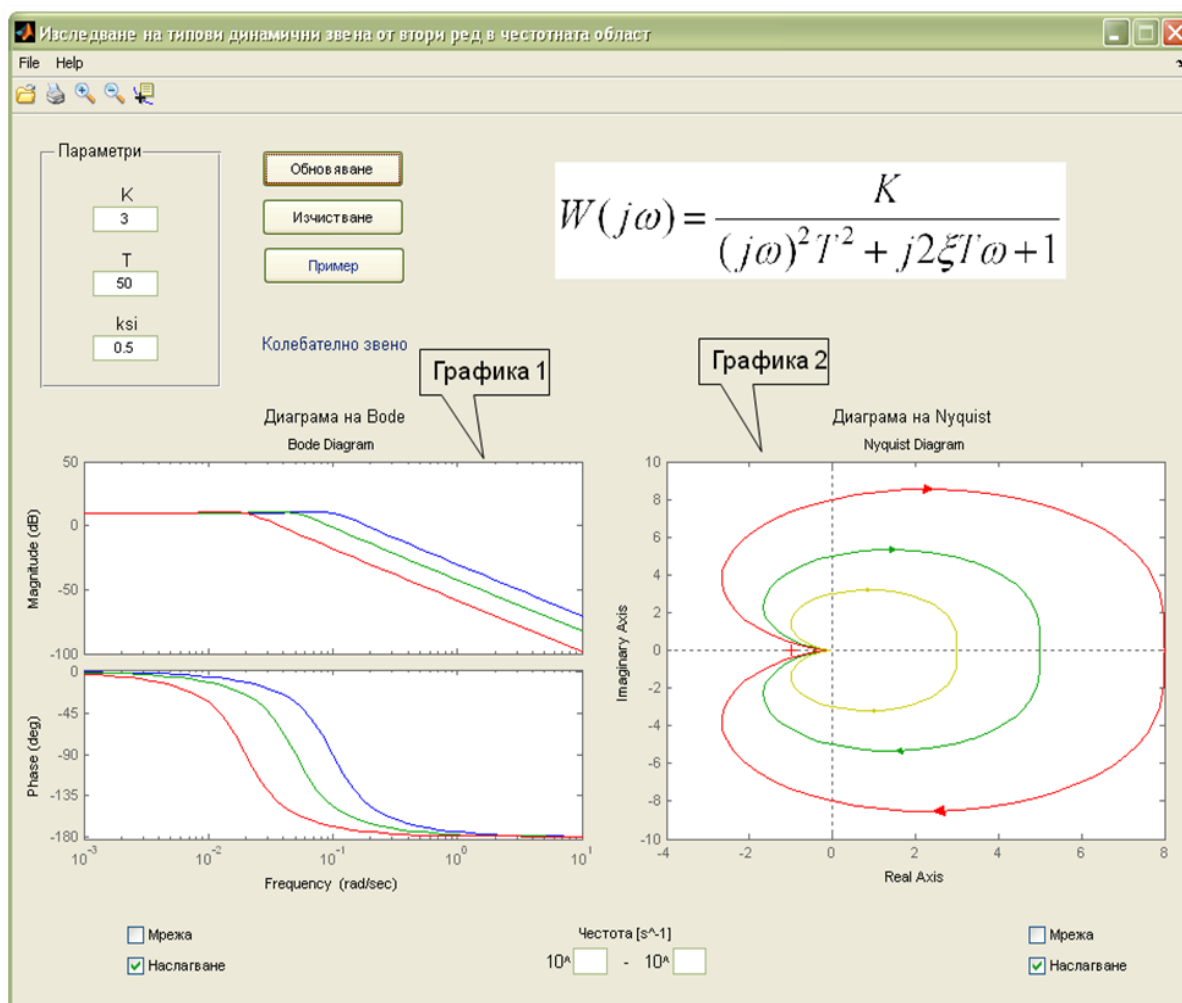
Предвиденият пример за реално диференциращото звено се състои от две диаграми на *Nyquist*. Неговата цел е да покаже на как едновременно коефициентите k и T влияят на диаграмата – фиг. 2.23.



Фиг. 2.23. Примерът за реално диференциращото звено

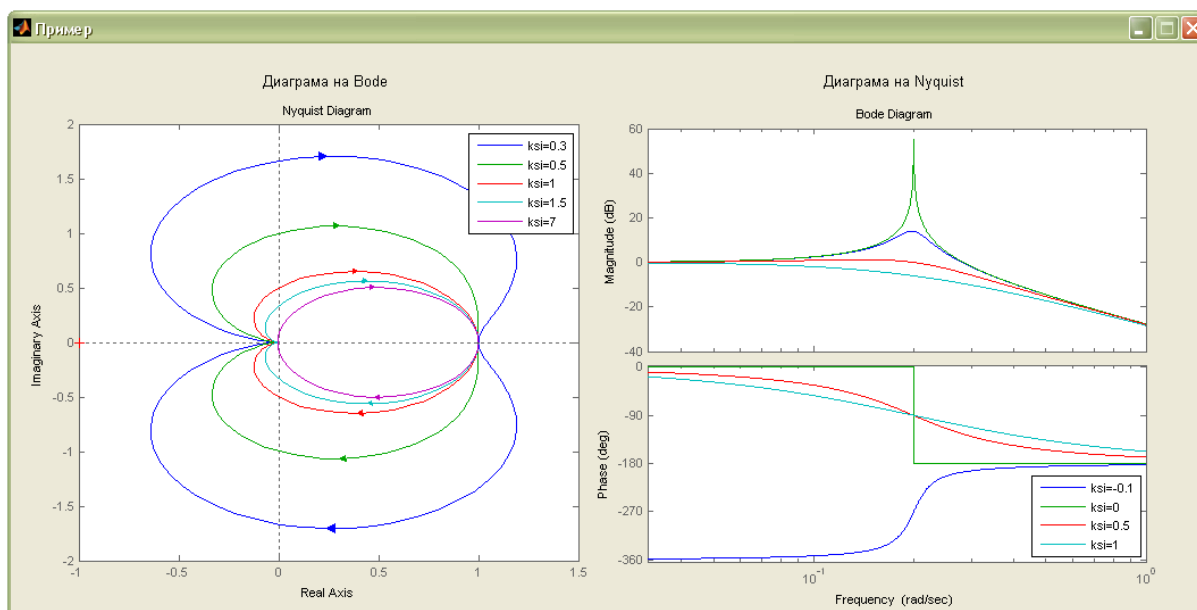
2.6. Прозорец „Типови динамични звена от втори ред в честотната област”

Интерфейсът на външен вид е подобен с този от изследване на системи във времевата област. Съществуват известни различия, които са с цел да покриват необходимите изисквания за пълното усвояване на материала от студентите. Прозорецът има две графики. На *Графика 1* се изчертава диаграма на *Bode*, а на *Графика 2* - диаграма на *Nyquist*, за различни звена от втори ред. Чрез изменение на коефициента k_{si} , студентите могат да се запознаят с честотните характеристики на апериодично звено от втори ред, колебателно звено, консервативно звено – фиг. 2.24.



Фиг. 2.24. Прозорец „Изследване на типови динамични звена от втори ред в честотната област”.

На фиг. 2.25 е изобразен примерът, предвиден в интерфейса. На една от графиките е изобразена диаграмата на **Bode** на колебателно звено и апериодично звено от втори ред. На другата графика е изобразена диаграмата на **Nyquist** за четирите гореспоменати типа. Предвидена е и легенда за по-добро ориентиране. Разбира се, студентите имат възможност да разгледат голям брой комбинации на коефициентите в основния интерфейс.

