



Министерство на Образованието и Науката
Университет “Проф. д-р Асен Златаров” - Бургас
Факултет по Технически науки
Катедра Компютърни Системи и Технологии

маг. инж. Иван Торлаков

**ПРИЛОЖЕНИЕ НА
ВИСОКО-ПРОИЗВОДИТЕЛНИ СИСТЕМИ ЗА
МОДЕЛИРАНЕ НА УСТОЙЧИВИ ПРОЦЕСИ
ПРИ НЕВРОННИ МРЕЖИ ОТ ТИП
КОЕН-ГРОЗБЕРГ**

АВТОРЕФЕРАТ

На дисертационен труд за придобиване на
образователна и научна степен “Доктор”
по научна специалност: Комуникационна и
компютърна техника

Научни ръководители:

Проф. д-р Гани Трендафилов Стамов
Проф. д-р Станислав Денчев Симеонов

Ноември 2022
гр. Бургас

Дисертационният труд е обсъден на разширен катедрен съвет при катедра Компютърни системи и технологии, Университет “Проф. д-р Асен Златаров” - гр. Бургас, на заседание, състояло се на 2022 г. и е насрочен за разкриване на процедура за защита пред жури със заповед 2022 г. на Ректора на Университета.

Дисертационният труд съдържа 149 страници, 8 таблици, 26 фигури и 41 алгоритъма. В библиографията са включени 123 заглавия.

Защитата на дисертационния труд ще се състои на 2022 г. в Университет “Проф. д-р Асен Златаров” - гр. Бургас.

Материалите за защитата са на разположение на интересуващите се в деловодството на Университет “Проф. д-р Асен Златаров” - гр. Бургас.

Автор: маг. инж. Иван Торлаков

Тема: Приложение на високо-производителни системи за моделиране на устойчиви процеси при невронни мрежи от тип Коен-Грозберг

Използвани съкращения

- API - Application Programming Interface
- BLAS - Basic Linear Algebra Subroutine
- BMU - Best Matching Unit
- CODINE - COmputing in DIstributed Networked Environments
- CSV - Comma Separated Values
- CUDA - Compute Unified Device Architecture
- FFS - Fast Fourier Transform
- GPGPU - General-Purpose computing on GPUs
- GPU - Graphics Processing Unit
- HIP - Heterogeneous-Computing Interface for Portability
- HLSL - High-Level Shared Language
- HPF - High Performance Fortran

- IPC - Inter Process Communication
- ISA - Instruction Set Architecture
- LAN - Local Area Network
- LSF - Load Sharing Facility
- MIMD - Multiple Instruction Multiple Data
- MISD - Multiple Instruction Single Data
- MPI - Message Passing Interface
- NOW - Network of Workstations
- OpenACC - Open Accelerators
- OpenCL - Open Computing Language
- OpenMP - Open Multi-Processing
- P2P - Point-to-Point
- PTX - Parallel Thread Execution
- PVM - Parallel Virtual Machine
- SDK - Software Development Kit
- SIMD - Single Instruction Multiple Data
- SISD - Single Instruction Single Data
- SM - Streaming Multiprocessors
- SOM - Self-organizing Map

- SP - Streaming Processors
- SSI - Single System Image
- SSH - Secure Shell
- STM - Short Term Memory

Увод

В нашето съвремие се наблюдават много приложения на изкуствени невронни мрежи в различни сфери от живота. Изучаването на моделите и тяхното комбиниране и развиване, дава възможност за все по-широк спектър на приложението им, а така също и за добавяне на нови възможности и тяхното усъвършенстване, особено когато се прилагат към внедряване на мащабни инфраструктури.

Подобряването на системите с интелигентно поведение и интеграцията им в сферата на информационните технологии предоставят нови методи за обработка на огромното количество данни без нуждата или минимална нужда от човешка намеса. Оптимизирането на софтуерните реализации на различни видове невронни мрежи може да доведе до усъвършенстване на моделите, чрез обратната трансформация от софтуера към модела.

