Studies in Computational Intelligence 521

Hiram Ponce-Espinosa
Pedro Ponce-Cruz
Arturo Molina

# Artificial Organic Networks

## Artificial Intelligence Based on Carbon Networks

Springer

# Studies in Computational Intelligence

Volume 521

*Series Editor*

J. Kacprzyk, Polish Academy of Sciences, Warsaw, Poland
e-mail: kacprzyk@ibspan.waw.pl

For further volumes:
http://www.springer.com/series/7092

## About the Series

The series “Studies in Computational Intelligence” (SCI) publishes new developments and advances in the various areas of computational intelligence – quickly and with a high quality. The intent is to cover the theory, applications, and design methods of computational intelligence, as embedded in the fields of engineering, computer science, physics and life sciences, as well as the methodologies behind them. The series contains monographs, lecture notes and edited volumes in computational intelligence spanning the areas of neural networks, connectionist systems, genetic algorithms, evolutionary computation, artificial intelligence, cellular automata, self-organizing systems, soft computing, fuzzy systems, and hybrid intelligent systems. Of particualr value to both the contributors and the readership are the short publication timeframe and the world-wide distribution, which enable both wide and rapid dissemination of research output.

Special offer: For all clients with a print standing order we offer free access to the electronic volumes of the Series published in the current year.

Indexed by DBLP, Ulrichs, SCOPUS, MathSciNet, Current Mathematical Publications, Mathematical Reviews, Zentralblatt Math: MetaPress and Springerlink.

Hiram Ponce-Espinosa
Pedro Ponce-Cruz · Arturo Molina

# Artificial Organic Networks

## Artificial Intelligence Based on Carbon Networks

Springer

Hiram Ponce-Espinosa
Pedro Ponce-Cruz
Arturo Molina
Instituto Tecnológico de Estudios
Superiores de Monterrey
Campus Ciudad de México
Tlalpan Distrito Federal
Mexico

ISSN 1860-949X ISSN 1860-9503 (electronic)
ISBN 978-3-319-02471-4 ISBN 978-3-319-02472-1 (eBook)
DOI 10.1007/978-3-319-02472-1
Springer Cham Heidelberg New York Dordrecht London

Library of Congress Control Number: 2013950734

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

*To Omar who makes me believe in the unbelievable*
*To my parents and grandfather*

—Hiram Ponce-Espinosa

*To Norma, Pedro and Jamie who are always very close to my heart*

—Pedro Ponce-Cruz

*To my lovely family: Silvia, Julio and Monserrat*

—Arturo Molina

# Preface

This book was written for undergraduate and graduate students as well as researchers and scientists interested in artificial intelligence techniques. In fact, the intention of this book is to introduce and fully describe the artificial organic networks technique, a novel machine learning method inspired on chemical carbon networks. In addition, an organic network-based algorithm named artificial hydrocarbon networks is presented to show the advantages and the scope of the technique.

On one hand, the book is complemented with several examples through chapters and the description of real-world applications using artificial hydrocarbon networks. On the other hand, the text is accompanied with an artificial organic networks toolkit implemented on LabVIEW$^{\text{TM}}$ allowing a hands-on experience to readers.

The organization of the book is as follows: Chapter 1 introduces an overview of machine learning and the modeling problem while Chap. 2 describes key concepts of organic chemistry in order to understand the technique, Chaps. 3 and 4 describe the artificial organic networks technique and the artificial hydrocarbon networks algorithm. Then, Chap. 5 offers some improvements to the basic artificial hydrocarbon networks algorithm. Finally, Chaps. 6 and 7 provide experimental results and discuss how to implement the algorithm in real-world applications like audio filtering, control systems and facial recognition.

Finally, we would like to express our gratitude to all those who provided support and reviewed details over and over, those who read and offered comments allowed us to quote their remarks, and those who assisted us in the editing, proofreading and designing stages. A special acknowledgement to the Tecnologico de Monterrey.

Mexico City, Mexico, August 2013

Hiram Ponce-Espinosa
Pedro Ponce-Cruz
Arturo Molina

# Contents

# Chapter 1
# Introduction to Modeling Problems

Computational algorithms for modeling problems are widely used in real world applications, such as: predicting behaviors in systems, describing of systems, or finding patterns on unknown and uncertain data. Interesting applications have been developed for engineering, biomedical, chemical, economics, physics, statistics, and so forth.

In that way, a large set of computational algorithms have been developed like: classical methods, probabilistic methods, parametric and nonparametric methods, probabilistic methods, heuristic and metaheuristic based methods, and naturally inspired methods. However, these modeling algorithms have several drawbacks. For instance, some algorithms assume partial knowledge of the system to model; others suppose specific dependences on variable instances; some others act as black boxes without concerning on how models work; like other algorithms are not stable due to heuristics or arbitrary parameters difficult for setting, except by trial-and-error or expertise.

It is well known that there is no algorithm that actually deal with all classes of modeling problems. In addition, some algorithms are ineffective respect to others and it depends on the domain problem. Hence, focusing on two key features of computational algorithms for modeling problems, i.e. stability in algorithms and partial knowledge of model behavior, a new paradigm of algorithms is emerging, called artificial organic networks, in order to enhance both features.

This framework is a chemically inspired technique based on carbon networks that aims to model engineering systems capturing stability in the topology of the algorithm and partial knowledge of the given system behavior. In particular, the idea behind artificial organic networks is to build a stable topology of mathematical units, so-called molecular units, that capture information of a system in small clusters. As chemical molecules do, these units can interact among them forming complex entities, i.e. artificial organic compounds, that capture nonlinearities among attribute variables in the system. At the end, artificial organic compounds can mix them up forming mixtures of nonlinear behaviors. Moreover, chemically inspired rules are applied in the interaction of molecular units and in the resultant topologies, exploiting

H. Ponce-Espinosa et al., *Artificial Organic Networks*,
Studies in Computational Intelligence 521, DOI: 10.1007/978-3-319-02472-1_1,

this "intelligent" chemical behavior to build stable and organized topologies that may help to explain systems in a better way.

Furthermore, artificial organic networks framework has been used for developing a computational algorithm for modeling problems named artificial hydrocarbon networks. In advance, that algorithm is inspired on chemical hydrocarbons (the most stable organic compounds in nature as well as the simplest compounds in terms of their structures) and it is considered as a class of supervised and parametric learning algorithms for modeling causal, continuous, multivariate systems.

Thus, in this chapter, an introduction to modeling problems and a brief review of machine learning are presented. Later on, naturally inspired computing is discussed to address chemically inspired algorithms where artificial organic networks framework fits. An outline of common methods are described as well as some aspects of computational complexity and stability that will be important in the next chapters. Then, a comparison between artificial organic networks and other learning algorithms is discussed to highlight the advantages of using artificial organic networks in modeling problems. Finally, the motivation of artificial organic networks summarizes this chapter.

## 1.1 The Modeling Problem

Consider a system $\Sigma$ that presents a response $Y$ for a stimulus $X$. Let also assume that a set of observations from $X$ and $Y$, denoted as an ordered pairs $(X, Y)$, are known. Then, it is possible to find a functional relationship $M$ between $X$ and $Y$ such that $M$ clearly describes the behavior of $\Sigma$ due to a stimulus $X$, denoted as $M_\Sigma$. Moreover, if $M_\Sigma(X)$ can estimate the observed values of $Y$ and can generalize it to other values of unknown pairs $(x_i, y_i) \notin (X, Y)$, then $M_\Sigma$ is considered a model of $\Sigma$.

In engineering, a model can be understood as a representation of a system. However, in real world applications there only exists a set of observations of the system, but it does not exist a functional relationship that helps to describe the system. In that sense, one may wish to build a model of that system. But, it is not an easy task to handle and several methods have been developed to solve the so-called modeling problem, as stated in Definition 1.1:

**Definition 1.1** (***modeling problem***) *Given an unknown system $\Sigma$ and a set of ordered pairs of observations $(X, Y)$ of the stimulus and the response of $\Sigma$. The* modeling problem *refers to find, or build, a functional relationship $M_\Sigma$ between $X$ and $Y$, such that, $M_\Sigma$ describes $\Sigma$ closely and generally. Moreover, $M_\Sigma$ is called the* model *of $\Sigma$, $X$ is known as the* set of attribute variables *and $Y$ as the* set of target variables.

As noted in Definition 1.1, build a model of a system requires to describe it closely, i.e. the error between the model response and the target has to be minimum. But also, it needs to describe it generally. In other words, the model has to describe closely

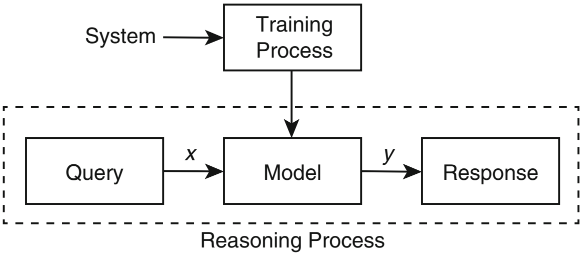


**Fig. 1.1** Block diagram of modeling systems implementing the training and reasoning processes

the target values and other values not seen in the set of data observations in order to get a predictable property to the model. As seen later in Sect. 1.2, some metrics of models in these terms are also described.

In any case, modeling systems have two steps: the training and the reasoning processes. The training process refers to the usage of datasets to build a model of a particular system. This is implemented as a learning algorithm. Once a model is trained, the reasoning process is the step in which a model is applied. Figure 1.1 shows a two step modeling system.

Learning algorithms are widely used in machine learning to develop systems that simulate human intelligence. In that sense, unknown systems or uncertain environments can be dealt using machine learning methods by building models of these systems or environments and then applying reasoning processes.

However, before running a training process, some assumptions to models are required. Actually, models can be classified as follows:

- Continuous and discrete models. A continuous model assumes that the system is continuous in its input domain, e.g. the indoor temperature in terms of time; while a discrete model assumes that the system is constrained to be discrete in its input domain, e.g. a set of different geometric figures.
- Linear and nonlinear models. A linear model considers that the relationship between attribute and target variables is linear; while a nonlinear model considers that the relationship between attribute and target variables is nonlinear, e.g. logarithmic, quadratic, etc.
- Recursive and nonrecursive models. A recursive model depends on attributes and target variables such that $M_{\Sigma}(X, Y)$; while a nonrecursive model only depends on attribute variables $M_{\Sigma}(X)$.

## 1.2 Review of Machine Learning

Machine learning refers to the study, build and analysis of systems that can learn from information [2, 18]. Computationally, these systems apply learning algorithms that mainly model a set of observations and generalize it to predict or infer new information.

Applications of machine learning are vast. For example, inference systems to predict future data, pattern recognition, audio and video signal processing, classification of data, natural language processing, search engines, data mining, financial analysis, modeling systems, artificial intelligence based approaches, medical diagnosis, recommender systems, intelligent tutoring systems, intelligent assistant systems, robotic approaches, scheduling, and so forth.

Focusing on modeling problems, literature reports different computational algorithms that allow suitable solutions for them [18]. In any direction, classical methods or machine learning techniques, might be classified in two different ways for solving modeling problems: learning algorithms and clustering algorithms. In a nutshell, given any system with suitable conditions, it can be modeled or inferred using learning algorithms; while given any set of unknown objects, it can be classified into several groups using clustering algorithms. Actually, the most important applications of these learning algorithms are inference and classification systems.

Since artificial organic networks framework falls into machine learning techniques, this section introduces the notion of these algorithms and presents a brief review of the most common methods used.

## *1.2.1 Learning Algorithms*

Interesting real world phenomena is highly complex in terms of their behavior. If delimited these phenomena to be any system (such as, physical, medical, economical, biological, etc.), it can be analyzed, predicted or described by a model. In fact, a model is a simple representation of a system or process. In suitable cases, representation of these systems as models is done by experts or by extracting information of amounts of data. However, the process is intractable if systems have complex, unpredictable behaviors or there are huge amounts of data to analyze. Thus, other techniques need to be applied. In that sense, learning algorithms intend to make this process easier. For example, Fig. 1.2 shows a typical situation in which a system (data points) has to be modeled using any learning algorithm and obtain a close function (continuous line) that best represents the system and allows generalization.

Depending on the way data observations are treated, learning algorithms can be classified as: supervised and unsupervised algorithms, parametric and nonparametric algorithms, probabilistic algorithms, or univariate and multivariate algorithms.

### 1.2.1.1 Supervised and Unsupervised Learning Algorithms

Let denote $\Sigma = (X, Y)$ to any given system $\Sigma$ with ordered pairs $(X, Y)$ of attribute and target variable observations, respectively. Supervised learning algorithms build a model $M_{\Sigma}$ of a system $\Sigma$ using past and present data or experience in the form of ordered pairs of attribute and target variables, also denoted inputs and outputs. The idea behind these algorithms is to minimize any metric that measures the distance

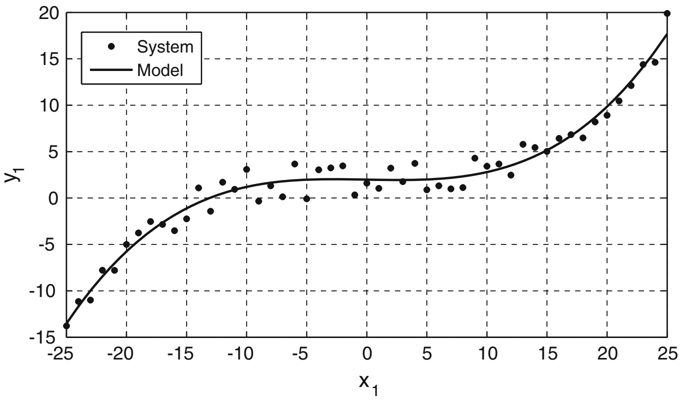


**Fig. 1.2** Example of building a model (*continuous line*) of a given system (*data points*)

between the target observations and the model. This metric is known as loss function or error function $E$, and measures how precise the model is in terms of target data. In optimization procedures, this loss function is called the objective function. Actually, the most common metric is the so-called squared loss function like (1.1); where, $q$ is the number of data observations (samples), $\mathbf{x}_k \in \mathbb{R}^n$ is the $k$-th sample of attribute variables $x_1, ..., x_n \in X$, $\mathbf{y}_k \in \mathbb{R}^m$ is the $k$-th sample of target variables $y_1, ..., y_m \in Y$ and $M_\Sigma(\mathbf{x}_k)$ is the response of the model due to $\mathbf{x}_k$.

$$E = \frac{1}{2}\sum_{k=1}^{q}(\mathbf{y}_k - M_\Sigma(\mathbf{x}_k))^2 \tag{1.1}$$

In contrast, unsupervised learning algorithms build models $M_\Sigma$ of a system $\Sigma$ only using attribute variables $X$. These learning processes recognize regularities in the input domain of the system. Different metrics are widely used, but a squared loss function $E$ can be computed as (1.2); where, $q$ is the number of input samples, $\mathbf{x}_k \in \mathbb{R}^n$ is the $k$-th sample of attribute variables $x_1, ..., x_n \in X$ and $\boldsymbol{\theta}_i$ is a set of parameters defined in the model.

$$E = \frac{1}{2}\sum_{k=1}^{q}(\mathbf{x}_k - \boldsymbol{\theta}_i)^2 \tag{1.2}$$

#### 1.2.1.2 Parametric and Nonparametric Learning Algorithms

On the one hand, parametric learning algorithms build a model $M_\Sigma$ that depends on suitable parameters $\boldsymbol{\theta}_i$, such that, $M_\Sigma(X|\boldsymbol{\theta}_i)$. In fact, parametric learning lagorithms are reduced to an optimization problem in which the set of optimal parameters $\boldsymbol{\theta}_i$

satisfies the minimization of $E$ in (1.1) or (1.2). The most known types of algorithms for continuous models fall into regression methods, and variants of it, like linear regression, quadratic regression, or polynomial regression. For example, artificial hydrocarbon networks algorithm uses the well-known least squares estimates method, a regression technique, as a supervised and parametric learning algorithm to find suitable parameters inside it.

