**УНИВЕРЗИТЕТ “Св. КИРИЛ И МЕТОДИЈ” - СКОПЈЕ**

**ФАКУЛТЕТ ЗА ЕЛЕКТРОТЕХНИКА И ИНФОРМАЦИСКИ ТЕХНОЛОГИИ**

ПРОЕКТНА ЗАДАЧА ПО ПРЕДМЕТОТ МОДЕЛИРАЊЕ, ИДЕНТИФИКАЦИЈА И СИМУЛАЦИЈА НА ТЕМА:

**АПРОКСИМАЦИЈА НА ДИНАМИЧКИОТ МОДЕЛ НА ЛИНЕАРЕН КОНТИНУАЛЕН СИСТЕМ СО КОРИСТЕЊЕ НА SIMULINK И MATLAB**

Благој Христов - 157/2016

*Скопје, Февруари 2019*

С О Д Р Ж И Н А

[**АПСТРАКТ** 4](#_Toc536755485)

[**ВОВЕД** 5](#_Toc536755486)

[Што е идентификација? 5](#_Toc536755487)

[Како од SIMULINK модел до динамички модел на системот? 6](#_Toc536755488)

[Цел на проектната задача 6](#_Toc536755489)

[**1. МОДЕЛИРАЊЕ НА СИСТЕМОТ И ПРЕТСТАВУВАЊЕ ВО SIMULINK** 7](#_Toc536755490)

[1.1 ДИНАМИЧКИ МОДЕЛ НА СИСТЕМОТ 7](#_Toc536755491)

[1.2 ПРЕТСТАВУВАЊЕ НА СИСТЕМОТ ВО SIMULINK 8](#_Toc536755492)

[1.3 ДОБИВАЊЕ НА ПОДАТОЦИ СО СИМУЛАЦИЈА НА SIMULINK МОДЕЛ 9](#_Toc536755493)

[**2. ИДЕНТИФИКАЦИЈА НА СИСТЕМОТ** 11](#_Toc536755494)

[2.1 АПРОКСИМАЦИЈА НА ДИСКРЕТЕН МОДЕЛ КОРИСТЕЈЌИ ГИ ПОДАТОЦИТЕ ОД СИМУЛАЦИЈАТА 11](#_Toc536755495)

[2.2 ИМПУЛСНА АНАЛИЗА НА ПОДАТОЦИТЕ 12](#_Toc536755496)

[2.3 УСОВРШУВАЊЕ НА АПРОКСИМАЦИЈАТА 13](#_Toc536755497)

[2.4 ПРЕТВОРАЊЕ НА АПРОКСИМИРАНИОТ ДИСКРЕТЕН МОДЕЛ ВО КОНТИНУАЛЕН 15](#_Toc536755498)

[**3. ДИРЕКТНА АПРОКСИМАЦИЈА НА КОНТИНУАЛЕН МОДЕЛ** 16](#_Toc536755499)

[3.1 АНАЛИЗА НА НЕИЗВЕСНОСТ НА ПАРАМЕТРИТЕ 17](#_Toc536755500)

[3.1.1 ЕСТИМАЦИЈА НА МОДЕЛОТ ВО КАНОНСКА ФОРМА 17](#_Toc536755501)

[3.1.2 АНАЛИЗА НА ГРАНИЦИТЕ НА ДОВЕРБА НА ОДЗИВОТ НА МОДЕЛОТ 18](#_Toc536755502)

[3.2 АНАЛИЗА НА ОДНЕСУВАЊЕТО НА АПРОКСИМИРАНИОТ МОДЕЛ ИЗМЕЃУ ПЕРИОДИТЕ НА ДИСКРЕТИЗАЦИЈА 19](#_Toc536755503)

[**ЗАКЛУЧОК** 21](#_Toc536755504)

[**КОРИСТЕНА ЛИТЕРАТУРА** 22](#_Toc536755505)

Л И С Т А Н А С Л И К И

[**Слика 1.** Скица на реалниот механички модел 7](#_Toc536755623)

[**Слика 2.** SIMULINK претстава на системот 8](file:///C:\Users\Blagoj\Desktop\Проект%20МИС%20-%20Благој%20Христов%20157_2016.docx#_Toc536755624)

[**Слика 3**. Графичка претстава на излезниот и управувачкиот сигнал 10](#_Toc536755625)

[**Слика 4.** Споредба на добиениот модел со реалните податоци 11](#_Toc536755626)

[**Слика 5.** Естимација на импулсен одзив на системот 12](#_Toc536755627)

[**Слика 6.** Споредба на апроксимираните модели со реалните податоци 13](#_Toc536755628)

[**Слика 7.** Графички приказ на нулите и половите на апроксимираниот модел 14](#_Toc536755629)

[**Слика 8.** Споредба на апроксимираните модели со реалните податоци 15](#_Toc536755630)

[**Слика 9.** Бодеов дијаграм на апроксимираните континуални модели   
и реалниот систем 18](#_Toc536755631)

[**Слика 10.** Отскочен одзив на апроксимираните континуални модели   
и реалниот систем 19](#_Toc536755632)

[**Слика 11.** SIMULINK претстава на системот без форматор од нулти ред 19](#_Toc536755633)

[**Слика 12.** Отскочен одзив на апроксимираниот модел и реалниот систем 20](#_Toc536755634)

[**Слика 13.** Отскочен одзив на апроксимираниот модел и реалниот систем 21](#_Toc536755635)

# АПСТРАКТ

Проблем со кој често се среќаваме при анализата на системите во реалноста е тоа што не секогаш може да се добие математичкиот модел на системот со постапките за моделирање. За таа цел потребно е да се изврши постапка на идентификација при која со набљудување на влезно-излезните податоци од системот, се добива проценка на неговиот динамички модел. Во овој труд се разгледува пример каде даден систем, кој е претставен со негова соодветна шема како SIMULINK модел, се апроксимира со користење на *System Identification Toolbox* алатката од програмскиот пакет MATLAB, и како добиениот модел може да се усоврши со понатамошна анализа.

# ВОВЕД

## Што е идентификација?

Во практиката, не секогаш може да се добие математичкиот динамички модел на системот со постапка на моделирање бидејќи не секогаш ни е позната внатрешноста на системот, како тој работи и кои се неговите состојбени променливи. Поради ова потребно е да се изврши некој тип постапка на идентификација при која се разгледува излезниот сигнал од системот во зависност од произволно избран влезен сигнал. Со анализа на добиените податоци и користејќи одредени методи и постапки може да се добие естимација за моделот на динамиката на системот, кој е доволно точен во однос на реалниот модел за да може да се користи за понатамошна работа.

Постојат повеќе разни постапки на идентификација кои можат да се применат за добивање на апроксимираниот модел на системот. За идентификација на континуалните детерминистички системи, каков што е системот што ќе се разгледува во овој труд, некои од постапките кои можат да се применат се:

* временска идентификација
* фреквенциска идентификација
* проценка со користење на методот на најмали квадрати

При временската идентификација се разгледуваат податоци добиени со директно отчитување на излезниот сигнал на системот во зависност од времето и притоа со примена на математички операции се добиваат половите на системот, а од нив и неговата динамика.

При фреквенциската идентификација се разгледуваат податоци добиени со мерење на фреквенциската карактеристика на системот во зависност од фреквенцијата, а со примена на конкретни методи и постапки се добива преносната функција на системот.

