



**БЪЛГАРСКА АКАДЕМИЯ НА НАУКИТЕ**

**Институт по информационни  
и комуникационни технологии**

маг. инж. Маргарита Николаева Терзийска

**НЕВРОННО-РАЗМИТИ МОДЕЛИ ЗА ЦЕЛИТЕ  
НА ПРЕДСКАЗВАЩОТО УПРАВЛЕНИЕ**

**А В Т О Р Е Ф Е Р А Т**

НА ДИСЕРТАЦИЯ

за присъждане на образователна и научна степен „доктор”

по научна специалност: 01.01.12 „Информатика”

професионално направление 4.6 „Информатика и компютърни науки”

Научни ръководители: доц. д-р Любка Дуковска  
проф. д-р Михаил Петров

**София, 2015 г.**

Дисертационният труд е разработен в общ обем от 189 страници, в които 58 фигури, 7 таблици и 17 страници библиография, включваща 187 литературни източника. Номерацията на фигурите и формулите не съвпада с тази в дисертацията.

Дисертацията е обсъдена и допусната до защита на разширено заседание на секция „Интелигентни системи“ на ИИКТ-БАН, състояло се на 05.10.2015 г. Защитата на дисертацията ще се състои на .....2015 г. от ... часа в зала 507 на блок 25 на ИИКТ-БАН на открито заседание на научно жури в състав:

1. акад. д.т.н. Иван Попчев
2. акад. д.т.н. Минчо Хаджийски
3. проф. д-р Андон Топалов - ТУ София, филиал Пловдив
4. доц. д-р Вера Ангелова - ИИКТ-БАН
5. доц. д-р Любка Дуковска - ИИКТ-БАН

Материалите за защитата са на разположение на интересуващите се в стая 215 на ИИКТ-БАН, ул. „Акад. Г. Бончев“, бл. 25А.

## **НЕВРОННО-РАЗМИТИ МОДЕЛИ ЗА ЦЕЛИТЕ НА ПРЕДСКАЗВАЩОТО УПРАВЛЕНИЕ**

Автор: *Маргарита Николаева Терзийска*

|                                                                                                                                                   |           |
|---------------------------------------------------------------------------------------------------------------------------------------------------|-----------|
| <b>Обща характеристика на дисертационния труд</b>                                                                                                 | <b>3</b>  |
| Увод                                                                                                                                              | 3         |
| Мотивация, цел и задачи                                                                                                                           | 3         |
| <b>Съкратено съдържание на дисертационния труд</b>                                                                                                | <b>4</b>  |
| Глава I. Съвременни алгоритми за нелинейно моделно предсказващо управление                                                                        | 4         |
| Глава II. Нелинейно невронно-размито обобщено предсказващо управление - теоретична постановка                                                     | 7         |
| Глава III. Проектиране на невронно-размити модели и алгоритми за регулатори с невронно-размито предсказващо управление                            | 10        |
| Глава IV. Експериментално изследване на предложените невронно-размити модели и алгоритми за регулатори с невронно-размито предсказващо управление | 22        |
| Глава V. Софтуерна реализация на предложените невронно-размити модели и алгоритми за регулатори с невронно-размито предсказващо управление        | 35        |
| Глава VI. Заключение и изводи                                                                                                                     | 36        |
| <b>Приноси в дисертационния труд</b>                                                                                                              | <b>38</b> |
| <b>Списък с публикации по дисертационния труд</b>                                                                                                 | <b>39</b> |
| <b>Участие в научно-изследователски проекти</b>                                                                                                   | <b>40</b> |
| <b>Библиография</b>                                                                                                                               | <b>40</b> |

## **Обща характеристика на дисертационния труд**

### **Увод**

Невронните мрежи, размитата логика и хибридните невронно-размити структури са в основата на системите с изкуствен интелект. Поотделно и в комбинация тези техники са известни като универсални апроксиматори, поради което са широко използвани за моделиране и идентификация на нелинейни обекти. Това ги прави подходящи за включване в предсказващите регулатори в ролята на предсказващи модели. Последните са съществена и много важна част от предсказващите регулатори, тъй като от точността им на предсказване зависи качеството на управление. Друга основна част от предсказващите регулатори е оптимизаторът, който определя стойността на управлението, решавайки оптимизационна задача. Предсказването на бъдещия изход на технологичния обект и определянето на оптималната стойност на управляващото въздействие са две изчислително тежки процедури, с които предсказващият регулатор трябва да се справи в рамките на един такт на дискретизация. Това е основната причина, поради която този вид регулатори се използват главно за бавни процеси. За да се преодолее този недостатък на предсказващите регулатори е необходимо да бъдат разработени начини за облекчаване на изчислителната процедура при моделирането и/или при оптимизацията. С оглед на това настоящият дисертационен труд е насочен към проектиране на алгоритми за нелинейни предсказващи регулатори, базирани на невронно-размити модели, които са с редуциран брой размити правила и съответно имат малък брой параметри за настройка.

### **Мотивация, цел и задачи**

Непрекъснато увеличаващият се брой на изследвания и разработки в областта на нелинейното моделно предсказващо управление (НМПУ), базирано на хибридни невронно-размити модели, говорят за засилен интерес и стремеж към усъвършенстване на тези регулатори, но така и за все още нерешени проблеми в тази област. Създаването на невронно-размити модели с намалена изчислителна тежест, бързи и надеждни методи за оптимизация, проучване на възможността за използване на интелигентни структури за настройката на предсказващите регулатори са само част от направленията за развитие на НМПУ. С оглед на това е формулирана следната **цел** на дисертационния труд, а именно: Разработване на невронно-размити предсказващи модели с механизъм на извеждане Такаги-Сугено с намалена изчислителна тежест (с оптимизиран брой размити правила), подходящи за целите на нелинейното предсказващо управление. За постигане на поставената цел следва да се решат и следните задачи:

- Да се разработи и софтуерно реализира разпределен невронно-размит модел, работещ с редуциран брой размити правила.
- Да се проектира, изследва и разработи програмен код на

изчислително ефективен невронно-размит модел с частично размиване на входните сигнали.

- Да се проектира, изследва и разработи програмно приложение за модифициран нео-размит модел.
- Да се разработи софтуерно и изследва модифициран нео-размит модел във вариант с Тип 2 размита логика и с Интуиционистка размита логика.
- Да се реализират програмно и изследват оптимизационни алгоритми от II ред (алгоритъм на Нютон и алгоритъм на Левенберг-Маргуард), които осигуряват бързодействие на предсказващия регулатор.
- Оптимизационните алгоритми от II ред да се реализират софтуерно във вариант с LU декомпозиция – изчислително ефективен метод, който позволява да се избегне намирането на обратната матрица на Хесе на всеки такт.
- Да се изследва възможността за използване на интелигентни структури в алгоритми за супервайзорна настройка на предсказващите алгоритми.

### **Съкратено съдържание на дисертационния труд**

Дисертационният труд е оформен в шест глави в съответствие с последователността на работа по него. В **Глава I** е дадена постановката на моделното предсказващо управление (МПУ) и е направен обзор на използваните в НМПУ невронни, размити и невронно-размити модели, въз основа на който са формирани целта и задачите на дисертационния труд. В **Глава II** е представена теоретичната постановка на невронно-размит обобщен предсказващ регулатор. Проектираните алгоритми за нелинейно невронно-размито обобщено предсказващо управление са детайлно описани в **Глава III**. Тяхната работоспособност и ефективност е удостоверена посредством резултатите от симулационни и реални експерименти, представени в **Глава IV**. Структурата и характерните особености на разработения програмен код са дискутирани в **Глава V**. В **Глава VI** са обобщени постигнатите резултати и са резюмирани направените изводи.

### **Глава I. Съвременни алгоритми за нелинейно моделно предсказващо управление**

Една от основните задачи, стоящи пред НМПУ, е изборът на модел, описващ възможно най-точно управлявания технологичен процес. Най-съществените изисквания при моделирането за целите на НМПУ, както и обзор на методите за моделиране и идентификация може да бъде намерен в (Lee, 2000). Трудът на (Pearson, 2003) представлява своеобразно ръководство за избора на подходяща структура на нелинеен модел. В него авторът описва някои основни класове нелинейни структури, класифицира ги и дава отговор на

въпроса какво представлява “добрият” модел. Понастоящем за целите на НМПУ се използват предимно: модели на Волтера; модели на Хамерщайн и Винер; невронни мрежи; модели, базирани върху размитата логика; хибридни модели (комбинация от изброените по-горе).

### **1.1 Невронни предсказващи модели**

Причината невронните мрежи да намерят широко практическо приложение в системите за управление се крие в техните особености, а именно: възможност за обучение, висока надеждност (при повреда на един или няколко неврона невронната мрежа продължава да работи ефективно) и висока скорост на обработка на информацията. В системите с НМПУ могат да се използват различни по структура невронни мрежи, но най-често използваните са тези с прави връзки. Наред с тях, в нелинейните предсказващи регулатори, се включват също радиално базисни невронни мрежи (Radial Basis Neural Networks), мрежи с авто-асоциативна памет (Cerebellar Model Arithmetic Computer - CMAC) и Wavelet Neural Networks. През последното десетилетие бурно развитие претърпяха рекурентните невронни мрежи. В тези мрежи, освен прави връзки се включват и обратни връзки (локални и/или глобални). Тези обратни връзки играят ролята на ”памет” и позволяват на модела по-добре да ”улови” динамиката на процеса. Ето защо се е наложило мнението, че рекурентните мрежи са по-добри апроксиматори от мрежите с прави връзки. Обучението на невронните мрежи е процес, свързан с адаптивната настройка на теглата на връзките и представлява нелинейна оптимизационна задача. Най-общо използваните методи за обучение могат да се класифицират като градиентни алгоритми от I-ви и II-ри ред. Към първата група се отнася алгоритъмът с обратно разпространение на грешката (backpropagation), както и неговите модификации (с въвеждане на моментум, с адаптивна скорост на обучение и др.), а втората група включва метода на Нютон, квази-Нютонови алгоритми, метод на Левенберг-Маркгуард. Като обучаващи алгоритми се използват също така и биологично инспирираните алгоритми за глобална оптимизация. Наред с вече утвърдилите се генетични алгоритми в последно време се използват също алгоритми за оптимизация по метода на мравките (Ant Colony Algorithm), по метода на пчелите (Artificial Bee Colony algorithm) и на рояците (Particle Swarm Optimization). Характерно за тази група алгоритми е, че при тях не се изисква изчисляване на производни, но пък са значително по-бавни от градиентните методи, което ги прави неподходящи за целите на предсказващото управление.

### **1.2 Размити предсказващи модели**

В предсказващото управление се използват също и размити модели във вариантите с механизъм на идвеждане на Мамдани и с механизъм на идвеждане на Такаги-Сугено. Тези модели се предпочитат, тъй като се проектират лесно, имат сравнително опростена структура и в тях лесно може

да се вгради човешка логика. Съществена трудност при изграждането им е определянето на размитите правила и функциите на принадлежност. (Liutkevičius et al., 2005). сравняват модел на Мамдани и на Такаги-Сугено. Според тях моделът на Мамдани е по-точен, но съдържа по-голям брой параметри, които трябва да бъдат определени при идентификацията. За целите на предсказващото управление по-широко се използват моделите с механизъм на извеждане на Такаги-Сугено. Характерно за тях е, че в следствената част на правилата изходната величина се дефинира с конкретна функция (линейна или нелинейна), чийто аргументи са входните променливи. През последното десетилетие много популярни станаха моделите, използващи размити множества (РМ) Тип 2, по-конкретно интервалните множества Тип 2. Съществен техен недостатък е, че имат по-голям брой параметри за настройка в сравнение с класическата (Тип 1) размита логика. Наред с теорията на РМ Тип 2 се развиват и тези на т.нар. множества с неравна форма (Rough Fuzzy Sets) и интуicionистки размити множества (Intuitionistic Fuzzy Sets). Характерно за последните е, че при тях освен степен на принадлежност  $\mu(x)$  се определя и степен на непринадлежност  $\nu(x)$ .

### 1.3 Невронно-размити предсказващи модели

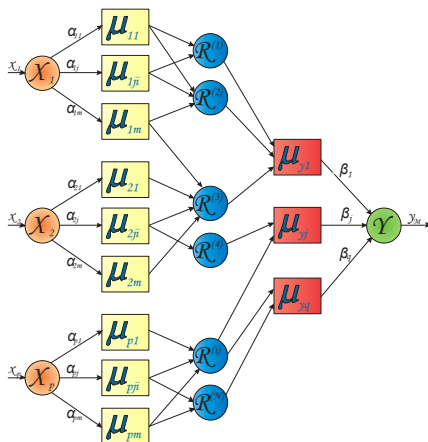
Като естествено продължение в развитието на невронните мрежи и системите с размита логика се появяват хибридните интелигентни архитектури. Те комбинират яснотата и прозрачността на размитите системи с възможността за обучение на невронните мрежи. Тяхната комбинация е сравнена от (Alavala, 2008) с човешкия мозък – невронната мрежа е структурата на мозъка, т.е. “хардуера“, докато размитата логика се грижи за “софтуера“. Един съществен недостатък на размитите модели, който те пренасят и в невронно-размитите структури е големият брой правила, с който работят. По принцип, броя на правилата зависи експоненциално от броя на входовете и от броя на използваните функции на принадлежност (ФП). Ако  $p$  е броя на входовете, а  $m$  е броя на ФП, то необходимия брой размити правила е  $m^p$ . Големият брой правила изисква определянето на още по-голям брой параметри по време на процедурата по обучение, което всъщност значително отежнява изчислителната процедура. Например, ако системата е с 10 входа, всеки от които е размит с по 2 ФП, това означава да се работи с 1024 ( $=2^{10}$ ) правила, което прави практическата реализация почти невъзможна. За да се намали броят на размитите правила без да се прави компромис с точността на моделиране в литературата са предложени различни методи.