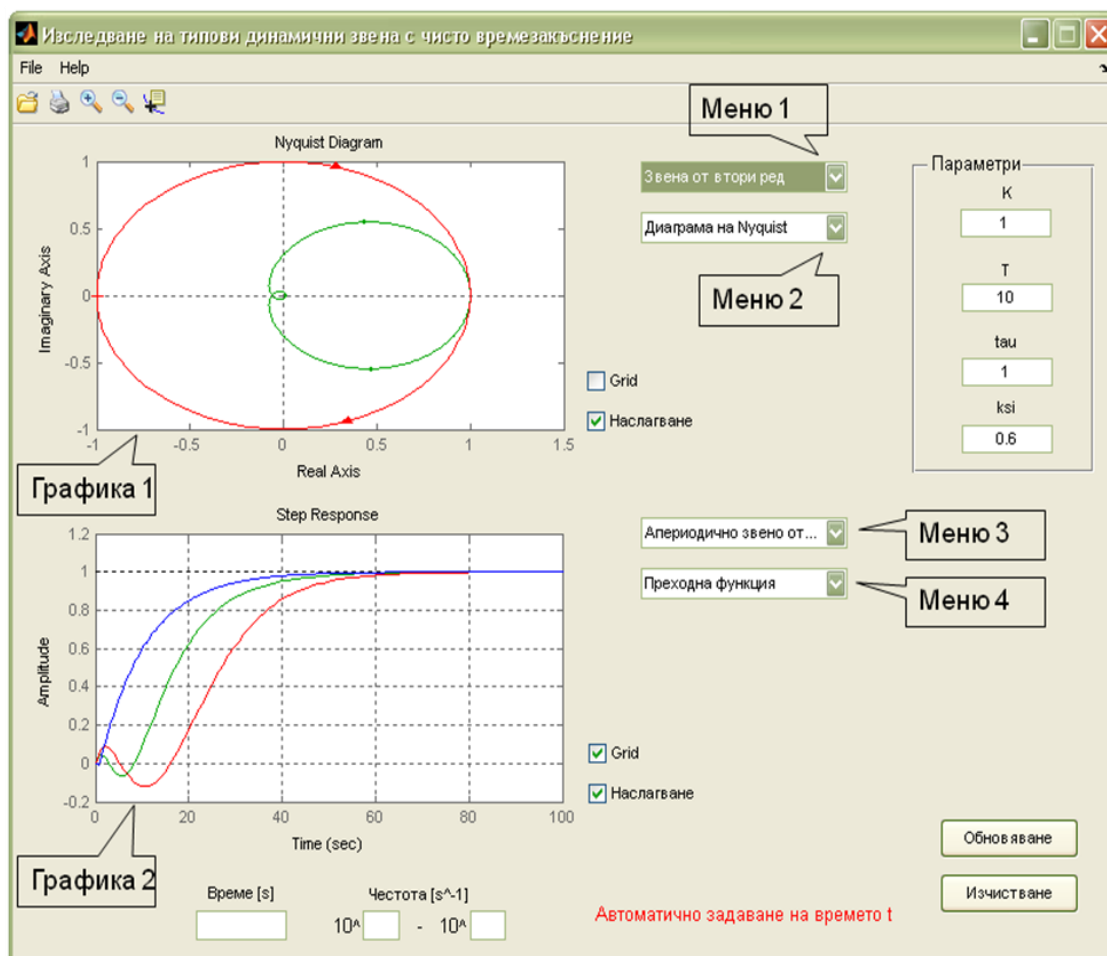


Фиг. 2.25. Пример за обекти от втори ред в честотната област.

2.7. Прозорец „Типови динамични звена с чисто закъснение“

След като студентите са се запознали последователно и поотделно с различните динамични звена без времезакъснение във времевата и честотната област, е предвиден интерфейс, който да ги въведе в изследването на типовите динамични звена с чисто времезакъснение.

На фиг. 2.26 е представен интерфейсът, който е компактен, но и многофункционален.



Фиг. 2.26. Прозорец „Изследване на типови динамични звена с чисто времезакъснение”.

Този интерфейс се състои от двете графики – *Графика 1* и *Графика 2*, четирите падащи менюта – *Меню 1-4*, полета за въвеждане на данни и два бутона. До всяка от графиките има поместени по две падащи менюта. От *Меню 1* и *Меню 3* студентите могат да избират типа на звеното:

- Апериодично звено от първи ред
- Инерционно интегриращо звено
- Реално диференциращо звено
- Звена от втори ред
- Звено чисто времезакъснение

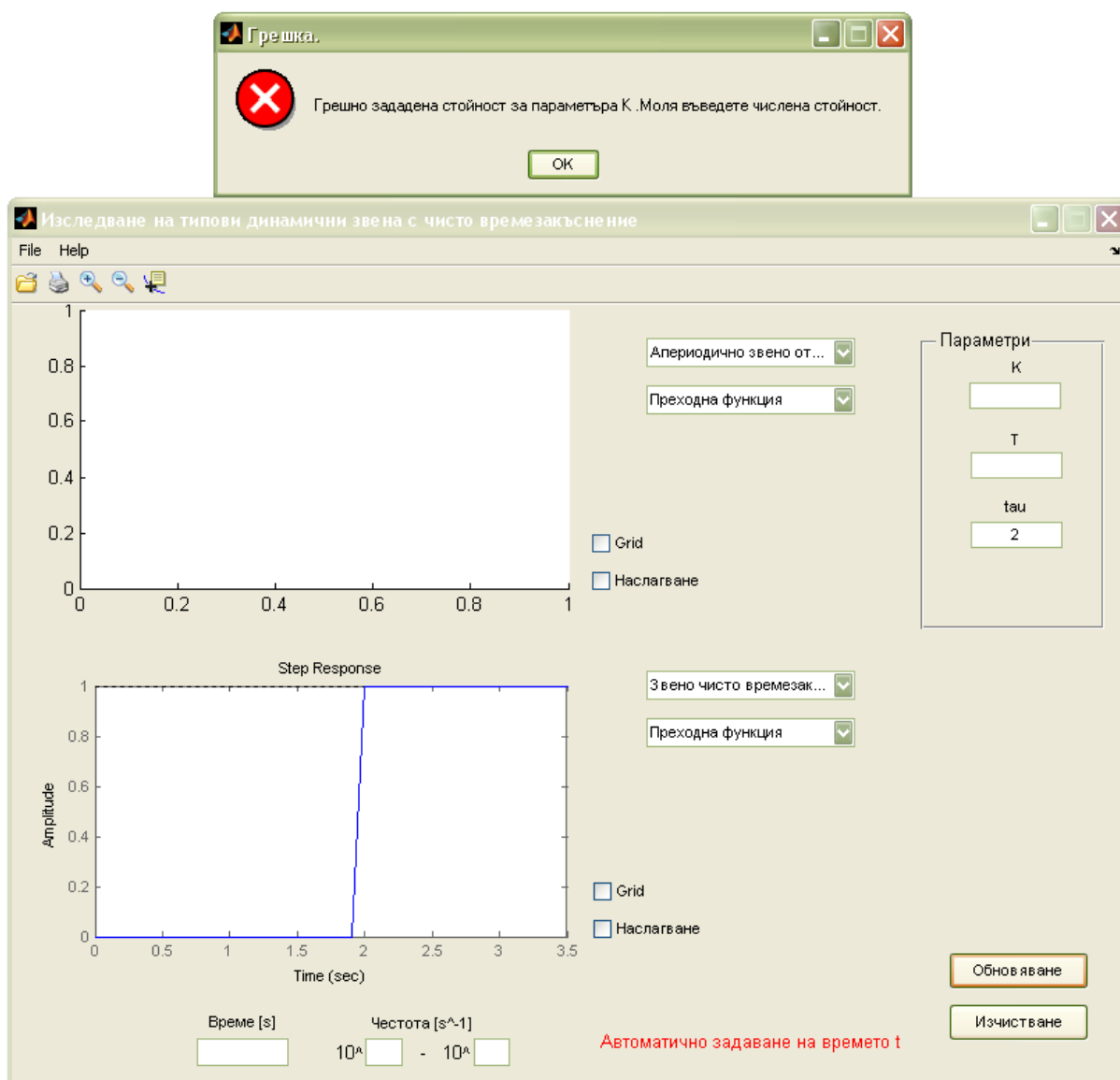
От **Меню 2** и **Меню 4** могат да избират желаната характеристика, която искат да се изчертае:

- Преходна функция
- Тегловна функция
- Диаграма на Nyquist
- Диаграма на Bode

Чрез функцията **наслагване**, студентите могат да изчертаят различни характеристики на различни звена с различни коефициенти, с или без закъснение.

Предвидените защиты от грешно въведени стойности се различават в известен смисъл, за да се гарантира безпроблемната работа. Ще бъде обърнато внимание на различията.