През последните няколко години се отделя голямо внимание на изследването на динамиката на невронните мрежи, дължащо се на обширното им приложение в голям набор от области. Анализът на динамичните поведения на невронните мрежи е важна стъпка към правилното практи-

ческо проектиране. Разглеждат се различни алгоритми, които бързо намират своето практическо приложение в сферите от живота.

В областта на информационните технологии се забелязва тенденция за изместване на интереса от увеличаване на мощността на изчислителните средства към повишаване на тяхната “интелигентност”. Създават се нови подходи, основани върху вече натрупани научни знания. Фундаменталният проблем тук не е създаването на нови мощни методи, а представянето на голям обем знания във вид, позволяващ тяхното ефективно използване и взаимодействие. Затова наред с техническите подходи в информационните технологии, прилагани през последните 20 години добавящи мощност, интензивно се развива ново научно направление на изкуствения интелект. Тази сравнително нова насока на развитие, предлага нови предизвикателства пред разработчици и проектантите и предполага паралелно развитие в много области на човешкото познание.

Предмет на дисертационния труд е получаване, изследване и анализиране на устойчиви процеси при невронни мрежи от тип Коен-Грозберг с двупосочна асоциативна памет с променливи във времето закъснения и променливи импулсни смущения чрез софтуерна реализация и използването на високопроизводителни системи. Създаването на такъв тип система е актуална тема поради ограничения брой реализирани софтуерни модели, библиотеки и системи за работа с конкретния модел изкуствени невронни мрежи. Използването на паралелна техника дава преимущество за скалируемост за получаване на резултати, което води до ускоряване на аналитичното обработване на получените

данни.

В дисертационния труд целта е да се моделират устойчиви процеси при невронни мрежи от тип Коен-Грозберг с двупосочна асоциативна памет с променливи във времето закъснения и променливи импулсни смущения, с помощта на високопроизводителна и паралелна техника. Да се имплементират програмно приложение с възможност за въвеждане на входни параметри, извличане на резултатни данни, съставяне на графики на получените резултати и изготвяне на доклади съгласно разглежданите необходими зависимости за стабилност на модела, демонстрираща работоспособността на модела.

За постигането на формулираните цели са поставени следните научни задачи:

- Да се проучат съществуващи софтуерни решения за тип Коен-Грозберг с двупосочна асоциативна памет или сходен модел невронни мрежи;
- Да се създаде математически модел и да се изследват условията за стабилност при мрежа от два неврона;
- Да се моделират условията за стабилност на разглеждания модел невронни мрежи;
- Да се създаде и реализира класически алгоритъм за невронни мрежи на разглеждания модел;
- Да се реализират и изследват паралелни алгоритми за условия за стабилност на съществуваща конфигурация;

- Да се реализира паралелен алгоритъм на база CUDA;
- Да се реализира паралелен алгоритъм на база OpenMPI;
- Да се изследват резултатите и тяхното представяне в предвид на техния обем.

Глава 1

Литературен обзор

1.1 Невронни мрежи

В настоящата глава са разгледани развитието на невронните мрежи през последните десетилетия. Представена е историческа ретроспекция на развитието от използването на един неврон за решаване на точни казуси към използването на едно и многослойни невронни мрежи за разширяване на техните възможност за решаване на по-голям набор от модели, от която може да се проследи тяхното развитие. Съществува тенденция към усъвършенстването на съществуващите модел и тяхното комбиниране с други типове невронни мрежи за постигане на по-голяма гама от възможности за приложения в практиката. Активно се използват невронни мрежи в сфери като авиотехниката, електрониката, информационните технологии, медицински и производствен сектор и други.

Направено е проучване на различни видове неврони, като персептрон и модел на МакКулох и Пийтс и техните особености, и едно- и многослойни невронни мрежи като модели с пряко предаване, семпли конкурентни, рекурентни мрежи, самоорганизиращи се карти, модели на мрежа на Грозберг и Коен-Грозберг.

Изводи:

Като резултат от направения обзор и проучване по темата могат да се направят следните изводи:

- Невронните мрежи участват все по-широко в различни сфери на практиката, като играят роля на допълнителен контрол и компаниите разчитат на получените резултати. В началото невронните мрежи са решават по-семпли задачи и с тяхното развитие се разширява и използването им;
- Актуална е темата за разработването на модел и софтуер за намирането на устойчиви процеси при невронни мрежи от тип Коен-Грозберг с двупосочна асоциативна памет с променливи във времето закъснения и променливи импулсни смущения, с помощта на високопроизводителна и паралелна техника.