Со методот на најмали квадрати поведението на системот се естимира со некоја функција која ја претставува зависноста на податоците добиени со мерење на влезниот и излезниот сигнал на системот. Типот на функцијата (линеарна, квадратна, кубна...) се претпоставува произволно, а се избира онаа функција која најдобро ја опишува зависноста на податоците. Оваа постапка уште е позната и под името регресиона анализа.

Функциите кои се користат во овој труд работат врз основа на временска идентификација и со нив се добива апроксимиран модел на системот во просторот на состојби.

## Како од SIMULINK модел до динамички модел на системот?

SIMULINK претставува графичка програмска околина за моделирање, симулација и анализа на повеќевеличински динамички системи. Заедно со програмскиот пакет MATLAB и услугите од алатката *System Identification Toolbox,* може да се изврши симулација како би се одвивала анализата и идентификацијата на еден реален систем.

*System Identification Toolbox* ни овозможува да користиме функции кои ни помагаат при анализата на системот. Идентификација на системот во овој труд ќе биде претставена според две различни постапки, базирани на следните чекори:

* *Постапка 1:*
  + претставување на системот и неговиот управувач како SIMULINK модел
  + добивање на вистинскиот модел на системот за понатамошна споредба
  + добивање на апроксимиран дискретен модел на системот во просторот на состојби
  + споредба и усовршување на апроксимираниот дискретен модел
  + претворање на дискретниот апроксимиран модел во континуален модел во просторот на состојби
* *Постапка 2:*
  + директно добивање на апроксимираниот континуален модел во просторот на состојби
  + анализа на неизвесност на параметрите
  + анализа на однесувањето на апроксимираниот модел измеѓу периодите на дискретизација

Добиениот апроксимиран модел не мора да значи дека е најдобриот модел кој што може да се добие со идентификација, но тој е доволно точен за секоја практична примена со што не е потребно да се извршува дополнителна анализа за подобрување на моделот.

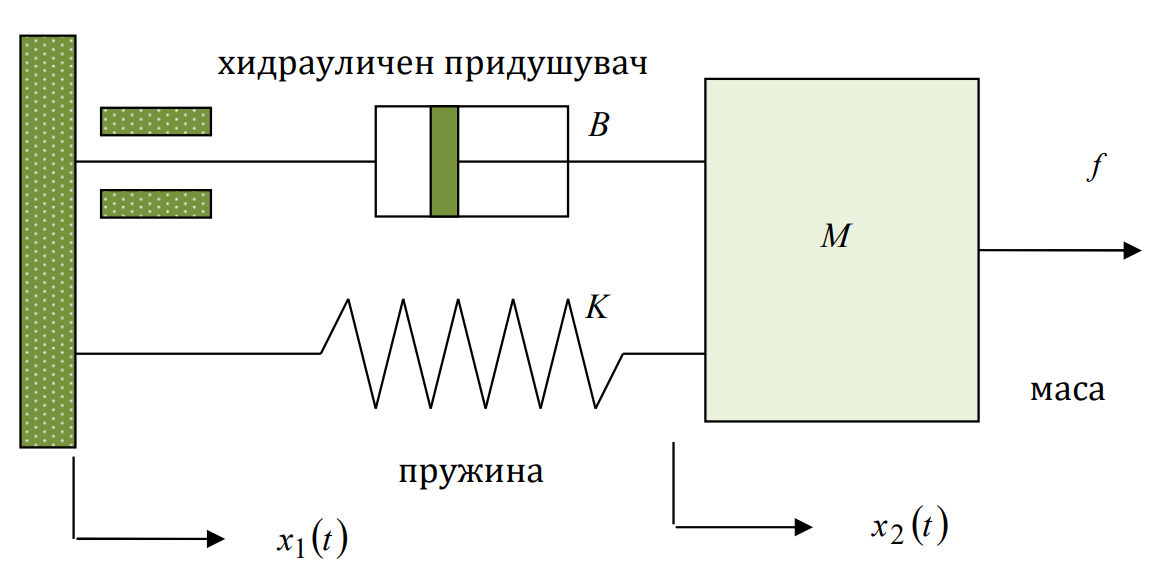
## Цел на проектната задача

Во оваа проектна задача е претставена симулација на реален процес на идентификација на некој систем чии што модел не е познат. Постапките и методите кои се прикажани подолу можат да се применат и на реален систем доколку може прецизно да се измери и отчита неговиот излезен сигнал. За поедноставување на анализата, системот врз кој се применува постапката на идентификација е земен произволно и тој претставува едноставен механички систем.

# 1. МОДЕЛИРАЊЕ НА СИСТЕМОТ И ПРЕТСТАВУВАЊЕ ВО SIMULINK

## 1.1 ДИНАМИЧКИ МОДЕЛ НА СИСТЕМОТ

Пред да ја започнеме постапката на идентификација, потребно е да се добие моделот на системот кој ќе се користи како референтен модел при анализата, како и тој да се претстави во SIMULINK. За поедноставена анализа, произволно како систем се избира следниот механички модел составен од маса, пружина и хидрауличен придушувач:



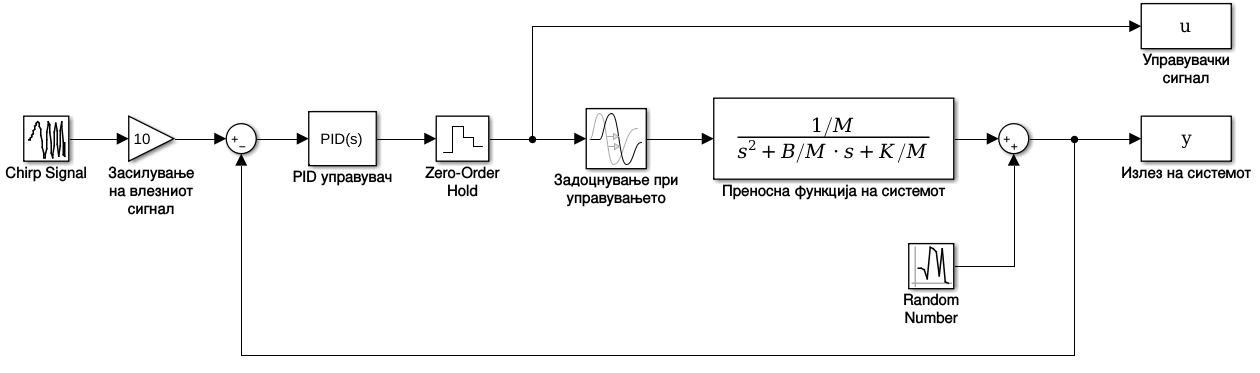
Слика 1. Скица на реалниот механички модел

Овој систем може едноставно да се моделира со користење на Њутнова механика, при што за неговиот модел, претставен преку диференцијална равенка, добиваме:

Со Лапласова трансформација на горната диференцијална равенка, под претпоставка на нулеви почетни услови, го добиваме моделот на системот во комплексното подрачје, а преку него ја добиваме и неговата преносна функција:

## 1.2 ПРЕТСТАВУВАЊЕ НА СИСТЕМОТ ВО SIMULINK

При претставување на системот како SIMULINK модел, покрај самиот систем се додава PID управувач, како и дополнителни блокови за реалното задоцнување кое е застапено во реалниот систем. На наредната слика е претставен моделот, а подолу е дадено објаснување каква функција има секој од блоковите и која е нивната улога во процесот на идентификација.