Нека имаме комбинация от изчертаване на преходна функция на апериодично звено от първи ред и на звено чисто времезакъснение. За правилното изчертаване са необходими коефициентите **k** , **T** и **τ** . След успешното изчертаване се отстранява **k** или **T** . Тъй като тези коефициенти не участват в предавателната функция на звено чисто времезакъснение, преходната функция ще бъде изчертана безпроблемно, докато тази на апериодичното звено изчистена. Това е направено с цел, да не се подвеждат студентите, че коефициентите **k** и **T** са необходими. Също така ще се появи съобщение за грешка с текст, подсказващ какви мерки да се предприемат, за да се отстрани проблема – фиг. 2.27. Същата е и ситуацията при комбинации на звено чисто времезакъснение с инерционно интегриращото звено и с реално диференциращото звено.



Фиг. 2.27. Пример за защиты от неправилно въвеждане на данни.

Друга особеност е появяването на текст *Автоматично задаване на времето t* само при изследването на времевите характеристики.

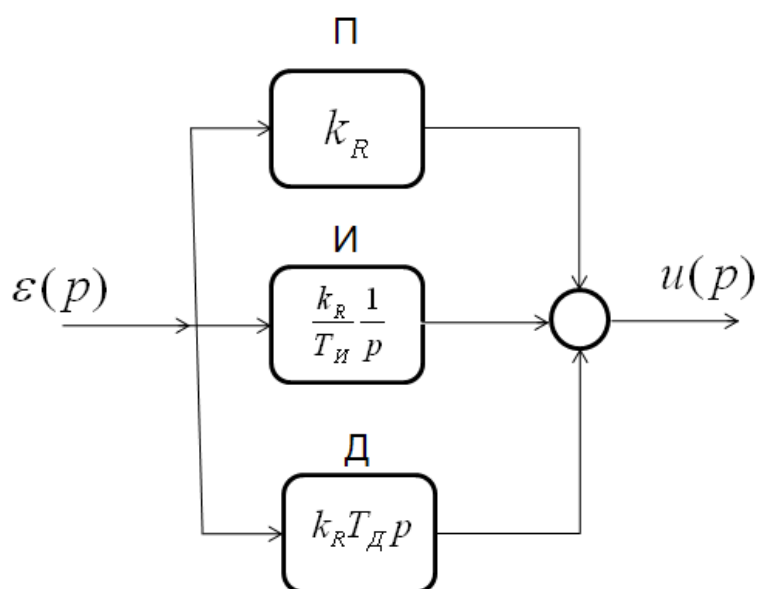
Като цяло този интерфейс обобщава предходните и добавя в изследването времезакъснението като параметър.

ТРЕТА ГЛАВА

3. ГРАФИЧЕН ИНТЕРФЕЙС ЗА ИЗСЛЕДВАНЕ НА ТИПОВИ ЗАКОНИ ЗА РЕГУЛИРАНЕ

3.1. Въведение

Най-разпространени в практиката и изучавани като еталон в теоретичните изследвания по управление са линейните закони за управление. Причината за тяхната важност са теоретичната и практическата простота на тези регулатори и все още голямото им приложение. Структурата на **ПИД** регулатор включва паралелно свързани пропорционална (**П**), интегрална (**И**) и диференциална (**Д**) съставка – фиг.3.1.

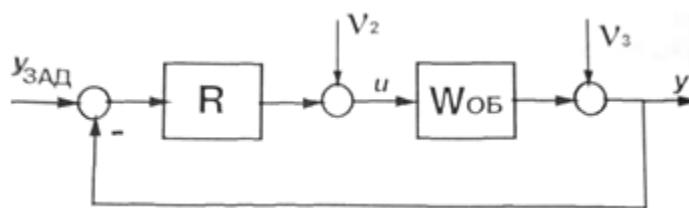


Фиг. 3.1. Схема на ПИД регулатор

П и **ПИ** се получават като частни случаи на по-общия **ПИД** регулатор, когато се премахнат съответно **Д**-съставката ($T_D=0$) и **И**-съставката ($T_I \rightarrow \infty$). Трите съставки на регулатора обработват текущата грешка в управлението (**П**), миналите грешки в управлението (**И**) и предсказаните бъдещи грешки в управлението (**Д**). **П**-съставката реализира общ коефициент на усилване на регулатора и доминира над другите съставки, когато грешката в управлението е голяма. **И**-съставката осъществява нулиране на грешките в установен режим при обработване на заданието и спомага за нискочестотно компенсиране на постоянни товарни смущения. **Д**-съставката подобрява преходното поведение на регулируемата величина чрез високочестотна компенсация. Най-често срещаните режими на работа на регулаторите са автоматичен, ръчен, каскаден и дистанционен.

Общите принципи на работа на линейните закони за регулиране [8, 9] се разглеждат при анализа на едноконтурни системи от вида на представената на фиг. 3.2. Свойствата на системата, включително видът на преходния процес и устойчивостта, се определят от характеристичния полином (знаменателят на предавателната функция на системата), а по-конкретно от типа и разположението на корените му, наречени полюси на системата. При по-сложни системи върху формата на преходния процес (но не и върху фундаменталните му свойства) оказват влияние и корените на полинома от числителя на системната предавателна функция (ако има такива), наречени нули на системата. Затова е необходимо да се определят предавателните функции на САР при различните възможни входни въздействия: задание, смущение на входа на обекта, смущение на изхода на обекта (фиг. 3.2).

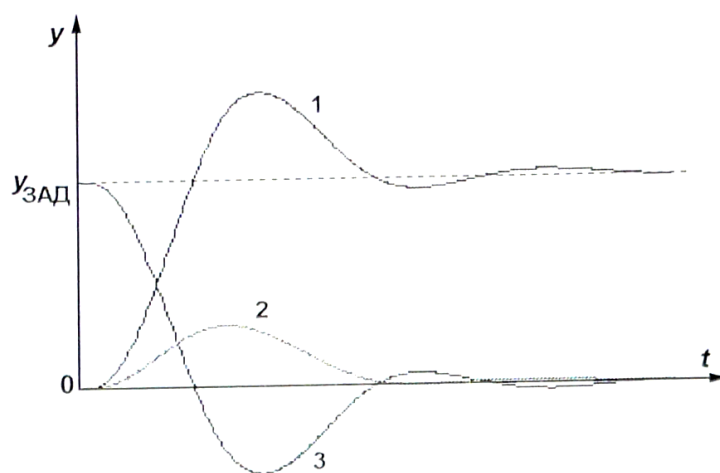
$$W_{y_{зад}}(p) = \frac{R(p)W_{об}(p)}{1 + R(p)W_{об}(p)}, \quad W_{v_2}(p) = \frac{W_{об}(p)}{1 + R(p)W_{об}(p)}, \quad W_{v_3}(p) = \frac{1}{1 + R(p)W_{об}(p)}$$



Фиг. 3.2. Едноконтурна САР

Характеристичният полином $H(p) = 1 + R(p)W_{ОБ}(p)$, включва само елементите на затворения контур и е един и същ и за трите предавателни функции, получавани при въздействие поотделно на всеки от различните входове в системата.

Гъвкавият елемент в системата е регулаторът. Тъй като полюсите на САР в трите различни случая са едни и същи, то основните свойства на преходния процес на всяка система ще са еднотипни – колебателен или апериодичен, устойчив или неустойчив. Съществува само едно принципно различие за трите случая, отнасящо се до началната и крайната стойност на преходния процес – фиг. 3.3. То се дължи на различията в числителя на предавателната функция.



Фиг. 3.3. Преходни процеси в САР според мястото на входното въздействие

3.1.1 П регулатор

Пропорционалният регулатор има променлив коефициент на регулиране, който представлява единствения параметър на настройка. Той е най-простият регулатор и представлява пропорционално звено. Поради това той не предизвиква фазови изменения в затворената система. Предавателната функция на **П** регулатора има вида $R(p) = k_R$.