On the other hand, nonparametric learning algorithms build a model $M_\Sigma$ that only depends on data observations from the system, such that, $M_\Sigma(X)$. Actually, nonparametric learning algorithms suppose similarities in input data. Thus, similar inputs may have similar outputs. Then, these methods are also called as smoothers. Common algorithms are the running mean smoother, the kernel smoother and the running-line smoother. Each of them makes a guess of the model $M_\Sigma(X)$ in terms of relative measurements among samples in $X$ in order to minimize (1.1).

#### 1.2.1.3 Probabilistic Learning Algorithms

Probabilistic learning algorithms claim for uncertainty in the input-output data observations $(X, Y)$. In that case, a model is built in terms of dependencies of units of information that are related with a conditional probability due to the Bayes' rule. Typical algorithms referred to Bayesian decision algorithms, such as Bayesian networks or belief networks. The maximum likelihood estimation and the bias and variance model are other learning algorithms using probabilities.

#### 1.2.1.4 Univariate and Multivariate Learning Algorithms

Univariate learning algorithms build models that only depend on an attribute variable $x \in X$ while multivariate learning algorithms build models that depend on several attribute variables $x_1, ..., x_n \in X$. In real world applications, multivariate learning algorithms are preferred more than univariate ones because complex systems depend on several input variables.

### *1.2.2 Classification Algorithms*

As said previously, unsupervised learning algorithms assume similarities on input data observations. Then, classification algorithms implement unsupervised learning algorithms for discriminating input data to be part or not of a category called a class. In the general case, classification algorithms can act as clustering algorithms in which the input data is partitioned into several classes or clusters. In order to do that, classification algorithms have a discriminant function which it decides if any input data is part of a class or not. Notice that parametric/nonparametric algorithms, probabilistic algorithms, and univariate/multivariate algorithms can be used

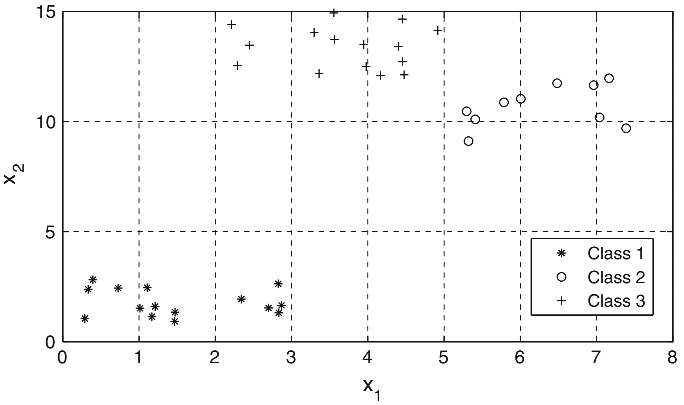


**Fig. 1.3** Example of clustering a set of data points

for classification algorithms, too. The significant change is that there is no objective function for comparison purposes.

As an application, clustering is a special case of classification algorithms. For example: mixture densities, the *k*-means clustering, fuzzy clustering means, and the expectation-maximization, are the most common algorithms for portioning the input space into several categories. Figure 1.3 shows an example of clustering given a dataset.

## 1.3 Nature-Inspired Computing

Many machine learning algorithms and other optimization procedures involved in learning and classification algorithms are inspired on nature, called naturally inspired algorithms [20, 21]. Roughly speaking, those are algorithms that use information about natural processes like water activities, social behavior, living organisms and their different levels, environmental actions, physical phenomena, and so forth. In fact, during the last decades, these algorithms have been developed and used more and more due to the advantages they bring on computing. In addition, naturally inspired, or simply nature-inspired, algorithms are related to artificial intelligence, a field in which theories of knowledge, soft-computing, mathematical models and biological or natural studies are joined together in order to give to an autonomous agent a specific behavior [20].

In computer sciences, nature-inspired algorithms are applied to achieve approximate solutions to problems like modeling, optimization, searching, scheduling, classification, clustering, etc. Actually, they employ heuristics that can handle with a large variety of problems but reaching approximate solutions, giving robustness. In some problems, it is preferable reaching an approximate solution more than an exact solution, if the algorithm gets the solution in minor time or solves different problems with minor changes.

Furthermore, two subclasses of algorithms can be found under nature-inspired algorithms. Biologically inspired, or simply bio-inspired, algorithms are those that emulate the way of reasoning, developing and solving problems using observations of biology processes. In a similar fashion, chemically inspired, or chemistry-inspired, algorithms are those that use chemical rules, interactions and organic agents to solve problems.

The most common nature-inspired algorithms are [9, 20, 21]: meta-heuristic algorithms, evolutionary algorithms, artificial neural networks, swarm intelligence, and DNA computing algorithms. Neurobiological systems, social behaviors, organic compounds, or adaptability of populations, are just a few examples in which these techniques are based on. Following, a brief description of these and other techniques is presented.

### *1.3.1 Metaheuristic Algorithms*

In optimization problems nature-inspired metaheuristic algorithms employ nature-based strategies to guide the search process, in an efficient way, to reach feasible solutions [31]. Usually, nature-inspired metaheuristics are approximate and solve non-specific problems. In addition, two search strategies are commonly used: intensification and diversification. The first refers to locally search domain spaces while the second refers to avoid local minima. Actually, metaheuristics emerge because they can get approximate solutions in a reasonable amount of time, in comparison with many optimization methods that are NP-hard (refer to Sect. 1.4).

General metaheuristics can be classified into trajectory-based algorithms and population-based algorithms [3]. Firstly, trajectory-based algorithms are those that have an initial state and reach a final state (also called the attractor), drawing a path of transient states during the search process. Simulated annealing is a good example of these kinds of nature-inspired metahauritic algorithms. Secondly, population-based algorithms deal with a set of solutions in each iteration of the algorithm that provides exploration in the search space. In contrast, single point search algorithms use one entity at a time to explore the domain space. For example, genetic algorithms and ant colony optimization are population-based nature-inspired metaheuristic algorithms.

Furthermore, some parametric learning and classification algorithms implement nature-inspired metaheuristics at the training process to find the optimal values of parameters. Some metaheuristics used are [3]: genetic algorithms, simulated annealing, ant colony optimization, particle swarm optimization, and others. In other cases, nature-inspired metaheuristic algorithms are directly applied to solve learning and classification problems like artificial neural networks do.

### 1.3.2 Evolutionary Algorithms

Nature evolves for adapting to an environment. Evolutionary algorithms are inspired on natural evolution in which the information is coded in chromosomes. In fact, the process of evolution uses two simple operations: mutation and recombination. Mutation meets the criterion of variability in a pool of basic information while recombination exchanges information among individuals. Since evolutionary algorithms are population-based metaheuristics, the above operations are done over a population or a set of individuals that evolves through epochs (or iterations). At last, natural selection causes extinction in non-adapted organisms and provides better resources to adapted organisms [18, 20].

In context, evolutionary algorithms can be classified into three families: evolutionary programming (aiming the adaptation of individuals rather than the evolution of them), evolution strategies (looking for optimization in engineering problems with some rules of adaptation in operations and individuals), and genetic algorithms (the most popular family of evolutionary algorithms which it primary uses recombination because independent solutions can derive better solutions to a suitable problem). Hybrid combinations of these algorithms can also be found. In that way, some other techniques that can be found are evolution programs, genetic programming, and mimetic algorithms.

Other techniques like gene expression and memetic algorithms are summed up to evolutionary algorithms. For instance, gene expression algorithms are bio-inspired by the central dogma of biology in which stands that DNA is the central part of the gene information and then it is translated to proteins via the mRNA and synthesis of proteins can do it well. In fact, genes are expressed in different ways depending on the transcription. Actually, in fitness functions of genetic algorithms, individuals are not properly prepared to be encoded and to be evaluated into these functions. Thus, genes in genetic algorithms can be evaluated in the fitness function via gene expression algorithms in which take into account the expression of the gene or chromosome in individuals and then process a phenotype to translate them properly. In contrast, memetic algorithms introduces cultural units, called memes, into genetic algorithms [20]. The idea is to use specific problem constrains in memes and generally run genetic algorithms. Then, at each iteration in genetic algorithms, a subset of population is evaluated using memes.

Some applications of evolutionary algorithms are focused to solve problems of [18, 20] combinatorial optimization, scheduling, classification, modeling, placement problems, searching, telecommunications, design, mechanical optimization, power planning, error coding design, and so forth.

### 1.3.3 Biologically Inspired Algorithms

Biologically inspired, or simply bio-inspired, algorithms are those that emulate the way of reasoning, developing and solving problems using observations of biology processes. Following, two common bio-inspired techniques are described.

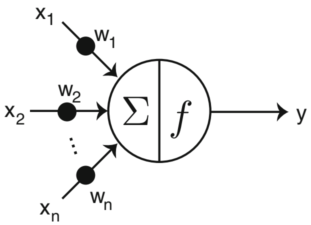


**Fig. 1.4** Simple neuron model of artificial neural networks

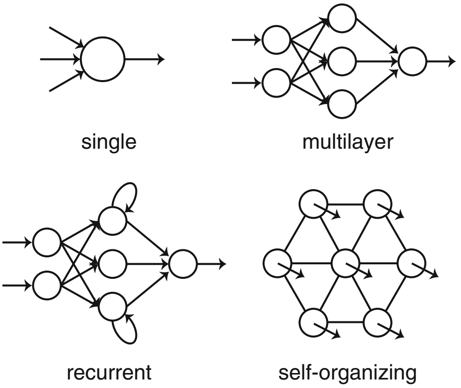


**Fig. 1.5** Different topologies of artificial neural networks

#### 1.3.3.1 Artificial Neural Networks

Artificial neural networks are biologically inspired algorithms that also fall into machine learning techniques. Roughly, they are mathematical models based on nervous system to emulate learning. They can improve their behavior applying a training process [4, 11]. Artificial neural networks define neurons as their basic entities in which they are cells that create relationships among them shaping networks. In fact, reasoning is perfomed in cells using the information coming from input messages or other output responses of neurons via an activation function, a mathematical model of a neural behavior. Figure 1.4 shows an simple neuron model of artificial neural networks.

Several types of neural networks are available. For example, the well-known supervised perceptron, singled or multilayered, was the first neural network implemented. It uses a threshold to stimulate or inhibit neurons. Another supervised neural networks is the general singled or multilayered artificial neural network, which differs from the previous one in the activation function. In contrast, examples of unsupervised neural networks are the feedforward and feedback associative memory networks like Hopfield's nets, and the competitive learning networks like self-organizing networks (e.g. Kohonen maps) [4, 11]. In addition, there are the recurrent neural networks that create linking cycles in neurons to perform dynamic behavior. Figure 1.5 shows the most common topologies in artificial neural networks.

One of the most appreciated features of artificial neural networks is the ability to model a process or learn the behavior of a system. In that sense, artificial neural networks have to be trained to meet a model or to learn behaviors. Supervised and unsupervised learning algorithms are applied in artificial neural networks. For instance, some algorithms for supervised learning [4, 20] are perceptron learning, linear autoassociators learning, iterative learning, Hopfield's model, means square error algorithms, and the Widow-Hoff rule or least mean squares algorithms. In addition, the most popular algorithm for multilayed neural networks is the back-propagation method [4]. Several unsupervised learning algorithms [4, 20] are *k*-means clustering, Kohonen clustering, self-organizing models, and adaptive resonance theory (e.g. ART1 or ART2).

Applications of artificial neural networks range from modeling, clustering, classification, pattern recognition, control, robotics, filtering, forecasting, fault detection, feature extraction, and so forth.

#### 1.3.3.2 Swarm Intelligence

Swarm intelligence is an artificial intelligence technique firstly inspired on social behavior in animals [20], i.e. insects. For instance, to insects, social behavior means the interaction of individuals in which they solve an optimization problem like finding the optimal path for food, task allocation, optimization based on communication, and so forth. Actually, swarm intelligence is one of the most useful bioinspired techniques for optimization problems. Nowadays, swarm intelligence also refers to agents that interact among them following bio-inspired metaheuristic rules, as explained below.

**Algorithm 1.1** AGENT-CYCLE: Main routine of an agent.

```
Determine the initial state of the environment
Determine the initial internal state of the agent
while goal is not reached do
    Sense the environment
    Update the internal state of the agent
    Define the next action to do using the internal state and the perception of the environment
    Execute the action over the environment
end-while
```

Formally, an agent is a computational system that is located in a given environment, and it has two important characteristics: it is autonomous in the environment and it can reach goals previously assigned [28]. For example, consider a mobile robot inside a maze; the maze is the environment and the robot is the agent that interact in the environment with one objective: solve the maze. In terms of swarm intelligence, ants are agents that find optimal paths for food between their colony and the source of food (space in the ground corresponds to the environment).

The interaction between the agent and the environment can be modeled as shown in Fig. 1.6. In particular, the agent can sense the environment and then it can do

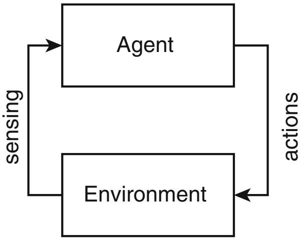


**Fig. 1.6** The agent-environment model

actions in the environment, until the goal is reached. The latter is called the agent cycle [28], as depicted in Algorithm 1.1.

Prominent algorithms in swarm intelligence are reported in literature like [20, 21, 31]: ant colony optimization, particle swarm optimization, honeybee forager allocation, intelligent water drops, artificial immune systems, and bee colony optimization. To illustrate swarm intelligence algorithms, particle swarm optimization and ant colony optimization will be present following.

Particle swarm optimization (PSO) is inspired on social behavior of bird flocking [16]. Because it is a population-based algorithm, a population of particles (possible solutions) is randomly initialized. In addition, each particle keeps track its own best position $\mathbf{p}_{best} = (p_1, ..., p_n)$ in the search space $\mathbb{R}^n$ and the best global position $\mathbf{g}_{global} = (g_1, .., g_n)$ of all particles is also stored. Iteratively, the position of each particle is updated based on rule (1.3); where, $x_i$ represents the position of a particle in the $i$-th dimension of the search space and $v_i$ is the velocity of a particle in the $i$-th dimension and computed as (1.4). The momentum parameter $\omega$ is used to follow the tendency of velocity in a particle, $\phi_p$ and $\phi_g$ are known as learning factors and commonly set up to a value of 2, and $r_p$ and $r_g$ are random values normally distributed over the interval [0, 1].

$$x_i = x_i + v_i \tag{1.3}$$

$$v_i = \omega v_i + \phi_p r_p \left(p_i - x_i\right) + \phi_g r_g \left(g_i - x_i\right) \tag{1.4}$$

If the position $\mathbf{x} = (x_1, ..., x_n)$ is better than the last one, it is stored as the best position of particle. After all current position are calculated, the global position is updated if one of the best position of all particles is better than the last global position stored so far. Eventually, the algorithm will reach an approximate solution $\mathbf{p}_{global}$ of the global optimum. In order to measure how good is a particle, a fitness function $f(\mathbf{p})$ must be design for a specific problem. Figure 1.7 shows the flow diagram of the simple PSO algorithm.