Слика 2. SIMULINK претстава на системот

Како влезен сигнал за системот е земен синусоидален “chirp” сигнал, претставен со блокот ***‘Chirp Signal’*** кој ја зголемува својата фреквенција со текот на времето се додека не достигне некоја претходно одредена посакувана фреквенција. Во конкретниот случај, за почетната фреквенција на сигналот е земена 0.001 Hz, крајната фреквенција е 0.2 Hz, а времето за кое ја достигнува е 500 секунди. Бидејќи сигналот е променливо периодичен, овозможува да се добие прецизна слика за динамиката на системот при различни фреквенции, а поради неговата периодичност овозможува да се внесат минимални промени на случајно избрани точки во излезниот сигнал, кои ќе бидат потребни за валидација при процесот на идентификација. Притоа, амплитудата на влезниот сигнал е засилена на 10 со цел да се добие попрегледен и постабилен излезен сигнал.

Управувачкиот сигнал на системот се остварува со користење на PID управувач претставен со блокот ***‘PID(s)’***, а тој претставува обичен интегратор со параметри *P = 0,   
I = 1, D = 0,* *N = 0.*Соодветно, преносната функција на управувачкиот блок ќе биде:

Сигналот потоа минува низ форматор од нулти ред, кој притоа внесува доцнење во системот. Вредностите од управувачкиот сигнал се зачувуваат во променлива *u.*

Блокот ***‘Задоцнување при управувањето’*** се користи за претставување на реалното задоцнување кое се внесува во системот поради управувањето, и тој внесува дополнително доцнење од 1 секунда.

Преносната функција на системот во комплексно подрачје е претставена со блокот ***‘Преносна функција на системот’.***

Блокот ***‘Random Number’*** се користи за додавање на некоја случајно избрана вредност на секоја точка од сигналот, така што доколку се промени “семето” на генераторот се менуваат и вредностите кои се додаваат. Ова овозможува со промена на “семето” да се добијат различни, но доволно слични, излезни сигнали со цел еден од можните сигнали да го искористиме за добивање на апроксимираниот модел на системот, додека друг да го користиме за да споредиме колку е добра апроксимацијата. Варијансата на броевите кои што се додаваат е мала, во конкретниот случај таа изнесува 0.01, додека времето на земање примероци е 0.5 секунди. Вредностите од излезниот сигнал се зачувуваат во променлива *y*.

Како што може да се забележи од моделот, намерно се внесуваат задоцнувања во системот бидејќи во реалноста тие постојат и тие мора да бидат претставени при симулацијата. Доколку тие не се вметнат, тогаш при постапката на идентификација грешката кумулативно ќе се зголемува поради нивното отсуство и добиениот апроксимиран модел нема да биде задоволително точен.

Со првата постапка на идентификација која ќе биде применета во овој труд се добива еквивалентниот дискретен модел на континуалниот систем, но потоа тој може да се претвори во континуален со одредена трансформација, додека со втората директно се добива континуалниот модел на системот, кој потоа се анализира и усовршува.

## 1.3 ДОБИВАЊЕ НА ПОДАТОЦИ СО СИМУЛАЦИЈА НА SIMULINK МОДЕЛ

Користејќи одредени функции од алатката *System Identification Toolbox*, може да се претстави системот како *idpoly* структура која ќе го претставува реалниот модел на линеарниот континуален систем. Тоа се остварува со наредните наредби:

M = 10;

B = 5;

K = 10;

num = [0 0 1/M];

den = [1 B/M K/M];

m0 = idpoly(1, num, 1, 1, den, ‘Ts’, 0, ‘InputDelay’, 1, ‘NoiseVariance’, 0.01)

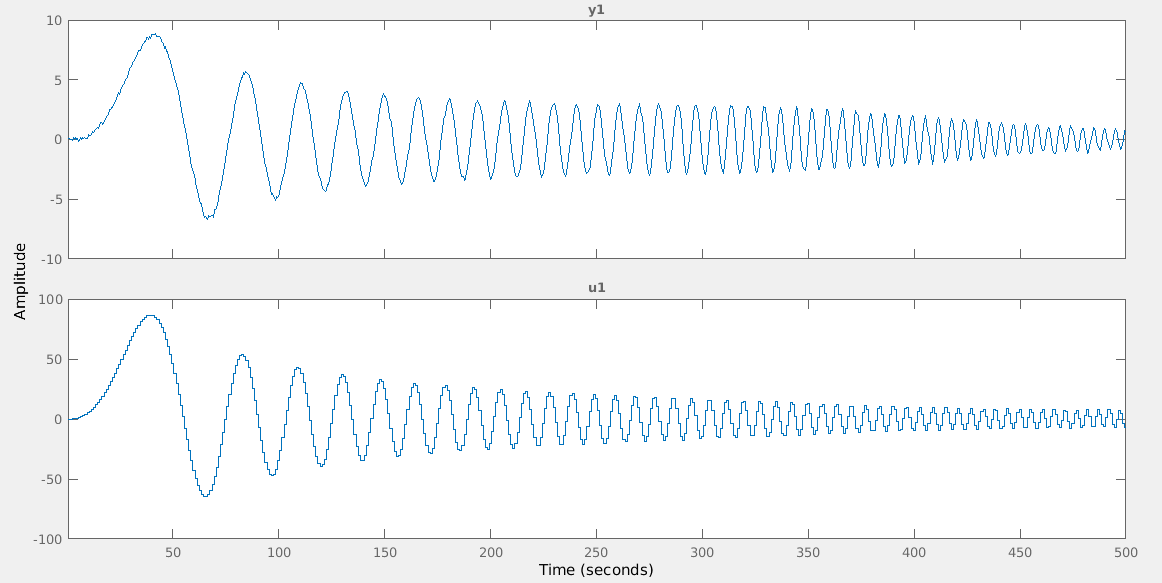
За да се добијат податоците од излезниот и управувачкиот сигнал на системот од SIMULINK моделот, треба да се отчитаат променливите *u* и *y*, и притоа да се создаде *iddata* структура. Тоа е претставено со следните наредби:

set\_param(‘SEMINARSKA\_MODEL/Random Number’, ‘seed’, ‘0’)

sim(‘SEMINARSKA\_MODEL’)

dat1\_est = iddata(y, u, 0.5) *% 0.5 e vreme na zemanje primeroci*

plot(dat1\_est)