Други съществени предизвикателства пред НМПУ са свързани с разработването на оптимизационни алгоритми и алгоритми за настройка на предсказващите регулатори. Основните изисквания, поставени пред оптимизационният алгоритъм, са бързата му сходимост и опростена изчислителна процедура, която да позволява реализацията му в реално време. Методи с намалена изчислителна тежест, гарантиращи в същото време

устойчивост и работоспособност на системата се предложени от (Chen and Allgöwer, 1998), (De Nicolao et al., 1996), (Tenny et al., 2002). Обширни трудове, посветени на изследването и разработването на алгоритми за решаването на нелинейната оптимизационна задача в реално време са предложени от (Diehl, 2001) и (Tenny, 2002). Що се отнася до настройката на предсказващите регулатори, нейната цел е осигуряване на добро отследяване на заданието, робастност при несъответствие на модела и намаляване влиянието на смущенията. Един съществен извод, който трябва да се направи е, че в областта на настройката на предсказващите регулатори не е работено задълбочено и също така не е проучена възможността за използването на интелигентни системи (в това число и невронно-размити структури) за настройката на предсказващите регулатори. Ето защо, наред с разработването на невронно-размити модели с намалена изчислителна тежест, като задачи пред настоящият дисертационен труд стоят проектирането на оптимизационни алгоритми и на алгоритми за настройка на предсказващите регулатори.

## Глава II. Нелинейно невронно-размито обобщено предсказващо управление - теоретична постановка

Отправна точка в процеса на търсене на решения на дискутираните вече проблеми се явява конкретен алгоритъм на нелинеен невронно-размит обобщен предсказващ регулатор. При него в ролята на предсказващ модел е използван невронно-размит авторегресионен модел (NNFARX), представен на фиг. 2.1.



Фиг. 2.1 Хибридна невронно-размита структура

Първият входен слой на NNFARX модела е непараметричен и служи за разпределение на входните сигнали. Вторият слой на предсказващия модел е

параметричен и реализира операцията размиване като за целта се използват Гаусови функции на принадлежност, описани със следната зависимост:

$$\mu_{x_{p,m}}^{(n)} = \exp \frac{-(x_p - c_{x_{p,m}})^2}{2\sigma_{x_{p,m}}^2} \quad (2.1)$$

В третият слой на NNFARX модела представлява своеобразен генератор на правила, тъй като в него се формират размити логически правила със следната форма:

$$R^{(N)}: \text{if } x_1(k+j) \text{ is } \tilde{X}_1^{(N)} \text{ and } x_2(k+j) \text{ is } \tilde{X}_2^{(N)} \dots \\ \text{if } x_p(k+j) \text{ is } \tilde{X}_p^{(N)} \text{ then } f_y^{(N)}(k+j) \quad (2.2)$$

Последните два слоя на NNFARX модела са непараметрични и формират размития механизъм за извеждане. В четвъртият слой се реализира операцията импликация като за целта се използва средно претеглено произведение:

$$\mu_{yq}^{(n)}(k+j) = \mu_{x_1,m}^{(n)}(k+j) * \mu_{x_2,m}^{(n)}(k+j) * \dots * \mu_{x_p,m}^{(n)}(k+j) \quad (2.3)$$

В петият последен слой се взема решението за извеждане, което се състои в определяне на стойността на изхода на модела посредством израза:

$$\hat{y}_M(k+j) = \frac{\sum_{i=1}^q f_y^{(i)}(k+j) \mu_y^{(i)}(k+j)}{\sum_{i=1}^q \mu_y^{(i)}(k+j)} \quad (2.4)$$

За определяне на управляващото въздействие в оптимизатора се решава итеративен оптимизационен алгоритъм от I ред. Използван е стандартния GPC квадратичен целеви критерий, като е прието, че не са наложени ограничения. Целевият критерий е преобразуван в матричен вид:

$$J[k, U(k)] = [R(k) - \hat{Y}(k)]^T [R(k) - \hat{Y}(k)] + \lambda \tilde{U}(k)^T \tilde{U}(k) \quad (2.5)$$

Минимизацията на GPC-критерия се основава на изчислението на вектор градиента на целевата функция в k-тия такт по отношение на предсказаните стойности на управляващото въздействие:

$$\nabla J[k, U(k)] = \left[ \frac{\partial J[k, U(k)]}{\partial u(k)}, \frac{\partial J[k, U(k)]}{\partial u(k+1)}, \dots, \frac{\partial J[k, U(k)]}{\partial u(k+N_u-1)} \right] \quad (2.6)$$

Спрямо матричното уравнение (2.2.1), елементите на вектор-градиента могат да се представят както следва:

$$\frac{\partial J[k, U(k)]}{\partial U(k)} = \left[ -2[R(k) - \hat{Y}(k)]^T \frac{\partial \hat{Y}(k)}{\partial U(k)} + 2\lambda \hat{U}(k)^T \frac{\partial \hat{U}(k)}{\partial U(k)} \right] \quad (2.7)$$

От този израз става ясно, че е необходимо е да се изчислят две групи частни производни. Първата група производни могат да бъдат записани в следната матрица:



$$\frac{\partial \hat{Y}(k)}{\partial U(k)} = \begin{bmatrix} \frac{\partial \hat{y}(k+N_1)}{\partial u(k)} & \dots & \frac{\partial \hat{y}(k+N_1)}{\partial u(k+N_u-1)} \\ \dots & \dots & \dots \\ \frac{\partial \hat{y}(k+N_2)}{\partial u(k)} & \dots & \frac{\partial \hat{y}(k+N_2)}{\partial u(k+N_u-1)} \end{bmatrix}_{[(Np-N_1+1) \times N_u]} \quad (2.8)$$

Втората група производни са дадени посредством матрицата:

$$\frac{\partial \hat{U}(k)}{\partial U(k)} = \begin{bmatrix} \frac{\partial \Delta u(k)}{\partial u(k)} & \dots & \frac{\partial \Delta u(k)}{\partial u(k+N_u-1)} \\ \dots & \dots & \dots \\ \frac{\partial \Delta u(k+N_u-1)}{\partial u(k)} & \dots & \frac{\partial \Delta u(k+N_u-1)}{\partial u(k+N_u-1)} \end{bmatrix}_{[N_u \times N_u]} \quad (2.9)$$

След като бъдат изчислени отделните градиентни елементи, те се приравняват на нула. В резултат на това се получава нова система от уравнения, която може да бъде решена по отношение на управляващите въздействия  $u(k)$ ,  $u(k+1)$ , ...,  $u(k+N_u+1)$ :

$$\frac{\partial J[k, U(k)]}{\partial u(k)} = -2\hat{e}(k+N_1) \frac{\partial \hat{y}(k+N_1)}{\partial u(k)} - \dots - 2\hat{e}(k+N_2) \frac{\partial \hat{y}(k+N_2)}{\partial u(k)} \quad (2.10)$$

$$+ 2\lambda \Delta u(k) - 2\lambda \Delta u(k+1) = 0$$

$$\frac{\partial J[k, U(k)]}{\partial u(k+1)} = -2\hat{e}(k+N_1) \frac{\partial \hat{y}(k+N_1)}{\partial u(k+1)} - \dots - 2\hat{e}(k+N_2) \frac{\partial \hat{y}(k+N_2)}{\partial u(k+1)} \quad (2.11)$$

$$+ 2\lambda \Delta u(k+1) - 2\lambda \Delta u(k+2) = 0$$

$$\frac{\partial J[k, U(k)]}{\partial u(k+N_u-2)} = -2\hat{e}(k+N_1) \frac{\partial \hat{y}(k+N_1)}{\partial u(k+N_u-2)} - \dots - \quad (2.12)$$

$$- 2\hat{e}(k+N_2) \frac{\partial \hat{y}(k+N_2)}{\partial u(k+N_u-2)} + 2\lambda \Delta u(k+N_u-2) - 2\lambda \Delta u(k+N_u-1) = 0$$

$$\frac{\partial J[k, U(k)]}{\partial u(k+N_u-1)} = -2\hat{e}(k+N_1) \frac{\partial \hat{y}(k+N_1)}{\partial u(k+N_u-1)} - \dots - \quad (2.13)$$

$$- 2\hat{e}(k+N_2) \frac{\partial \hat{y}(k+N_2)}{\partial u(k+N_u-1)} + 2\lambda \Delta u(k+N_u-1) = 0$$

Тази система от уравнения може да бъде решена много просто като се започне в обратен ред от последното уравнение (2.2.16), от което се изчислява управляващото въздействие  $\Delta u(k+N_u-1)$ . Получената стойност се замества в предходното уравнение и от него се определя  $\Delta u(k+N_u-2)$ . По подобен начин се изчислява цялата последователност от управляващи въздействия в рамките на хоризонта на управление.

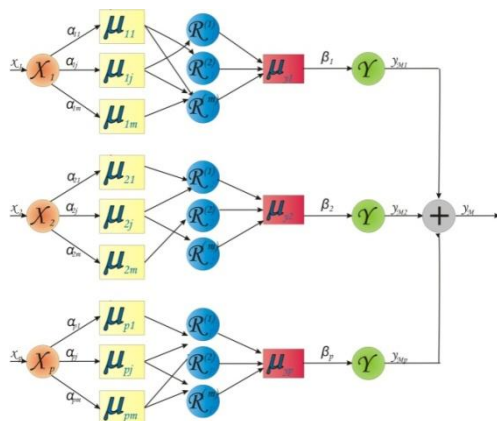
За така описаният нелинеен невронно-размит обобщен предсказващ регулаторе е прието да има четири входа и всеки от тях да се размива с по три Гаусови множества. Оттук произтича и един от основните му недостатъци, а именно големият брой размити правила, с които работи предсказващия модел. Вече беше споменато, че техният брой се определя съгласно израза  $N=m^p$  и следователно в конкретния случай броят на правилата е 81. Това от своя страна означава, че на всеки такт по време на обучението на този модел трябва да бъдат определени стойностите на 1053 параметъра! Това го прави практически неприложим за работа в реално време, особено за процеси с бързо изменяща се динамика. Поради това е необходимо да се търсят нови невронно-размити модели, работещи с редуциран брой размити правила и същевременно осигуряващи висока точност на предсказване. Освен по отношение на модела, работата на този регулатор би могла да се подобри и с включването на градиентни алгоритми от II ред в оптимизатора. Върху качеството на управление на нелинейния невронно-размит обобщен предсказващ регулатор, описан в Глава II, би могло да се повлияе и чрез подходящата му настройка.

### **Глава III. Проектиране на алгоритми за регулатори с невронно-размито предсказващо управление**

#### **3.1. Разпределен невронно-размит модел**

Разпределеният невронно-размит модел (Distributed Adaptive Neuro Fuzzy Architecture – DANFA) е представлява шестслойна адаптивна архитектура с механизъм на извеждане на Такаги-Сугено, чиято структура е представена на фиг. 3.1. Основната идея при него е входните сигнали да бъдат "разпределени" към отделни невронно-размити структури от типа, представен на фиг. 2.1. По този начин се получава мрежа от невронно-размити структури, като всяка от тях играе ролята на отделен подмодел, а обобщеният (глобален) DANFA модел е съвкупност от  $q$  на брой подмодела. Основното предимство на DANFA модела е, че при него броят на размитите правила е значително редуциран в сравнение с модела, описан в т.2.1. При последният броя на размитите правила се определя от израза  $N=m^p$ , докато при DANFA модела това става посредством формулата:

$$N = m_1^{p_1} + m_2^{p_2} + \dots + m_q^{p_q} \quad (3.1)$$



фиг. 3.1 Структура на DANFA модел

Отделните слоеве на DANFA модела изпълняват както следва:

**Слой 1:** Невроните от този слой приемат входните сигнали и ги предават към следващия слой.

**Слой 2:** Тук се осъществява операцията размиване, като за целта е необходимо да се определи вида и броя на използваните функции на принадлежност. В конкретния случай са използвани три Гаусови функции на принадлежност, описани с формула (2.1).

**Слой 3:** Този слой представлява своеобразен генератор на правила, тъй като именно тук се формират размитите правила от типа (2.2).

**Слой 4:** Реализира се операцията импликация посредством израза (2.3).

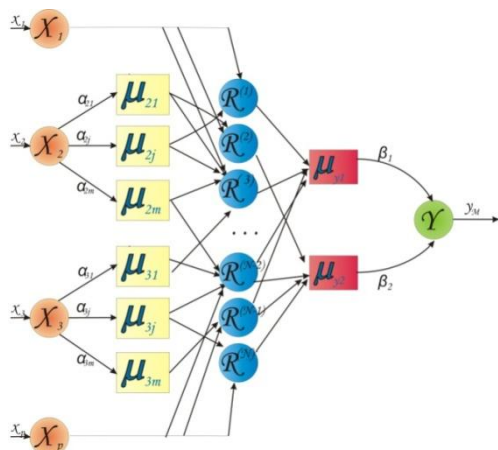
**Слой 5:** Формират се изходите на отделните подмодели  $\hat{y}_{M1}$ ,  $\hat{y}_{M2}$ , ...,  $\hat{y}_{Mq}$ , съгласно изразите (2.4). С  $q$  е означен броят на използваните подмодели.

**Слой 6:** Получава се изхода на DANFA модела като сума от изходите на отделните подмодели  $\hat{y}_{M1}$ ,  $\hat{y}_{M2}$ , ...,  $\hat{y}_{Mq}$ .