On the other hand, ant colony optimization (ACO) is inspired on ants that find optimal paths between their colony and a source of food [6]. Typically, it is implemented to optimize trajectories. Again, this algorithm is a population-based metaheuristic technique that initializes a population of ants. Each ant has a pheromone level assigned. Then, each ant can move in the search space iteratively sparsing its pheromone, leaving a path of pheromone trail. Then, an ant moves from position $x$

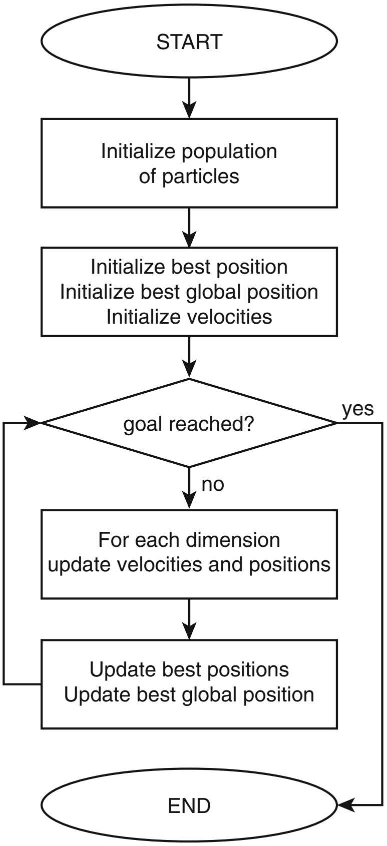

START
Initialize population of particles
Initialize best position
Initialize best global position
Initialize velocities
goal reached?
yes
no
For each dimension update velocities and positions
Update best positions
Update best global position
END

**Fig. 1.7** Flow diagram of the simple particle swarm optimization algorithm

to position $y$ based on the probability of movement $p_{xy}$ calculated by (1.5); where, $\eta_{xy}$ is the attractiveness level of pheromone between positions $x$ and $y$, $\tau_{xy}$ is the trail level indicating how well this path from $x$ and $y$ is so far, $\alpha \geq 0$ is the influence of pheromone and $\beta \geq 1$ is the influence of attractiveness. Commonly, $\eta_{xy} = \frac{1}{d_{xy}}$ where $d_{xy}$ is the distance between $x$ and $y$.

$$p_{xy} = \frac{\tau_{xy}^{\alpha} \cdot \eta_{xy}^{\beta}}{\sum_{y} \tau_{xy}^{\alpha} \cdot \eta_{xy}^{\beta}} \tag{1.5}$$

In each iteration, the trail level of each $k$ ant is updated following rule (1.6); where, $\sigma$ is the pheromone evaporation coefficient, $C$ is a gain value of pheromone and $L_k$ is the length of path that the $k$ ant has travelled.

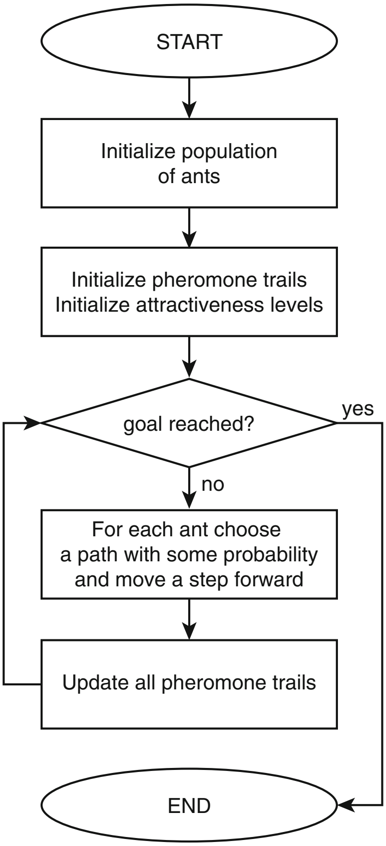

START
Initialize population of ants
Initialize pheromone trails
Initialize attractiveness levels
goal reached?
yes
no
For each ant choose a path with some probability and move a step forward
Update all pheromone trails
END

**Fig. 1.8** Flow diagram of the simple ant colony optimization algorithm

$$\tau_{xy} = (1-\sigma)\tau_{xy} + \sum_k \Delta\tau_{xy}^k$$
$$\Delta\tau_{xy}^k = \begin{cases} C/L_k & \mathit{if\ k\ use\ path\ xy} \\ 0 & \mathit{otherwise} \end{cases} \tag{1.6}$$

At last, the path that has the highest pheromone trail level is the optimal trajectory. Figure 1.8 shows the flow diagram of the simple ACO algorithm.

## *1.3.4 Chemically Inspired Algorithms*

Chemically inspired, or chemistry-inspired, algorithms are those that use chemical rules, interactions and organic agents to solve problems. Following, some chemically inspired techniques are described.

#### 1.3.4.1 DNA Computing

This technique is a (bio-) chemically inspired (i.e. organic chemistry, biochemistry and molecular biology) on the deoxyribonucleic acid (DNA) [1, 9]. The idea is to represent information as DNA strands or chains and use chemical reactions, as operations, over DNA strands. In that way, parallel computation is done using several strands and applying the same operation at once. In fact, DNA is a string of bases, i.e. adenine, guanine, cytosine, and thymine; and then data can be coded in these bases.

In that case, bit strings (or strands) of bases are used to codify information, via ligation reaction to produce an initial random set of strands. Secondly, these strands are amplified by polymerase chain reaction in order to find all strands that start and end in specific positions of the search space, and then they run on an agarose gel to purify them to get only specialized strands. Affinity of these strands is also implemented with a biotin-avidin magnetic beads system. Finally, another amplification via polymerase chain reaction is done. Several methods are presented in literature as the sticker based model.

Solving NP-hard optimization problems (see Sect. 1.4), implementations in molecular computing or parallel computing approaches are examples of applications of DNA computing.

#### 1.3.4.2 Organic Computing

Since 2004, the German Research Foundation (DFG, by its German acronym) proposes organic computing [30]. This is an emerging paradigm of developing large collections of agent systems inspired on living organisms aiming the following characteristics: self-organizing, self-configuring, self-optimizing, self-healing, self-protecting, self-explaining, and context aware. Thus, organic computing systems dynamically adapt from the current state of the environment. It is primary focuses on autonomous systems. Several artificial intelligence techniques and control system methods are used on it for achieving the final objective.

#### 1.3.4.3 Chemical Reaction Optimization

This is an emerging algorithm proposed in 2010 by Lam and Li [15]. In fact, chemical reaction optimization (CRO) algorithm loosely mimics what happens to molecules microscopically when they are subjected to chemical reactions, following the energy minimization rule in the potential energy surface (PES). Figure 1.9 summarizes the simple CRO algorithm.

As noted in Fig. 1.9, there are three stages: initialization, iteration and finalization. The first step considers the initialization of parameters and control variables that model the minimization problem and determine the size of the set of molecules in the chemical reaction. The second step loops until a stop criterion is reached. In this stage, molecules are subjected to chemical reactions: inter-molecular ineffective collision,

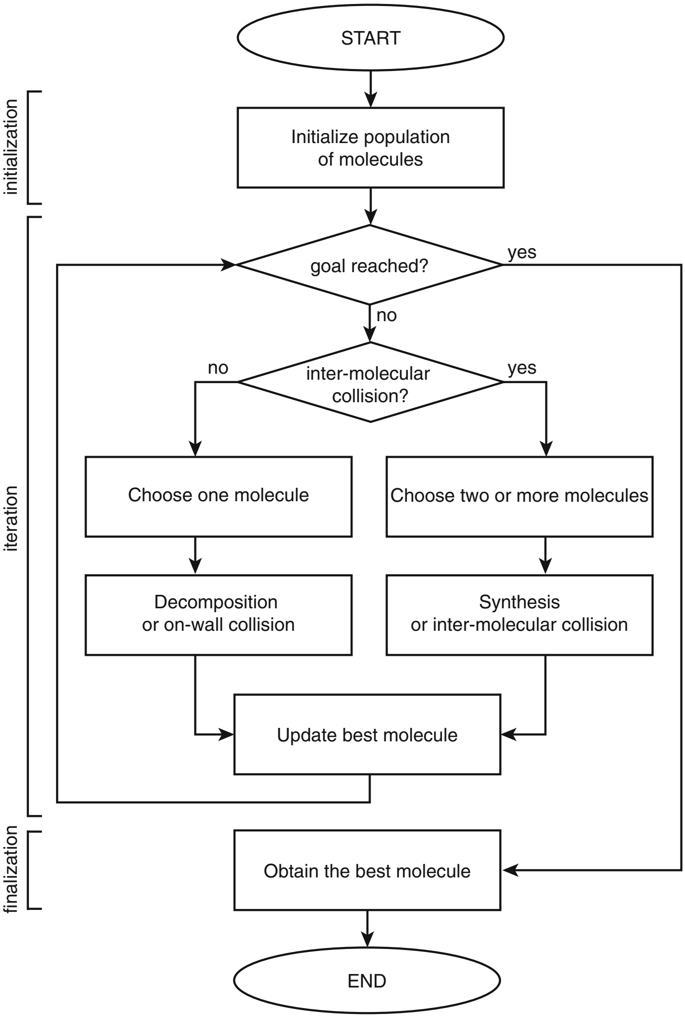

START
initialization
Initialize population of molecules
iteration
goal reached?
yes
no
inter-molecular collision?
no
yes
Choose one molecule
Choose two or more molecules
Decomposition or on-wall collision
Synthesis or inter-molecular collision
Update best molecule
finalization
Obtain the best molecule
END

**Fig. 1.9** Flow diagram of the simple chemical reaction optimization algorithm

synthesis, decomposition, or on-wall ineffective collision. Those cause that each molecule reacts changing its inner structure. The algorithm stops in hte finalization step when a stop criterion is reached. In order to determine the minimization of energy in the PES, the objective function of the problem is evaluated with each molecular structure, storing the best molecular structure so far.

## 1.4 Comparison of Algorithms for Modeling Problems

In general, computational algorithms for modeling problems are considered nondeterministic polynomial time (*NP*) problems which means that the solution might not be calculated with deterministic algorithms in polynomial time (with respect to the inputs), but it is easy to check the solution in polynomial time. For instance, consider the modeling system in Fig. 1.1. The training process is hard to solve computationally because finding the proper topology and parameters of models (i.e. building and optimizing) requires too much effort in terms of time and non-deterministic algorithms have to be used. However, once the model is obtained, the reasoning process is easy to compute because the response to a query is totally deterministic. Thus, different techniques have been studied and proposed in order to reduce the time in the learning phase, for example: randomization, conditional probabilities, heuristics and metaheuristics, naturally inspired algorithms, or parallel computing.

For instance, randomization refers to use random events to probably obtain good solutions in minor time, while conditional probabilities are implement in techniques that have to compute the joint probability of events. Heuristics and metaheuristics, as described in Sect. 1.3.1, guide the algorithm to find an approximate solution rapidly. In the same way, naturally inspired algorithms assume that nature implement optimal rules that might be used to guide algorithms to a feasible solution. In contrast, parallel computing accelerates the learning process by dividing an algorithm into small tasks that run in multiple threads and cores, simultaneously.

In a nutshell, computational algorithms for modeling problems have the following weaknesses in terms of the computational time:

- Most of the algorithms are in the class of nondeterministic polynomial time complexity.
- Fast and efficient algorithms reach at least an approximate solution.
- Some algorithms depend on (meta) heuristics or randomization.
- Some algorithms are considered as black boxes (they cannot offer an interpretation of the system behavior).
- Fast and efficient algorithms are designed for specific problems.

In some cases, it is useful to know which modeling algorithms are better than others. In that sense, learning algorithms can be compared in terms of: computational time complexity, stability (generalization), characteristics of the learning algorithm, characteristics of the models built, types of problems that solve, and so forth.

### *1.4.1 Complexity and Stability in Modeling Problems*

Two important metrics in learning algorithms are the complexity and the stability. In this section, both of them are briefly introduced.

**Table 1.1** Computational time complexities of algorithms

| Running time | Representation in $O$-notation |
|---|---|
| Constant | $O(1)$ |
| Logarithmic | $O(\log n)$ |
| Linear | $O(n)$ |
| $k$-order Polynomial | $O(n^k)$ |
| Exponential | $O(2^n)$ |

#### 1.4.1.1 Computational Time Complexity

Complexity is a metric of how well or how efficient is an algorithm, and it can be used for comparing computational algorithms that solve the same class of problems. Formally, the computational time complexity, denoted by $O(t(n))$, measures the time taken by an algorithm to run in terms of the size of the instance $n$; where $t(n)$ is a given function [7]. Table 1.1 shows the most important time complexities of algorithms, in ascending order, and their representations. Notice that very large algorithms that run in exponential time will not converge [7].

On the other hand, three important time complexity notations have been developed: the upper bound, the lower bound and the time average complexities. In that way, the $O$-notation measures the running time of algorithms in the worst case, also known as the upper bound computational time complexity. The $\Omega$-notation represents the lower bound complexity and assumes that an algorithm runs proportionally at least as $\Omega(t(n))$ for input size $n$. Finally, the $\Theta$-notation represents the time average complexity of an algorithm, if for some values of $n$, the lower and upper bounds of the time complexity of the algorithm are equivalent [19].

In computational complexity theory, problems can be classified depending on the running time that their algorithms take to solve them with respect to an input size $n$. For instance, in complexity classes, $P$ and $NP$ are two fundamental classes [7, 19]. All problems that can be solved at most in polynomial time with deterministic algorithms are $P$ problems. In constrast, all problems that can be solved with non-deterministic algorithms and their solutions can be checked in polynomial time are $NP$ problems. In fact, $P \subset NP$ (it remains open the question if $P = NP$). As noted, $P$ problems are preferred more than $NP$ problems, when $n$ tends to be large.

The hardest problems in the $NP$ class are considered $NP$-complete problems. Furthermore, problems that can be solved with non-deterministic algorithm but their solutions cannot be checked in polynomial time are $NP$-hard problems. Roughly speaking, $NP$-hard problems are at least as hard as $NP$-complete problems. Figure 1.10 shows a complexity class diagram of $P$, $NP$, $NP$-complete and $NP$-hard classes. Moreover, it is known, by Ladner's theorem [23], that there exist some problems in $NP$ that are neither as easy as $P$ problems nor as hard as $NP$-complete problems, if and only if $P \neq NP$. Figure 1.10 also shows these types of problems.