Слика 3. Графичка претстава на излезниот и управувачкиот сигнал

Од горниот график може да се забележи дека излезниот сигнал како и управувачкиот имаат променлива периодична природа која зависи од влезниот “chirp” сигнал. Со ова се добива едно множество на влезно-излезни податоци и за излезниот и за управувачкиот сигнал. Со цел да може во понатамошната анализа да се спореди колку е добар апроксимираниот модел што ќе се добие со постапката на идентификација, потребно е да се извлече уште едно множество на влезно-излезни податоци од истиот модел, но со различно “семе” на генераторот за случајни вредности:

set\_param(‘SEMINARSKA\_MODEL/Random Number’, ‘seed’, ‘157’)

sim(‘SEMINARSKA\_MODEL’)

dat1\_val = iddata(y, u, 0.5) *% 0.5 e vreme na zemanje primeroci*

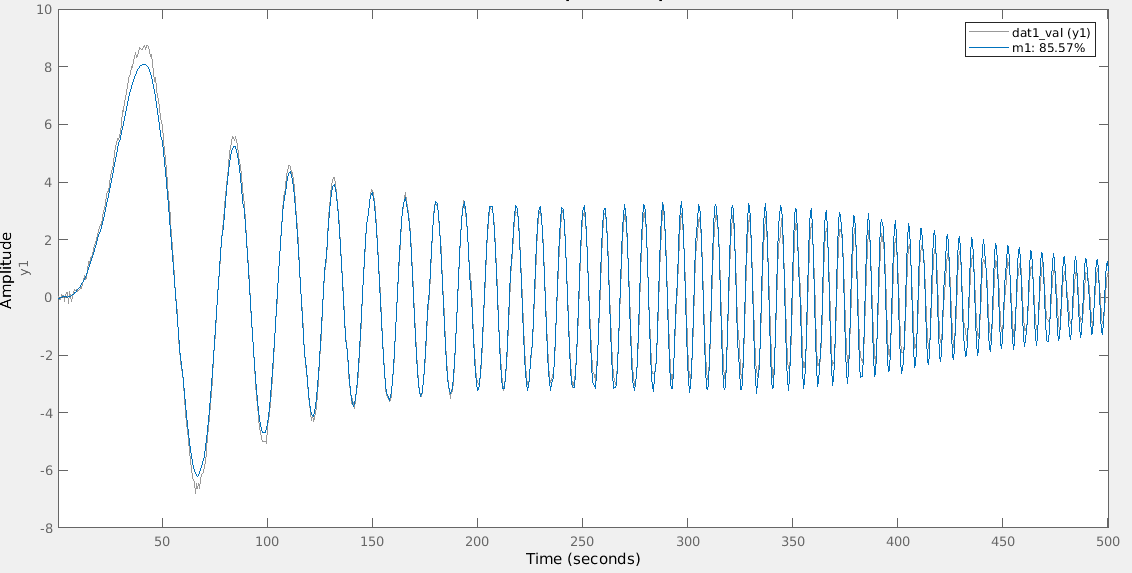
# 2. ИДЕНТИФИКАЦИЈА НА СИСТЕМОТ

## 2.1 АПРОКСИМАЦИЈА НА ДИСКРЕТЕН МОДЕЛ КОРИСТЕЈЌИ ГИ ПОДАТОЦИТЕ ОД СИМУЛАЦИЈАТА

За идентификација на системот се користи функцијата *n4sid* од алатката *System Identification Toolbox.* Таа извршува проценка на моделот на системот од множество на влезно-излезни податоци врз база на *“subspace method”* при кој се добива модел на системот во просторот на состојби. Во конкретниот случај, добиениот модел ќе биде дискретен иако системот е континуален. Функцијата има повеќе променливи параметри, но може да се остави самата автоматски да ги нагоди параметрите со користење на повикот *‘best’*. По добивање на апроксимираниот модел на системот се извршува графичка споредба на податоците за валидација кои ги добивме во претходната глава и одзивот на новодобиениот модел со наредбата *compare*.

m1 = n4sid(dat1\_est, ‘best’)

compare(dat1\_val, m1)

****

Слика 4. Споредба на добиениот модел со реалните податоци

Од графикот може да се воочи дека апроксимираниот модел релативно добро ги следи измерените податоци, со точност од 85.57 %, но не значи дека истата точност ќе се постигне при модификација на влезниот сигнал или при друг тип на влезен сигнал. За таа цел, се извршува дополнителна анализа и усовршување на моделот.

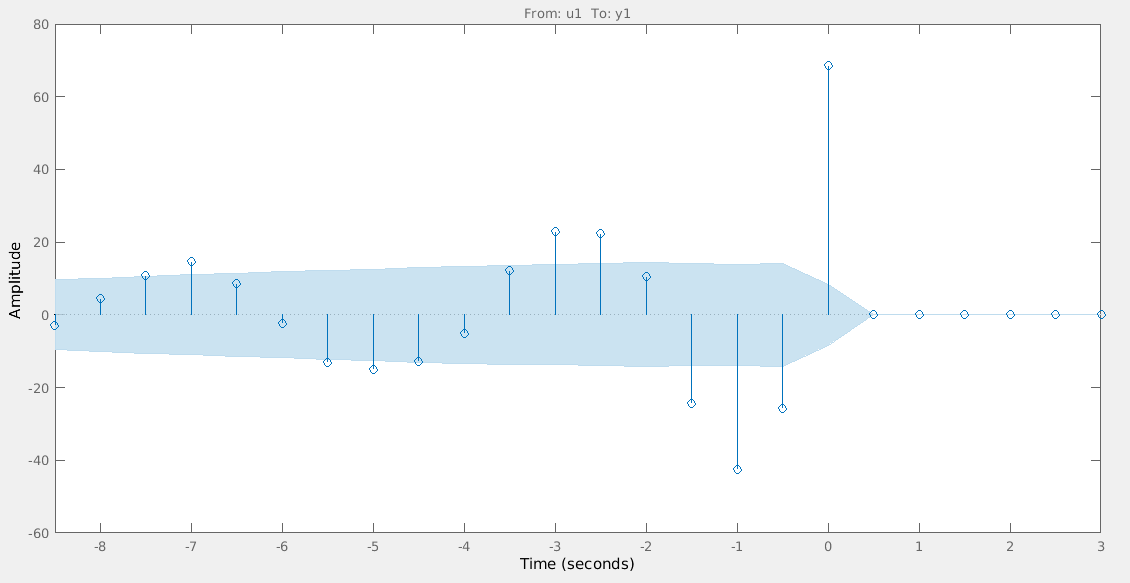
## 2.2 ИМПУЛСНА АНАЛИЗА НА ПОДАТОЦИТЕ

Пред да се обидеме да го подобриме моделот, ќе извршиме импулсна анализа на влезно-излезните податоци, односно како тие се однесуваат при импулсен влез. Импулсниот модел се добива со наредбата *impulseest*, а неговата графичка претстава се добива со наредбата *impulseplot.* Во графикот е вметне и регион на доверливост кој одговара на 3 стандардни девијации:

impulse\_model = impulseest(dat1\_est, [], ‘negative’)

h = impulseplot(impulse\_model)

showConfidence(h, 3)

****

Слика 5. Естимација на импулсен одзив на системот

Добиениот модел е конечен импулсен модел чии коефициенти автоматски се определуваат. Целта на импулсниот модел е за да се одредат негативните задоцнувања кои потоа дополнително ќе се анализираат. Бидејќи влијанијата на негативните задоцнувања не се занемарливи поради управувачкиот сигнал во системот (повратната врска), естимацијата на импулсниот одзив не е доволна за да се одреди вкупното временското задоцнување во системот. Поради ова се користат повеќе ARX – модели од низок степен со различни задоцнувања со цел да се најде оној кој најдобро се совпаѓа со реалното задоцнување:

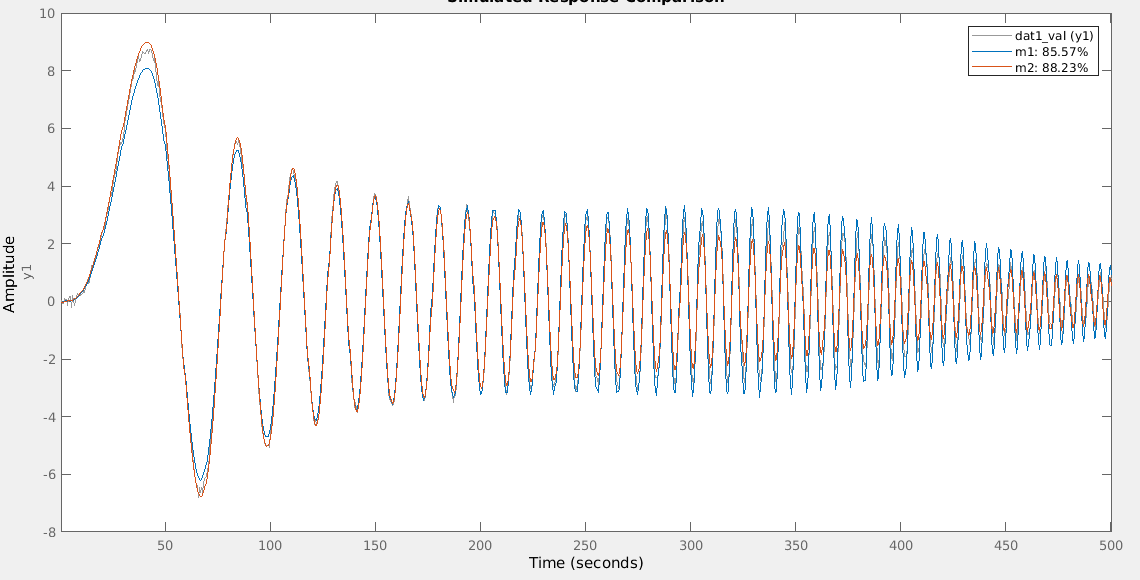
V = arxstruc(dat1\_est, dat1\_val, struc(1:2, 1:2, 1:10));

nn = selstruc(V, 0) % zadocnuvanjeto e tretiot element od nn

Од добиената променлива *nn* се воочува дека задоцнувањето во системот е 3 единици (што е точно, од блокот за задоцнување 1 секунда што е еквивалентно на две единици од по 0.5 секунди – времето на земање примероци, и од форматорот од нулти ред уште една). Потоа, соодветниот ARX – модел може да се добие и да се спореди со претходниот апроксимиран модел *m1:*

m2 = arx(dat1\_est, nn)

compare(dat1\_val, m1, m2);

****

Слика 6. Споредба на апроксимираните модели со реалните податоци

Од графикот се воочува дека новодобиениот апроксимиран модел *m2* иако уште поточно ги следи влезно-излезните податоци, со точност од 88.23%, сепак може дополнително да се усоврши.

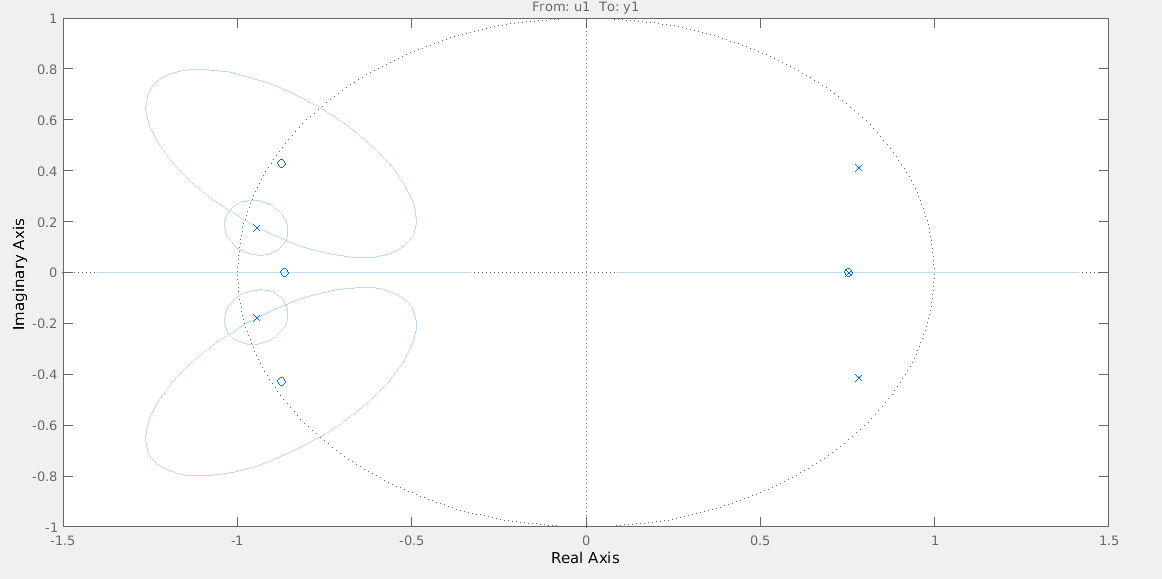
## 2.3 УСОВРШУВАЊЕ НА АПРОКСИМАЦИЈАТА

И двата модела *m1* и *m2* имаат релативно слична точност до реалниот модел при симулацијата. Апроксимацијата може уште повеќе да се усоврши со прецизно подесување на редот на апроксимираниот модел и задоцнувањата. За моделот *m1* при користење на наредбата *n4sid* не внесовме параметри со кои да се изврши апроксимацијата, туку го користевме повикот *‘best’* кој автоматски ги одреди параметрите. Со новите податоци кои ги дознавме за задоцнувањата во системот, може одново да се создаде модел со истата наредба, но притоа параметрите кои го диктираат задоцнувањето да се подесат така што да во системот постои доцнење од точно три временски единици. Параметарот кој го диктира редот на системот се уште ќе оставиме автоматски да биде избран од страна на самата функција.

m3 = n4sid(dat1\_est, ‘best’, ‘InputDelay’, 2, ‘Feedthrough’, false)

m3 = pem(dat1\_est, m3)

Задоцнувањето се внесува преку параметарот *‘InputDelay’*, каде внесуваме доцнење од 2 единици, а останатата 1 единица се внесува со параметарот *‘Feedthrough’* кој се поставува на *false*. Со наредбата *pem* се извршува дополнително усовршување на апроксимираниот модел користејќи постапка за минимизација на грешката. Доколку се погледне добиениот модел, ќе забележиме дека функцијата автоматски избрала динамика од петти степен, иако системот е од втор ред. Бидејќи веќе ја знаеме динамиката на системот, знаеме дека моделот не е точен. Но, во реалноста каде не ни е познат редот на системот, може да провериме дали функцијата добро го избрала редот на системот со анализа на нулите и половите на апроксимираниот модел, претставени во комплексната рамнина заедно со региони на доверливост кои одговараат на 5 стандардни девијации, колку што е редот на моделот:

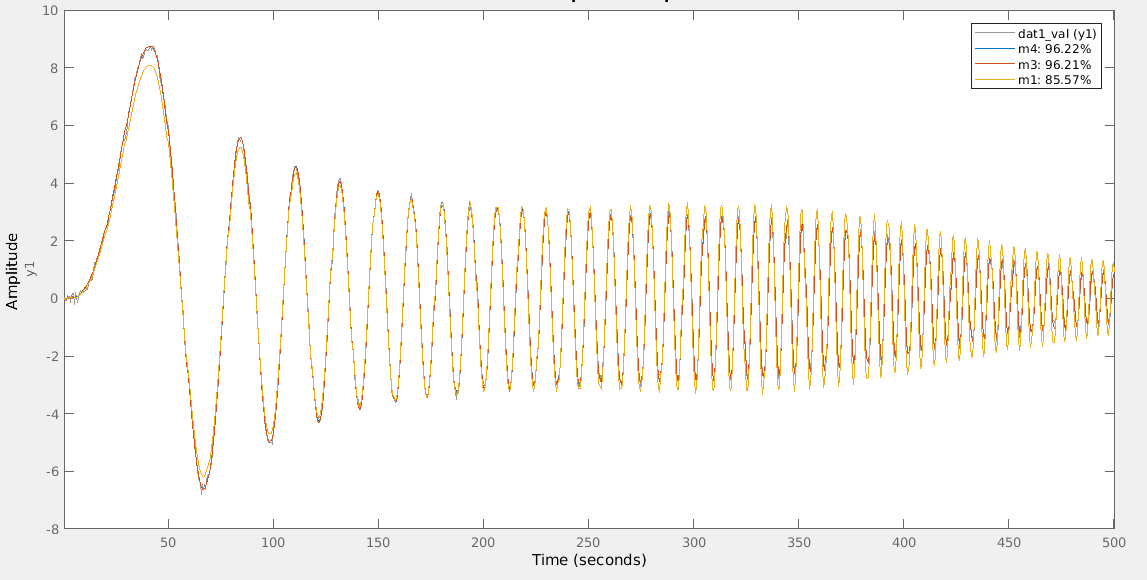
****

Слика 7. Графички приказ на нулите и половите на апроксимираниот модел

Од графикот може да се заклучи дека регионите на доверливост на три од половите и три од нулите се преклопуваат, па тие може да се поништат без настане голема промена во динамиката на системот, со што ќе се добие модел кој има два полови и една нула. Од ова дознаваме дека системот може да се претстави (релативно точно) со динамика од втор ред, а во нашиот случај знаеме и дека тоа е точниот ред на реалниот систем. Крајниот апроксимиран модел и графичката споредба со реалните податоци ги добиваме со следните наредби:

m4 = ssest(dat1\_est, 2, ‘InputDelay’, 2, ‘Feedthrough’, false,‘Ts’, dat1\_est.Ts)

compare(dat1\_val, m4, m3, m1)

****

Слика 8. Споредба на апроксимираните модели со реалните податоци

Анализирајќи го графикот заклучуваме дека моделите *m3* и *m4* најдобро го опишуваат реалниот систем и иако нивната точност е речиси потполно иста, моделот *m4* има динамика од ист ред со реалниот систем. Поради ова, заклучуваме дека моделот *m4* е најдобрата и најточната апроксимација на системот.

## 2.4 ПРЕТВОРАЊЕ НА АПРОКСИМИРАНИОТ ДИСКРЕТЕН МОДЕЛ ВО КОНТИНУАЛЕН

Преостанува уште добиениот дискретен модел *m4* да го претвориме во континуален. Тоа се извршува многу едноставно со повикување на функцијата *d2c*, со што го добиваме еквивалентниот континуален модел во просторот на состојби на апроксимираниот систем *m4*, а за да ја добиеме неговата преносна функција ја користиме наредбата *idtf*. Конечната апроксимирана преносна функција која се добива со постапката на идентификација е:

Доколку ја споредиме добиената преносна функција со реалната:

Може да се воочи дека коефициентот пред *s* членот во броителот е доволно мал за да може да се изедначи со нула, а останатите коефициенти се доволно блиски до вистинските вредности на реалниот систем, па апроксимираниот модел ги задоволува бараните критериуми на точност.

# 3. ДИРЕКТНА АПРОКСИМАЦИЈА НА КОНТИНУАЛЕН МОДЕЛ

## 

Со претходната постапка го естимиравме еквивалентниот дискретен модел на реалниот континуален систем, па потоа го трансформиравме добиенот модел во континуален. Постои и постапка директно да се апроксимира континуален модел со користење на наредбата *ssest*:

m5 = ssest(dat1\_est, 2, ‘Feedthrough’, false, ‘InputDelay’, 1);

present(m5)

Дискретниот модел *m4* содржеше две единици доцнење кои заедно претставуваа задоцнување од една секунда, поради што параметарот *‘InputDelay*’ го подесуваме да биде 1.

m5 =

Continuous-time identified state-space model:

dx/dt = A x(t) + B u(t) + K e(t)

y(t) = C x(t) + D u(t) + e(t)

A =

x1 x2

x1 -0.5397 +/- 2.687e+12 1.262 +/- 3.009e+12

x2 -0.81 +/- 1.614e+12 0.04033 +/- 2.687e+12

B =

u1

x1 0.001379 +/- 1.972e+10

x2 0.003202 +/- 1.366e+10

C =

x1 x2

y1 26.85 +/- 1.011e+14 -11.34 +/- 1.661e+14

D =

u1

y1 0

Estimated using SSEST on time domain data "dat1\_est".

Fit to estimation data: 96.28% (prediction focus)

## 3.1 АНАЛИЗА НА НЕИЗВЕСНОСТ НА ПАРАМЕТРИТЕ

Добиениот модел содржи голема неизвесност на параметрите иако тој има точност од 96.28%. Ова настанува поради тоа што моделот користи повеќе параметри од што му се апсолутно потребни, што доведува до губење на уникатноста на проценката на параметрите. За да се види вистинскиот ефект од оваа неизвесност, можни се два пристапи:

* да се разгледа неизвесноста како граници на доверба на одзивот на моделот наместо на параметрите
* да се естимира моделот во канонска форма

### 3.1.1 ЕСТИМАЦИЈА НА МОДЕЛОТ ВО КАНОНСКА ФОРМА

За естимација на канонската форма на моделот повторно се користи наредбата *ssest*, но сега се додава повикот *‘Form’:*

m5\_canon = ssest(dat1\_est, 2, ‘Feedthrough’, false, ‘InputDelay’, 1,

‘Form’, ‘canonical’ );

m5\_canon =

Continuous-time identified state-space model:

dx/dt = A x(t) + B u(t) + K e(t)

y(t) = C x(t) + D u(t) + e(t)

A =

x1 x2

x1 0 1

x2 -1.001 +/- 0.0033 -0.4993 +/- 0.002094

B =

u1

x1 0.0006886 +/- 0.0005252

x2 0.09975 +/- 0.0005839

C =

x1 x2

y1 1 0

D =

u1

y1 0

Моделот *m5\_canon* претставува канонска параметризација на моделот *m5*. Неговата точност е иста како и на самиот модел *m5*. Но, за разлика од него, неизвесноста на параметрите е забележително многу помала, што дава доказ за неговата доверливост. Сепак, како што видовме кај моделот *m5*, голема неизвесност не мора да значи дека моделот е “лош”.