Търсим реакцията на системата при единично стъпално въздействие - $y_{зад}^o = 1(t)$. Ако обекта са представя чрез апериодично звено от първи ред, предавателната функция на затворената система е:

$$\Phi(p) = \frac{k_R \frac{k_{OB}}{T_{OB}p + 1}}{1 + k_R \frac{k_{OB}}{T_{OB}p + 1}} = \frac{k_R k_{OB}}{1 + k_R k_{OB}} \cdot \frac{1}{\frac{T_{OB}}{1 + k_R k_{OB}} p + 1} = \frac{k_e}{T_e p + 1},$$

$$\text{където: } k_e = \frac{k_R k_{OB}}{1 + k_R k_{OB}} \quad T_e = \frac{T_{OB}}{1 + k_R k_{OB}}$$

От формулите се вижда се, че $k_e < 1$, а $k_e \approx 1$, когато $k_R \rightarrow \infty$. Големият коефициент на усилване подобрява работата на системата, но намалява запасите на устойчивост.

$$\text{Статичната грешка в този случай ще има стойност: } \Delta = \frac{1}{1 + k_R k_{OB}}.$$

Пропорционалният регулатор не може да гарантира отсъствие на статична грешка в установен режим, нито при смущения по задание, нито при смущения на входа на обекта. Това е основния недостатък на **П** регулатора.

3.1.2 ПИ регулатор

Пропорционално-интегралният регулатор има два параметъра за настройка - k_R и T_I . Пропорционалният регулатор е частен случай на **ПИ**, когато имаме голямо време на интегриране.

Предавателната функция на **ПИ** регулатора има вида:

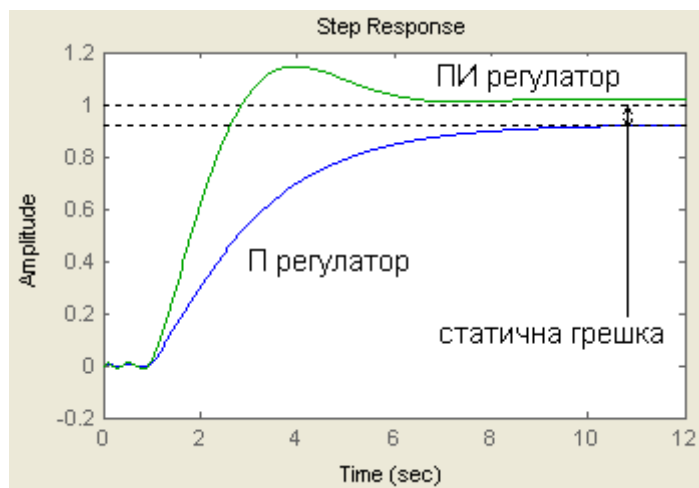
$$R(p) = k_R \left(1 + \frac{1}{T_I p} \right)$$

Предавателната функция на затворената система е:

$$\Phi(p) = \frac{k_R \left(1 + \frac{1}{T_I p} \right) \frac{k_{OB}}{T_{OB} p + 1}}{1 + k_R \left(1 + \frac{1}{T_I p} \right) \frac{k_{OB}}{T_{OB} p + 1}} = \frac{k_R k_{OB} (T_I p)}{T_I T_{OB} p^2 + T_I p (1 + k_R k_{OB}) + k_R k_{OB}}$$

В този случай $\lim_{p \rightarrow 0} \Phi(p) = \frac{k_R k_{OB}}{k_R k_{OB}} = 1$ при единично стъпаловидно

изменение на заданието, на изхода на системата се получава единица. В система с **ПИ** регулатор няма статична грешка поради наличието на интегрираща съставяща. **ПИ** регулаторът прави системата по-бавна в сравнение с **П** регулатор (интегриращата съставяща въвежда допълнително закъснение в системата), но нулира статичната грешка – фиг. 3.4.



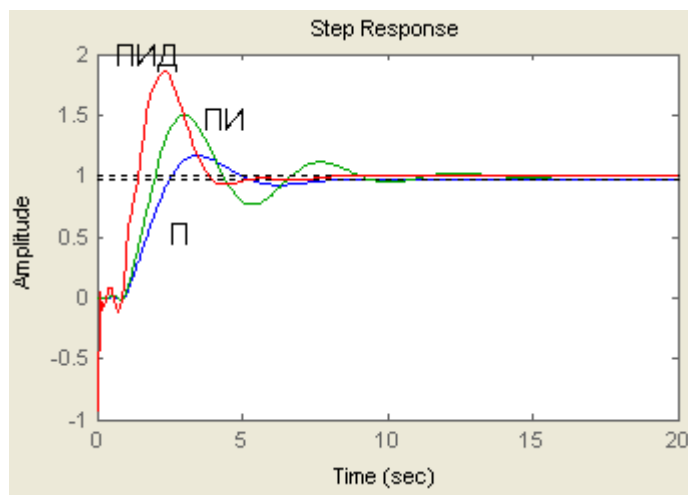
Фиг. 3.4. Регулиране с П и ПИ регулатор.

3.1.3 ПИД регулатор

Предавателната функция на **ПИД** регулатора има вида:

$$R(p) = k_R \left(1 + \frac{1}{T_I p} + T_D p \right)$$

Диференциращата съставяща внася положителен ъгъл на изпреварване, който до известна степен неутрализира въздействието на интегриращата, като отдалечава АФЧХ от - 1. Това означава по-голяма точност и бързодействие. Поради наличието на **И**- съставяща в регулатора, системата запазва свойството да няма статична грешка – фиг. 3.5.



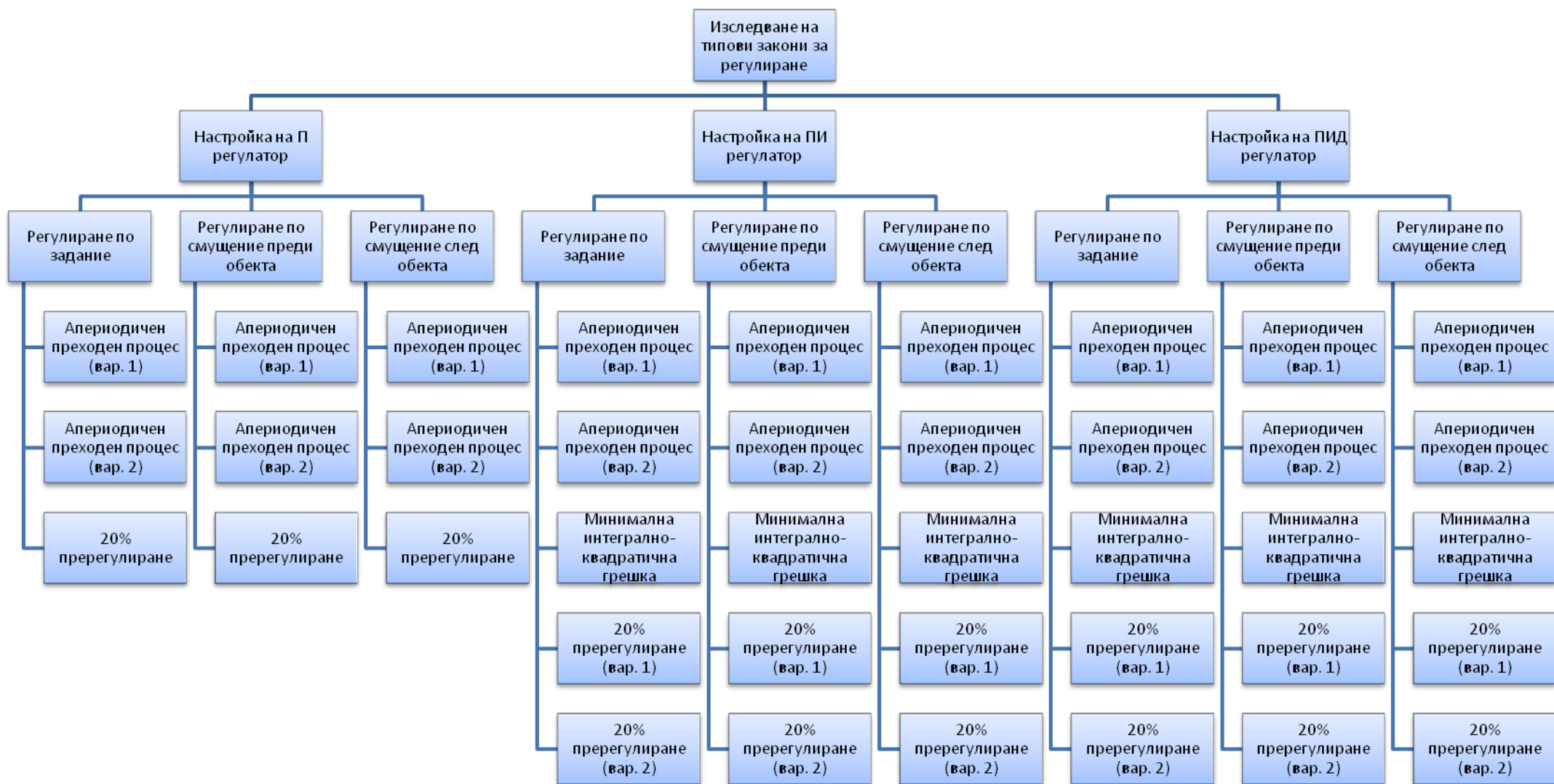
Фиг. 3.5. Регулиране с П,ПИ и ПИД регулатор

3.2. Графичен интерфейс в средата на Matlab за настройка и анализ на системи с П, ПИ и ПИД регулатори.