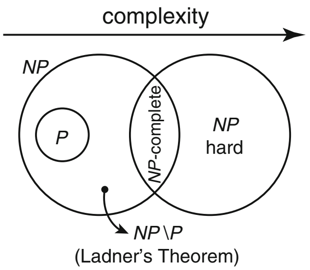


**Fig. 1.10** Diagram of complexity classes

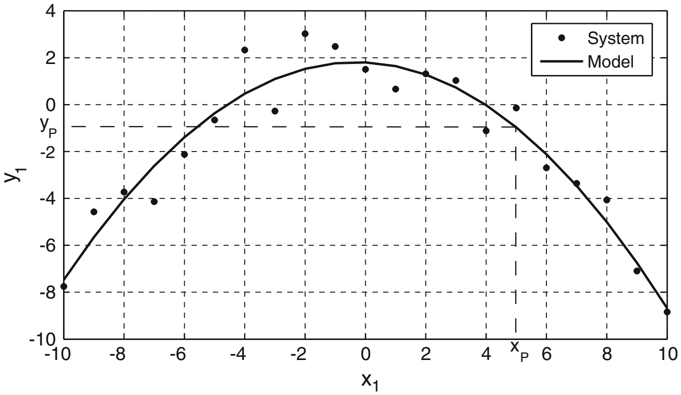


**Fig. 1.11** Example of generalization in learning algorithms

#### 1.4.1.2 Stability and Generalization

Stability is the property of algorithms that claims whenever possible that for small changes in input data, algorithms produce small changes in the final results [2, 18]. If this property is not satisfied, then an algorithm is said to be unstable. Furthermore, if an algorithm is stable for certain choices of input data, then it is called conditionally stable [5].

In learning algorithms, stability is important because this property is closely related to generalization [2, 18]. The latter is considered the ability of an algorithm to return an accurate output value for a new, unseen data point. For example, consider the training set (data points) and the model response (continuous line) depicted in Fig. 1.11. As seen, the model responses from the continuous input domain giving acceptable points (output range) closely to the training set; then, the model is considered stable. In contrast, Fig. 1.12 shows the same training set but with another model response (continuous line). Clearly, the model does not respond closely to the training set; thus, the model is considered unstable.

In order to perform generalization in learning algorithms, two loss functions have to be minimized: the error due to bias and the error due to variance [8]. The first

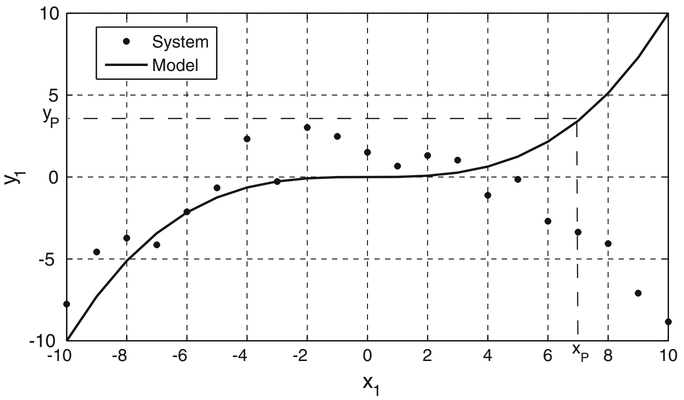


**Fig. 1.12** Example of an unstable model in learning algorithms

minimizes (1.1) allowing the model to fit the target data. The second minimizes the variability of a model response for a given target data, if this model have different realizations, as (1.7); where, $E$ is the error due to variance, $q$ is the size of observations, $\mathbf{x}_k \in \mathbb{R}^n$ is the $k$-th observation of attribute variable $\mathbf{x} = (x_1, ..., x_n)$, $M_\Sigma$ is the model and $\overline{M_\Sigma}$ is the average prediction of model $M_\Sigma$ in a given point $\mathbf{x}_k$.

$$E = \frac{1}{2}\sum_{k=1}^{q}\left(M_\Sigma(\mathbf{x}_k) - \overline{M_\Sigma}(\mathbf{x}_k)\right)^2 \tag{1.7}$$

This tradeoff is important because models might be over-fitting or under-fitting [8]. Roughly, an over-fitted model predicts training data points very precise, but new data points are not well generalized. In comparison, an under-fitted model predicts training and new data points loosely. At last, both behaviors are not prefer.

### *1.4.2 Artificial Organic Networks and Modeling Problems*

In this section, the most common algorithms in machine learning are compared in terms of computational time complexity [10, 12–14, 17, 22, 24–27, 29, 32], characteristics of learning algorithms, characteristics of built models and types of problems solved. In addition, artificial organic networks technique is also contrasted with these learning algorithms.

For instance, Table 1.2 summarizes the time complexity of learning algorithms as well as characteristics of those like: supervised or unsupervised, parametric or nonparametric, deterministic or nondeterministic, probabilistic, and univariate or multivariate algorithms. In addition, Table 1.2 also presents the direct type of problems that algorithms can solve (approximation or prediction, classification, optimization),

**Table 1.2** Characteristics of common learning algorithms

| Method/Algorithm | Time Complexity | Supervised | Unsupervised | Parametric | Nonparametric | Deterministic | Nondeterministic | Probabilistic | Univariate | Multivariate | Approximation | Classification | Optimization |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *General* | | | | | | | | | | | | | |
| Linear regression (LR) | $O(c^2n)$ | X | | X | | X | | | X | X | X | | |
| General regression (GR) | $O(c^2n)$ | X | | X | | X | | | X | X | X | | |
| Running mean smoother (RMS) | $O(n)$ | X | | | X | X | | | X | X | X | | |
| Kernel smoother (KS) | $O(nd)$ | X | | | X | X | | | X | X | X | | |
| Decision trees (DT) | $O(nc^2)$ | X | X | | X | X | X | X | X | X | | X | |
| Random forest (RF) | $O(Mcn \log n)$ | X | X | | X | X | X | X | X | X | | X | |
| Naive Bayes classifier (NBC) | $O(nc)$ | X | | X | | | X | X | X | X | | X | |
| Bayesian networks (BN) | $O(cd^e)$ | X | X | X | | | X | X | X | X | | X | |
| Gaussian mixture models (GMM) | $O(Sikn)$ | X | X | X | | | X | X | X | X | | X | |
| Support vector machine (SVM) | $O(n^3)$ | X | | X | | | X | | X | X | X | X | |
| $k$-nearest neighbor (kNN) | $O(knd)$ | X | X | | X | X | | | X | X | X | X | |
| $k$-means algorithm (kM) | $O(n^{dk+1} \log n)$ | | X | X | | | X | | X | X | | X | |
| Fuzzy clustering means (FCM) | $O(indk)$ | | X | X | | | X | | X | X | | X | |
| Expectation-maximization (EM) | $O(Sikn)$ | | X | X | | | X | X | X | X | | | X |
| Simulated annealing (SA) | OP | * | * | * | * | | X | X | X | X | | | X |
| Tabu search (TS) | OP | * | * | * | * | | X | | X | X | | | X |
| *Evolutionary* | | | | | | | | | | | | | |
| Genetic algorithms (GA) | NA | * | * | * | * | | X | | X | X | | | X |
| Gene expression algorithms (GE) | NA | * | * | * | * | | X | | X | X | | | X |

(continued)

**Table 1.2** (continued)

| Method/Algorithm | Time Complexity | Supervised | Unsupervised | Parametric | Nonparametric | Deterministic | Nondeterministic | Probabilistic | Univariate | Multivariate | Approximation | Classification | Optimization |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Memetic algorithms (MA) | NA | * | * | * | * | | X | | X | X | | | X |
| *Artificial neural networks* | | | | | | | | | | | | | |
| Backpropagation (BP) | TD | X | | X | | | X | | X | X | X | X | |
| Generalized Hebbian algorithm (HA) | TD | | X | X | | | X | | X | X | X | X | |
| Hopfield's nets (HN) | TD | | X | X | | | X | | X | X | | X | |
| Kohonen maps (SOM) | TD | | X | X | | | X | | X | X | | X | |
| *Swarm intelligence* | | | | | | | | | | | | | |
| Particle swarm optimization (PSO) | NA | * | * | * | * | | X | | X | X | | | X |
| Ant colony optimization (ACO) | NA | * | * | * | * | | X | X | X | X | | | X |
| Bees colony optimization (BCO) | NA | * | * | * | * | | X | X | X | X | | | X |
| Intelligent water drops (IWD) | NA | * | * | * | * | | X | | X | X | | | X |
| Cuckoo search (CS) | NA | * | * | * | * | | X | | X | X | | | X |
| Bacterial foraging optimization (BFO) | NA | * | * | * | * | | X | | X | X | | | X |
| *Chemically inspired* | | | | | | | | | | | | | |
| DNA computing (DNA) | NA | * | * | * | * | | X | | X | X | | | X |
| Chemical reaction optimization (CRO) | NA | * | * | * | * | | X | | X | X | | | X |
| Artificial hydrocarbon networks (AHN) | $O(Cmn \ln \frac{1}{\varepsilon})$ | X | | X | | | X | | X | X | X | X | |

*X*—marks stand for a characteristic found in the method/algorithm. *—marks refer that a method/algorithm does not present that characteristic. *OP*—marks refer to a method/algorithm which its time complexity varies depending on the specific-problem and/or optimization algorithm. *TD*—marks stand for a method that its time complexity is topology-dependant. *NA*—marks refer to a method/algorithm that does not build a model directly

**Table 1.3** Symbols used in Table 1.2

| Symbol | Description |
|---|---|
| $n$ | Number of samples |
| $c$ | Number of attribute variables |
| $d$ | Dimensionality of inputs |
| $k$ | Number of clusters |
| $i$ | Number of iterations |
| $e$ | Maximum number of parents in Bayesian networks |
| $M$ | Number of trees in random forests |
| $S$ | Number of random repetitions |
| $C$ | Number of compounds in AHNs |
| $m$ | Number of molecules in AHNs |
| $\varepsilon$ | Tolerance greater than zero in AHNs |

i.e. the main purpose of algorithms; however, it does not mean that algorithms might be used in other problems. Furthermore, it is remarkable to say that the chart reports the time complexity of classical or simple algorithms unless otherwise noted. Table 1.3 shows the symbols used in Table 1.2.

Focusing on artificial organic networks, i.e. the artificial hydrocarbon network algorithm (AHN), it is close related to backpropagation-based multilayer neural networks (BP) and support vector machines (SVM) in terms of supervised learning, non-probabilistic models used for approximation/classification problems. It is important to note because it gives a general view of where artificial hydrocarbon networks is located, as revealed in Fig. 1.13. Notice that artificial hydrocarbon networks is also located between regression algorithms, like linear regression (LR) and general regression (GR), and clustering algorithms like $k$-nearest neighbor (kNN), $k$-means algorithm (kM) and fuzzy clustering means (FCM). Smoothers are not too far away from regression algorithms and AHNs. Roughly speaking, the above discussion means that artificial hydrocarbon networks builds approximation and classification models as well as like-smoothers (i.e. filtering systems).

Unfortunately, in terms of time complexity, artificial hydrocarbon networks cannot be easily compared with support vector machines or backpropagation-based multilayer neural networks. However, if the number of compounds and molecules (units of AHNs) are fixed, artificial hydrocarbon networks would be less complex than support vector machines, when the number of training samples is large. The backpropagation algorithm depends on the topology of the artificial neural network, thus a comparison of AHNs and BP cannot be computed in terms of computational time complexity (see Table 1.2). At most, backpropagation algorithm is based on gradient descent methods that actually has time complexity $O(\ln \frac{1}{\varepsilon})$ where $\varepsilon > 0$ is a small tolerance value that is used as stop criterion.

Finally, it is important to highlight that some time complexities summarized in Table 1.2 are specialized. For example, the time complexity of linear/general regression is based on the least squares estimates algorithm, the time complexity of

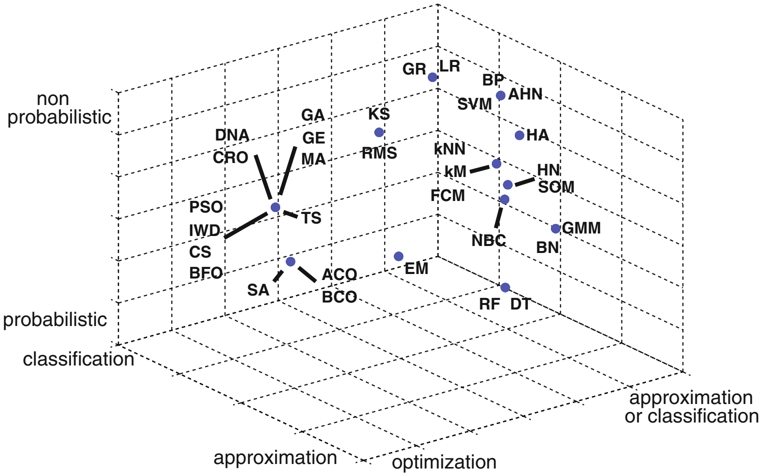


**Fig. 1.13** Clustering map of learning algorithms/methods of Table 1.2. Labels are acronyms of algorithms/methods

decision trees is computed with the standard $C4.5$ training algorithm, time complexity of Bayesian networks is computed in polytrees, time complexity of kernel smoothers is calculated with easiest known nearest neighbor smoother. In addition, time complexity of expectation-maximization (EM) is computed for random EM algorithms, as well as in Gaussian mixture models.

On the other hand, Table 1.4 reports characteristics of models when they are built from some learning algorithms. In this chart, the simple artificial hydrocarbon networks algorithm build continuous, nonlinear and static models representing white or gray boxes of the system. Notice that nonrecursive models are trained, thus dynamic models cannot be handled. However, artificial organic networks framework does not limit training models, allowing recursive models for dynamic systems.

## 1.5 Motivation of Artificial Organic Networks

In a nutshell, most of computational algorithms for modeling problems, also known are learning algorithms, are $NP$-complete which means that, if $P \neq NP$, they have to be improved. For instance, naturally inspired algorithms have been designed for that purpose, using heuristics and metaheuristics to face $NP$-complete problems. In that sense, there are important key features in learning algorithms that have to be improve: computational time complexity, stability or generalization in algorithms, and easiness of model interpretation to understand systems.