### 3.2. Невронно-размит модел с частично размиване на входните

Невронно-размитият модел с частично размиване на входните сигнали (Semi Fuzzy Neural Network - SFNN) е петслойна структура с механизъм на извеждане на Такаги-Сугено. При него [5] част от входните сигнали не се размиват, а влизат с техните реални стойности, претеглени със съответен тегловен коефициент, директно в третия слой (слоя с размитите правила), т.е. директно в следствената част на функциите на Сугено. По този начин от една страна се редуцира броят на размитите правила, с които модела работи, а от друга - се намалява и броят на нелинейните параметри, които трябва да се определят в процеса на обучението му. Така например, ако SFNN модела има 4 входа (2 размити и 2 неразмити, които са съответно изхода на обекта и управлението), то тогава общият брой правила, които трябва да се генерират е 9. Общият брой параметри (линейни и нелинейни), които трябва да бъдат

определени при обучението на модела е 64.

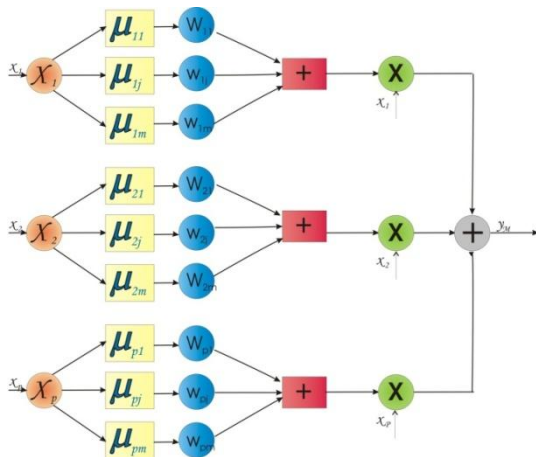


Фиг. 3.2 Структура на Невронно-размит модел с частично размиване на входните сигнали

Основна трудност, която се явява при използването на SFNN модела, е начинът, по който да се определи кои от входните сигнали да бъдат размити и кои да се включат директно в следствената част на размитите правила. В търсене на подходящо решение на този проблем са разработени и изследвани ефективността и точността на моделиране на три разновидности на SFNN модела – SFNN Type I, при който се размиват миналите стойности на изхода на обекта, а стойностите на управлението влизат директно в следствената част на размитите правила, SFNN Type II, който е обратния вариант на SFNN Type I и SFNN Type III, наречен още модел с кръстосани връзки, при който са размити по една стойност от управлението и от изхода на обекта. Трябва да се подчертае, че тези три разновидности са разработени при допускането, че се реализира NNFXR модел с 4 входа, от които само два са размити.

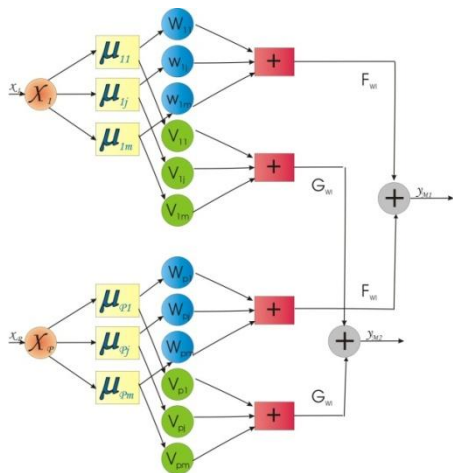
### 3.3. Модифициран нео-размит модел

Нео-размитият неврон е предложен от проф. Т. Yamakawa и екипът му (Yamakawa et al. , 1992). По същество той представлява радиално базисна невронна мрежа с механизъм на извеждане Такаги-Сугено от нулев ред. За целите на настоящата дисертация е реализиран модифициран нео-размит модел, чиято структура е представена на фиг. 3.3.



Фиг. 3.3. Структура на модифициран нео-размит модел

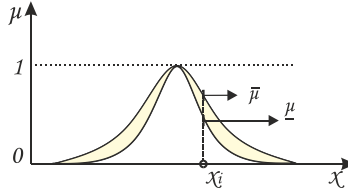
Освен това модифицираният нео-размит модел е реализиран в многомерен вариант [4] като за целта е използвана структурата, представена на фиг. 3.4. Особено в случая е, че е необходимо да се минимизират две моментни грешки и съответно да се актуализират двете групи тегловни коефициенти  $w_{ji}(k)$  и  $v_{ji}(k)$ , но не се прави обучение на параметрите на размитите множества.



Фиг. 3.4 Структура на многомерен модифициран нео-размит модел

### 3.4 Реализация на предложените невронно-размити модели с Тип 2 размита логика

При реализацията на предложените невронно-размити модели с Тип 2 размита логика са използвани функции на принадлежност, които имат вида, представен на фиг. 3.5, от която се вижда, че центърът на гаусовата функция е един, но ширината е различна, при което се получават съответно и две степени на принадлежност в съответствие с израза (3.1). Оцветената област на фиг. 3.5 се нарича област на неопределеност (Footprint of uncertainty - FOU).



Фиг. 3.5. Тип 2 Гаусово размито множество

$$\mu_{ij}(x_i) = \exp\left(\frac{-(x_i - c_{ij})^2}{2\sigma_{ij}^2}\right) = \begin{cases} \bar{\mu}_{ij} \text{ за } \sigma_{ij} = \bar{\sigma}_{ij} \\ \underline{\mu}_{ij} \text{ за } \sigma_{ij} = \underline{\sigma}_{ij} \end{cases} \quad (3.1)$$

т.е. има степен на принадлежност  $\bar{\mu}$  към горното множество и степен на принадлежност  $\underline{\mu}$  към долното множество. При това положение изразът (2.3) за размита импликация се трансформира до вида:

$$\mu_{ij}^*(x_i) = \begin{cases} \bar{\mu}_{ij}^* = \prod_{i=1}^n \bar{\mu}_{ij} \\ \underline{\mu}_{ij}^* = \prod_{i=1}^n \underline{\mu}_{ij} \end{cases} \quad (3.2)$$

Стойността на изхода на модела при Тип 2 невронно-размита структура се определя съгласно следните формули:

$$\hat{y}_M(k+j) = \frac{1}{2} \left( \frac{\sum_{i=1}^q \bar{f}_y^{(i)}(k+j) \bar{\mu}_y^{(i)}(k+j)}{\sum_{i=1}^q \bar{\mu}_y^{(i)}(k+j)} + \frac{\sum_{i=1}^q f_y^{(i)}(k+j) \underline{\mu}_y^{(i)}(k+j)}{\sum_{i=1}^q \underline{\mu}_y^{(i)}(k+j)} \right) \quad (3.3)$$

$$\hat{y}_M(k+j) = \frac{1}{2} \left( \sum_{i=1}^q \bar{f}_y^{(i)}(k+j) \bar{\mu}_y^{*(i)}(k+j) + \sum_{i=1}^q f_y^{(i)}(k+j) \underline{\mu}_y^{*(i)}(k+j) \right) \quad (3.4)$$

От горните изрази става ясно, че при използване на Тип 2 невронно-размита структура е необходимо на всеки такт да се определят по-голям брой параметри, което се явява техен недостатък и „цена”, която трябва да се плати, за да се получи по-точно моделиране в условията на неопределености.

### 3.5 Реализация на предложените невронно-размити модели с Интуиционистка размита логика

Интуиционистката размита логика е друг начин за справяне с неопределеностите в системата. Характерна особеност на Интуиционистката размита логика е, че при нея освен степен на принадлежност  $\mu(x)$  се определя и степен на непринадлежност  $\nu(x)$

$$\mu_{ij}(x_i) = \exp\left(\frac{-(x_i - c_{ij})^2}{2\sigma_{ij}^2}\right) \quad (3.5)$$

$$\nu_{ij}(x_i) = \left(1 - \exp\left(\frac{-(x_i - c_{ij})^2}{2\sigma_{ij}^2}\right)\right)^k, k \geq 1 \quad (3.6)$$

където  $x_i$  е входния сигнал,  $i=1:p$  е броя на входовете на интуиционистката невронно-размита структура,  $c_{ij}$  и  $\sigma_{ij}$  са центъра и ширината на Гаусовите функции на принадлежност,  $j=1:m$  като  $m$  е броя на използваните функции на принадлежност,  $k$  е параметър, който трябва да бъде избран (Zhоu, 2013). Определя се също така и област на неопределеност като се използва следната зависимост:

$$\pi(x) = 1 - \mu(x) - \nu(x) \quad (3.7)$$

При интуиционистки невронно-размити модели изходът се определя в съответствие с:

$$\hat{y}_M(k+1) = (1 - \pi(x))y_\mu + \pi(x)y_\nu \quad (3.8)$$

където  $y_\mu$  и  $y_\nu$  се изчисляват както следва:

$$y_\mu(k+j) = \frac{\sum_{i=1}^q f_\mu^{(i)}(k+j)\mu_y^{(i)}(k+j)}{\sum_{i=1}^q \mu_y^{(i)}(k+j)} \quad (3.9)$$

$$y_\nu(k+j) = \frac{\sum_{i=1}^q f_\nu^{(i)}(k+j)\nu_y^{(i)}(k+j)}{\sum_{i=1}^q \nu_y^{(i)}(k+j)} \quad (3.10)$$

И тук, както и при Тип 2 невронно-размитите структури, броят на параметрите, които следва да се определят по време на обучението е завишен. Като съществено предимство на интуиционистките невронно-размити модели пред тези с Тип 2 размита логика може да се изтъкне, че не е необходимо двукратно реализиране на операцията размиване. След като веднъж вече е определена степента на принадлежност, то от израза (3.6) лесно може да бъде определена степента на непринадлежност.

### 3.6 Алгоритми за обучение на предложените невронно-размити модели

Безспорно когато става дума за обучение на невронни мрежи и/или невронно-размити структури най-популярния метод е този с обратно разпространение на грешката. По същество той е градиентен метод от първи ред и като такъв има някои недостатъци – бавна сходимост и възможност от засядане в локален екстремум. С цел преодоляване на тези недостатъци в настоящата дисертация за обучението на предложените DANFA, SFNN и модифициран нео-размит модели освен алгоритъма с обратно разпространение на грешката са разработени също така рекурентен метод на най-малките квадрати, метод на Нютон и метод на Левенберг-Маркгуард.

#### 3.6.1 Двустъпков градиентен алгоритъм

Двустъпковият градиентен алгоритъм представлява модифициран алгоритъм с обратно разпространение на грешката. Основната разлика между двата алгоритъма е в броя на настройваемите параметри. При класическият алгоритъм за обучение грешката се разпространява в обратна посока последователно през всички слоеве на невронната (невронно-размита) мрежа и се актуализират теглата на връзките във всички слоеве. При двустъпковия градиентен алгоритъм се актуализират теглата на връзките само в два от слоевете. Това са т.нар. параметрични слоеве. Обучението с двустъпковия градиентен метод се основава на минимизирането на моментната моделна грешка (грешка между изхода на обекта и изхода на невронно-размития модел):

$$\varepsilon(k) = y(k) - \hat{y}_M(k) \quad (3.11)$$

където  $y$  е реалната стойност на изходната величина, а  $\hat{y}$  е стойността на изхода на обекта изчислена от невронно-размития модел (DANFA, SFNN или модифицирания нео-размит модел). Формираният целеви критерий е:

$$E = \frac{\varepsilon^2(k)}{2} \quad (3.12)$$

Най-напред се актуализират теглата на връзките в третия слой. Това са коефициентите в следствената част на размитите правила. За да се опрости записването въвеждаме векторът  $\beta$ , в който са включени търсените параметри:

$$\begin{aligned} \beta &= \{a_1^{(n)}, \dots, a_{ny}^{(n)}, b_0^{(n)}, b_1^{(n)}, \dots, b_{nu}^{(n)}\} = \\ &= \{\beta_1^{(n)}, \beta_2^{(n)}, \dots, \beta_{ny}^{(n)}, \beta_{ny+1}^{(n)}, \dots, \beta_{ny+nu+1}^{(n)}\} \end{aligned} \quad (3.13)$$

По този начин обобщеното обучаващо правило за коефициентите във функцията на Сугено може да се представи със следния израз:

$$\beta(k+1) = \beta(k) + \eta(k) \left( -\frac{\partial E}{\partial \beta} \right) \quad (3.14)$$

където с  $\eta(k) = \eta = \text{const}$  е означена скоростта на обучение. Производната от обучаващото правило може да се представи по следния начин:



$$\frac{\partial E}{\partial \mathbf{\beta}} = \frac{\partial E}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial \mathbf{\beta}} \quad (3.15)$$

където:

$$\frac{\partial E}{\partial \hat{y}} = -(y(k) - \hat{y}(k)) = -\varepsilon(k) \quad (3.16)$$

Стойността на втората частна производна е различна за свободния член и останалите коефициенти от функцията на Сугено и се определя съгласно изразите:

- за свободния член

$$\frac{\partial \hat{y}}{\partial \beta_{0j}} = \bar{\mu}_y^{(n)}(k) \quad (3.17)$$

- за останалите коефициенти

$$\frac{\partial \hat{y}}{\partial \beta_{ij}} = \bar{\mu}_y^{(n)}(k) x_p(k) \quad (3.18)$$

Замествайки (3.17) и (3.18) в (3.14) се получават окончателните изрази за параметрите в следствието:

$$\beta_{0j}(k+1) = \beta_{0j}(k) + \eta(k) \varepsilon(k) \bar{\mu}_y^{(n)}(k) \quad (3.19)$$

$$\beta(k+1) = \beta(k) + \eta(k) \varepsilon(k) \bar{\mu}_y^{(n)}(k) x_p(k) \quad (3.20)$$

Втората група настройваеми параметри са тези на размитите Гаусови множества  $c_{Xp,m}$  и  $\sigma_{Xp,m}$ . Отново с цел опростяване на записа се въвежда вектора  $\alpha$ , в който са включени нелинейните параметри:

$$\begin{aligned} \alpha &= \{c_{11}^{(n)}, \dots, c_{pm}^{(n)}, \sigma_{11}^{(n)}, \dots, \sigma_{pm}^{(n)}\} = \\ &= \{\alpha_1^{(n)}, \alpha_2^{(n)}, \dots, \alpha_{pm}^{(n)}, \alpha_{pm+1}^{(n)}, \dots, \alpha_{2pm}^{(n)}\} \end{aligned} \quad (3.21)$$