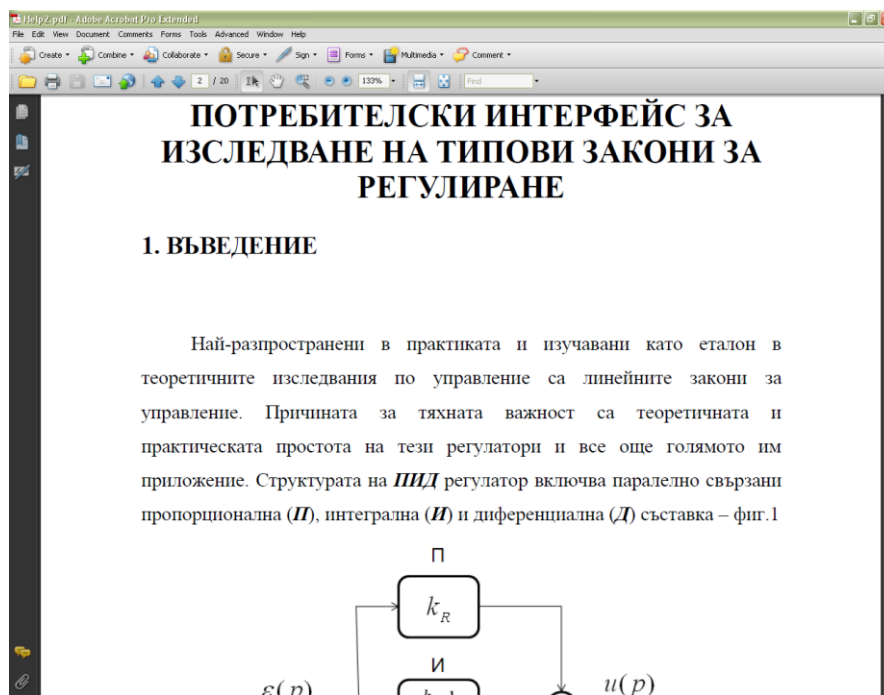
За изследване на типови закони за регулиране е създаден графичен интерфейс, като функциите му са представени на фиг. 3.6. Стартирането на интерфейса се осъществява като в работното пространство на *Matlab* се изпише *reg* последвано от натискане на клавиш *Enter* от клавиатурата.

Графичният интерфейс е показан на фиг. 3.8.

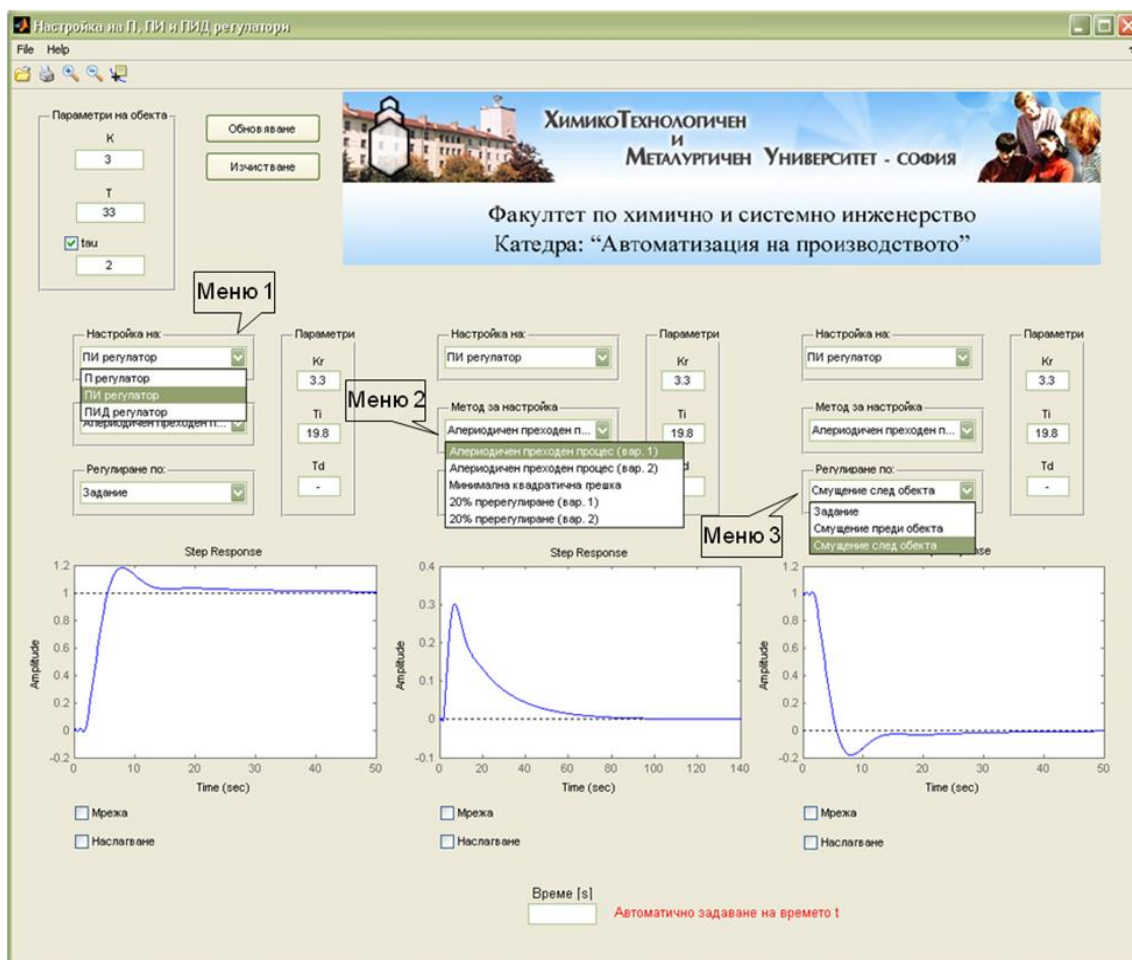
Той съдържа стандартните менюта *File* и *Help*, като подменюта на *File* са *Open*, *Print* и *Close*. Подменюта на *Help* са *About*, *GUI Help* и *Matlab Help*. При избор на *GUI Help* се отваря текст с описание на системата на български език – фиг. 3.7. Това до голяма степен ще улесни работата на студентите. Добавени са и пет икони за бърз достъп от панела с инструменти. Те са: *Open*, *Print*, *ZoomIn*, *ZoomOut* и *DataCursor*. С *DataCursor* потребителят може да получи информация за числени стойности от графиките в произволна посочена точка.