The above features motivate artificial organic networks as a learning algorithm that is chemically inspired on carbon networks. It assumes that the easiness of topology,

**Table 1.4** Characteristics of models built by common learning algorithms

| Method/Algorithm | Continuous | Discrete | Linear | Nonlinear | Static | Dynamic | Recursive | Nonrecursive | White/Gray Box | Black Box |
|---|---|---|---|---|---|---|---|---|---|---|
| *General* | | | | | | | | | | |
| Linear regression (LR) | X | X | X | | X | X | | X | | X |
| General regression (GR) | X | X | X | X | X | X | | X | | X |
| Running mean smoother (RMS) | X | | X | X | X | | | X | | X |
| Kernel smoother (KS) | X | | X | X | X | | | X | | X |
| Decision trees (DT) | X | X | X | X | X | | | X | X | |
| Random forest (RF) | X | X | X | X | X | | | X | | X |
| Naive Bayes classifier (NBC) | | X | X | X | X | | | X | X | |
| Bayesian networks (BN) | X | X | X | X | X | X | X | X | X | |
| Gaussian mixture models (GMM) | X | X | X | X | X | | | X | | X |
| Support vector machine (SVM) | X | X | X | X | X | X | X | X | | X |
| *k*-nearest neighbor (kNN) | X | X | X | X | X | | | X | | X |
| *k*-means algorithm (kM) | X | X | X | X | X | | | X | | X |
| Fuzzy clustering means (FCM) | X | X | X | X | X | | | X | | X |
| Expectation-maximization (EM) | X | X | X | X | X | | | X | X | |

(continued)

**Table 1.4** (continued)

| Method/Algorithm | Continuous | Discrete | Linear | Nonlinear | Static | Dynamic | Recursive | Nonrecursive | White/Gray Box | Black Box |
|---|---|---|---|---|---|---|---|---|---|---|
| Simulated annealing (SA) | | | | | | NA | | | | |
| Tabu search (TS) | | | | | | NA | | | | |
| *Evolutionary* | | | | | | | | | | |
| Genetic algorithms (GA) | | | | | | NA | | | | |
| Gene expression algorithms (GE) | | | | | | NA | | | | |
| Memetic algorithms (MA) | | | | | | NA | | | | |
| *Artificial neural networks* | | | | | | | | | | |
| Backpropagation (BP) | X | X | X | X | X | X | X | X | | X |
| Generalized Hebbian algorithm (HA) | | X | X | X | X | X | X | X | | X |
| Hopfield's nets (HN) | X | X | X | X | X | X | X | | | X |
| Kohonen maps (SOM) | X | X | X | X | X | | | X | | X |
| *Swarm intelligence* | | | | | | | | | | |
| Particle swarm optimization (PSO) | | | | | | NA | | | | |
| Ant colony optimization (ACO) | | | | | | NA | | | | |

(continued)

**Table 1.4** (continued)

| Method/Algorithm | Continuous | Discrete | Linear | Nonlinear | Static | Dynamic | Recursive | Nonrecursive | White/Gray Box | Black Box |
|---|---|---|---|---|---|---|---|---|---|---|
| Bees colony optimization (BCO) | | | | | | NA | | | | |
| Intelligent water drops (IWD) | | | | | | NA | | | | |
| Simulated annealing (SA) | | | | | | NA | | | | |
| Tabu search (TS) | | | | | | NA | | | | |
| Cuckoo search (CS) | | | | | | NA | | | | |
| Bacterial foraging optimization (BFO) | | | | | | NA | | | | |
| *Chemically inspired* | | | | | | | | | | |
| DNA computing (DNA) | | | | | | NA | | | | |
| Chemical reaction optimization (CRO) | | | | | | NA | | | | |
| Artificial hydrocarbon networks (AHN) | X | | X | X | X | | | X | X | |

*X*—marks stand for a characteristic found in the model. *NA*—marks refer that a method/algorithm does not train a model directly i.e. the multilevel arrangement of molecular units as well as mixtures of organic compounds, captures the behavior of a given system; but also, that topology brings other properties like stabilization, encapsulation, inheritance and robustness. At last, the topology can also be built on-line using chemical rules.

Thus, in the rest of the book, artificial organic networks technique and artificial hydrocarbon networks algorithm are introduced and fully described. In particular, the book is organized as follows:

Chapter 2 introduces basic knowledge of organic chemistry in order to understand the inspiration of artificial organic networks. The chapter begins with an overall description of organic chemistry and its importance in real-world. Then, basic concepts like atoms, molecules, mixtures and interactions in chemical compounds are described. In particular, energy minimization in organic compound structures is discussed because it will be very important in the design phase of artificial organic networks. At last, an overview of classification of organic compounds and characteristics that artificial organic networks can mimic are finally described.

Chapter 3 fully describes artificial organic networks technique from its metaphor of carbon networks to definitions of concepts treated. Actually, artificial organic networks is also defined from three different views: the levels of energy in the topology, the mathematical definition and the framework of the technique. Finally, implementation issues of training procedures and on-line building models are discussed.

Chapter 4 introduces and describes artificial hydrocarbon networks algorithm, a chemically inspired learning method based on artificial organic networks and chemical hydrocarbon compounds. The algorithm is described deeply. For instance, two theorems about time complexity and stability of the algorithm are stated. At last, the chapter presents a discussion about how to interpret parameters and obtained topologies of artificial hydrocarbon networks in order to understand the behavior of a system.

On the other hand, Chap. 5 describes three enhancements of the artificial hydrocarbon networks algorithm. First, the chapter describes how to optimize the number of molecules in artificial hydrocarbon compounds using the enthalpy rule of molecules. Second, it extends the algorithm to the $n$-dimensional case in which several attribute variables can be taken into account in artificial hydrocarbon networks. Finally, artificial aromatic compounds algorithm is introduced, as the recursive version of artificial hydrocarbon networks., which it can be used for dynamic systems.

Chapter 6 presents experimental results of artificial hydrocarbon networks. It summarizes examples of approximation problems, inference applications, clustering problems and classification applications. The last section discusses some guidelines to implement artificial hydrocarbon networks in real-world applications.

Finally, Chap. 7 encloses real applications of artificial hydrocarbon networks. For example, it describes two different digital signal processing systems in audio and image, the design of fuzzy-molecular control systems in direct current motors, the design of a hybrid algorithm using genetic algorithms, and finally, the development of multi-agent based systems using artificial hydrocarbon networks.

# References

1. Adleman L (1994) Molecular computation of solutions to combinatorial problems. Science 266:1021–1024
2. Alpaydin E (2004) Introduction to Machine Learning. MIT Press, United States of America
3. Blum C, Roli A (2003) Metaheuristics in combinatorial optimization: overview and conceptual comparison. ACM Comput Surv 35(3):268–308
4. Boden M (1996) Artificial Intelligence. Academic Press, United States of America
5. Burden R, Faires J (2005) Numerical Analysis. Cengage Learning, United States of America
6. Dorigo M, Gambardella LM (1997) Ant colony systems: a cooperative learning approach to the traveling salesman problem. IEEE Trans Evol Comput 1(1):53–66
7. Evans J, Minieka E (1992) Optimization algorithms for networks and graphs. Marcel Dekker, United States of America
8. Hastie T, Tibshirani R, Friedman JH (2009) The elements of statistical learning: data mining, inference, and prediction. Springer, New York
9. Hromkovic J (2001) Algorithms for hard problems: introduction to combinatorial optimization, randomization, approximation, and heuristics. Springer-Verlag, Germany
10. Hung MC, Yang DL (2001) An efficient fuzzy C-means clustering algorithm. In: Proceedings of IEEE international conference on data mining. California, San Jose, pp 225–232
11. Irwin G, Warwick K, Hunt K (1995) Neural network applications in control. The Institution of Electrical Engineers, England
12. Jana PK, Sinha BP (1997) Fast parallel algorithms for forecasting. ELSEVIER Comput Math Appl 34(9):39–49
13. Kolahdouzan MR, Shahabi C (2004) Continuous K nearest neighbor queries in spatial network databases. In: Proceedings of the 2nd workshop on spatio-temporal database management. Toronto, Canada
14. Kramer KA, Hall LO, Goldgof DB (2009) Fast support vector machines for continuous data. IEEE Trans Syst Man Cybern 39(4):989–1001
15. Lam AYS, Li VOK (2010) Chemical-reaction-inspired metaheuristic for optimization. IEEE Trans Evol Comput 14(3):381–399
16. Lazinica A (ed) (2009) Particle swarm optimization. InTech
17. Memisevic R (2003) Unsupervised kernel regression for nonlinear dimensionality reduction. Ph.D. thesis, Universitat Bielefeld
18. Mitchell T (1997) Machine Learning. McGraw Hill, United States of America
19. Moret B, Saphiro H (1991) Algorithms from P to NP. The Benjamin/Cummings Publishing Company, United States of America
20. Olariu S, Zomaya A (2006) Handbook of bioinspired algorithms and applications. CRC Press, United States of America
21. Pazos A, Sierra A, Buceta W (2009) Advancing artificial intelligence through biological process applications. Medical Information Science Reference, United States of America
22. Robnik-Sikonja M (2004) Improving random forests. In: Boulicaut JF (ed) Proceedings of ECML machine learning. Springer, Berlin
23. Rudich S, Wigderson A (eds) (2004) Computational complexity theory. American Mathematical Society, Providence
24. Schoukens J, Rolain Y, Gustafsson F, Pintelon R (1998) Fast calculation of least-squares estimates for system identification. In: Proceedings of the 37th IEEE conference on decision and control, vol 3. Tampa, Florida, pp 3408–3410
25. Sreenivasarao V, Vidyavathi S (2010) Comparative analysis of fuzzy C-mean and modified fuzzy possibilistic C-mean algorithms in data mining. Int J Comput Sci Technol 1(1):104–106
26. Su J, Zhang H (2006) A fast decision tree learning algorithm. In: Proceedings of the 21st national conference on artificial intelligence, vol 1, pp 500–505
27. Vens C, Costa F (2011) Random forest based feature induction. In: Proceedings of IEEE 11th international conference on data mining. Vancouver, pp 744–753

28. Wooldridge M (2002) An introduction to multiagent systems. John Wiley and Sons, England
29. Wu D, Butz C (2005) On the complexity of probabilistic inference in singly connected bayesian networks. In: Proceedings of the 10th international conference on rough sets, fuzzy sets, data mining, and granular computing, vol Part I. Springer, Berlin, pp 581–590
30. Wurtz RP (ed) (2008) Organic computing. Springer, Berlin
31. Yang XS (2010) Nature-inspired metaheuristics algorithms. Luniver Press,University of Cambridge, United Kingdom
32. Zhao Q, Hautamaki V, Karkkainen I, Franti P (2012) Random swap EM algorithm for gaussian mixture models. ELSEVIER Pattern Recognit Lett 33:2120–2126

# Chapter 2
# Chemical Organic Compounds

For centuries, human beings have found inspiration in nature, from macro-scale to micro-scale. Animals have been inspiring designs of cars, robotics, and even computational algorithms based on their behaviors. Some new super tough materials got inspired in deer antlers. Environmental analysis of pressure, temperature or humidity have been inspiring new ways of greenhouses. Shapes of nature have been inspiring on painting, digital art, or sculpture. Chemical products like waterproofing sprays were inspired on specific nanostructures of lotus leaves. Also, burdock seeds inspired the well-known hook-and-loop fastener.

In particular, scientists and researchers take advantage of natural inspiration because nature has shown that it can adapt itself to better response of changes and can reach feasible solutions to problems like better configurations of structure of matter, or animal survival in ecosystems. In fact, nature tends to optimality in all different ways. For instance, consider atom-structures that tend to minimize energy in bonds, but also preserve particular characteristics depending on atom relationships. To this end, the notion of the latter will be highly important through this book because the study of chemical organic compounds inspires artificial organic networks.

For instance, consider a given physical system with some input and output signals. Then, the supervised artificial organic networks technique can model the system using signal information to build a structure made of atoms that are clustered in molecules. In fact, these molecules will be used to enclose related information found in the signals of the system. Moreover, if molecules cannot approximate completely the behavior of the system, they can be joined together forming complex molecules (referred also as compounds). At last, compounds can also mix them up forming mixtures of molecules that represent linear combinations of behaviors of subsystems, giving an organized structure approximating the overall system. Actually, this structure will be referred as an artificial organic network because it is inspired on carbon based networks, especially studied on organic chemistry, that present highly stable molecules due to the electronegativity of carbon atoms. Interesting, an artificial organic network presents characteristics like structural organization, clustering

H. Ponce-Espinosa et al., *Artificial Organic Networks*,
Studies in Computational Intelligence 521, DOI: 10.1007/978-3-319-02472-1_2,

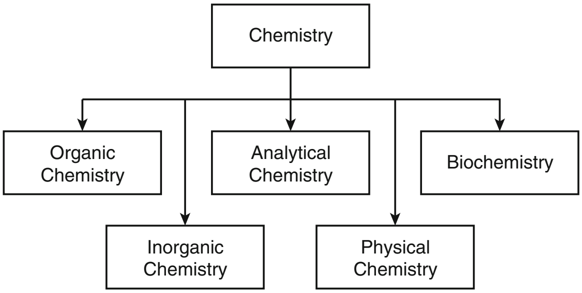


**Fig. 2.1** Branches of chemistry

information, inheritance of behavior, encapsulation of data, and stability in structure and response.

Thus, this chapter introduces fundamentals on organic chemistry to deeply understand artificial organic networks. In particular, basic concepts of organic compounds are described from the point of view of structural stability and energy minimization. In addition, classification of organic compounds is outlined. Finally, hydrocarbons, the most stable organic compounds in nature, are described.

## 2.1 The Importance of Organic Chemistry

Chemistry studies the matter, its properties and the laws that govern it. As known, chemistry is divided into five branches: organic chemistry, inorganic chemistry, analytical chemistry, physical chemistry, and biochemistry, as shown in Fig. 2.1. For instance, organic chemistry is the study of the compounds of carbon while inorganic chemistry is the study of all other compounds [10]. Analytical chemistry is the study of methods that determine and identify elements in compounds. Physical chemistry applies physics to chemistry and it is the study of termodynamics and kinetics of chemical reactions, and biochemistry is the study of chemical processes inside living organisms.

Centering on organic chemistry, it is very important in different ways. Looking around, organic products are present daily like cleaning accesories as soaps, shampoos or perfumes; they are also present in food as proteins, lipids and carbohydrates that give and store energy, that are included in structural formation, that transport other molecules, or that regulate growth in living organisms. In fact, all compounds responsible of life are organic substances denominated biomolecules, e.g. proteins, lipids, carbohydrates; but also nucleic acids, complex molecules involved in genetics, are considered part of them [1, 3, 4].

It is interesting how compounds made of carbon atoms can define millions and millions of different organic compounds. Actually, carbon atoms can interact among them to form chains and rings, giving the opportunity of changing chemical properties of compounds [4]. This is possible because their bonds with other atoms are very strong in comparison with other atomic interactions, as explained later in this chapter.

Moreover, organic chemistry is not only present in living organisms, it is also involved in human health technologies like the development of new drugs or the study of materials, e.g. hypoalergenic materials, used in protesis or cookware coating; production of food; development of cancer treatments as the so-called "sharpshooter" drugs. Also, it is widely used for developing new materials applied in space. Other industrial applications of organic compounds are: solvents like degreasers or dry cleaning products, chemical gums for wood like latex or white glue, plastics like toys or plastic coatings, etc.

Then, organic chemistry is generating new tendencies in technology and applied sciences. For example, consider the development of alternative energy sources as biofuels that can be produced from plants or organic waste; or consider organic photo-sensors that convert light in electrical signals to capture images in cameras. Furthermore, in computer sciences, organic chemistry is inspiring DNA-computing to solve hard problems with strands of DNA, or the organic computing paradigm that inherits organic properties to multiagent systems.

In addition, current trends of organic chemistry are related to green chemistry that encourages the usage minimization of hazardous substances to the environment; applications of fullerenes (molecules completely made of carbon atoms) in strength of materials, biology, treatments for industrial wastewater in the field of nanotechnology, architecture inspiration, conductivity materials, antiviral production, etc.; other trends like solid state materials, organic nanostructures, liquid crystals, organic dyes, organometallic applications to polymeric production, agrochemistry, and so forth.

On the other hand, it is well known that organic chemistry influences the global economy with hydrocarbons, specially in the production of petroleum, and the industry of polymers [3]. Thus, organic chemistry is highly important in the daily life, from the study of living organisms to medical applications, through the economical and technological impact.

## 2.2 Basic Concepts of Organic Compounds

Organic chemistry is the study of the compounds of carbon [3, 10]. Interestingly, most of these compounds consist of carbon and a few of other elements like hydrogen, oxygen, nitrogen, phosphorus, sulfur, and halogens (fluorine, chlorine, bromine, iodine, astatine). In fact, organic compounds have important physical and chemical properties derived from carbon bonding to itself and the specialization of the few other elements, like the strength of structures or the effecting of heat.

In the following sections, physical properties are associated to the structure of organic compounds while chemical properties to the behavior of them. Notice that this review is organized in order to highlight the role of energy stabilization in organic compounds.