За определянето им се използва аналогичен на (3.14) израз:

$$\alpha(k+1) = \alpha(k) + \eta(k) \left( -\frac{\partial E}{\partial \alpha} \right) \quad (3.22)$$

където производната може да се представи като:

$$\frac{\partial E}{\partial \alpha} = \frac{\partial E}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial \mu_{xp,m}} \cdot \frac{\partial \mu_{xp,m}}{\partial \alpha} \quad (3.23)$$

Първата частна производна от (3.23) се определя с израза (3.16), а останалите две са както следва:

$$\frac{\partial \hat{y}}{\partial \mu_{xp,m}} = \frac{f_y(k) - \hat{y}(k)}{\sum_{i=1}^q \mu_{yq}^{(n)}(k)} \quad (3.24)$$

- за центъра  $c_{Xp,m}$  на размитите множества

$$\frac{\partial \mu_{xp,m}}{\partial \mathbf{a}} = 2\mu_{xp,m} \frac{(x_p - c_{pm})}{\sigma_{pm}^2} \quad (3.25)$$

- за ширината  $\sigma_{xp,m}$  на размитите множества

$$\frac{\partial \mu_{xp,m}}{\partial \mathbf{a}} = \mu_{xp,m} \frac{(x_p - c_{pm})^2}{\sigma_{pm}^3} \quad (3.26)$$

Замествайки изразите (3.16), (3.24), (3.25) и (3.26) в (3.22) се получава окончателния рекурентен израз за всеки един входен параметър за настройка (център  $c_{xp,m}$  или ширина  $\sigma_{xp,m}$ ):

$$c_{xp,m}(k+1) = c_{xp,m}(k) + \dots + \eta \varepsilon(k) \bar{\mu}_y^{(n)}(k) [f_y^{(n)}(k) - \hat{y}(k)] \frac{[x_p(k) - c_{xp,m}(k)]}{\sigma_{xp,m}^2(k)} \quad (3.27)$$

$$\sigma_{xp,m}(k+1) = \sigma_{xp,m}(k) + \dots + \eta \varepsilon(k) \bar{\mu}_y^{(n)}(k) [f_y^{(n)}(k) - \hat{y}(k)] \frac{[x_p(k) - c_{xp,m}(k)]^2}{\sigma_{xp,m}^3(k)} \quad (3.28)$$

### 3.6.2 Рекурентен метод на най-малките квадрати

Методът на най-малките квадрати (МНК) е един от най-широко използваните методи за оценка на параметрите на динамични модели. В настоящата дисертация той е използван, за да се реализира хибриден алгоритъм за обучение, при който посредством рекурентен МНК (РМНК) е направена оценка на параметрите във функцията на Сугено, а чрез алгоритъма с обратно разпространение на грешката се актуализират параметрите на Гаусовите функции на принадлежност (център и ширина). За реализацията му първо се определят параметрите във функцията на Сугено, което налага преобразуване й в следната форма:

$$y(k) = a_1^{(i)} y(k-1) + a_2^{(i)} y(k-2) + \dots + a_{n_y}^{(i)} y(k-n_y) + \dots + b_1^{(i)} u(k) + b_2^{(i)} u(k-1) + \dots + b_{n_u}^{(i)} u(k-n_u) + b_0^{(i)} \quad (3.29)$$

Последният израз от своя страна може да бъде записан във векторна форма по следния начин:

$$y(k) = \psi^T(k) \theta \quad (3.30)$$

където:

$$\theta = [b_0, b_1, \dots, b_m, a_1, a_2, \dots, a_n]^T$$

$$\psi(k) = [u(k-1), \dots, u(k-n_u), y(k-1), \dots, y(k-n_y)]^T$$

При подобна постановка рекурсивната оценка на параметрите по МНК става с използването на следните изрази:

$$\hat{\theta}(k) = \hat{\theta}(k-1) + K(k)[y(k) - \psi^T(k)\hat{\theta}(k-1)] \quad (3.31)$$

$$K(k) = P(k-1)\psi(k)[I + \psi^T(k)P(k-1)\psi(k)]^{-1} \quad (3.32)$$

$$\begin{aligned} P(k) &= P(k-1) - K(k)\psi^T(k)P(k-1) = \\ &= P(k-1) - P(k-1)\psi(k)[I + \psi^T(k)P(k-1)\psi(k)]^{-1}\psi^T(k)P(k-1) \end{aligned} \quad (3.33)$$

За определянето на параметрите от предпоставката на размитите правила се използват изразите (3.21 – 3.28). РМНК име сравнително проста и лесна реализация. Оценката се получава посредством прибавянето на претеглената предсказана грешка към предходната стойност на оценката  $\phi$ . Векторът  $K(k)$  се нарича Калманово усиление, а  $P(k)$  представлява симетрична ковариационна матрица. Изборът на началната ѝ стойност е свързан с неопределеността в стойностите на оценките на параметрите.

### 3.6.3 Алгоритми за обучение, базирани на градиентни методи от втори ред

За целите на дисертационния труд са реализирани хибридни алгоритми за обучение, които представляват комбинация от градиентен метод от втори ред (Нютон или Левенберг-Маркгуард метод) и алгоритъм с обратно разпространение на грешката, т.е. параметрите във функцията на Сугено се определят посредством градиентен метод от втори ред, а нелинейните параметри на финкциите на принадлежност – с градиентен метод от първи ред. Това е направено с цел да се избегнат възможни осцилации при използването на алгоритъм за обучение, основан изцяло на градиентен метод от втори ред. Актуализирането на стойностите в следствената част на размитите правила по метода на Нютон става чрез изразите:

$$\beta_{0j}(k+1) = \beta_{0j}(k) + \Delta\beta_{0j}(k) \quad (3.34)$$

$$\beta_{ij}(k+1) = \beta_{ij}(k) + \Delta\beta_{ij}(k) \quad (3.35)$$

където

$$\Delta\beta_{0j}(k) = -[J^T(\beta_{0j})J(\beta_{0j})]^{-1}J^T(\beta_{0j})\varepsilon(k) \quad (3.36)$$

$$\Delta\beta_{ij}(k) = -[J^T(\beta_{ij})J(\beta_{ij})]^{-1}J^T(\beta_{ij})\varepsilon(k) \quad (3.37)$$

и тъй като якобиянът за свободния член и за останалите коефициенти има вида:

$$J^T(\beta_{0j}) = \bar{\mu}_y^{(j)} \quad J(\beta_{0j}) = [\bar{\mu}_y^{(j)}]^T \quad J^T(\beta_{ij}) = \bar{\mu}_y^{(j)}x_i \quad J(\beta_{ij}) = [\bar{\mu}_y^{(j)}x_i]^T$$

то окончателно изразите (3.6.23) и (3.6.24) приемат следната форма:

$$\beta_{0j}(k+1) = \beta_{0j}(k) + \eta[\bar{\mu}_y^{(j)}(\bar{\mu}_y^{(j)})^T]^{-1}\varepsilon(k)\bar{\mu}_y^{(j)} \quad (3.38)$$

$$\beta_{ij}(k+1) = \beta_{ij}(k) + \eta[\bar{\mu}_y^{(j)}x_i(k)(\bar{\mu}_y^{(j)}x_i(k))^T]^{-1}\varepsilon(k)\bar{\mu}_y^{(j)}x_i(k) \quad (3.39)$$

По аналогичен начин се получават рекурентните зависимости за параметрите във функцията на Сугено при използване на алгоритъма на Левенберг-Маркгуард. В този случай изразите (3.36) и (3.37) се преобразуват до вида:

$$\Delta\beta_{0j}(k) = -[J^T(\beta_{0j})J(\beta_{0j}) + \lambda I]^{-1} J^T(\beta_{0j})\varepsilon(k) \quad (3.40)$$

$$\Delta\beta_{ij}(k) = -[J^T(\beta_{ij})J(\beta_{ij}) + \lambda I]^{-1} J^T(\beta_{ij})\varepsilon(k) \quad (3.41)$$

където  $I$  е единична матрица, а параметърът  $\lambda$  регулира направлението на движение към екстремума. И тъй като стойността на якобияна е същата както при алгоритъма на Нютон, то окончателните изрази за определяне на параметрите в следствената част на размитите правила са както следва:

$$\beta_{0j}(k+1) = \beta_{0j}(k) + \eta [\bar{\mu}_y^{(j)} (\bar{\mu}_y^{(j)})^T + \lambda I]^{-1} \varepsilon(k) \bar{\mu}_y^{(j)} \quad (3.42)$$

$$\beta_{ij}(k+1) = \beta_{ij}(k) + \eta [\bar{\mu}_y^{(j)} x_i(k) (\bar{\mu}_y^{(j)} x_i(k))^T + \lambda I]^{-1} \varepsilon(k) \bar{\mu}_y^{(j)} x_i(k) \quad (3.43)$$

За определянето на параметрите от предпоставката на размитите правила отново се използват изразите (3.21 – 3.28).

### 3.7. Оптимизационни алгоритми за определяне стойността на управлението

#### 3.7.1. Метод на Нютон

При използване на квадратична целева функция и при липса на наложени ограничения оптимизацията може да бъде осъществена отново аналитично. За целта е при метода на Нютон е необходимо да се определят вторите частни производни на целевия критерий, а това става по следния начин:

$$H[k, U(k)] = \left[ \frac{\partial^2 J[k, U(k)]}{\partial u^2(k)}, \frac{\partial^2 J[k, U(k)]}{\partial u^2(k+1)}, \dots, \frac{\partial^2 J[k, U(k)]}{\partial u^2(k+N_u-1)} \right] \quad (3.44)$$

Всеки елемент от този вектор се определя както следва:

$$H[k, U(k)] = \left[ -2 \left[ R(k) - \hat{Y}(k) \right]^T \frac{\partial \hat{Y}(k)}{\partial U(k)} + 2\rho \hat{U}(k)^T \frac{\partial \hat{U}(k)}{\partial U(k)} \right]' = \left[ \begin{aligned} & -2 \left( \left[ R(k) - \hat{Y}(k) \right]^T \right)' \frac{\partial \hat{Y}(k)}{\partial U(k)} - 2 \left[ R(k) - \hat{Y}(k) \right]^T \left( \frac{\partial \hat{Y}(k)}{\partial U(k)} \right)' + \\ & + 2\rho \left( \hat{U}(k)^T \right)' \frac{\partial \hat{U}(k)}{\partial U(k)} + 2\rho \hat{U}(k)^T \left( \frac{\partial \hat{U}(k)}{\partial U(k)} \right)' \end{aligned} \right] \quad (3.45)$$

Тъй като стойностите на матриците  $\left( \frac{\partial \hat{Y}(k)}{\partial U(k)} \right)'$  и  $\left( \frac{\partial \hat{U}(k)}{\partial U(k)} \right)'$  са нулеви, то за горния израз може да се опрости до следния вид [2]:

$$H[k, U(k)] = -2 \left( \frac{\partial \hat{Y}(k)}{\partial U(k)} \right)^2 + 2\rho \left( \frac{\partial \hat{U}(k)}{\partial U(k)} \right)^2 \quad (3.46)$$

След определяне на матрицата на Хесе е необходимо тя да бъде инвертирана, което внася допълнителна изчислителна тежест в метода на Нютон и се явява съществен негов недостатък. Разполагайки с обратната матрица на Хесе изразът (3.7.3) се преобразува до следните рекурентни зависимости:

$$\frac{\partial^2 J[k, U(k)]}{\partial U^2(k)} = \left[ -2 \left( \frac{\partial \hat{Y}(k)}{\partial U(k)} \right)^2 + 2\rho \left( \frac{\partial \hat{U}(k)}{\partial U(k)} \right)^2 \right]^{-1} * \left[ -2 [R(k) - \hat{Y}(k)]^T \frac{\partial \hat{Y}(k)}{\partial U(k)} + 2\rho \hat{U}(k)^T \frac{\partial \hat{U}(k)}{\partial U(k)} \right] \quad (3.47)$$

$$\Delta u(k + N_u - 2) = \Delta u(k + N_u - 1) + \rho^{-1} H^{-1} \left[ e(k + N_1) \frac{\partial \hat{y}(k + N_1)}{\partial u(k + N_u - 2)} + \dots + e(k + N_2) \frac{\partial \hat{y}(k + N_2)}{\partial u(k + N_u - 2)} \right] \quad (3.48)$$

$$\Delta u(k) = \Delta u(k + 1) + \rho^{-1} H^{-1} \left[ e(k + N_1) \frac{\partial \hat{y}(k + N_1)}{\partial u(k)} + \dots + e(k + N_2) \frac{\partial \hat{y}(k + N_2)}{\partial u(k)} \right] \quad (3.49)$$

### 3.7.2. Метод на Левенберг-Маркгуард

При метода на Левенберг-Маркгуард рекурентни зависимости за определяне на управляващото въздействие имат вида:

$$\Delta u(k + N_u - 1) = \rho^{-1} [H^{-1} + \lambda I] \left[ e(k + N_1) \frac{\partial \hat{y}(k + N_1)}{\partial u(k + N_u - 1)} + \dots + e(k + N_2) \frac{\partial \hat{y}(k + N_2)}{\partial u(k + N_u - 1)} \right] \quad (3.50)$$

$$\Delta u(k + N_u - 2) = \Delta u(k + N_u - 1) + \rho^{-1} [H^{-1}(x^{(k)}) + \lambda I] * \left[ e(k + N_1) \frac{\partial \hat{y}(k + N_1)}{\partial u(k + N_u - 2)} + \dots + e(k + N_2) \frac{\partial \hat{y}(k + N_2)}{\partial u(k + N_u - 2)} \right] \quad (3.51)$$

$$\Delta u(k) = \Delta u(k + 1) + \rho^{-1} [H^{-1}(x^{(k)}) + \lambda I] * \left[ e(k + N_1) \frac{\partial \hat{y}(k + N_1)}{\partial u(k)} + \dots + e(k + N_2) \frac{\partial \hat{y}(k + N_2)}{\partial u(k)} \right] \quad (3.52)$$

където с  $I$  е означена единичната матрица, а параметърът  $\lambda$  регулира направлението на движение към екстремума, от градиентно – при голямо  $\lambda$  ( $\lambda > 10^3$ ) до направление определено по .... от метода на Нютон при  $\lambda = 0$ . В началото на търсенето параметърът  $\lambda$  се избира голям (например  $\lambda = 10^4$ ), при

което се подтилка влиянието на матрицата на Хесе и процесът на търсене се определя от градиентното направление. Ако още при първата стъпка се получи подобряване на целевия критерий, стойността на  $\lambda$  се намалява и се прави нова стъпка. Изчисляването на вектор градиента и матрицата на Хесе става по същия начин както при метода на Нютон.