Цели и задачи на дисертационния труд

Основна целта на дисертационния труд е да се моделират устойчиви процеси при невронни мрежи от тип Коен-Грозберг с двупосочна асоциативна памет с променливи във времето закъснения и променливи импулсни смущения, с помощта на високопроизводителна и паралелна техника. Да се реализира програмно приложение с възможност за въвеждане на входни параметри, извличане на резултатни данни, съставяне на графики на получените резултати и изготвяне на доклади съгласно разглежданите необходими зависимости за стабилност на модела, демонстриращо работоспособността на модела.

Така формулираната цел може да бъде постигната чрез изпълненото на следните задачи:

- Да се проучат съществуващи софтуерни решения за тип Коен-Грозберг с двупосочна асоциативна памет или сходен модел невронни мрежи;
- Да се създаде математически модел и да се изследват условията за стабилност при мрежа от два неврона;

- Да се моделират условията за стабилност на разглеждания модел невронни мрежи;
- Да се създаде и реализира класически алгоритъм за невронни мрежи на разглеждания модел;
- Да се реализират и изследват паралелни алгоритми за условия за стабилност на съществуваща конфигурация;
- Да се реализира паралелен алгоритъм на база CUDA;
- Да се реализира паралелен алгоритъм на база OpenMPI;
- Да се изследват резултатите и тяхното представяне в предвид на техния обем.

Посочените по-горе задачи и възможности определят и актуалността на дисертационната работа. Надеждността на научните резултати се потвърждава от теоретични изследвания, данни от разработената система, както и от сравняване на получените резултати с резултатите, дадени в научната литература. Практическото значение на изследването се определя от факта, че прилагането на разработения подход може да бъде използван за намирането на устойчиви точки на невронни мрежи чрез използването на паралелна техника. Създаването на такъв тип система е актуална тема поради ограничения брой реализирани софтуерни модели, библиотеки и системи за работа с конкретния модел изкуствени невронни мрежи. Използването на паралелна техника дава преимущество за скалируемост за получаване на резултати, което води до ускоряване на аналитичното обработване на получените данни.

Глава 2

Математически модел за изследване на качествени свойства на импулсни невронни мрежи от тип Коен-Грозберг

В тази глава се разглеждат математическия апарат от понятия и теореми, на базата на които се изгражда теоретичен модел на невронни мрежи от тип Коен-Грозберг с двупосочна асициативна памет с променливи във времето закъснения и променливи импулсни смущения. Разглеждат се основните резултати за h -устойчивост за равновесното състояние на модела. Дефинира се теорема и доказателствена част за равновесието z^* на импулсна Коен-Грозберг невронна мрежа с двупосочна асоциативна памет със закъснение по посочения модел 2.1 за глобална експоненциална

устойчивост по отношение на функцията h .

$$\left\{ \begin{array}{l} \dot{x}_i(t) = -a_i(x_i(t)) \left[b_i(x_i(t)) - \sum_{j=1}^m c_{ij} f_j(y_j(t)) \right. \\ \left. - \sum_{j=1}^m d_{ij} g_j(y_j(t - \sigma_j(t))) - I_i \right], t \neq \tau_k(x(t), y(t)), \\ \dot{y}_j(t) = -\hat{a}_j(y_j(t)) \left[\hat{b}_i(x_i(t)) \sum_{j=1}^n p_{ij} \hat{f}_j(x_i(t)) \right. \\ \left. - \sum_{i=1}^n q_{ij} \hat{g}_j(x_i(t - \hat{\sigma}_j(t))) - J_j \right], t \neq \tau_k(x(t), y(t)), \\ (x_i(t^+), y_j(t^+))^T = (x_i(t) + P_{ik}(x_i(t), y_j(t) + Q_{jk}(y_j)))^T, \\ t \neq \tau_k(x(t), y(t)), \end{array} \right. \quad (2.1)$$

Демонстрират се валидността на получените резултати в Теорема 5 критерии за глобална експоненциална устойчивост по отношение на многообразието.