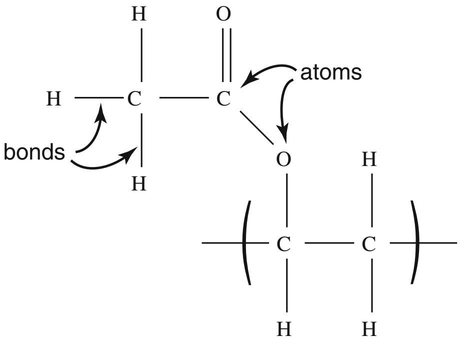


**Fig. 2.2** Example of the molecular structure of polyvinyl acetate, an organic compound used in white glue

## *2.2.1 Structural Definitions*

Organic compounds are constituted with basic units called atoms. When atoms interact among them, they form molecules and compounds, as shown in Fig. 2.2. Thus, the structure of organic compounds considers the set of atoms and the ways they are bonded. But also, energy minimization and geometric configuration play an important role in the structure [5]. Following, basic components and interactions of chemical organic compounds are summarized.

### 2.2.1.1 Atoms

The basic unit of organic compounds is the atom [4, 5]. Each atom is a set of charged particles including protons, electrons and the neutral particles so-called neutrons. Roughly, any atom has protons and neutrons joined together at its nucleus and the latter is surrounded by a set of electrons. The sum of protons and neutrons is equals to the mass number of an atom. In addition, the number of protons in the nucleus is referred as the atomic number. Hence, different atoms have different atomic numbers. The well-known periodic table summarizes this and other concepts about atoms. Thus, different atoms have distinct physical characteristics. In the sense of this book, the distinction of an atom is limited to the atomic number, and different atoms refer to different behaviors due to their physical characteristics.

For instance, the model of atoms assures that they have a particular way to distribute their electrons known as electron configuration [3, 4]. It assumes that electrons can only orbit in specific spaces called energy shells. Additionally, shells are divided into subshells (i.e. $s$, $p$, $d$ and $f$) in which electrons are grouped in orbitals. Finally, each orbital can be occupied by at most two electrons. However, electrons might represent an infinite set of configurations. In that sense, studies in chemistry and quantum mechanics reveal that there is an electron configuration in which electrons preserve the minimum energy of the atomic system, so-called the ground-state (see Sect. 2.2.2.3).

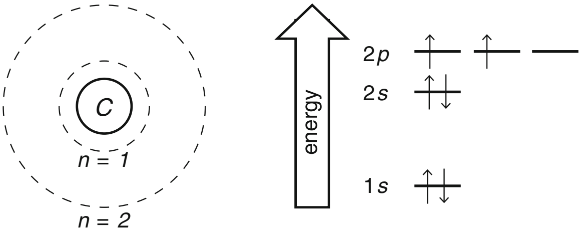


**Fig. 2.3** Simple model of the carbon atom and its energy-level diagram

Atoms are also characterized by valence electrons. These kinds of electrons are the ones which conforms the last energy shell defined in the atom. Particularly, this energy shell is called outer shell. Since electrons need some energy to be in shells, the ground-state assures that energy minimization occurs when electrons follow the quantum chemistry principles (see Sect. 2.2.2.3). Otherwise, atoms are in an excited state.

Basically, noble gases (e.g. helium, neon, argon, krypton, xenon, radon) are the only atoms with minimum energy per shell. Then, atoms must satisfy a chemical rule that states atoms in ground-state which do not have an entire outer shell (there are some valence electrons but not all of them) need more electrons in order to complete it. This is the basis of atomic interactions in organic chemistry: ionic interactions and covalent bonds, treated later in this chapter. Figure 2.3 shows a simple model and the energy-level diagram of carbon atom.

#### 2.2.1.2 Molecules

A set of two or more atoms acting as a single unit is called a molecule. As mentioned previously, atoms need to complete the outer shell with electrons; thus, creation of molecules is one mechanism to do that. Three important types of molecules are defined following:

**Elements.** These molecules are made of like atoms, e.g. hydrogen element, because atoms are not found isolated in nature. Elements can be classified into three categories:

*Metals.* Physically, they are solid at room temperature (except mercury), they can also be shiny, malleable and ductile, and they serve as conductors of electricity. Chemically, they react losing electrons.

*Nonmetals.* Physically, they are the opposite of metals. Chemically, they react gaining electrons. These elements are the only ones present in chemical organic compounds.

*Metalloids.* They have a hybrid behavior combining metal and nonmetal properties.

**Compounds.** They are molecules made of other molecules when reacting two or more of them. In the general case, a compound is made up elements proportionally to its mass. For instance, organic compounds are made of carbon atoms. If there are no carbon atoms, then compounds are considered as inorganic.

**Functional groups.** These molecules are made of a carbon atom bonded with one of the other atoms allowed in organic compounds. Functional groups are important because organic compounds are classified based on them; chemical reactions act on them in order to form other stable compounds, simple or complex; and, they are basis for naming organic compounds.

#### 2.2.1.3 Chemical Bonding

In molecules, chemical bonding refers to the relationship among atoms. In fact, this process is based on charged particle attraction. In chemistry, two main chemical bondings are present: ionic interactions and covalent bonds.

**Ionic interactions.** A metal element tends to lose electrons while a nonmetal element tends to gain electrons. Thus, ionic interaction is present into a metal-and-nonmetal reaction. Particularly to organic compounds, these chemical bondings are not present.

**Covalent bonds.** When two nonmetal atoms are interacting, they cannot lose or gain electrons. Thus, they share electrons of their outer shells. Due to the set of atoms in organic compounds, covalent bonds will be considered through this book.

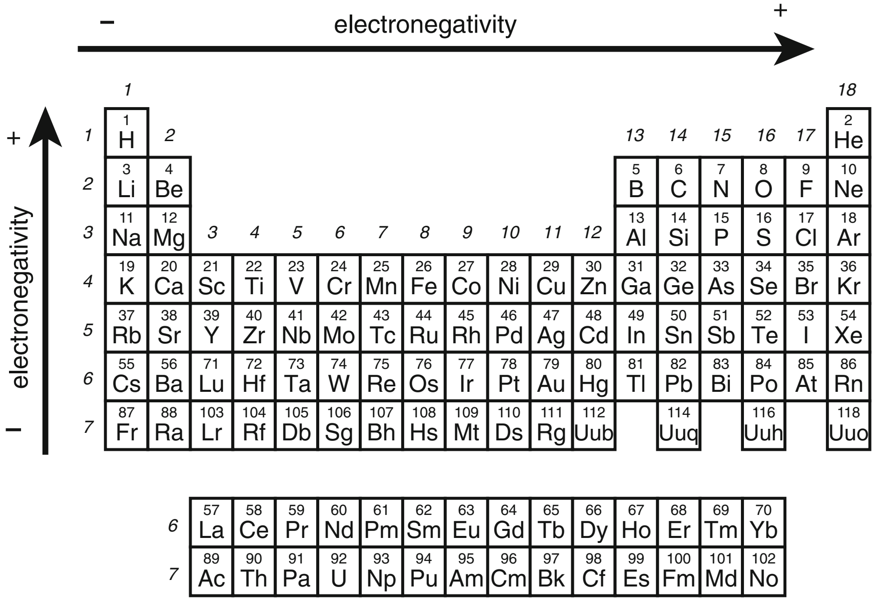
In both cases, chemical bonding is explained by electronegativity which it is presented in Sect. 2.3.

### *2.2.2 Chemical Definitions*

The study of interactions of atoms and molecules are based on chemical rules and properties. Depending on them, organic compounds have specific characteristics and behaviors. In the following section, chemical principles are depicted in order to understand the behavior of molecules and chemical bonding [5].

#### 2.2.2.1 Electronegativity

In chemistry, the notion of tendency of atoms to attract electrons is a chemical property associated to atoms named electronegativity [4, 5]. For instance, consider two atoms closely located. If they interact making a chemical bond, then both atoms have high electronegativity. Otherwise, atoms will not relate at all. Interestingly,



| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1<br>H | | | | | | | | | | | | | | | | | 2<br>He |
| 2 | 3<br>Li | 4<br>Be | | | | | | | | | | | 5<br>B | 6<br>C | 7<br>N | 8<br>O | 9<br>F | 10<br>Ne |
| 3 | 11<br>Na | 12<br>Mg | | | | | | | | | | | 13<br>Al | 14<br>Si | 15<br>P | 16<br>S | 17<br>Cl | 18<br>Ar |
| 4 | 19<br>K | 20<br>Ca | 21<br>Sc | 22<br>Ti | 23<br>V | 24<br>Cr | 25<br>Mn | 26<br>Fe | 27<br>Co | 28<br>Ni | 29<br>Cu | 30<br>Zn | 31<br>Ga | 32<br>Ge | 33<br>As | 34<br>Se | 35<br>Br | 36<br>Kr |
| 5 | 37<br>Rb | 38<br>Sr | 39<br>Y | 40<br>Zr | 41<br>Nb | 42<br>Mo | 43<br>Tc | 44<br>Ru | 45<br>Rh | 46<br>Pd | 47<br>Ag | 48<br>Cd | 49<br>In | 50<br>Sn | 51<br>Sb | 52<br>Te | 53<br>I | 54<br>Xe |
| 6 | 55<br>Cs | 56<br>Ba | 71<br>Lu | 72<br>Hf | 73<br>Ta | 74<br>W | 75<br>Re | 76<br>Os | 77<br>Ir | 78<br>Pt | 79<br>Au | 80<br>Hg | 81<br>Tl | 82<br>Pb | 83<br>Bi | 84<br>Po | 85<br>At | 86<br>Rn |
| 7 | 87<br>Fr | 88<br>Ra | 103<br>Lr | 104<br>Rf | 105<br>Db | 106<br>Sg | 107<br>Bh | 108<br>Hs | 109<br>Mt | 110<br>Ds | 111<br>Rg | 112<br>Uub | | 114<br>Uuq | | 116<br>Uuh | | 118<br>Uuo |

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 6 | 57<br>La | 58<br>Ce | 59<br>Pr | 60<br>Nd | 61<br>Pm | 62<br>Sm | 63<br>Eu | 64<br>Gd | 65<br>Tb | 66<br>Dy | 67<br>Ho | 68<br>Er | 69<br>Tm | 70<br>Yb |
| 7 | 89<br>Ac | 90<br>Th | 91<br>Pa | 92<br>U | 93<br>Np | 94<br>Pu | 95<br>Am | 96<br>Cm | 97<br>Bk | 98<br>Cf | 99<br>Es | 100<br>Fm | 101<br>Md | 102<br>No |

**Fig. 2.4** Tendency of electronegativity in the periodic table

**Table 2.1** Electronegativity values of atoms related to chemical organic compounds

| Element | Electronegativity (Pauling scale) |
|---|---|
| *H* | 2.1 |
| *C* | 2.5 |
| *O* | 3.5 |
| *N* | 3.0 |
| *P* | 2.1 |
| *S* | 2.5 |
| *F* | 4.0 |
| *Cl* | 3.0 |
| *Br* | 2.8 |
| *I* | 2.6 |
| *At* | 2.2 |

elements in the periodic table are arranged so that electronegativity increases from left to right and from top to bottom, except for the hydrogen atom that has similar electronegativity as atoms related to organic compounds. Figure 2.4 shows tendency of electronegativity in the periodic table. In addition, electronegativity values of the set of atoms related to organic compounds [2, 3] are also shown in Table 2.1.

Electronegativity is measured using the Pauling scale. It represents the absence of electronegativity with a zero-value and higher values represent higher electronegativity. For example from Table 2.1, hydrogen and astatine elements have smaller electronegativity than fluorine which has the highest electronegativity even in the whole periodic table.

Moreover, electronegativity is useful when classifying chemical bondings. If the difference in electronegativity is less than 0.5 in the Pauling scale, then the chemical bonding is considered as covalent bond. If the difference in electronegativity is between 0.5 and 1.7, then it is a polar covalent bond (see Sect. 2.3). Finally, if the difference in electronegativity is greater than 1.7, then it is an ionic interaction.

#### 2.2.2.2 Quantum Chemistry Principles

As mentioned previously, the ground-state principle assures energy minimization in chemical structures [1, 8]. However, three important quantum chemistry principles need to be stated before introducing ground-state formally. These quantum chemistry principles refer to theoretical observations that regulate chemical interactions among atoms.

**Aufbau Principle.** This principle states that electrons must fill lowest energy shells first. As an exception, electrons in *d* orbital have more stability when this orbital is half filled or full filled. Thus, one electron from the above *s* orbital is placed on the *d* orbital.

**Pauli exclusion.** This principle states that a pair of electrons in an orbital have to have opposite spins.

**Hund's rule.** It states that filling orbitals with electrons, these must not be in pairs on orbitals until each orbital contains one electron except when filling *s* orbitals.

Figure 2.5 shows the diagram for filling electrons in order to reach a ground-state in atoms [1, 8]. The arrows mean the path for picking up orbitals in order to filling them. Notice that the above three quantum chemistry rules must be satisfied.

*Example 2.1* Determine the ground-state electron configuration of: (a) the carbon atom, and (b) the copper atom.

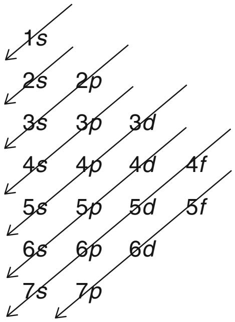


**Fig. 2.5** Diagram model for writing the ground-state electron configuration of atoms

**Solution 2.1** Using Fig. 2.5 and the atomic number of each atom, the ground-state electron configuration is:

(a) Carbon atom has atomic number six, thus six electrons are needed: $C : 1s^2 2s^2 2p^2$.
(b) Copper atom has atomic number 29, thus 29 electrons are needed: $Cu : 1s^2 2s^2 2p^6 3s^2 3p^6 4s^1 3d^{10}$.

#### 2.2.2.3 Ground-State Principle

Structures of chemical organic compounds preserve stability while minimizing energy. In practice, electrons have infinite number of configurations for each atom, and for each electron configuration it has an specific energy level. The electron configuration that has the minimum energy level is named ground-state [8]. Other electron configuration different from the ground-state is called excited-state. In organic chemistry, the ground-state electron configuration is considered for all chemical structures because it aims to study and understand physical and chemical properties.

On the other hand, ground-state has theoretical applications in energy minimization and chemical bonding. For instance, noble gases are the only elements with the minimum energies for each shell. It means that noble gases have filled full with electrons in the outer shell. Thus, atoms tend to fill the outer shell with all possible valence electrons. In organic chemsitry, this rule is called the octet rule. In general, not all atoms claim for the rule, but in organic chemistry it is perfectly possible.

#### 2.2.2.4 Lewis Dot Structure

A visual interpretation of ground-state, chemical bonding and electronegativity concepts all together, is the so-called Lewis dot structure. This diagram considers the chemical symbol of a given atom and some dots around it representing the valence electrons of that atom. Thus, sharing, gaining or losing electrons are seen explicitly. Figure 2.6 shows the Lewis dot structure of a carbon atom and the Lewis dot structure of the interaction between carbon and hydrogen atoms.

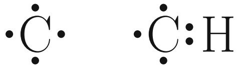


**Fig. 2.6** Examples of Lewis dot structures. (*left*) Diagram of carbon atom. (*right*) Diagram of interaction between carbon and hydrogen atoms

#### 2.2.2.5 Chemical Balance Interaction

After atoms find ground-state electron configuration and molecules are formed via the octet-rule, another chemical interaction can be applied to minimize energy in structures. This is referring to the chemical balance theory [5, 8, 10]. Notice that it does not use atomic interactions.