### 3.7.3. LU декомпозиция

Характерно за описаните по-горе оптимизационни алгоритми за определяне стойността на управлението (метод на Нютон и метод на Левенберг-Маркгуард) е, че на всеки такт е необходимо намирането на обратната матрица на Хесе. Това е процедура, която изисква значителни изчисления и следователно води до намаляване на бързодействието на предсказващия регулатор. За да се избегне този недостатък е използван метода на LU декомпозицията, вид Гаусова елиминация, която в програмната среда Matlab се осъществява с операцията дясно деление [2].

### 3.8. Алгоритъм за супервайзорна адаптивна донастройка на предсказващ регулатор

Разработен е алгоритъм за супервайзорна адаптивна донастройка на тегловния коефициент в целевия критерий [3], като се запазват постоянни стойностите на хоризонтите на управление и на предсказване. Супервайзорът е реализиран в два варианта – с използване на размита логика с механизъм на извеждане на Мамдани и чрез невронно-размита структура с механизъм на извеждане на Сугено.

## Глава IV. Експериментални изследвания с проектираните алгоритми

### 4.1. Изследване способността на предложените модели за предсказване на хаотични времеви серии

Проектираните DANFA, SFNN и нео-размит модел първоначално са тествани да предсказват хаотични времеви серии. Направено е сравнение на тези модели с невронно-размития модел, описан в т.2.1. Получените резултати са обобщени и е определен най-добрият модел в съответствие със заложените критерии. Тестването на предложените невронно-размити модели е направено с помощта на някои превърнали се в класически хаотични времеви серии, а именно Макгий-Глас (Mackey-Glass), Рослер (Rossler) и Лоренц (Lorenz), описани съответно с изразите (4.1), (4.2) и (4.3):

$$x(i+1) = \frac{x(i) + ax(i-s)}{(1+x^c(i-s)) - bx(i)} \quad (4.1)$$

където  $a=0.2$ ;  $b=0.1$ ;  $C=10$ ; а началните условия са  $x(0)=0.1$  и  $s=17s$ .

$$\begin{aligned}\frac{\partial x}{\partial t} &= -y - z \\ \frac{\partial y}{\partial t} &= x + ay \\ \frac{\partial z}{\partial t} &= b + z(x - c)\end{aligned}\tag{4.2}$$

където  $a=0.2$ ;  $b=0.4$ ;  $c=5.7$ ; при следните начални условия  $x_0=0.1$ ;  $y_0=0.1$ ;  $z_0=0.1$

$$\begin{aligned}\frac{\partial x}{\partial t} &= s(y - x) \\ \frac{\partial y}{\partial t} &= rx - y - xz \\ \frac{\partial z}{\partial t} &= xy - bz\end{aligned}\tag{4.3}$$

където  $s=16$ ;  $b=4$ ;  $r=45.92$ ; при следните начални условия  $x_0=0.1$ ;  $y_0=0.1$ ;  $z_0=0.1$ .

За оценка на точността на предсказване е прието да се използва средноквадратична грешка (Mean Squared Error - MSE), дефинирана посредством израз:

$$MSE = \frac{1}{n} \sum_{i=1}^n (\hat{Y}_i - Y_i)^2\tag{4.4}$$

и коренуваната средноквадратична грешка, определена чрез:

$$RMSE = \frac{1}{n} \sqrt{\sum_{i=1}^n (\hat{Y}_i - Y_i)^2}\tag{4.5}$$

За оценка на предложените невронно-размити модели са използвани няколко критерии, а именно: брой размити правила, използвани от модела; брой параметри, които трябва да се определят в процеса на обучението му; времето, необходимо за реализиране на всички изчисления в рамките на един такт (стъпка) и точност на предсказване, определена в съответствие с изразите (4.4) и (4.5). Обобщение на тези критерии за изследваните модели е представено в табл. 4.1.

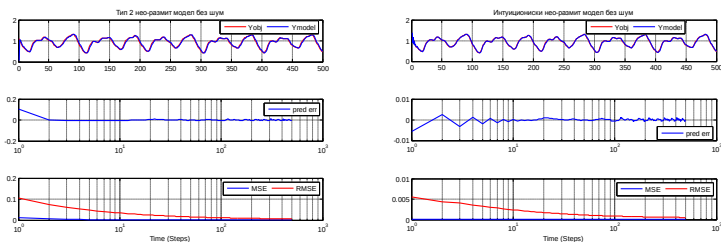
Таблица 4.1.

| Модел                           | Бр. правила | Бр. параметри | Точност |           |          | t            |
|---------------------------------|-------------|---------------|---------|-----------|----------|--------------|
|                                 |             |               | стъпки  | MSE       | RMSE     |              |
| Невронно-размит модел, описан в | 81          | 1053          | 50      | 0.0091595 | 0.095705 | $2,82e^{-4}$ |
|                                 |             |               | 100     | 0.0077459 | 0.088011 |              |
|                                 |             |               | 150     | 0.0079593 | 0.089513 |              |
|                                 |             |               | 200     | 0.077992  | 0.088313 |              |

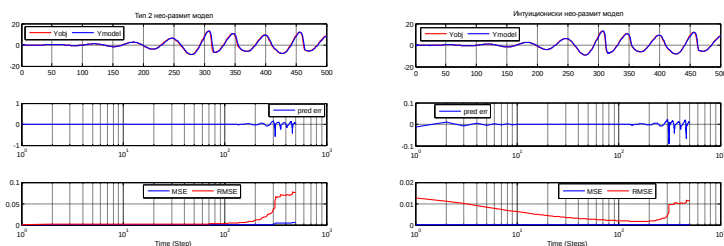
|                                                                                              |    |     |     |                     |          |                     |
|----------------------------------------------------------------------------------------------|----|-----|-----|---------------------|----------|---------------------|
| т.2.1                                                                                        |    |     | 250 | 0.0077583           | 0.088081 |                     |
|                                                                                              |    |     | 300 | 0.0079122           | 0.08895  |                     |
|                                                                                              |    |     | 350 | 0.0077017           | 0.087759 |                     |
|                                                                                              |    |     | 400 | 0.007838            | 0.088471 |                     |
|                                                                                              |    |     | 450 | 0.007659            | 0.087516 |                     |
|                                                                                              |    |     | 500 | 0.007769            | 0.088143 |                     |
| DANFA<br>модел                                                                               | 18 | 126 | 50  | 0.053553            | 0.23141  | 1.83e <sup>-4</sup> |
|                                                                                              |    |     | 100 | 0.027756            | 0.1659   |                     |
|                                                                                              |    |     | 150 | 0.01893             | 0.13759  |                     |
|                                                                                              |    |     | 200 | 0.014637            | 0.12098  |                     |
|                                                                                              |    |     | 250 | 0.012033            | 0.1097   |                     |
|                                                                                              |    |     | 300 | 0.010343            | 0.10156  |                     |
|                                                                                              |    |     | 350 | 0.0090874           | 0.095328 |                     |
|                                                                                              |    |     | 400 | 0.0081819           | 0.090454 |                     |
|                                                                                              |    |     | 450 | 0.0074437           | 0.086277 |                     |
|                                                                                              |    |     | 500 | 0.0068844           | 0.082973 |                     |
| SFNN<br>Type III<br>модел                                                                    | 9  | 63  | 50  | 1,6e <sup>-4</sup>  | 0,0129   | 1.58e <sup>-4</sup> |
|                                                                                              |    |     | 100 | 8,7e <sup>-5</sup>  | 0,0093   |                     |
|                                                                                              |    |     | 150 | 6,04e <sup>-5</sup> | 0,0078   |                     |
|                                                                                              |    |     | 200 | 4,8e <sup>-5</sup>  | 0,0069   |                     |
|                                                                                              |    |     | 250 | 3,97e <sup>-5</sup> | 0,0063   |                     |
|                                                                                              |    |     | 300 | 3,5e <sup>-5</sup>  | 0,0059   |                     |
|                                                                                              |    |     | 350 | 3,1e <sup>-5</sup>  | 0,0056   |                     |
|                                                                                              |    |     | 400 | 2,8e <sup>-5</sup>  | 0,0053   |                     |
|                                                                                              |    |     | 450 | 2,6e <sup>-5</sup>  | 0,0051   |                     |
|                                                                                              |    |     | 500 | 2,3e <sup>-5</sup>  | 0,0049   |                     |
| Нео-<br>размит<br>модел с<br>обучаване<br>параметри<br>те на<br>размитите<br>множеств<br>а   | 12 | 36  | 50  | 9,35e <sup>-5</sup> | 0,0097   | 1.67e <sup>-4</sup> |
|                                                                                              |    |     | 100 | 7,06e <sup>-5</sup> | 0,0084   |                     |
|                                                                                              |    |     | 150 | 5,6e <sup>-5</sup>  | 0,0075   |                     |
|                                                                                              |    |     | 200 | 4,5e <sup>-5</sup>  | 0,0067   |                     |
|                                                                                              |    |     | 250 | 3,9e <sup>-5</sup>  | 0,0062   |                     |
|                                                                                              |    |     | 300 | 3,4e <sup>-5</sup>  | 0,0058   |                     |
|                                                                                              |    |     | 350 | 3,12e <sup>-5</sup> | 0,0056   |                     |
|                                                                                              |    |     | 400 | 2,92e <sup>-5</sup> | 0,0054   |                     |
|                                                                                              |    |     | 450 | 2,33e <sup>-5</sup> | 0,0048   |                     |
|                                                                                              |    |     | 500 | 2,17e <sup>-5</sup> | 0,0047   |                     |
| Нео-<br>размит<br>модел без<br>обучаване<br>параметри<br>те на<br>размитите<br>множеств<br>а | 12 | 12  | 50  | 9,5e <sup>-5</sup>  | 0,0097   | 1.14e <sup>-4</sup> |
|                                                                                              |    |     | 100 | 7,3e <sup>-5</sup>  | 0,0085   |                     |
|                                                                                              |    |     | 150 | 5,9e <sup>-5</sup>  | 0,0077   |                     |
|                                                                                              |    |     | 200 | 4,8e <sup>-5</sup>  | 0,0069   |                     |
|                                                                                              |    |     | 250 | 4,1e <sup>-5</sup>  | 0,0064   |                     |
|                                                                                              |    |     | 300 | 3,5e <sup>-5</sup>  | 0,0059   |                     |
|                                                                                              |    |     | 350 | 3,2e <sup>-5</sup>  | 0,0057   |                     |
|                                                                                              |    |     | 400 | 2,9e <sup>-5</sup>  | 0,0054   |                     |
|                                                                                              |    |     | 450 | 2,5e <sup>-5</sup>  | 0,0050   |                     |
|                                                                                              |    |     | 500 | 2,25e <sup>-5</sup> | 0,0047   |                     |



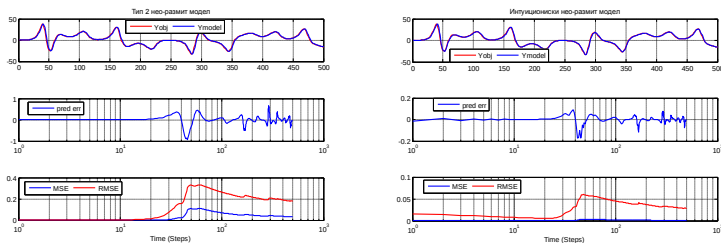
Въз основа на обобщените данни от табл. 4.1. може да се каже, че модифицирания нео-размит модел е най-добър по смисъла на зададените критерии и е реализиран във вариант с Тип 2 размита логика и с Интуиционистка размита логика в присъствието на шум, за да се види как се справя в условия на неопределености, а също и в МИМО вариант, тъй като едно от предимствата на предказващите регулатори е способността им да се справят с многомерни обекти.



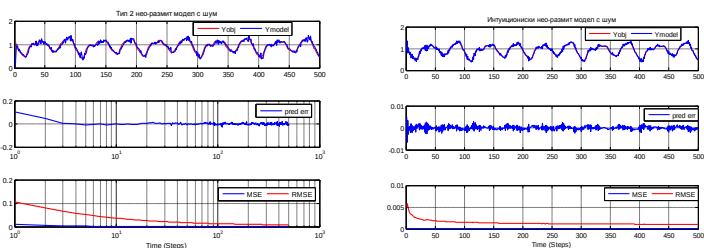
Фиг.4.1 Предказване на Макий-Глас хаотична времева серия с Тип 2 (вляво) и с Интуиционистки нео-размит модел



Фиг.4.2 Предказване на Рослер хаотична времева серия с Тип 2 (вляво) и с Интуиционистки нео-размит модел



Фиг.4.3 Предказване на Лоренц хаотична времева серия с Тип 2 (вляво) и с Интуиционистки нео-размит модел



Фиг.4.4 Предсказване на Макий-Глас хаотична времева серия с шум с Тип 2 (вляво) и с Интуиционистски нео-размит модел

За точността на моделиране се съди по стойностите на MSE и RMSE, които са обобщени в табл. 4.2. От получените резултати става ясно, че Интуиционистският нео-размит модел работи с по-малка грешка в сравнение с Тип 2 нео-размит модела. Времето за осъществяване на необходимите изчисления за една стъпка от обучението при Интуиционисткия нео-размит модел е съответно  $1.48e-4$  за случая без шум и  $1.50e-4$  със шум. При Тип 2 нео-размит модела това време е по-голямо – съответно  $1.93e-4$  и  $1.96e-4$  за случаите без и със шум. Следователно Интуиционистският нео-размит модел е по-точен и по-бърз и това го прави подходящ за включване в предсказващ регулатор.