Фиг. 3.6. Функционална схема на разработения графичен интерфейс



Фиг. 3.7. Пример за *GUI Help*



Фиг. 3.8. Графичен интерфейс за изследване на типови закони за регулиране

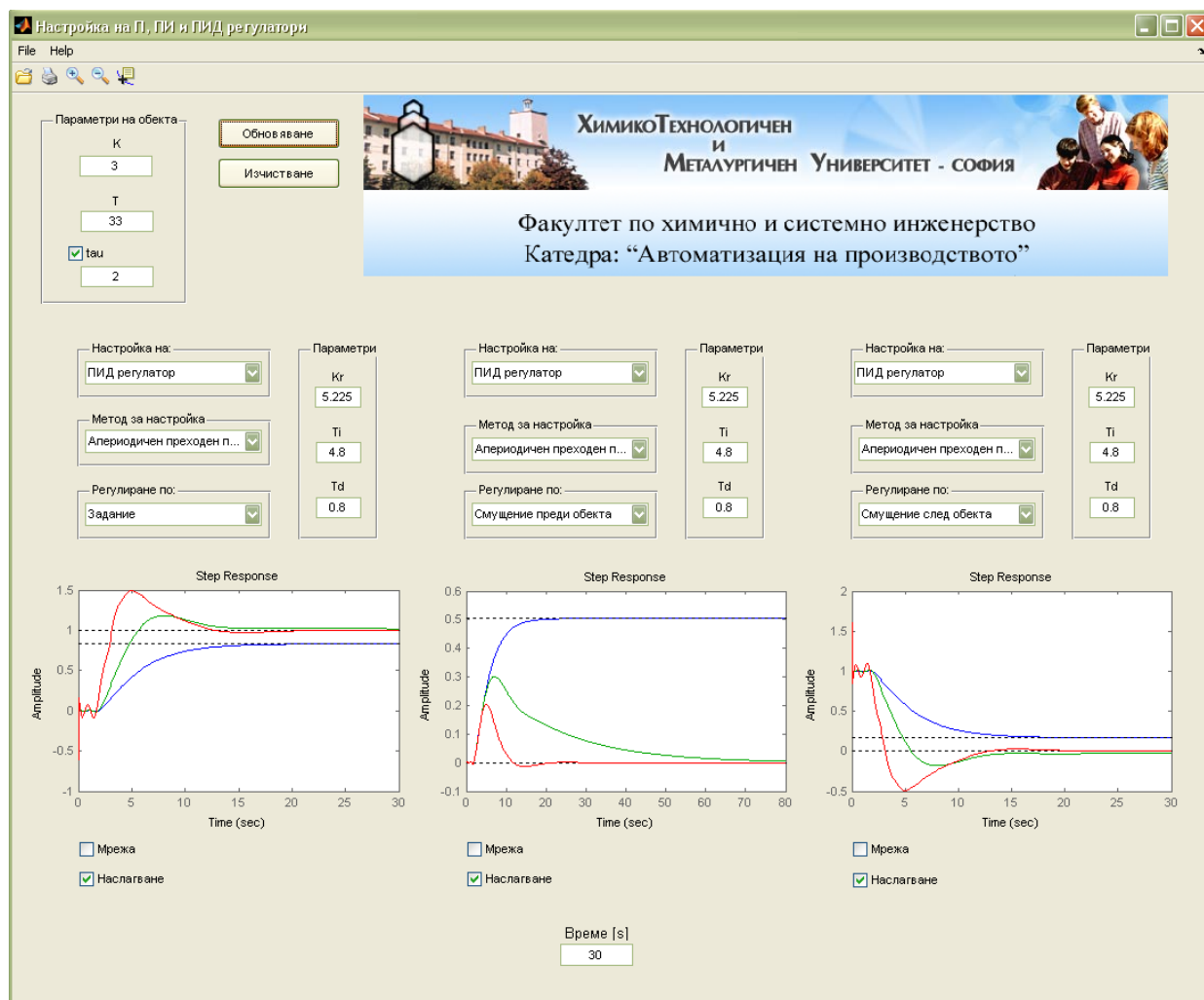
От **Меню 1** (фиг.3.8) се избира типа на желанния регулатор за настройка - **П**, **ПИ** или **ПИД**.

Предвидени са по три менюта от типа **Меню 3** (фиг.3.8) за изобразяване на трите разглеждани случая – реакция на затворената система по задание, по смущение на входа на обекта и по смущение на изхода на обекта. Предвидена е и възможност потребителите да могат да избират различни комбинации.

На фиг. 3.9 е представен пример за настройка на трите вида регулатори за трите възможни смущаващи въздействия, а именно:

- Със син цвят е показан **П** регулатор
- Зеленият отговаря на **ПИ** регулатор
- Червеният цвят е за **ПИД** регулатор

Същите цветове се използват и за останалите примери в създадения потребителски интерфейс.



Фиг. 3.9. Интерфейс за настройка и изследване на типови промишлени регулатори

Така разработената система включва три метода за настройка на **П** регулатор и пет метода за настройка на **ПИ** и **ПИД** регулатори. Изборът на метод за настройка се осъществява от **Меню 2** (фиг. 3.8). Системата използва приблизителни формули за настройка на типови промишлени регулатори.

Използваните методи за настройка на системи с **П** регулатор са приблизителни формули, основани на следните критерии за качество:

- Аperiодичен преходен процес (вар. 1)
- Аperiодичен преходен процес (вар. 2)
- 20% пререгулиране

Използваните методи за настройка на системи с **ПИ** и **ПИД** регулатор също са приблизителни формули, основани на следните критерии за качество:

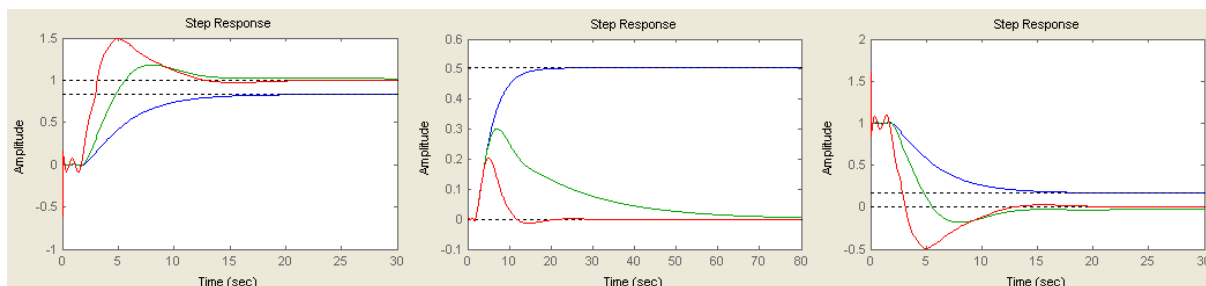
- Аперiodичен преходен процес (вар. 1)
- Аперiodичен преходен процес (вар. 2)
- Минимална интегрално-квадратична грешка
- 20% пререгулиране (вар. 1)
- 20% пререгулиране (вар. 2)

Използваните приблизителни формули [9] за настройка на регулаторите са представени в таблици от 3.1 до 3.5.

За използваните приблизителни методи, на фигури 3.10, 3.11, 3.12, 3.13 и 3.14 са представени и примери за всеки използван метод за настройка.

Таблица 3.1. Аперiodичен преходен процес с минимално време на регулиране (вар. 1).

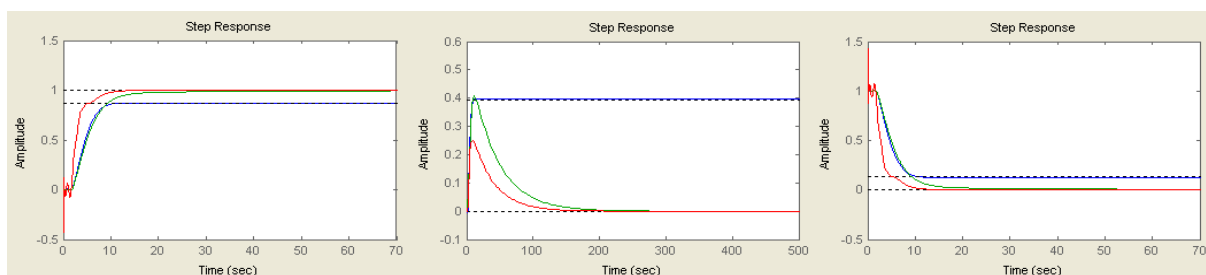
Регулатор	K_P	T_I	T_D
П	$\frac{0.3T_{OB}}{K_{OB}\tau_{OB}}$	-	-
ПИ	$\frac{0.6T_{OB}}{K_{OB}\tau_{OB}}$	$0.6T_{OB}$	-
ПИД	$\frac{0.95T_{OB}}{K_{OB}\tau_{OB}}$	$2.4\tau_{OB}$	$0.4\tau_{OB}$



Фиг. 3.10. Регулиране при използване на формулите от таблица 1

Таблица 3.2. Аperiодичен преходен процес с минимално време на регулиране (вар. 2).