Глава 3

Паралелни алгоритми за изследване на импулсни невронни мрежи от тип Коен-Грозберг

В тази глава е направено теоретично, технически и функционално описание на апаратна и програмна реализация за високопроизводителни и паралелни системи.

3.1 Апаратна реализация

Разгледани са класификацията на апаратната реализация. Това, което дава на високопроизводителните изчислителни решения предимство по мощност и скорост пред стандартните компютри, е техният хардуер и дизайн на системата. Използват се три **High-Performance Computing** дизайна: паралелни изчисления, клъстерни изчисления и мрежови и разпределени изчисления. Представена е теоретична и

практична обосновка на високопроизводителна машина на база CUDA реализация. Разгледани са архитектурата на високопроизводителна машина на база компютърен клъстер, класификацията на компютърните клъстъри, различните нива на компютърните клъстъри.

3.2 Програмно осигуряване

В тази секция са представени две различни по реализации програмни реализации за работа с високопроизводителна паралелна техника. Разгледани са библиотечния софтуер CUDA и стандарта OpenMPI.

3.2.1 CUDA

CUDA идва със софтуерна среда, която позволява на разработчиците да използват C++ като език за програмиране на високо ниво. Поддържат се и други езици, интерфейси за програмиране на приложения или базирани на директиви подходи, като Fortran, DirectCompute, OpenACC. CUDA C++ разширява C++, като позволява на програмиста да дефинира C++ функции, наречени kernel функции (функции на ядра), които когато бъдат извикани, се изпълняват N пъти паралелно от N различни CUDA нишки, за разлика от само веднъж като обикновените C++ функции.

Представени са йерархията при нишките и паметта в програмното осигуряване. За удобство `threadIdx` е три компонентен вектор, така че нишките могат да бъдат идентифицирани с помощта на едномерен, двумерен или тримерен

индекс на нишка, образувайки едномерен, двумерен или тримерен блок от нишки, наречени нишков блок. Това осигурява естествен начин за извикване на изчисление между елементите в даден домейн като вектор, матрица или 3D масив. CUDA нишките могат да имат достъп до данни от множество пространства на паметта по време на тяхното изпълнение. Всяка нишка има частна локална памет. Всеки блок на нишки има споделена памет, видима за всички нишки на блока и със същия живот като блока. Всички нишки имат достъп до една и съща глобална памет. Съществуват и две допълнителни пространства за памет само за четене, достъпни от всички нишки: константните и текстовните памет.

3.2.2 OpenMPI

OpenMPI е софтуерна реализация с отворен код на стандарта Message Passing Interface (MPI). MPI дефинира API, който се използва за специфичен тип преносима, високопроизводителна междупроцесна комуникация: предаване на съобщения. Документът описва надеждното прехвърляне на дискретни, въведени съобщения между MPI процесите. Представени са типовете комуникация между процесите - `point-to-point` и колективна.

`Point-to-point` комуникациите са разделени на две основни операции - изпращане и получаване. Най-основните форми на P2P комуникация се наричат блокиращи комуникации. Процесът, изпращащ съобщение, изчаква то да бъде получено от процеса-получател докато завърши с получаването на цялата информация.

Колективните комуникации позволяват да се обменят по-лесно информации във всички процеси на фиксиран комуникатор. Съществуват различни видове колективни комуникации, подходящи за много различни цели. Те биват Излъчване, Редукция, Разпръскване и Събиране. Подобно на комуникацията P2P също има имплементации за блокиращо и неблокиращо изпращане на данни.

Представена е работа с разделение на комуникатори в MPI. Съществува възможност да се създават собствени комуникатори и да се използват вместо стандартния MPI_COMM_WORLD. Комуникаторите са сложен предмет в MPI и в стандартната документация на MPI са разглеждани като отделна глава на тях и на свързаните с тях понятия. Съществуват група, контекстът и вътрешен комуникатор.