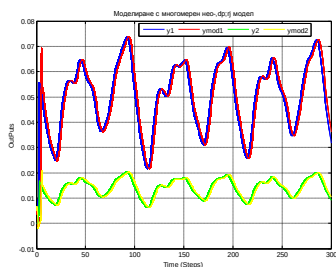
Таблица 4.2

| Стъпки | Тип 2 нео-размит модел |        |              |        | Интуиционистски нео-размит модел |              |              |        |
|--------|------------------------|--------|--------------|--------|----------------------------------|--------------|--------------|--------|
|        | Без шум                |        | С шум        |        | Без шум                          |              | С шум        |        |
|        | MSE                    | RMSE   | MSE          | RMSE   | MSE                              | RMSE         | MSE          | RMSE   |
| 50     | $2.38e^{-4}$           | 0.0156 | $2.98e^{-4}$ | 0.0173 | $1.26e^{-6}$                     | 0.0011       | $3.42e^{-6}$ | 0.0019 |
| 100    | $1.26e^{-4}$           | 0.0112 | $1.69e^{-4}$ | 0.0130 | $7.05e^{-7}$                     | $8.39e^{-4}$ | $2.25e^{-6}$ | 0.0015 |
| 150    | $8.85e^{-5}$           | 0.0094 | $1.31e^{-4}$ | 0.0115 | $5.19e^{-7}$                     | $7.2e^{-4}$  | $1.87e^{-6}$ | 0.0013 |
| 200    | $7.01e^{-5}$           | 0.0084 | $1.07e^{-4}$ | 0.0104 | $4.32e^{-7}$                     | $6.57e^{-4}$ | $1.61e^{-6}$ | 0.0012 |
| 250    | $5.89e^{-5}$           | 0.0077 | $8.4e^{-5}$  | 0.0091 | $3.75e^{-7}$                     | $6.12e^{-4}$ | $1.48e^{-6}$ | 0.0011 |
| 300    | $5.19e^{-5}$           | 0.0072 | $7.22e^{-5}$ | 0.0085 | $3.41e^{-7}$                     | $5.85e^{-4}$ | $1.37e^{-6}$ | 0.0011 |
| 350    | $4.63e^{-5}$           | 0.0068 | $6.72e^{-5}$ | 0.0082 | $3.13e^{-7}$                     | $5.6e^{-4}$  | $1.26e^{-6}$ | 0.0010 |
| 400    | $4.26e^{-5}$           | 0.0065 | $5.91e^{-5}$ | 0.0077 | $2.96e^{-7}$                     | $5.44e^{-4}$ | $1.19e^{-6}$ | 0.0010 |
| 450    | $3.92e^{-5}$           | 0.0063 | $5.33e^{-5}$ | 0.0073 | $2.78e^{-7}$                     | $5.28e^{-4}$ | $1.13e^{-6}$ | 0.0010 |
| 500    | $3.69e^{-5}$           | 0.0061 | $4.76e^{-5}$ | 0.0069 | $2.67e^{-7}$                     | $5.17e^{-4}$ | $1.09e^{-6}$ | 0.0010 |

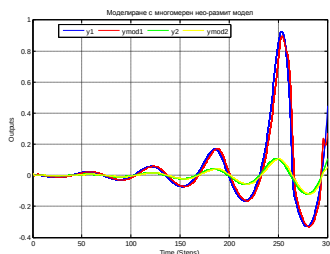
За тестването на многомерния нео-размит модел е използвана многомерна нелинейна система, представена от Pan (2012) и описана с изразите:

$$\begin{aligned}
y_1(k) &= \frac{y_1^2(k-1)}{y_1^2(k-1)+1} + 0.5y_2(k-1) \\
y_2(k) &= \frac{y_1^2(k-1)}{y_2^2(k-1)+y_3^2(k-1)+y_4^2(k-1)} + u_1(k-1) \\
y_3(k) &= \frac{y_3^2(k-1)}{y_3^2(k-1)+1} + 0.3y_4(k-1) \\
y_4(k) &= \frac{y_3^2(k-1)}{y_1^2(k-1)+y_2^2(k-1)+y_4^2(k-1)} + 0.5u_2(k-1)
\end{aligned}
\tag{4.6}$$

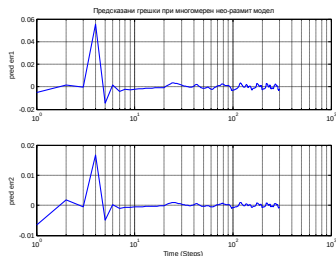
където  $u_1(k)$  и  $u_2(k)$  са входовете, а  $y_1(k)$  и  $y_3(k)$  са изходите. В ролята на входове са използвани хаотичните времеви серии на Макий-Глас и Рослер. Получените резултати са представени на фиг. 4.5 до фиг. 4.8.



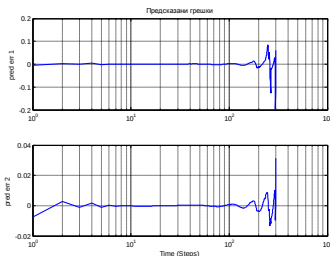
Фиг.4.5 Моделиране на Макий-Глас хаотична времева серия с многомерен нео-размит модел



Фиг.4.6 Моделиране на Рослер хаотична времева серия с многомерен нео-размит модел



Фиг.4.7 Предсказани грешки при многомерен нео-размит модел



Фиг.4.8 Предсказани грешки при многомерен нео-размит модел

## 4.2. Реализиране на обобщен предсказващ регулатор с използване на проектираните модели

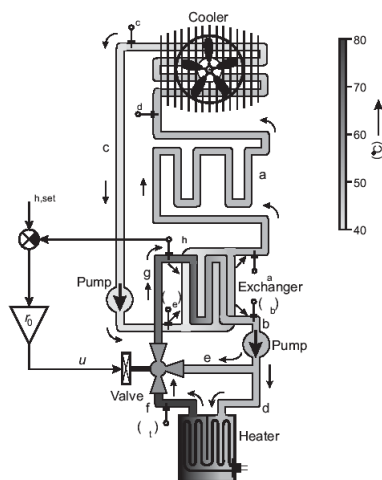
При втората част от изследванията тестваните вече невронно-размити модели с редуциран брой размити правила са вградени в алгоритми на обобщени

предсказващи регулатори, които са използвани за управлението на химически реактор и лабораторна топлинна система. Използваният химичен реактор представлява опростен модел на полимерен реактор (Continuous Stirred Tank Reactor - CSTR). Той може да бъде описан със следната система нелинейни диференциали уравнения:

$$\begin{aligned} \dot{x}_1 &= x_1 + D_a(1-x_1)e^{\left(\frac{x_2}{1+\frac{x_2}{\varphi}}\right)} \\ \dot{x}_2 &= -(1+\delta)x_2 + BD_a(1-x_1)e^{\left(\frac{x_2}{1+\frac{x_2}{\varphi}}\right)} + u\delta \end{aligned} \quad (4.7)$$

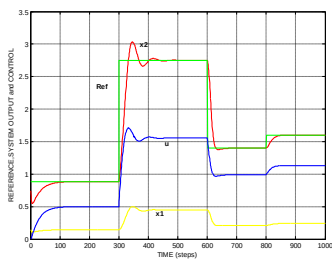
където  $x_1$  е концентрацията на продуктите в реактора,  $x_2$  е температурата в реактора, и е температурата на охлаждането на реактора, а  $B$ ,  $D_a$ ,  $\delta$ ,  $\varphi$  са физически параметри на реактора и имат следните стойности:  $D_a = 0,072$ ;  $\varphi = 20$ ;  $B = 8$ ;  $\delta = 0,69$ .

Схемата на използваната лабораторна топлинна система е представена на фиг. 4.9. Тя се намира в Машинния факултет на Техническия университет в Прага и се състои от два топлинни контура, през които циркулира вода. Движението на водата се осъществява посредством две помпи, съответно по една за двете вериги. Топлинният източник на системата е електрически нагревател, разположен по главния (първичен) контур. В топлообменника, който се намира в центъра на схемата се осъществява топлинния обмен между двата контура. Друг основен компонент е охладителя, който принадлежи към втория контур. Както се вижда от схемата, отделните части на системата са свързани по между си чрез тръби, което е причина за наличието на чисти закъснения. Измерваните температури в системата са следните:  $\mathcal{Q}_h$  - температура на водата на входа на топлообменника;  $\mathcal{Q}_a$  - температура на водата на изхода на топлообменника;  $\mathcal{Q}_d$ ,  $\mathcal{Q}_c$  - входната и изходна температура на водата към вентилатора. Управляващия вход на системата е напрежението  $u$ . То се подава към сервомеханизъм, регулиращ позицията на вентил. Положението на вентила определя съотношението на вода от нагревателя и вода от топлообменника, като по този начин се регулира температурата. Целта е управлението на температурата на водата на входа на топлообменника-  $\mathcal{Q}_h$ . Други входове на системата са скоростта на перката на вентилатора и топлинното изпълнение на нагревателя.

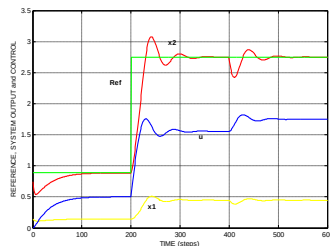


Фиг. 4.9 Структурна схема на топлинната система

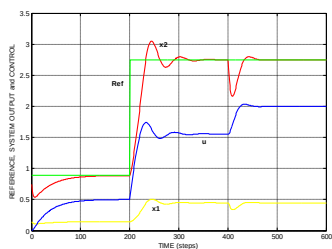
Всеки един от представените по-горе модели е включен в обобщен предсказващ регулатор, реализиран чрез итеративният оптимизационен алгоритъм от първи ред. GPC регулатор базиран на DANFA модела със следните параметри  $N1=1$ ;  $Np=7$ ;  $Nu=3$ ;  $\rho=0.1$  е използван за управление на химичен реактор. Получените резултати са представени на фиг. 4.10 до фиг. 4.13. Всички фигури показват поведението на обекта и стойността на управлението. На фиг. 4.10 е представен преходния процес при променливо задание. Симулационните резултати при промяна на параметрите на обекта са показани на фиг. 4.12 В този случай на стъпка 400 стойността на топлинния коефициент  $\delta$  и на  $B$  са едновременно променени на 0.8 и 7.5, съответно. Изследвана е възможността на регулатора да се справя със смущения (фиг. 4.11), като за целта на стъпка 400 е подадено стъпално смущение с амплитуда 0.5. На фиг. 4.13 е показано как се справя регулатора в най-тежкия случай, когато едновременно имаме промяна в параметрите на обекта и добавено смущение.



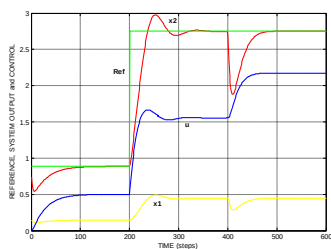
Фиг. 4.10 Преходен процес при променливо задание



Фиг. 4.11 Преходен процес при променливо задание и добавяне на смущение в стъпка 400



Фиг. 4.12 Преходен процес при променливо задание и промяна в параметрите на обекта -  $B=7.5$ ,  $\delta=0.8$  at the step 400



Фиг. 4.13 Преходен процес при променливо задание, промяна в параметрите на обекта и добавяне на смущение в стъпка 400

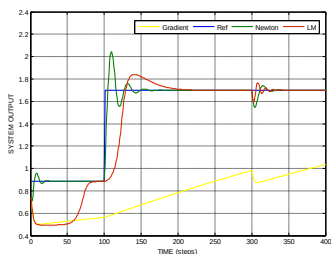
Подобни резултати се получават при управлението на химичния реактор с обобщен предсказващ регулатор, при който за модел са използвани и другите предложени модели - SFNN в трите му варианта и нео-размит, реализиран с Тип 2 и Интуиционистка размита логика. Поради това е счтено за излишно представянето им на отделни графики. Следователно може да се направи извода, че структурата на използвания невронно-размит модел не оказва влияние върху качеството на управление. Друг интересен извод може да се направи по отношение на времето  $t$ , необходимо за реализиране на всички изчисления в оптимизатора за един такт. Стойностите на  $t$  при обобщен предсказващ регулатор с използване на различни невронно-размити модели са представени в Таблица 4.3. Става ясно, че времето за работа на обобщен предсказващ регулатор с невронно-размит модел, описан в Глава II, е приблизително два пъти по-голямо в сравнение с това на регулатор, базиран на някой от предложените в настоящата дисертация модели. Трябва да се отбележи, че това време  $t$  нараства с увеличаване на хоризонтите на предсказване  $N_p$  и на управление  $N_u$ .