In fact, chemical balance interaction determines different ways to mix two or more substances, e.g. molecules or compounds, by reacting and forming products, i.e. new compounds. In the simplest form, compounds might be determined as a linear mixture of substances in definite ratios. To this end, these ratios are named stoichiometric coefficients. In fact, if optimal stoichimetric coeficcients were found, then chemical balance interaction can be viewed as an energy stabilization process. In chemistry, these structures are called mixtures.

## 2.3 Covalent Bonding

In Sect. 2.2.1.3, covalent bonds were defined. Since, they are important to organic chemistry, this section introduces the classification and an accepted chemical model for covalent bonds.

Rewriting, covalent bonds appear when two nonmetal atoms are sharing electrons of their outer shells because they cannot lose or gain electrons. In general, chemical interactions of two atoms with difference in electronegativity less than 1.7 in Pauling scale, are considered covalent bonds. An usual classification of covalent bonds are the following:

**Polar covalent bonds.** It appears when two different atoms are interacting, e.g. in $C - H$ interaction.

**Nonpolar covalent bonds.** It appears when two similar atoms are interacting, e.g. in $C - C$ interaction.

Below, there is a description of an accepted model of covalent bonds. In particular, it introduces the empirical model and two important parameters which are: the length and the minimum energy of bond.

### *2.3.1 Characterization of Covalent Bonds*

In organic chemistry, the model of chemical covalent bonds refers to a mathematical expression that characterizes the chemical covalent bond between two atoms [5, 8]. In fact, this model relates the energy of bond and the interatomic distance. For instance, the energy of bond is the value of energy necessary to break the atomic bond; while the interatomic distance is the length between the nuclei of the two atoms participating in the chemical interaction. Additionally, the notion of covalent bonds between atoms can be extended to covalent bonds between molecules, relating the energy of bond and the intermolecular distance which it refers to the interatomic distance of the two atoms located on distinct molecules participating in the interaction.

Without loss of generalization, interatomic distances are considered following [8]. On one hand, if the interatomic distance is too large, there is no chemical bond between atoms; thus, the energy of bond is equal to zero. On the other hand, let $E_{attraction}$ be the energy related to the attraction force between electrons of one atom and the nucleus of the other which it increases rapidly when the distance is short. However, if atoms are too close, a repulsive force appears due to the electrons of both atoms and the energy associated to this force, let say $E_{repulsion}$, increases negatively. Then, the sum of both energies $E_{attraction}$ and $E_{repulsion}$ gets the total energy of bond $E_{bond}$ as (2.1):

$$E_{bond} = -E_{attraction} + E_{repulsion} \tag{2.1}$$

In fact, the empirical model of chemical covalent bond (2.1) [8] can be rewritten in terms of the interatomic distance as expressed in (2.2):

$$E_{bond} = -\frac{A}{r^m} + \frac{B}{r^n}\ ,\ (m < n) \tag{2.2}$$

Where, $A$ and $B$ are constants related to the charge of atoms, $r$ is the interatomic distance, and $m, n$ are two positive empirical coefficients defining the attractive and repulsive forces, typically $n \approx 12$ and $m \geq 6$. Figure 2.7 shows the relationship between $E_{attraction}$ and $E_{repulsion}$.

Actually, two interesting parameters of covalent bonds are the length of bond and the minimum energy, and one parameter for specific nonpolar covalent bonds is called the order of bond:

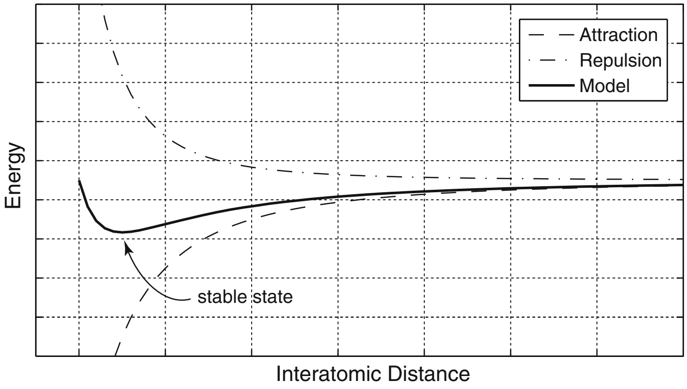


**Fig. 2.7** Empirical model of chemical covalent bonds

**Length of bond.** It is the measurement of interatomic or intermolecular distance where the minimum energy of bond is reached.

**Minimum energy of bond.** It is the minimum energy required to break the bond, and it also represents the most stable state of chemical covalent bond.

**Order of bond.** (Nonpolar covalent bonds) It is the number of pairs of electrons shared in the covalent bond. While the order of bond increases, the covalent bond is physically stronger and more stable. Orders are:

**Simple bonds.** One pair of electrons are shared.
**Double bonds.** Two pairs of electrons are shared.
**Triple bonds.** Triple pairs of electrons are shared.

## 2.4 Energy in Organic Compounds

Energy is an important concept concerning into physics, chemistry, thermodynamics and generally in engineering. It can be defined as the ability of a system to do work. Referring to organic compounds, chemical energy is the responsible of structural transformation, combustion, electricity due to nuclear fission, batteries, etc.

Typically, energy is classified into kinetic and potential energies. Roughly speaking, kinetic energy is the one used when any mass is in movement while potential energy is referred to as stored energy. Considering the latter, potential energy determines stability in chemical structures. If potential energy is low, structures are stable. If potential energy increases, chemical structures are unstable. Since potential energy could transform it to kinetic energy, the higher the potential energy is, the greater motion of chemical structures is, assuming rearranging in geometric structures of atoms, molecules and compounds. Thus, energy minimization refers to chemical structural stability.

In that sense, organic compounds have some strategies to minimize energy in their chemical structures, summarized in the following energy level scheme. Finally, three measurements of energy in thermodynamics are introduced.

### *2.4.1 Energy Level Scheme*

In this section, it is proposed a model of energy minimization in chemical structures named the energy level scheme. In fact, this is a synthesis of concepts of quantum chemical structures, molecular formation and mixture formation.

Stability in organic compounds comes from the fact that every structural unit claims for energy minimization Then, three levels of energy are considered. The first level of energy minimization refers to the ground-state electron configuration. Before any atom interact among others, atoms tend to minimize energy following

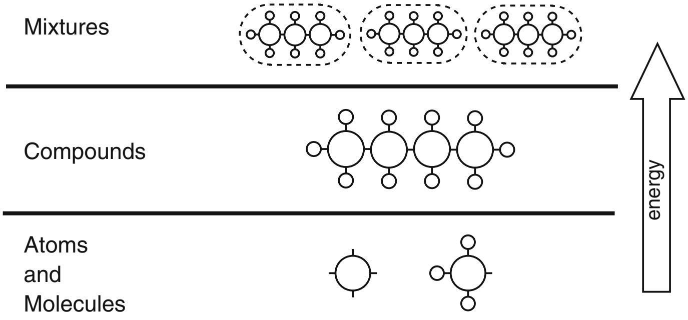


**Fig. 2.8** Representation of the energy level scheme

quantum chemical structures (see Sect. 2.2.2.2). In addition, since atoms aim to hold the octet-rule, they interact among them via chemical bonds forming molecules.

The next level of energy minimization considers chemical reactions on molecules to create compounds. These compounds are made of different molecules but interaction of them needs minimizing energy to conform bonds. In particular, these interactions release energy in structures by breaking and making bonds. Finally, the last level of energy minimization is presented in chemical balance interaction while creating mixtures of molecules or compounds.

As notice, atoms are more stable than molecules, molecules are more stable than compounds, and the latter are more stable than mixtures. Figure 2.8 represents visually this energy level scheme. It concludes that stability is preserved easily in the bottom level and structures are more unstable at the top level.

### *2.4.2 Measures of Energy*

In thermodynamics—a branch of physics which describes the relationship between heat and work, changes in temperature, and transformation of energy—, transference and conversion of energy in chemical reactions are studied [1, 4]. Actually, three important thermodynamic properties are considered to measure energy in chemical reactions: enthalpy, Gibbs free energy, and entropy. In practice these properties cannot be measured directly to the system, thus the difference of them are useful, as mentioned later.

Consider a chemical reaction from reactants ($R$) to products ($P$) as depicted in Fig. 2.9. At the beginning, reactants have some energy and at a given time a chemical reaction is produced and the products have less energy than reactants. Visually, this is known as reaction coordinate diagram.

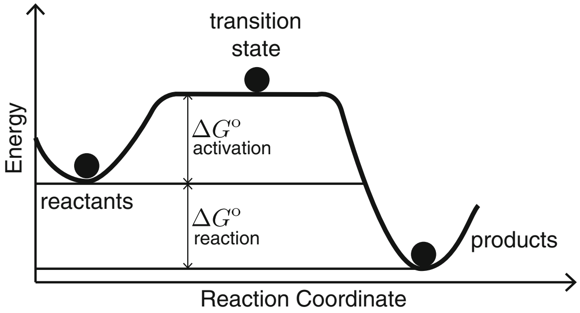


**Fig. 2.9** Example of a reaction coordinate diagram

#### 2.4.2.1 Enthalpy

Enthalpy $H$ is a thermodynamic property of chemical bonds [6]. In practice, the energy involved in these bonds is directly related to the chemical energy at constant pressure. Thus enthalpy can be measured using the heat in reaction. When computing the enthalpy in chemical reactions, Hess' law [6] states that enthalpy of a reaction is independent of the pathway between the initial and final states, then the heat of reaction [1] $\Delta H^{\circ}$ is used instead of $H$, defined as (2.3); where, $H_i^{\circ}$ is the initial enthalpy (e.g. energy of reactants) and $H_f^{\circ}$ is the final enthalpy (e.g. energy of products). In laboratory, the heat $Q$ measured with a calorimeter is equal to $\Delta H$ such that (2.4) holds.

$$\Delta H^{\circ} = H_f^{\circ} - H_i^{\circ} \tag{2.3}$$

$$\Delta H^{\circ} = Q \tag{2.4}$$

In organic chemistry, several enthalpies of chemical bonds and molecules are already obtained. Table 2.2 summarizes enthalpies of chemical bonds related to important atomic interations in organic compounds [4, 6, 8, 9].

Finally, heat of reaction determine if the reaction releases heat known as exothermic ($\Delta H^{\circ} < 0$) or absorbs heat known as endothermic ($\Delta H^{\circ} > 0$).

**Table 2.2** List of enthalpies of chemical bonds in organic compounds

| Chemical bond | Enthalpy (kJ/mol) |
|---|---|
| $C - C$ | 350 |
| $C - H$ | 415 |
| $C = C$ | 611 |
| $C \equiv C$ | 837 |

[1] The superscript (°) means that the property is measured at stardard states of 298 K in temperature and 1atm in pressure.

#### 2.4.2.2 Gibbs Free Energy

The Gibbs free energy $G$ is another thermodynamic property of chemical reactions that measures the velocity (rate) of reaction and its equilibrium [7]. At standard states, the change of Gibbs free energy $\Delta G^\circ$ is related to the equilibrium constant $K_{eq}$ as (2.5); where, $R$ is the gas constant ($8.314\,\text{J/mol} \cdot \text{K}$), and $T$ is the temperature. If $\Delta G^\circ < 0$ refers to an spontaneous reaction. Otherwise, reactions need external energy to produce it.

$$\Delta G^\circ = -RT \ln(K_{eq}) \tag{2.5}$$

Interestingly, enthalpy, entropy and Gibbs free energy are expressed together in the Gibbs-Helmholtz equation as (2.6); where, $T$ is temperature, $\Delta G^\circ$ is the change in Gibbs free energy, $\Delta H^\circ$ is heat of reaction, and $\Delta S^\circ$ is the change in entropy.

$$\Delta G^\circ = \Delta H^\circ - T\Delta S^\circ \tag{2.6}$$

#### 2.4.2.3 Entropy

Entropy $S$ is the measurement of the number of processes or possible conformations in chemical reactions [7]. Roughly speaking, entropy measures disorder in reactions. It is an statistical property associated to stability which it is associated to the second law of thermodynamics that states stable systems are those with maximum entropy. For instance, in standard states, if the change in entropy $\Delta S^\circ > 0$, then the system is more stable; and if the change in entropy $\Delta S^\circ < 0$, then the system is more unstable. This behavior is also observed in (2.6).

In practice, the change in entropy is calculated with (2.6) because there are no instruments to measure it directly in the system.

## 2.5 Classification of Organic Compounds

As discussed previously, chemical organic compounds are based on their own functional group (see Sect. 2.2.1.2). In fact, the most studied organic compounds are [4]: hydrocarbons, alcohols, amines, aldehydes, ketones, carboxylic acids, polymers, carbohydrates, lipids, amino acids, proteins, and nucleic acids. Following, there is a brief description of each type of compound.

### *2.5.1 Hydrocarbons*

Hydrocarbons are the simplest organic compounds with functional group $CH$ [10]. Hydrocarbons are classified as alkanes, alkenes and alkynes, depending on the order of bonds among carbon atoms of their functional group.

On one hand, alkanes are the simplest hydrocarbons formed by one polar covalent bond between two carbon atoms [10]. The general formula can be expressed as $C_nH_{2n+2}$, in which $n$ is the number of carbon atoms in the molecule. Structural isomers are the different forms that alkanes can be expressed in terms of the topology of the molecules with the same number of elements. Structural isomers have particular physical and chemical properties.

Another type of hydrocarbons is the cyclic hydrocarbons in which carbons are ring-shaped bonded [10]. If all carbons in cyclic hydrocarbons are bonded with hydrogen fulfilling the octet rule, they are called cycloalkanes. In nature, the most abundant are cyclopentane and cyclohexenes.

When alkanes and cycloalkanes have low molecular weight tend to be gases and when it grows its molecular weight they are solids. This depends on the intramolecular forces and the melting and boiling points, where they become smaller as the dispersion forces become weak. The average density of alkanes is 0.8 g/ml, so they float in water [4]. Also, the more compact isomers (fewer branches) have a boiling point higher than the highly branched isomers. These organic compounds are more stable by strong *CH* bonds, but may react to form oxygen and oxidation. This oxidation can release heat that can be used as an energy source. Mainly, they come from fossil fuels.

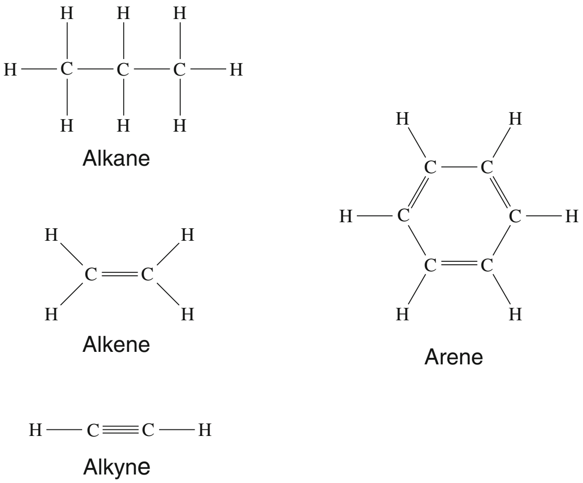
Other types of hydrocarbons are alkenes and alkynes. Alkenes are unsaturated hydrocarbons having carbon atoms with double bond and alkynes are those with carbon atoms of triple bond. Also, the arenes are the functional groups formed by a ring of carbon with double bond. In general, they share the same physical and chemical properties of alkanes, but are not soluble in water, only on themselves. The most common processes in nature are the formation of *CH* skeletons that allow biodiversity, mainly in plants, where the structures are iterative and the ramifications are taken for enzymatic reactions. If any of the hydrogen atoms of an alkane is removed is known as alkyl and if one-hydrogen atom is replaced by a halogen is known as halo alkanes.