Всяка програма може да бъде разделена на четири основни категории според таксономия на Флин:

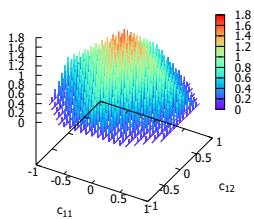
- SISD - Single Instruction, Single Data
- SIMD - Single Instruction, Multiple Data
- MISD - Multiple Instruction, Single Data
- MIMD - Multiple Instruction, Multiple Data

Глава 4

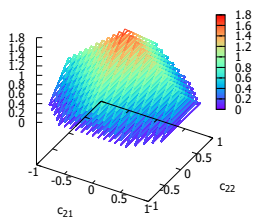
Алгоритми и имплементация на модел за стабилност при невронни мрежи тип Коен-Грозберг на ВППС

В тази глава е представено обобщението на използвания модел за разработването на софтуерните реализации.

Софтуерната реализация на математическия модел е разработена на програмния език С. Разработката премина през три етапа. Първоначално бе реализиран проекта без използването на паралелна техника, за да бъде разгледан и подобрен метода, по който се правят изчисленията. Графично обобщените данни от резултати са представени на фигури 4.1, 4.2, 4.3 и 4.4.

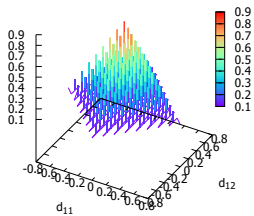


(а) μ данни за c_{1j}

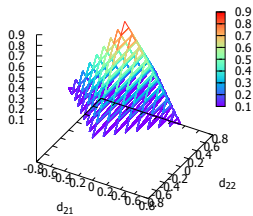


(б) μ данни за c_{2j}

Фигура 4.1: Резултати за c_{ij}

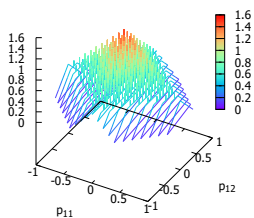


(а) μ данни за d_{1j}

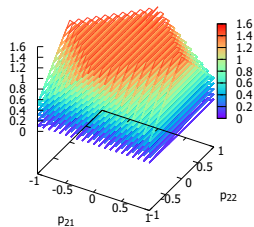


(б) μ данни за d_{2j}

Фигура 4.2: Резултати за d_{ij}

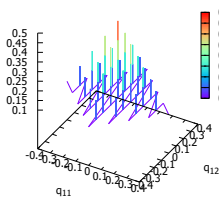


(a) μ данни за p_{1j}

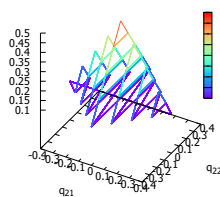


(б) μ данни за p_{2j}

Фигура 4.3: Резултати за p_{ij}



(a) μ данни за q_{1j}



(б) μ данни за q_{2j}

Фигура 4.4: Резултати за q_{ij}

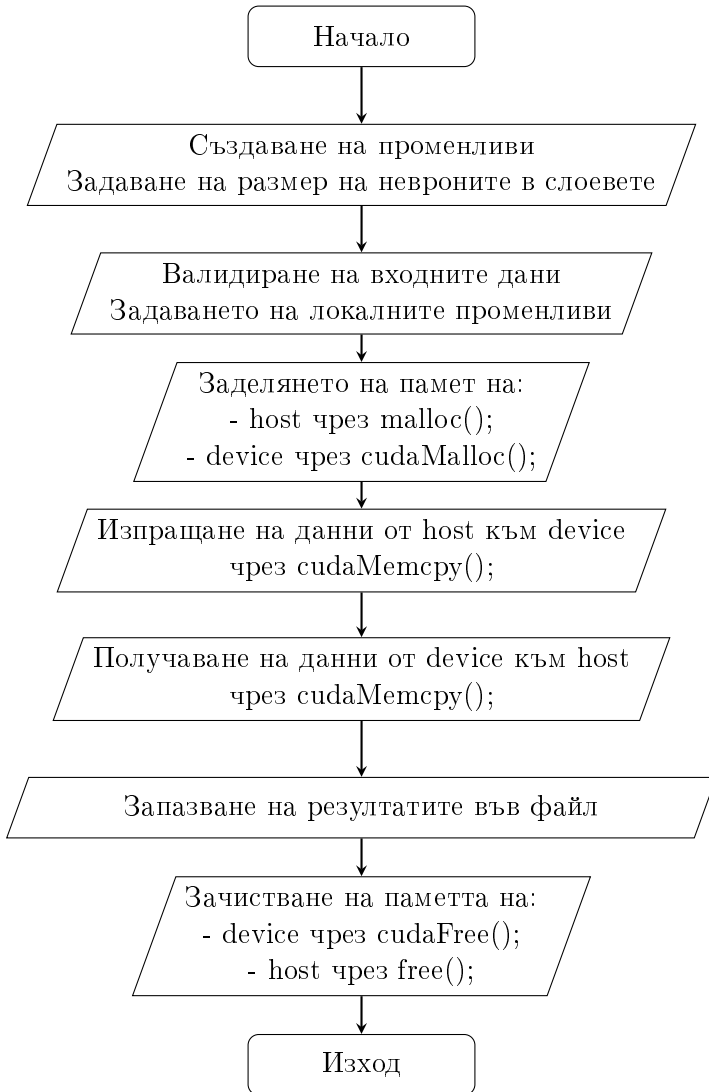
Във втория етап от разработката, софтуерната реализация бе постигната посредством технологията на NVidia наречена Compute Unified Device Architecture (CUDA).

Алгоритъма на `main` функцията има за цел да свърже CPU изпълнимия код с този на външния GPU код. Друга необходима функция е управлението, разпределението и почистването на паметта, както и валидирането на външните променливи. В базов коментар на алгоритъма на работа на текущата разработка е представен на 4.5.

Създадена е допълнителна помощна функция за обработка на грешките възникнали по време на изпълнение на програмата 4.1. Нейната функционалност се състои в проверка за наличието на дефинирана константа за `debug` режим на програмата. При зададен такъв режим се проверява резултата на структурата `cudaError_t`. При стойност различна от `cudaSuccess`, на стандартния изход за грешки `stderr` се извежда помощния текст на грешката чрез използването на `cudaGetErrorString(cudaError_t result)`.