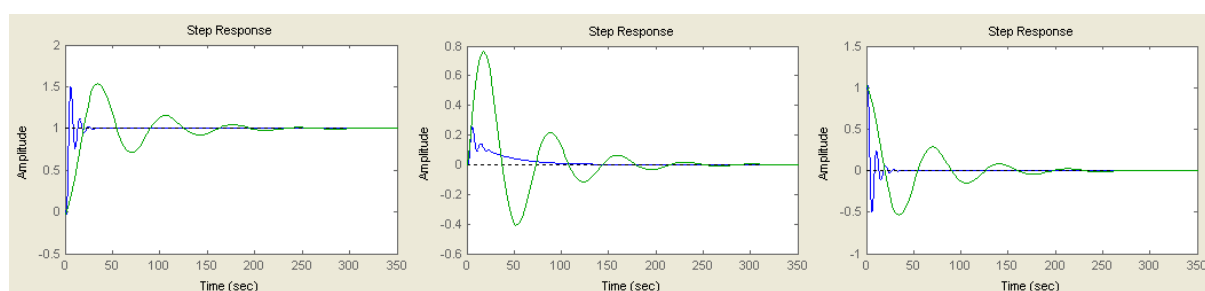
Регулатор	K_P	T_I	T_D
П	$\frac{0.4T_{OB}}{K_{OB}\tau_{OB}}$	-	-
ПИ	$\frac{0.35T_{OB}}{K_{OB}\tau_{OB}}$	$1.17T_{OB}$	-
ПИД	$\frac{0.6T_{OB}}{K_{OB}\tau_{OB}}$	T_{OB}	$0.5\tau_{OB}$



Фиг. 3.11. Регулиране при използване на формулите от таблица 2

Таблица 3.3. Минимална интегрално-квадратична грешка.

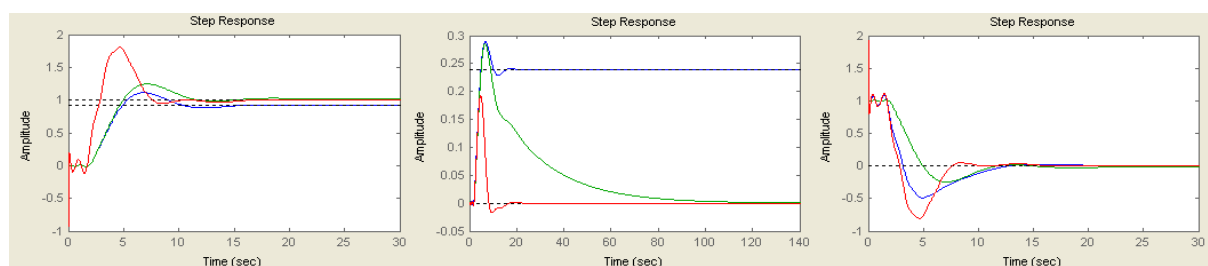
Регулатор	K_P	T_I	T_D
П	-	-	-
ПИ	$\frac{T_{OB}}{K_{OB}\tau_{OB}}$	T_{OB}	-
ПИД	$\frac{1.4T_{OB}}{K_{OB}\tau_{OB}}$	$1.3\tau_{OB}$	$0.5\tau_{OB}$



Фиг. 3.12. Регулиране при използване на формулите от таблица 3

Таблица 3.4. Колебателен преходен процес с 20% пререгулиране и минимално време на регулиране (вар. 1).

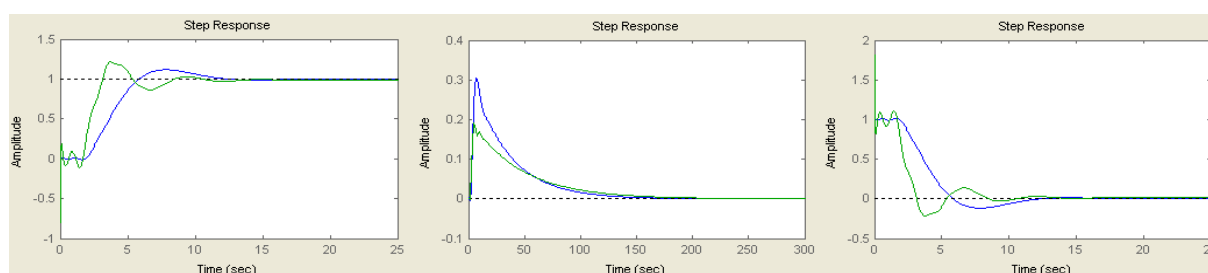
Регулятор	K_P	T_I	T_D
П	$\frac{0.7T_{OB}}{K_{OB}\tau_{OB}}$	-	-
ПИ	$\frac{0.7T_{OB}}{K_{OB}\tau_{OB}}$	$0.7T_{OB}$	-
ПИД	$\frac{1.2T_{OB}}{K_{OB}\tau_{OB}}$	$2\tau_{OB}$	$0.4\tau_{OB}$



Фиг. 3.13. Регулиране при използване на формулите от таблица 4

Таблица 3.5. Колебателен преходен процес с 20% пререгулиране и минимално време на регулиране (вар.1).

Регулатор	K_P	T_I	T_D
П	-	-	-
ПИ	$\frac{0.6T_{OB}}{K_{OB}\tau_{OB}}$	T_{OB}	-
ПИД	$\frac{0.95T_{OB}}{K_{OB}\tau_{OB}}$	$1.36T_{OB}$	$0.47\tau_{OB}$



Фиг. 3.14. Регулиране при използване на формулите от таблица 5

За да е възможно изчертаването на преходния процес, студентът трябва да въведе параметрите на обекта в полетата оградени в панел със заглавие *Параметри на обекта*.

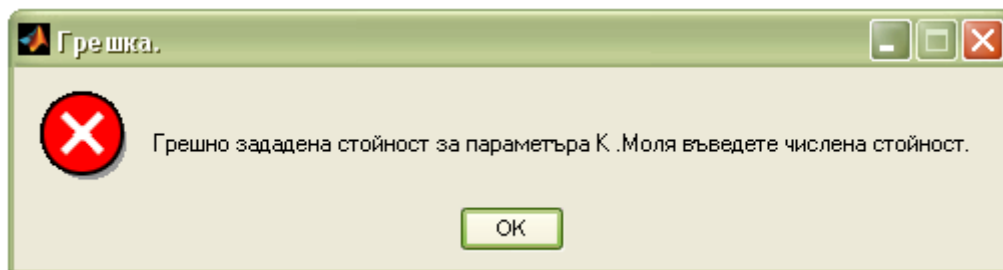
При отсъствие или неправилно зададена стойност за времето се появява текст *Автоматично задаване на времето t*. След правилното въвеждане последвано от бутон *Обновяване*, съобщението се скрива и се изчертават графиките за въведения времеви диапазон.

За да имат ефект въведените промени от страна на потребителите е необходимо да се натисне бутон *Обновяване*. Бутонът *Изчистване* служи за изчистване на:

- вече изчертаните графики;
- на параметрите на регулатора;
- на съобщението за автоматично задаване на времето, ако е имало такова;

- на отметките указващи наслагването и мрежата за двете графики.

При неправилно въвеждане на параметрите на обекта се показва съобщение за грешка – фиг. 3.15.



Фиг. 3.15. Съобщение за грешка при неправилно въвеждане на коефициента K

Работата може да се продължи единствено, когато съобщението бъде опознато от бутона **ОК**. За да не се формират грешни представи у студентите се изпълняват следните действия:

- Графиките се изчистват;
- Параметрите на регулатора се изчистват;
- Съобщението за автоматично задаване на времето се изчиства, ако е имало такова.

В панел със заглавие **Параметри на регулатора** се изобразяват изчислените настроечни параметри. Тези полета не могат да бъдат редактирани от потребителя. Те са само с информационна цел.

Допълнителната функция **наslagване** предоставя възможността да се сравняват едновременно преходни процеси по задание и по смущение:

- При различни регулатори;
- При еднакви регулатори с различни приблизителни формули за настройка;
- При настройки с различни или еднакви параметри на обекта и др.

ЗАКЛЮЧЕНИЕ

В дипломната работа са разработени две системи от графични интерфейси (GUI) в средата на *Matlab*, за целите на обучението по дисциплината Теория на Управлението:

- за изследване на линейни динамични системи във времевата и честотната област
- за настройка и изследване на типови аналогови закони за регулиране.