Таблица 4.3

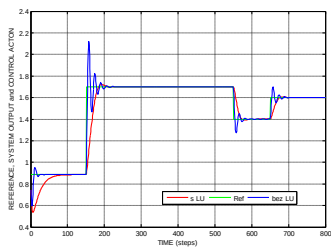
| Регулатор | $N_1=1; N_u=3; N_p=5$ | $N_1=1; N_u=3; N_p=10$ | $N_1=1; N_u=7; N_p=10$ |
|-----------|-----------------------|------------------------|------------------------|
|           | $t$ [s]               | $t$ [s]                | $t$ [s]                |
| DANFAGPC  | 0.0010                | 0.0020                 | 0.0045                 |
| SFNNGPC   | 0.0012                | 0.0022                 | 0.0037                 |
| NFNGPC    | 0.0008                | 0.0016                 | 0.0038                 |
| T2NFNGPC  | 0.0010                | 0.0021                 | 0.0039                 |
| INFNGPC   | 0.0009                | 0.0017                 | 0.0035                 |
| NFMGPC    | 0.0019                | 0.0038                 | 0.0063                 |

Добре известен факт е, че методът за оптимизация оказва съществено влияние върху изчислителната ефективност на оптимизатора. В настоящата дисертация са разработени алгоритми от втори ред, а именно тези на Нютон и на Левенберг-Маркгуард, тъй като при тях е намален броя на итерациите, свързани със сходимостта на алгоритъма, а това от своя страна води до по-

голямо бързодействие на предсказващия регулатор. Последното е видно от фиг.4.14, на която е направено сравнение между обобщен предсказващ регулатор с SFNN модел, използващ различни методи за оптимизация – градиентен от първи ред, Нютон и Левенберг-Маркгуард. При едни и същи стойности на хоризонтите на управление и предсказване и на тегловният коефициент  $\rho$  ( $N_1=1$ ;  $N_u=5$ ;  $N_p=5$ ;  $\rho=0.002$ ) в рамките на 100 стъпки регулаторът с градиентен метод не може да достигне заданието. В същото време за регулатор с метод на Нютон са необходими около 25 стъпки, а на регулатор с метод на Левенберг-Маркгуард около 70 стъпки. На стъпка 300 е подадено смущение, което много бързо е отработено от регулаторите с Нютон и Левенберг-Маркгуард оптимизация, докато регулатора с градиентен метод изпитва затруднение.



Фиг. 4.14 Сравнение на обобщено предсказващо управление, реализирано с различни оптимизационни алгоритми



Фиг. 4.15 Преходен процес при променливо задание, промяна в параметрите на обекта и добавяне на смущение в стъпка 400

Основен проблем при градиентните методи от втори ред е необходимостта на всеки такт да се намира обратната матрица на Хесе. Това е процедура, която изисква допълнителни изчисления и следователно ще доведе до увеличаване на времето  $t$ , необходимо за реализиране на изчисленията в оптимизатора за един такт. Данни за това са представени в Табл. 4.4, като на хоризонтите на управление и на предсказване са еднакви ( $N_1=1$ ;  $N_u=N_p=5$ ;  $\rho=0.002$ ).

Таблица 4.4

| Регулатор | Метод на Нютон | Метод на Левенберг-Маркгуард | Метод на Нютон с LU декомпозиция | Метод на Левенберг-Маркгуард с LU декомпозиция |
|-----------|----------------|------------------------------|----------------------------------|------------------------------------------------|
|           | t [s]          | t [s]                        | t [s]                            | t [s]                                          |
| DANFAGPC  | 0.0029         | 0.0029                       | 0.0026                           | 0.0026                                         |
| SFNNGPC   | 0.0026         | 0.0026                       | 0.0022                           | 0.0022                                         |
| NFNGPC    | 0.0023         | 0.0023                       | 0.0019                           | 0.0019                                         |
| T2NFNGPC  | 0.0027         | 0.0027                       | 0.0021                           | 0.0021                                         |
| INFNGPC   | 0.0022         | 0.0022                       | 0.0018                           | 0.0018                                         |
| NFMGPC    | 0.0055         | 0.0055                       | 0.0047                           | 0.0047                                         |

Резултатите от табл. 4.4 показват, че в изчислителен план разлика между метода на Нютон и този на Левенберг-Маркгуард на практика не съществува. Що се отнася до качеството на управление метода на Нютон осигурява по-бързо достигане до заданието, но и преходен процес с пререгулиране. За да се избегне изчислително тежката процедура по определянето на обратната матрица на Хесе на всеки такт при градиентните методи от втори ред е използван метода на LU декомпозицията, която в програмната среда Matlab се осъществява с операцията дясно деление. На фиг. 4.15 са представени преходните процеси при управление на химичния реактор с обобщен предсказващ регулатор с Нютон метод за оптимизация с изчисляване на обратната матрица на Хесе и с LU декомпозиция. Вижда се, че с въвеждането на LU декомпозиция колебанията в преходния процес се „изглаждат” и характерът му се свежда до аperiодичен. Освен това, от данните в табл.4.4 става ясно, че при модифицираните алгоритми на Нютон и на Левенберг-Маркгуард времето  $t$  за реализиране на изчисленията в оптимизатора за един такт намалява.

Реализиран е алгоритъм на обобщен предсказващ регулатор, при който се осъществява адаптивна супервайзорна донастройка на този коефициент. Той е изграден на базата на невронно-размития модел, описан в Глава II и Нютонова оптимизационна процедура, и е тестван в реални условия за управление на топлинна система. Експериментите са направени при следните стойности на параметрите:

- Параметри на регулатора:  $N1=1$   $Nu=3$   $Np=5$
- Постоянно задание:  $r=65$  °C
- Променливо задание:  $r=65$  °C при  $t=[0:2000]s$ ;  $r=70$  °C при  $t=[2000:3000]s$ ;  $r=55$  °C при  $t=[3000:5000]s$

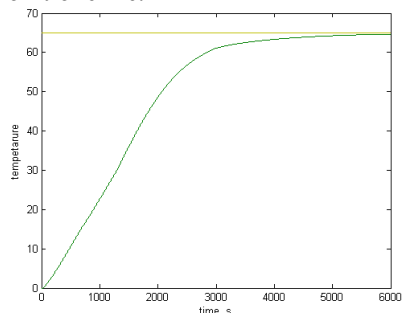
• Скорост на обучение на невронно-размития модел:  $\eta=0.09$

• Скорост на обучение на невронно-размития модел в супервайзора:  $\eta_{sup}=0.07$ .

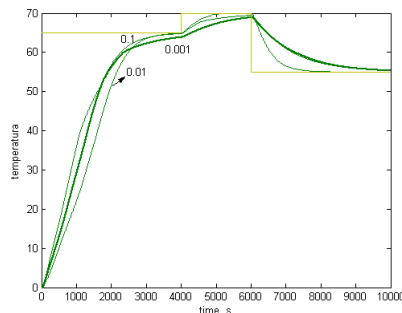
Супервайзорният елемент е реализиран в два варианта - с механизъм на извеждане Мамдани и с механизъм на извеждане Такаги-Сугено. Резултатите, получени в първия случай са показани на фиг. 4.16 и фиг. 4.17. Поведението на системата при еднакъв брой размити правила в структурата на супервайзора и различен начален коефициент  $r$  показват проява на сходна динамика на процесите. Вижда се, че изборът на начална стойност на този параметър е определящ за динамиката на преходния процес. Увеличаването му води до по-големи стойности на пререгулирането и по-бързо достигане на желаната стойност. Трябва да се отбележи и фактът, че наличието на модификации в схемата на предсказващия регулатор свързани с допълнителни изчисления води до значително влияние върху времето на преходния процес. От друга страна това забавяне получено от наличието на супервайзора в управляващия контур може да се компенсира с по-голям начален коефициент  $r$ . Това запазва динамиката на процеса, като влияе благоприятно върху намаляване на времето на преходния процес и максималното динамично



отклонение.

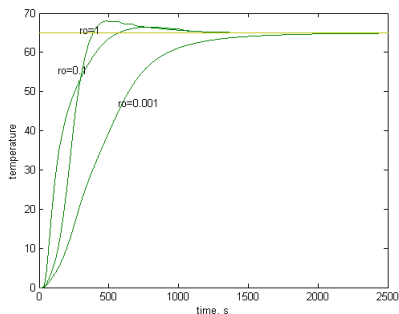


Фиг. 4.16 Преходен процеси при постоянно задание и при начално  $\rho=0.001$

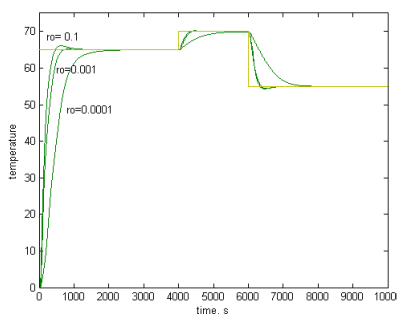


Фиг. 4.17 Преходни процеси при променливо задание и различни начални стойности на  $\rho$

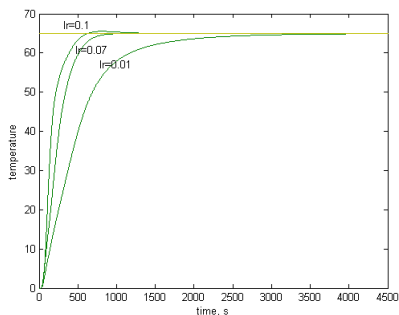
По-долу са представени резултатите с използване на супервайзорен елемент с механизъм на извеждане Такаги-Сугено. На фиг. 4.18 са дадени преходните процеси при постоянно задание и при различни начални стойности на  $\rho$ , а на фиг. 4.19 при променливо задание и различни начални стойности на  $\rho$ . Използваните различни начални стойности на тегловния коефициент са както следва:  $\rho=1$ ;  $\rho=0.1$ ;  $\rho=0.001$  при постоянно задание и при променливо:  $\rho=0.1$ ;  $\rho=0.001$ ;  $\rho=0.0001$ . Скоростта на обучение на тегловния коефициент в случая е избрана да бъде постоянна  $\eta_{\text{sup}}=0.07$ . На фиг. 4.20 са представени резултати при различни стойности на скоростта на обучение на невронно-размития модел в супервайзора. Началния коефициент е приет да бъде  $\rho=0.001$  съответно при  $\eta_{\text{sup}}=0.1$ ;  $\eta_{\text{sup}}=0.07$ ;  $\eta_{\text{sup}}=0.01$ . Изменението на тегловния коефициент  $\rho$  при различни начални стойности е показано на фиг. 4.21, където е прието:  $\eta_{\text{sup}}=0.07$  и  $\rho=0.1$ ;  $\rho=0.001$ ;  $\rho=0.0001$ .



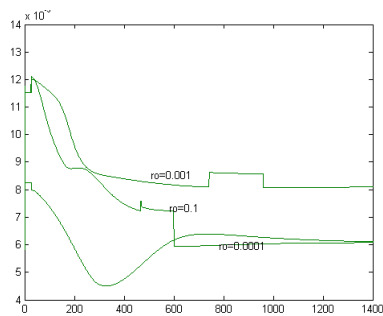
Фиг. 4.18 Преходни процеси при постоянно задание и при различни начални стойности на  $\rho$



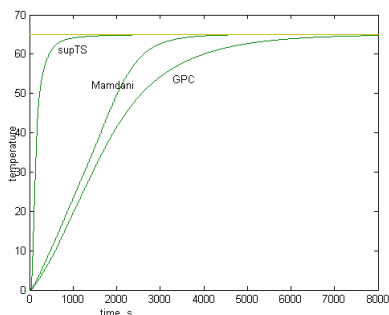
Фиг. 4.19 Преходни процеси при променливо задание и при различни начални стойности на  $\rho$



Фиг. 4.20 Преходни процеси при различни стойности на скоростта на обучение



Фиг. 4.21 Изменението на  $\rho$  при различни начални стойности



Фиг. 4.22 Сравнение между различните методи на управление

Сравнението между различните типове схеми на управление е направено при еднакви условия (фиг.4.22). Стойността на тегловния коефициент при GPC управлението е приет да бъде  $\rho=0.001$ , съответно и за началните стойности при двата метода със супервайзор. От показаните преходни процеси се вижда, че метода с допълнително супервайзорно ниво използвава размитите структури на Такаги-Сугено дава най-добри резултати. Адаптивната донастройка на тегловния коефициент осигурява бързодействие и качество на процеса. Въпреки фактът, че наличието на модификации в схемата на предсказващия регулатор свързани с допълнителни изчисления води до значително влияние върху времето на преходния процес, в случая не се отразява на бързодействието за достигане на желаната стойност. При метода използващ подхода на Мамдани за донастройката на  $\rho$  това обаче не е така. Ако се сравни със метода използващ NewtonGPC управление, лесно може да се забележи забавянето получено от наличността на допълнителното супервайзорно ниво. То обаче може да се компенсира със задаването на по-висока начална стойност на тегловния коефициент, което намалява времето на

преходния процес. Така от направеното сравнение може да се стигне до заключението че, системата със супервайзорно донастройващо ниво използващо Такаги-Сугено модел е един по-добър вариант за управление на изследваната система, който осигурява по-добро качество и ефективност на процеса.

## **Глава V. Софтуерна реализация на предложените невронно-размити модели и алгоритми за регулатори с невронно-размито предсказващо управление**

Софтуерната реализация на предложените невронно-размити модели с редуциран брой размити правила и алгоритми за предсказващи регулатори е осъществена в програмната среда Matlab. Програмите, реализиращи отделните невронно-размити модели, като цяло имат сходна структура и работят по един и същ начин. Те могат да бъдат условно разделени на следните части: инициализация, невронно-размит модел, обучение и визуализация на графичните резултати. В софтуерната реализация на отделните модели различия има главно във втора и трета част. Те са свързани с типа на програмираната невронно-размитата структура и с избрания метод за обучението ѝ. В същото време инициализацията и представянето на резултатите се осъществяват по идентичен начин.