```
inline cudaError_t cudaCheck(cudaError_t result) {
    #if defined(DEBUG) || defined(_DEBUG)
    if (result != cudaSuccess) {
        fprintf(stderr, "CUDA Runtime Error: %s\n",
        ↪ cudaGetErrorString(result));
        assert(result == cudaSuccess);
    }
    #endif
    return result;
}
```

Програмен код 4.1: Функция за обработка на грешките



Фигура 4.5: Блок схема на main функцията

Третата стъпка се извърши чрез използването на технологията OpenMPI върху клъстър от осем машини, всяка от които е снабдена с четири процесора Intel® Xeon® Gold 5218R (27.5M Cache, 2.10 GHz).

Софтуерното приложение може да бъде разделено на няколко отделни секции. От MPI_COMM_WORLD се взема ранга на текущия изпълнимия процес и общата бройка на всички налични възли за дадената задача. Създават се променливи, необходими за неблокиращата комуникация между отделните възли, за извеждане на информация и за статут на направена заявка в комуникационния “свят”. Извършва се проверка за коректни данни, като се проверява дали подадените стойности са логически коректни. Валидират се дали са предоставени всички задължителни параметри към програмата. Всяка стойност на параметър се преобразува към съответния тип данни, който е необходим за процеса. Псевдо описание на кода е представено на 4.2. При липса на подадени параметри, по-малък брой от необходимите или грешни стойности се извежда помощно меню на терминалния прозорец, което да насочи потребителя към правилното използване на компилираната програма.

```
if rank not 0
  if argument_count < 8
    print usage comment
    abort()
  else
    for i ← 1 to argument_count
      if i + 1 is not argument_count
        if argv[i] = "--step"
```

```

        step ← argv[++i]
    else if argv[i] = "--min"
        min ← argv[++i]
    else if argv[i] = "--max"
        max ← argv[++i]
    else if argv[i] = "--fpath"
        fpath ← argv[++i]
    else
        print "Not enough arguments provided"
        abort()

    if min < max || step > (max - min) || fpath is
↪ empty
        print "Wrong input parameters"
        abort()