### 3.1.2 АНАЛИЗА НА ГРАНИЦИТЕ НА ДОВЕРБА НА ОДЗИВОТ НА МОДЕЛОТ

Со цел да се процени вистинскиот квалитет на овие модели, потребно е да се разгледаат нивните одзиви во временски и фреквентен домен со региони на доверливост кои одговараат на 3 стандардни девијации. Во анализата е вклучен и реалниот оригинален модел *m0* за споредба. Најпрво се разгледува Бодеовиот дијаграм на моделите:

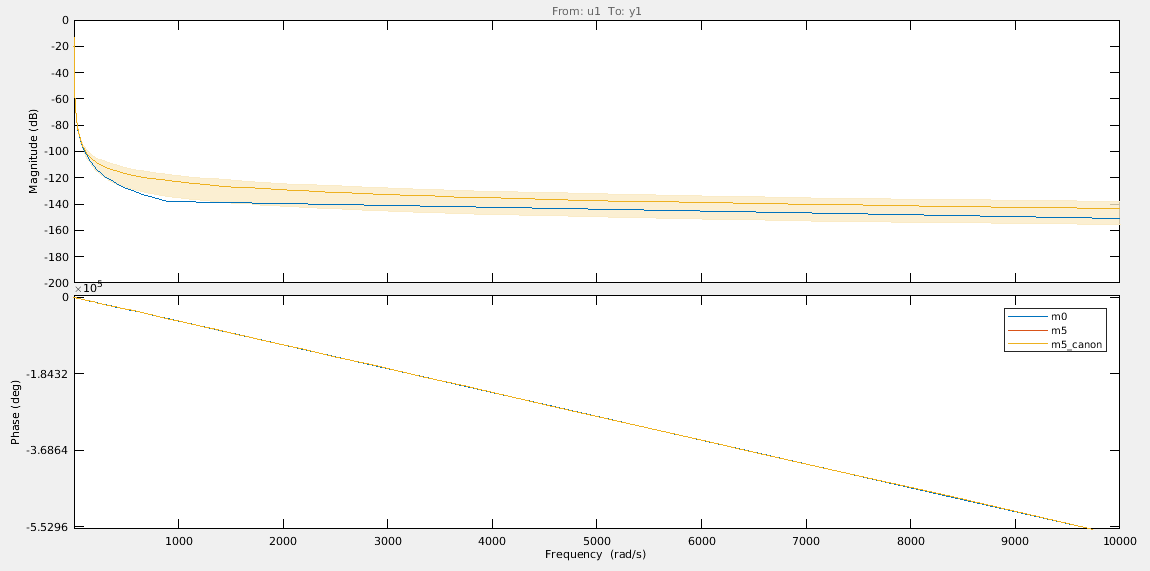
opt = bodeoptions;

opt.FreqScale = ‘linear’;

h = bodeplot(m0, m5, m5\_canon, opt);

showConfidence(h, 3)

legend show

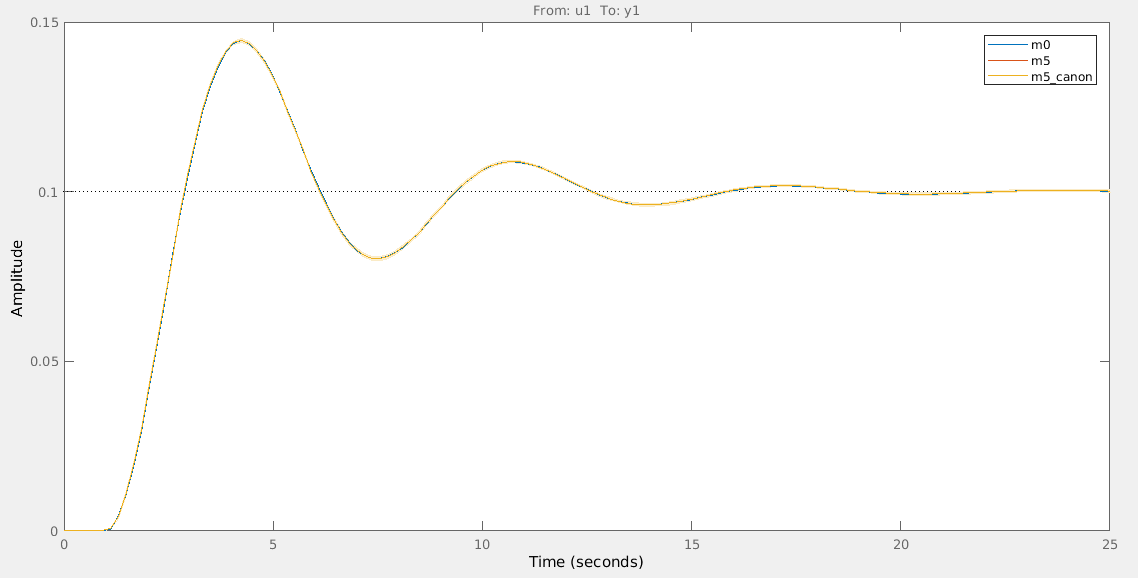
****

Слика 9. Бодеов дијаграм на апроксимираните континуални модели и реалниот систем

Одзив на моделите при Хевисајдова влезна возбуда:

showConfidence(stepplot(m0, m5, m5\_canon),3)

legend show

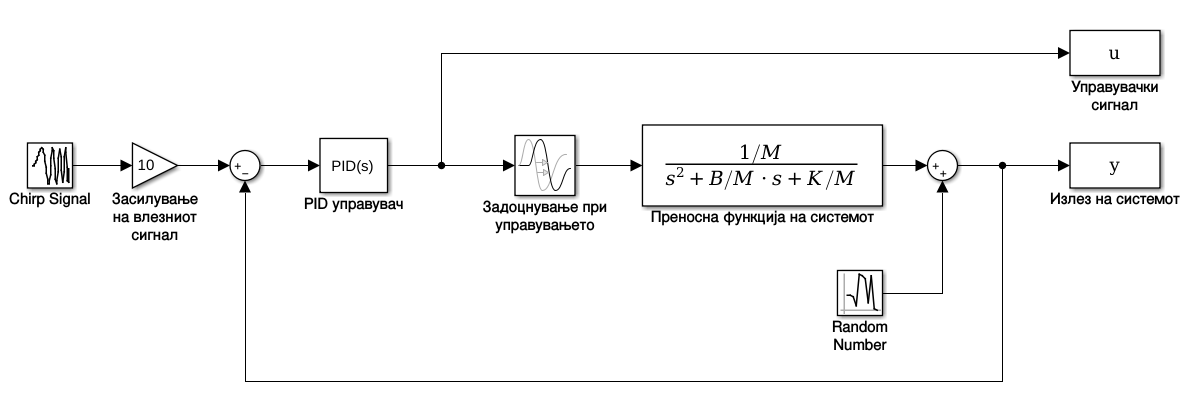


Слика 10. Отскочен одзив на апроксимираните континуални модели и реалниот систем

Од двата графици можеме да забележиме дека границите на неизвесност на двата модели се виртуелно идентични, особено кај отскочната влезна возбуда.

## 3.2 АНАЛИЗА НА ОДНЕСУВАЊЕТО НА АПРОКСИМИРАНИОТ МОДЕЛ ИЗМЕЃУ ПЕРИОДИТЕ НА ДИСКРЕТИЗАЦИЈА

Кога се споредуваат континуални временски модели естимирани од временски дискретизирани податоци, битно е да се земе во предвид однесувањето на влезниот сигнал измеѓу периодите да дискретизација. Во конкретниот пример, влезот во системот измеѓу периодите на дискретизација имаше константна форма, поради форматорот од нулти ред во управувачот. Доколку се отстрани форматорот, добиваме вистински континуален систем. Времето на земање податоци е се уште 0.5 секунди и сите останати делови остануваат исти. Новиот SIMULINK модел е следниот:

****

Слика 11. SIMULINK претстава на системот без форматор од нулти ред

На ист начин се симулира и овој SIMULINK модел, и притоа се создава соодветен апроксимиран модел. Доколку создадеме дискретен модел, тој се уште ќе се однесува релативно точно во однос на влезно-излезните податоци, бидејќи тој е добро подесен во однос на мерењата и автоматски ќе ги инкорпорира својствата на дискретизацијата како и однесувањето на влезниот сигнал измеѓу периодите на дискретизација. Но, проблем настанува при создавањето на апроксимиран континуален модел, бидејќи потребно е да бидат познати особините на сигналите измеѓу периодите на дискретизација.