Софтуерната реализация на предсказващ регулатор включва програмнен код на невронно-размит модел и на оптимизатор. Последният получава като вход стойностите от предсказващия модел. Оптимизаторът е реализиран в няколко варианта: с градиентен алгоритъм от I ред, с алгоритъм на Нютон и с алгоритъм на Левенберг-Маркгуард. Последните два принадлежат към групата градиентни алгоритми от II ред. Характерно за тях е необходимостта от изчисляването на обратна матрица на Хесе на всеки такт на дискретизация, което внася в алгоритъма допълнително изчислително натоварване. За да се преодолее този недостатък тези два алгоритъма са модифицирани чрез въвеждане на LU декомпозиция.

Програмният код на алгоритъм за супервайзорна донастройка на регулатор с невронно-размито предсказващо управление до голяма степен се припокрива с този на невронно-размит предсказващ регулатор. В него се включва конкретен невронно-размит модел и оптимизационен алгоритъм. Съществено различие се явява добавянето на супервайзорния елемент. В ролята на супервайзор би могъл да се използва всеки един от предложените невронно-размити модели. Особеното в случая е, че като входове се използват стойностите на системната грешка, а изход е стойността на тегловния коефициент  $p$ , който се подава към оптимизатора.

## Глава V. Заключение и изводи

Настоящият дисертационен труд е посветен на разработването на невронно-размити модели за целите на предсказващото управление, т.е. такива, които гарантират висока точност на моделиране и намалена изчислителна тежест. В търсене на конкретни решения на поставените задачи са предложени три невронно-размити модела, работещи с редуциран брой размити правила и подходящи за включване в системи с НМПУ. Това са *Разпределен невронно-размит модел (DANFA)*, *Невронно-размит модел с частично размиване на входните сигнали (SFNN)* и *Модифициран нео-размит модел*. За да могат да се справят успешно в условията на неопределености DANFA, SFNN и Модифицираният нео-размит модел са реализирани във вариант с Тип 2 размита логика и с Интуиционистка размита логика. Това повишава тяхната точност на предсказване, но е за сметка на завишен брой параметри, които трябва да се актуализират по време на обучението.

Едно от основните предимства, на които предсказващите регулатори дължат широкото си приложение, е възможността им да се справят с многомерни обекти с неизвестно или променливо времезакъснение. За да бъде възможно това е нужно да се разработят многомерни невронно-размити модели, за които също е в сила изискването да осигуряват висока точност на предсказване и да са с малък брой параметри за настройка. Това е постигнато с реализираният в многомерен вариант модифициран нео-размит модел.

Проектирани са и различни алгоритми за обучение на предложените невронно-размити структури. Наред с превърналия се в класически метод с обратно разпространение на грешката, са разработени също и хибридни алгоритми, комбиниращи рекурентен метод на най-малките квадрати, метод на Нютон и метод на Левенберг-Маркгуард за обучение на параметрите във функцията на Сугено с градиентен метод от I ред за обучение на параметрите на размитите множества.

При системите с НМПУ методът за оптимизация е от особено значение, тъй като оказва съществено влияние върху изчислителната ефективност на оптимизатора. С оглед на това в настоящата дисертация са предпочетени алгоритми от втори ред, а именно тези на Нютон и на Левенберг-Маркгуард, тъй като при тях е намален броят на итерациите, свързани със сходимостта на алгоритъма, а това от своя страна води до по-голямо бързодействие на предсказващия регулатор.

Изследвана е възможността за използване на интелигентни невронно-размити структури за настройката на предсказващите регулатори. За целта е реализиран алгоритъм за супервайзорна адаптивна донастройка на тегловния коефициент в целевия критерий, като се запазват постоянни стойностите на хоризонтите на управление и на предсказване. Една интересна идея за бъдещи изследвания е използване на многомерния модифициран нео-размит модел за

цялостна настройка на предсказващите регулатори, т.е. за определянето както на тегловния коефициент, така и на хоризонтите на управление  $N_u$  и на предсказване  $N_p$ .

Работоспособността и ефективността на предложените алгоритми и изследвана по два начина. Първоначално е изследвана способността на DANFA, SFNN и модифицирания нео-размит модел да предсказват хаотични времеви серии. За обучението им е използван двустъпков градиентен алгоритъм с постоянна скорост на обучение. Направено е сравнение на тези

модели с невронно-размития модел, описан в т.2.1, при което за най-добър модел в съответствие със заложените критерии е определен модифицирания нео-размит модел. Въпреки, че работи с по-голям брой размити правила в сравнение с SFNN Type III, модифицирания нео-размит модел има по-малък брой параметри за актуализиране в процеса на обучение (дори и в случая с обучаване параметрите на размитите множества), по-точен е и по-бърз. Ето защо именно модифицирания нео-размит модела е реализиран във вариант с Тип 2 размита логика и с Интуиционистка размита логика в присъствието на шум, за да се види как се справя в условия на неопределености. От получените резултати става ясно, че Интуиционисткият нео-размит модел работи с по-малка грешка в сравнение с Тип 2 нео-размит модел. Времето за осъществяване на необходимите изчисления за една стъпка от обучението при Интуиционисткия нео-размит модел е съответно  $1.48e-4$  за случая без шум и  $1.50e-4$  със шум. При Тип 2 нео-размит модела това време е по-голямо – съответно  $1.93e-4$  и  $1.96e-4$  за случаите без и със шум. Следователно Интуиционисткият нео-размит модел е по-точен и по-бърз и това го прави подходящ за включване в предсказващ регулатор. Модифицираният нео-размит модел е разработен и в ММО вариант, тъй като едно от предимствата на предсказващите регулатори е способността им да се справят с многомерни обекти.

При втората група експерименти всеки един от предложените невронно-размити модели е използван за реализиране на обобщен предсказващ регулатор с итеративен оптимизационен алгоритъм от първи и от втори ред. От получените резултати може да се заключи, че структурата на използвания невронно-размит модел не оказва влияние върху качеството на управление. С използването на предложените невронно-размити модели се съкращава приблизително два пъти времето  $t$ , необходимо за реализиране на всички изчисления в оптимизатора за един такт. Това време зависи също и от хоризонтите на предсказване и на управление и нараства с увеличаването им.

Бързодействието на невронно-размит обобщен предсказващ регулатор значително се увеличава когато се използва итеративен оптимизационен алгоритъм от втори ред. Основната трудност при тези алгоритми - изчисляването на обратната матрица на Хесе на всеки такт – е преодоляна с въвеждането на LU декомпозиция. От получените резултати се вижда, че с

въвеждането на LU декомпозиция колебанията в преходния процес се „изглаждат“ и характерът му се свежда до аperiодичен. Става ясно също, че при модифицираните с помощта на Гаусовата елиминация алгоритми на Нютон и на Левенберг-Маркгуард времето  $t$  за реализиране на изчисленията в оптимизатора за един такт намалява. Що се отнася до настройката на предсказващите регулатори един съществен извод, който може да се направи е, че тегловният коефициент  $\rho$  оказва съществено влияние върху качеството на управление, но не и върху времето за работа на оптимизатора. Действието му може да бъде сравнено с това на пропорционалната съставка на ПИД регулатора.

В заключение може да се каже, че предложените DANFA, SFNN, Модифициран нео-размит модел и техните варианти с Тип 2 размита логика и с Интуиционистка размита логика удовлетворяват напълно поставените в дисертационния труд цел и задачи. От представените резултати е видно, че те осигуряват предсказване с висока точност като същевременно са с намалена изчислителна тежест. Включването им в обобщен предсказващ регулатор води до намаляване на времето за изчисления приблизително два пъти. В същото време обаче, приложението на DANFA, SFNN и Модифицирания нео-размит модел не трябва да се ограничава само до предсказващите регулатори. Те биха могли успешно да се използват при разпознаване на образи и обработка на изображения, в ролята на класификатори, за обработка на данни и извличане на знания от тях, в системите за сигурност и др.

### **Приноси в дисертационния труд**

В резултат от изпълнението на поставената цел, а именно разработване на алгоритми за нелинейно предсказващо управление на базата на невронно-размит предсказващ модел с механизъм на извеждане Такаги-Сугено с намалена изчислителна тежест, са формирани следните научно-приложни и приложни приноси:

1. Предложен и софтуерно реализиран е DANFA модел, който работи с редуциран брой размити правила и има намален брой параметри.
2. Предложен е втори изчислително ефективен невронно-размит модел, а именно SFNN модел, който е програмно реализиран в три варианта.
3. Разработен е програмен код за модифициран нео-размит модел, работещ с минимален брой размити правила и осигуряващ висока точност на моделиране.
4. Софтуерно е реализиран модифициран нео-размит модел във вариант с Тип 2 размита логика и с Интуиционистка размита логика, с оглед постигане на по-добро управление в условия на неопределеност.
5. Реализирани са програмно итеративни градиентни оптимизационни алгоритми от втори ред, осигуряващи бързодействие на нелинеен невронно-размит обобщен предсказващ регулатор.

6. Оптимизационните алгоритми от II ред са реализирани софтуерно във вариант с LU декомпозиция – изчислително ефективен метод, който позволява да се избегне намирането на обратната матрица на Хесе на всеки такт.
7. Разработен е програмно алгоритъм, за супервайзорна адаптивна донастройка на тегловния коефициент в целевия критерий за оптимизация.

### **Списък с публикации по дисертационния труд**

#### **Доклади в списания**

1. **Terziyska, M.**, A Distributed Adaptive Neuro-Fuzzy Network for Chaotic Time Series Prediction, Cybernetics and Information Technologies. Vol. 15, Issue 1, Pages 24–33, ISSN (Online) 1314-4081, DOI: 10.1515/cait-2015-0003, March 2015.
2. **Terziyska, M.**, Y. Todorov, Fuzzy-neural predictive control using fast optimization polices, International Journal of Intelligent Reasoning-Based Systems, vol. 6, issue 3/4, pp. 136-144, ISSN online: 1755-0564, ISSN print: 1755-0556, 2014. DOI: 10.1504/IJIRIS.2014.066250.
3. **Terziyska, M.**, Y. Todorov, M. Petrov, Real-time Supervisory tuning of Predictive Controller, International Conference TechSys'2013, Plovdiv Bulgaria, Published in Journal of Technical University-Sofia, branch Plovdiv, Bulgaria, “Fundamental Sciences and Applications” Vol. 19, pp. 155-160, ISSN 1310-8271, 2013.
4. **Terziyska, M.**, Y. Todorov, L. Doukovska, Neo-fuzzy Network for Modeling of Nonlinear MIMO Dynamics, International Conference TechSys'2015, Plovdiv Bulgaria, Published in Journal of Technical University-Sofia, branch Plovdiv, Bulgaria, “Fundamental Sciences and Applications” Vol. 21, book 1, pp. 65-70, ISSN 1310-8271, 2015.

#### **Доклади от международни конференции**

5. **Terziyska, M.**, L. Doukovska, M. Petrov, Implicit GPC based on Semi Fuzzy Neural Network Model, IEEE Intelligent Systems IS'14, Warsaw, Poland, September 24-26, 2014.
6. Todorov, Y., M. **Terziyska**, S. Ahmed, M. Petrov, Fuzzy-Neural Predictive Control using Levenberg-Marquardt optimization approach, In Proceedings of International IEEE conference on Innovations in Intelligent Systems and Applications, INISTA'2013, Albena, Bulgaria, pp. 1-5, ISBN 978-1-4799-0659-8, 2013.
7. Todorov, Y., M. **Terziyska**, Modeling of Chaotic time series by Interval Type-2 NEO-Fuzzy Neural Network, International Conference on Artificial Neural Networks (ICANN'2014), Hamburg, Germany, Springer Lecture Notes on Computer Science, vol. 8681, pp. 643-650, ISBN 978-3-319-11178-0, ISSN 0302-

### **Участие в научно-изследователски проекти**

1. Договор № ДО1-192 за съфинансиране на научен проект "AComIn: Advanced Computing for Innovation" №316087 по Седма рамкова програма на Европейската комисия, ръководител проф. дмн Галя Ангелова.
2. Договор № ДФНИ-И02/5 от 12.12. 2014 г. на тема: "Интеркритериален анализ - нов подход за вземане на решения", ръководител чл.-кор. проф. дмн д-р Крaсимир Атанасов.

### **Библиография**

Alavala, C. (2008). Fuzzy Logic and Neural Networks: Basic Concepts and Applications. New Age International Pvt Ltd.

Chen, H. and F. Allgöwer (1998). A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability. Automatica, Vol. 34, Issue 10, pp. 1205-1218.

De Nicolao, G., L. Magni and R. Scattolini (1996). Stabilizing nonlinear receding horizon control via a nonquadratic terminal state penalty. In Symposium on Control, Optimization and Supervision, CESA'96 IMACS Multiconference, pp. 185-187, Lille.

Diehl, M. (2001). Real-time Optimization for Large Scale Nonlinear Processes. PhD Thesis, University of Heidelberg.

Lee, J. H., (2000). Modeling and Identification for Nonlinear Model Predictive Control: Requirements, Current Status and Future Research Needs. Progress in Systems and Control Theory, Vol. 26, Basel, Boston, Berlin: BirkhauserVerlag.

Liutkevičius, R. and S. Dainys (2005). Hybrid Fuzzy Model of Nonlinear Plant. Information Technology and Control, Vol.34, No.1.

Pan, Y. and J. Wang (2012). Model predictive control of unknown nonlinear dynamical systems based on recurrent neural networks. IEEE Trans. Ind. Electron., vol. 59, no. 8, pp. 3089–3101.

Pearson, R. K., (2003). Selecting nonlinear model structures for computer control. Journal of Process Control, Vol. 13, Issue 1, pp. 1-26.

Tenny, M. (2002). Computational Strategies for Nonlinear Model Predictive Control. PhD Thesis, University of Wisconsin-Madison.

Yamakawa, T., E. Uchino, T. Miki, and H. Kusanagi (1992). A neo fuzzy neuron and its applications to system identification and prediction of the system behavior. In Proc. 2-nd Int. Conf. on Fuzzy Logic and Neural Networks - IIZUKA-92, Japan, pages 477–483.