```

Програмен код 4.2: Псевдо код на обработката на входните данни

Приноси

Приносите към дисертационния труд могат да бъдат разделени на научни и научно практически. Те могат да бъдат обобщени както следва:

- Изследвани са съвременните технологии при невронните мрежи от тип Коен-Грозберг с двупосочна асоциативна памет с променливи във времето закъснения и променливи импулсни смущения;
- Концепцията за устойчивост по отношение на многообразиата дефинирана по определена функция h обобщава множество от понятия за устойчивост. Следователно Теорема 5 може да бъде приложена към редица конкретни ситуации в зависимост от избора на нормата $\|z\|$ и Ляпуновата функция $V(t, z)$. Предложеният резултат разширява и обобщава съществуващите резултати за устойчивост на импулсни Коен-Грозберг невронни мрежи с двупосочна асоциативна памет с променливи във времето закъснения. (Публикувано в *Mathematics*. 2020; 8(3):335. doi: 10.3390/math8030335)

- Критериите за устойчивост изведени в Теорема 5 също обобщават резултатите от други научни автори като се взема предвид променливите импулсни смущения и h -многообразия. Като се има предвид, че импулсните смущения в променливо време в импулсните невронни мрежи са по-естествени и реалистични, следователно новите резултати предлагат разширен хоризонт за приложения. (Публикувано в *Mathematics*. 2020; 8(3):335. doi: 10.3390/math8030335)
- Реализиран е приложен метод, на базата на обобщените критерии за устойчивост изведени в Теорема 5, на който с помощта на паралелен софтуер се търсят стабилни зони на настоящия модел. (Публикувано в *Proceedings of Seventh International Congress on Information and Communication Technology. Lecture Notes in Networks and Systems*, vol 465. Springer, Singapore. doi: 10.1007/978-981-19-2397-5_34)
- Направено е изследване за прилагането на представения метод за направата на софтуерен анализ, чрез използването на паралелна техника, по-специално CUDA и OpenMPI. Даденият тип невронни мрежи дават възможност за моделиране и изследване на биологични проблеми. (Публикувано в *BioInfoMed'2022, Second International Symposium on Bioinformatics and Biomedicine*, October 5-7, 2022)

Публикации

1. Stamov, G., Stamova, I., Simeonov, S., and Torlakov, I.. 2020. “On the Stability with Respect to H-Manifolds for Cohen–Grossberg-Type Bidirectional Associative Memory Neural Networks with Variable Impulsive Perturbations and Time-Varying Delays” *Mathematics* 8, no. 3: 335. doi: 10.3390/math8030335;
2. Stamov, G., Simeonov, S., Torlakov, I. (2022). “Visualization on Stability of Impulsive Cohen-Grossberg Neural Networks with Time-Varying Delays”. In: Sotirov, S.S., Pencheva, T., Kacprzyk, J., Atanassov, K.T., Sotirova, E., Staneva, G. (eds) *Contemporary Methods in Bioinformatics and Biomedicine and Their Applications. BioInfo-Med 2020. Lecture Notes in Networks and Systems*, vol 374. Springer, Cham. doi: 10.1007/978-3-030-96638-6_21;
3. Stamov, G., Simeonov, S., Torlakov, I. (2023). “Software Analysis of Bidirectional Associative Memory (BAM) Cohen-Grossberg-Type Impulsive Neural Networks with Time-Varying Delays”. In: Yang, X.S., Sherratt, S., Dey, N., Joshi, A. (eds) *Proceedings of Seventh International*

Congress on Information and Communication Technology. Lecture Notes in Networks and Systems, vol 465. Springer, Singapore. doi: 10.1007/978-981-19-2397-5_34;

4. Stamov, G., Simeonov, S., Torlakov, I., Yaneva, M. “Parallel technique on Bidirectional Associative Memory Cohen–Grossberg neural network”. BioInfoMed’2022, Second International Symposium on Bioinformatics and Biomedicine, 2022.