Најпрво се создава моделот следејќи ја претходната постапка:

sim(‘SEMINARSKA\_MODEL\_KONT’)

dat2\_est = iddata(y, u, 0.5);

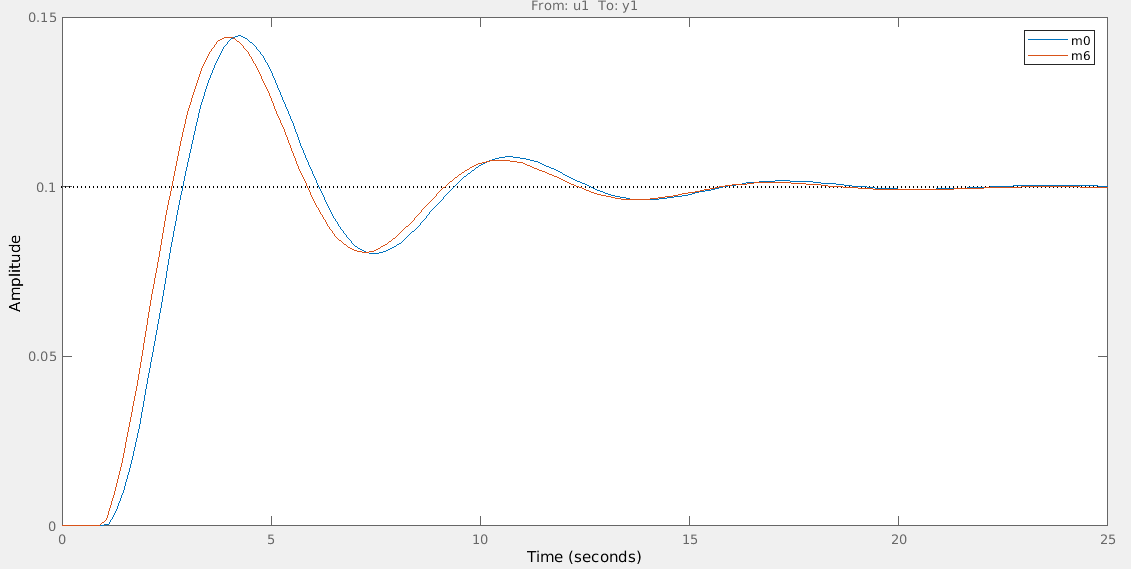
m6 = ssest(dat2\_est, 2, ‘Feedthrough’, false, ‘InputDelay’, 1,

‘Form’, ‘canonical’ );

idtf(m6)

Преносната функција на добиениот модел е следната:

Од каде веднаш може да се воочи дека е полоша од преносната функција од моделот што се доби со форматорот од нулти ред. Истото може да го заклучиме и со набљудување на графикот на одзивот на соодветните модели при отскочна влезна возбуда:



Слика 12. Отскочен одзив на апроксимираниот модел и реалниот систем

Ова отстапување на апроксимираниот модел може да се надомести со тоа што во самите податоци кои се добиваат со симулација на SIMULINK моделот се внесе информација за нивното однесување измеѓу периодите на дискретизација. Како претпоставка, внесуваме линеарна врска помеѓу периодите на дискретизација, што се претставува со форматор од прв ред. На крај, одново се создава апроксимиран модел *m7,* кој ќе се однесува многу поточно од претходниот модел.

dat2\_est.Intersample = ‘foh’;

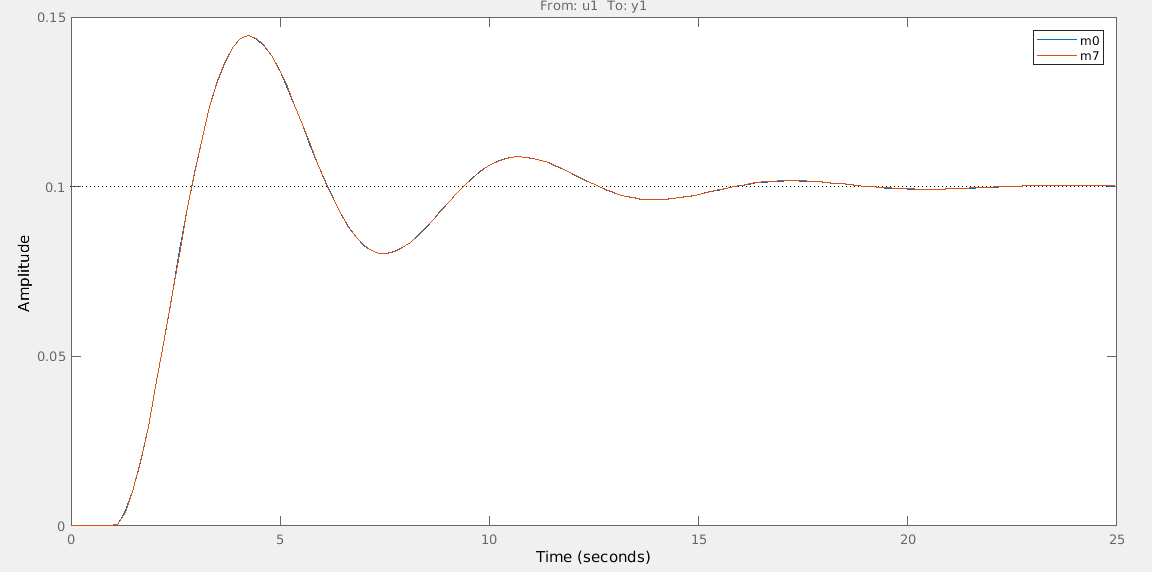
m7 = ssest(dat2\_est, 2, ‘Feedthrough’, false, ‘InputDelay’, 1,

‘Form’, ‘canonical’ );

idtf(m7)

Преносната функција на добиениот модел е следната:

Може да се забележи дека добиениот модел е најточен од сите други модели добиени до сега, а доколку се спореди неговиот отскочен одзив со реалниот систем тие практично целосно ќе се поклопуваат:



Слика 13. Отскочен одзив на апроксимираниот модел и реалниот систем

# ЗАКЛУЧОК

Со разни постапки на идентификација можат да се добијат различни модели со релативно слична точност во однос на реалниот. Иако во овој труд се стремеше да се усоврши добиениот апроксимиран модел, виртуелно сите од добиените модели се доволно прецизни за практична примена во реалноста.

# КОРИСТЕНА ЛИТЕРАТУРА

[1] https://www.mathworks.com/help/ident/examples/estimating-continuous-time-models-using-simulink-data.html#d120e4881

[2] http://ukim.edu.mk/e-izdanija/FEIT/Resheni\_zadachi\_od\_oblasta\_modeliranje\_simulacija\_i\_identifikacija\_na\_dinamichki\_sistemi.pdf

[3] Елизабета Лазаревска, *"Моделирање, симулација и идентификација на динамички системи"*, 2011.