В дипломната работа е извършено следното:

1. Изучени са възможностите на *Matlab* за създаване, програмиране и настройка на потребителски интерфейси. Методиката за работа със системата е структурирана и обяснена подробно в първа глава на дипломната работа.
2. Във втора глава е разгледан подробно разработеният графичен интерфейс за анализ на динамични системи във времевата и честотната области, както и системи с чисто закъснение. Интерфейсът позволява да се сравняват еднотипни характеристики на различни системи и еднотипни системи при вариране на параметрите им.
3. В трета глава от дипломната работа е представен разработеният графичен интерфейс за настройка и анализ на едноконтурни системи за управление с аналогови П, ПИ и ПИД регулатори. Този интерфейс предоставя на студентите широк спектър от методи за синтез на типови аналогови регулатори,

както и възможности за изследване на едноконтурни системи с различни видове регулатори

4. Чрез примери са показани резултати от работата на двете системи.
5. Разработени са *Help* – файлове, в които е поместено описание на разработените системи на български език.

Чрез средствата на *Matlab* за създаване, програмиране и настройка на потребителски интерфейси, в дипломната работа са проектирани програмни системи за целите на обучението. Системите биха могли да бъдат използвани също и за решаване на типови проектантски и изследователски задачи.

ЛИТЕРАТУРА

1. MATLAB Creating Graphical User Interfaces, MathWorks Web site
2. Matlab Help ver. 7.8.0.347 (R2009a)
3. Burns, R. S., Advanced Control Engineering, Butterworth-Heinemann, 2001
4. Love, J., Process Automation Handbook, Springer-Verlag London, 2007
5. Chau, P.C., Chemical Process Control a first Course with MATLAB, 2001
6. Stephanopoulos, G., Chemical Process Control: An Introduction to Theory and Practice, 1984
7. Еленков, Г., Ръководство за упражнения по основи на автоматичното управление, ИК Юрапел, 1996
8. Хаджийски, М. Б, Автоматизация на технологичните процеси в химическата и металургичната промишленост, Техника, София, 1989
9. Цанев, А., Автоматизация на технологичните процеси, учебни записки, 2008

ПРИЛОЖЕНИЕ 1

```
function varargout = tau1(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @tau1_OpeningFcn, ...
                  'gui_OutputFcn',  @tau1_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
function tau1_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;
axes(handles.axes1);
uctm=imread ('uctm.bmp');
image(uctm)
axis off
axis image
axes(handles.axes2);
G=tf(1,[10 1]);
step(G);
axes(handles.axes4);
ksi=0.2;
k=1;
T=10;
W=tf(k,[T^2 2*ksi*T 1]);
step(W)
guidata(hObject, handles);

function varargout = tau1_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;

function pushbutton1_Callback(hObject, eventdata, handles)
part1

function pushbutton2_Callback(hObject, eventdata, handles)
part2

function pushbutton3_Callback(hObject, eventdata, handles)

function pushbutton5_Callback(hObject, eventdata, handles)
part3

function pushbutton8_Callback(hObject, eventdata, handles)
close
taula
```

ПРИЛОЖЕНИЕ 2

```
function varargout = part1(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn',  @part1_OpeningFcn, ...
                  'gui_OutputFcn',   @part1_OutputFcn, ...
                  'gui_LayoutFcn',   [], ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before part1 is made visible.
function part1_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;
guidata(hObject, handles);
if strcmp(get(hObject, 'Visible'), 'off')
    axes(handles.axes8);
    a=imread ('Aper.bmp');
    image(a)
    axis off
    axis image
    axes(handles.axes9);
    b=imread ('inerc.bmp');
    image(b)
    axis off
    axis image
    axes(handles.axes10);
    c=imread ('difer.bmp');
    image(c)
    axis off
    axis image
end

function varargout = part1_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;

function pushbutton1_Callback(hObject, eventdata, handles)
k = str2double(get(handles.edit1, 'String'));
T = str2double(get(handles.edit2, 'String'));
b=isnan(k);
c=isnan(T);
t1 = str2double(get(handles.edit3, 'string'));
t=0:0.1:t1;
a1=isnan(t);
if b~=1 && c~=1
if t1<=0
t=NaN;
set(handles.text6, 'Visible', 'On')
set(handles.text6, 'String', 'Автоматично задаване на времето t')
```

```

elseif a1>=1
    t=NaN;
    set(handles.text6,'Visible','On')
    set(handles.text6,'String','Автоматично задаване на времето t')
else
    set(handles.text6,'Visible','Off')

end
end
if b==1
    msgbox('Грешно зададена стойност за параметъра К .Моля въведете числена
стойност.', 'Грешка.','error', 'modal');
    cla(handles.axes1)
    cla(handles.axes2)
    set(handles.text6,'Visible','Off')
elseif c==1
    msgbox('Грешно зададена стойност за параметъра Т .Моля въведете числена
стойност.', 'Грешка.','error', 'modal');
    cla(handles.axes1)
    cla(handles.axes2)
    set(handles.text6,'Visible','Off')
end

if (get(handles.radiobutton1,'Value') == get(handles.radiobutton3,'Max'))
    W=tf(k,[T 1]);
end
if (get(handles.radiobutton2,'Value') == get(handles.radiobutton3,'Max'))

    W=tf(k,[T 1 0]);

end
if (get(handles.radiobutton3,'Value') == get(handles.radiobutton3,'Max'))
    W=tf([k 0],[T 1]);
end

if (get(handles.checkbox1,'Value'))==1
    clc
    W
    pause(0)
    workspace
    pause(3)
end
if b~=1 && c~=1
    axes(handles.axes1);
    step(W,t)
    axes(handles.axes2);
    impulse(W,t)
end
if (get(handles.checkbox2,'Value'))==1
    axes(handles.axes1);
    grid on
end
if (get(handles.checkbox4,'Value'))==1
    axes(handles.axes2);
    grid on
end

function radiobutton1_Callback(hObject, eventdata, handles)
function radiobutton2_Callback(hObject, eventdata, handles)

```

```

function radiobutton3_Callback(hObject, eventdata, handles)
function edit1_Callback(hObject, eventdata, handles)

function edit1_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit2_Callback(hObject, eventdata, handles)

function edit2_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function pushbutton2_Callback(hObject, eventdata, handles)
cla(handles.axes1)
cla(handles.axes2)
set(handles.checkbox2,'Value',0)
set(handles.checkbox3,'Value',0)
set(handles.checkbox4,'Value',0)
set(handles.checkbox5,'Value',0)
set(handles.text6,'Visible','Off')

function edit3_Callback(hObject, eventdata, handles)

function edit3_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function checkbox1_Callback(hObject, eventdata, handles)

function checkbox2_Callback(hObject, eventdata, handles)
if (get(hObject,'Value') == get(hObject,'Max'))
    axes(handles.axes1);
    grid on
else
    axes(handles.axes1);
    grid off
end

function checkbox3_Callback(hObject, eventdata, handles)
if (get(hObject,'Value') == get(hObject,'Max'))
    axes(handles.axes1);
    hold on
else
    axes(handles.axes1);
    hold off
end

function checkbox4_Callback(hObject, eventdata, handles)
if (get(hObject,'Value') == get(hObject,'Max'))
    axes(handles.axes2);
    grid on

```

```

else
    axes(handles.axes2);
    grid off
end

function checkbox5_Callback(hObject, eventdata, handles)
if (get(hObject, 'Value') == get(hObject, 'Max'))
    axes(handles.axes2);
    hold on
else
    axes(handles.axes2);
    hold off
end

function pushbutton3_Callback(hObject, eventdata, handles)
function FileMenu_Callback(hObject, eventdata, handles)
function HelpMenu_Callback(hObject, eventdata, handles)

function AboutMenuItem_Callback(hObject, eventdata, handles)
About

function MatlabHelpMenuItem_Callback(hObject, eventdata, handles)
helpbrowser

function OpenMenuItem_Callback(hObject, eventdata, handles)
file = uigetfile('*.fig');
if ~isequal(file, 0)
    open(file);
end

function PrintMenuItem_Callback(hObject, eventdata, handles)
printdlg(handles.figure1)

function CloseMenuItem_Callback(hObject, eventdata, handles)
selection = questdlg(['Close ' get(handles.figure1, 'Name') '?'], ...
                    ['Close ' get(handles.figure1, 'Name') '...'], ...
                    'Yes', 'No', 'Yes');
if strcmp(selection, 'No')
    return;
end

delete(handles.figure1)

function pushbutton5_Callback(hObject, eventdata, handles)
example2

function pushbutton4_Callback(hObject, eventdata, handles)
example1

function pushbutton6_Callback(hObject, eventdata, handles)
example3

function GUIHelpMenuItem_Callback(hObject, eventdata, handles)
open('gra.pdf')

```