Figure 2.10 shows the difference among alkanes, alkenes, alkynes and arenes.

### 2.5.2 Alcohols, Ethers, and Thiols

On one hand, alcohols [4] are considered the central organic compounds because they can be transformed in other organic compounds or can be expressed as a product of reactions of other compounds. Their functional group is the hydroxyl *–OH* that is bonded to a carbon atom. In general, alcohols are more soluble in water than hydrocarbons since their molecular weight comes to grow. In addition, alcohols can form weak acids in presence of water; and in presence of strong acids, alcohols can form weak bases. Several compounds that alcohols can be converted are alkyl halides, alkenes, aldehydes, and carboxylic acids, among others.

On the other hand, ethers [4] have a functional group with oxygen bonded to two-hybrid carbon atoms $sp^3$. They are more soluble in water than hydrocarbons



Alkane

Alkene

Arene

Alkyne

**Fig. 2.10** Different types of hydrocarbons

because oxygen may form hydrogen bonds (weaker bond between a hydrogen and an electronegative element). In fact, ethers are very stable as hydrocarbons and they cannot react with the majority of organic compounds. Actually, ethers can help to chemical reactions in other organic compounds.

At last, thiols [4] are organic compounds with sulfhydryl *–SH* as functional group. Physically, they cannot form hydrogen bonds; thus, the boiling point of thiols is lower than alcohols or ethers. Moreover, thiols may not be too soluble in water.

### *2.5.3 Amines*

The functional group of amines is the *amino*, a compound formed with nitrogen and one, two, or three groups of carbon atoms [3, 4]. Depending on the ammonia, amines can be classified as primary amines if its ammonia has one-hydrogen atom replaced; secondary amines if its ammonia has two-hydrogen atoms replaced; or tertiary ammines if its ammonia has three-hydrogen atoms replaced. In fact, amines can present hydrogen bonds with nitrogen atoms. The greater the molecular weight amines present, the less soluble in water they can be.

### *2.5.4 Aldehydes, Ketones, and Carboxylic Acids*

First, aldehydes [3] have a carbonyl functional group $C = O$, and a union of one or two hydrogen atoms. Ketones [3] have the same carbonyl functional group union with two-carbon atoms. In general, the boiling point of aldehydes and ketones is higher than the boiling point of nonpolar compounds. Aldehydes can react in order to form carboxyl acids because they are the one of the easiest compounds that can oxidize. In contrast, ketones cannot oxidize easily.

On the other hand, carboxylic acids [3] have the carboxyl functional group $-CO_2H$. This functional group is a hybrid between a carbonyl and a hydroxyl. In reactions, carboxyl acids can be converted into acid chlorides, esters and amides. Because carboxyl acids have oxygen and hydrogen atoms, they are quite stables and can form hydrogen bonds. In comparison with aldehydes, ketones or amines, carboxyl acids have large boiling points.

### *2.5.5 Polymers*

In organic compounds, polymers are molecules of long chains formed by monomers (simplest organic compound) bonded [3, 4]. In general, polymers have different structures, the most common are: linear, branched, comb-shaped, ladder, star, cross-linked network, and dendritic. The properties of polymers depend on the size and topology of the molecules. Physical properties of polymers are related to resistance, elasticity, and so forth. In general, monomers used in polymers are iteratively repeated through the whole network, and relations are done by covalent bonds.

There exist two processes of polymer synthesis. Step-growth polymerization is a laboratory synthetic process in which monomers are added to the chain one at a time. Since, polymerization is the process in which small molecules are added to the chain in one chemical reaction without loss of atoms. The most important polymer structures are polyamides, polyesters, polycarbonates, polyurethanes, and epoxies.

### *2.5.6 Carbohydrates, Lipids, Amino Acids, and Proteins*

These are considered as biomolecules and they are associated to a chemical function [1]. In that context, carbohydrates [1] are compounds that can store energy, form part of structural tissues, and they are part of nucleic acids. The majority of carbohydrates are based on the formula $C_n(H_2O)_m$.

Lipids [1] are organic compounds typically known as the energy source of living beings. In contrast, amino acids [1] are organic compounds formed with a carboxyl and an amino, used in the transportation of enzymes. Alpha-amino acids are monomers of proteins.

Finally, proteins [1] are organic compounds with one or more chains of polypeptides, macromolecules containing ten or more amino acids joined together with peptide bonds. In general, proteins are the structural basis of organisms, the growth regulators, the transportation of other molecules, etc.

### 2.5.7 Nucleic Acids

Other biomolecules are the nucleic acids [1, 4]. They form the basis of information over the organization, maintenance, and regulation of cellular functions. This information is expressed in genes via deoxyribonucleic acids (DNA) translated by ribonucleic acids (RNA) in the synthesis of proteins. Roughly speaking, the structure of DNA is based on deoxyribose units and phosphate, in which simple bases of aromatic heterocyclic amines mate to them: adenine, guanine, thiamine, and cytosine. The entire structure has two helices. Observations summarized the following:

- Composition of bases in any organism is the same in every cells of the organism, and it is unique to it.
- Molar percentages of adenine and thiamine are equal.
- Molar percentages of guanine and cytosine are equal, too.
- Molar percentages of purine bases (adenine and guanine) and the pyrimidine bases (cytosine and thiamine) are equal.
- In comparison to DNA, RNA is based in one structure in which uracil and cytosine are bases rather than thiamine and cytosine.

## 2.6 Organic Compounds as Inspiration

As reviewed, organic compounds are represented physically and chemically. However, these representations are intimately related because structures are conformed under chemical rules. In addition, these compounds aim to reach stability via energy minimization. In fact, this point of view inspired artificial organic networks. Thus, this section introduces the motivation of that inspiration and the characteristics studied.

### 2.6.1 Motivation

When looking at nature, and more specifically at the structure of matter, some characteristics of chemical compounds are rapidly noticeable. For instance, just around eleven elements can derived in more than twenty millions of different organic compounds. In addition, compounds are made of other simple molecules that it can be described as organized units. Also, this organization is based on chemical rules that are applied over and over again from bottom to top in the energy level scheme. But the must important observation is that chemical structures look for stabilization via energy minimization. Thus, organic compounds are made of minimal resources and optimal ways to do that.

Actually, notice that the relationship of atoms makes possible molecular units. Different atoms and different arrangements of those make different molecules. Notice that the differentiation of molecules may be observable from their physical and chemical properties. In that sense, physical properties may refer to the structure of the molecule while chemical features may refer to the behavior of that molecule. Thus, molecules can be seen as basic units of information characterized by the atoms in the inside. Moreover, molecules might be seen as encapsulation and potential inheritance of information. However, relationships among atoms cannot be performed without two possible interactions: chemical bonds and chemical reactions. The first interaction forms basic and complex molecules, and the second one forms mixtures of molecules.

## *2.6.2 Characteristics*

From the above motivation, systems inspired on chemical organic compounds will have the characteristics described below. In Fig. 2.11 is shown a diagram explaining these characteristics assuming a system inspired on chemical organic compounds.

### 2.6.2.1 Structural and Behavioral Properties

On one hand, structural properties refer to units or subsystems with specific properties. For instance, these units are molecules while specific properties are atoms. On the other hand, behavioral properties refer to functions or processes inside units. These processes are chemical interactions and chemical behaviors of compounds.

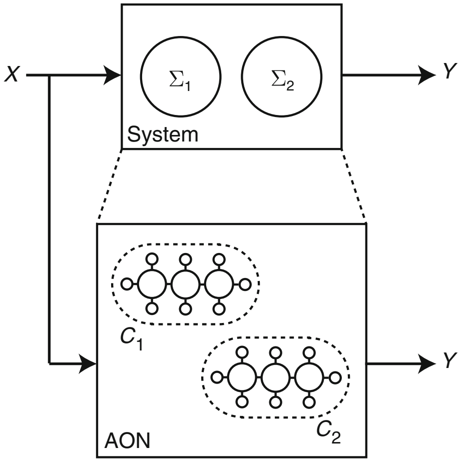


**Fig. 2.11** Example of a system inspired on chemical organic compounds

#### 2.6.2.2 Encapsulation

Structural units with processes can be encapsulated. Thus, information of subsystems can be easily clustered. In fact, inheritance and organization are derived from this characteristic.

#### 2.6.2.3 Inheritance

Consider a given, known system that is inspired on chemical organic compounds. If another, unknown system has similar behavior to the given one; then, structural units from the known system might be inherited to the unknown system. This characteristic comes from the encapsulation property.

#### 2.6.2.4 Organization

Since structural units encapsulate information, these can be organized from basic to complex units. Basic units are the ones that have elemental information, e.g. with low energy, and complex units are made of combination of basic units, e.g. with high energy. The energy level scheme can be used in order to organize information into molecular units, compound units and mixture units. For example, mixtures can contain several compounds, each one assimilating different characteristics of the whole system inspired on chemical organic compounds.

#### 2.6.2.5 Mixing Properties

Organization can derive in mixing properties. As described previously, molecular units can be seen as pieces of a puzzle; thus molecules can chemically interact among them in order to create complex units as compounds or mixtures.

#### 2.6.2.6 Stability

Since it is the central observation of organic compounds, stability is one of the most important characteristics of systems inspired on chemical organic compounds. Structural units, encapsulation, inheritance, mixing properties and organization are possible due to energy minimization. Structures of these kinds of systems will be made of minimal units, and optimality will be found. Thus, energy is an important property of systems inspired on chemical organic compounds. As explained before, chemical rules (i.e. Aufbau principle, Pauli exclusion and Hund's rule) help molecular structures to be stable since atoms and molecules are arranged suitably such that the overall structure has the minimum energy.

### 2.6.2.7 Robustness

The notion of the finite small set of atoms and the huge amount of different organic compounds characterizes systems inspired on chemical organic compounds in robustness. This characteristic can be seen as the power of finite set of parameters to model a large set of different systems.

## References

1. Bettelheim FA, Brown WH, Campbell MK, Farrell SO (2008) Introduction to general, organic and biochemistry. Cengage Learning, Belmont
2. Bhushan B (2007) Springer handbook of nanotechnology. Springer, Berlin
3. Brown TL, Lemay HE, Bursten BE, Murphy CJ, Woodward PM (2009) Chemistry: the central science. Pearson Education, Upper Saddle River
4. Brown WH, Foote CS, Iverson BL, Anslyn EV (2011) Organic chemistry. Cengage Learning, Belmont
5. Carey FA, Sundberg RJ (2007) Advanced organic chemistry: part A: structure and mechanisms. Springer, New York
6. Ganguly J (2009) Thermodynamics in earth and planetary sciences. Springer, Berlin
7. Greiner W, Neise L, Stocker H (1995) Thermodynamics and statistical mechanics. Springer, Berlin
8. Klein DR (2011) Organic chemistry. Wiley, Hoboken
9. Lide DR (2008) CRC handbook of chemistry and physics. Taylor and Francis, Boca Raton
10. Quinkert G, Egert E, Griesinger C (1996) Aspects of organic chemistry: structure. Wiley, New York

# Chapter 3
# Artificial Organic Networks

Chemical organic compounds are based on a finite, small set of elements that can create more than twenty million known compounds. These organic compounds are the most stable ones in nature primary due to chemical rules aiming energy minimization. In fact, they have characteristics like: structural and behavioral properties, encapsulation, inheritance, organization, stability, robustness, and complexity. Because of these features, the artificial organic networks technique is inspired on chemical organic compounds.

For instance, consider any unknown system that has to be described, analyzed, or predicted. Modeling techniques would be selected to solve that problem. However, as explained later in this chapter, some of these techniques have drawbacks, specially in stability and structural modeling understandings. Then, artificial organic networks is proposed to enhance these characteristics in computational algorithms for modeling problems.

In this chapter, artificial organic networks are introduced and fully described, from its components and interactions to the formal definition. In order to better understand it, an overview of the technique and the metaphor of it are discussed. Finally, some implementation issues and the outline of their solutions are described.

## 3.1 Overview of Artificial Organic Networks

Let $\Sigma$ be an unknown given system like Fig. 3.1. Also, suppose that $X$ is a set of excited signals to $\Sigma$ such that $\forall x \in X$ represents input signals, and $Y$ is a set of output signals $\forall y \in Y$ of $\Sigma$. Moreover, suppose that $\Sigma$ needs to be described, analyzed, or predicted. Then, a model $M_\Sigma$ might be used to represent $\Sigma$ (see Fig. 3.1).

In that way, it would be interesting that $M_\Sigma$ represents $\Sigma$ as closer as it is in real. Thus, $M_\Sigma$ cannot be a black or a gray box. Contrasting, the model would try to understand what happens inside the unknown system like shown in Fig. 3.2. However, in Chap. 1, it was explained that modeling the inner behavior of a system is difficult

H. Ponce-Espinosa et al., *Artificial Organic Networks*,
Studies in Computational Intelligence 521, DOI: 10.1007/978-3-319-02472-1_3,

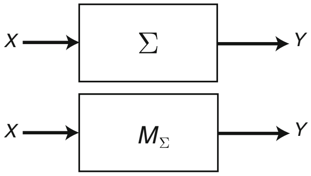


**Fig. 3.1** Example of the model of an unknown system

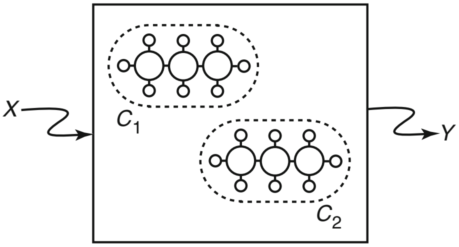


**Fig. 3.2** The idea behind modeling with artificial organic networks

and in some cases intractable due to the initial guess of the system, possibility of local minima, uncertainty in data, etc., deriving in an unstable algorithm.

To this end, let suppose that there exists a model $M_\Sigma$ representing $\Sigma$ that can encapsulate and organize information coming from $X$ and $Y$ signals, and it can stay stable for suitable conditions. Moreover, let suppose that $M_\Sigma$ is a model inspired on chemical organic compounds such that all characteristics of these compounds are also presented on it. Then, intuitively, the model $M_\Sigma$ is considered an artificial organic network, as shown in Fig. 3.2.

Notice that in order to $M_\Sigma$ be considered as a model of $\Sigma$ inspired on chemical organic compounds, its structure, behavior and rules must satisfy chemical conditions. In the next section, some key concepts from chemical organic compounds are clearly matching to concepts related to artificial organic networks; so that in following sections, the modeling technique be easily understandable.

### *3.1.1 The Metaphor*

In nature, the environment plays an important role when creating or modifying chemical organic compounds. It starts with a punch of atoms subjected to conditions that allows favorable interactions on them. Chemically, the relationships in that atoms are searching for optimal solutions, like energy minimization. This optimality makes possible organized structures, seen as modules or units of information. Moreover, the whole assembly responds to the environment. At last, the obtained chemical structure not only satisfies environmental constraints and stability, but also it develops a proper behavior.

In fact, similar chemical organic compounds have similar structures and behaviors. In other words, molecules can package information and inherit that to other compounds, representing the same structure and behavior. For instance, consider Fig. 3.3. It shows the structure of a chemical organic compound that has similar structure of another chemical organic compound having one equal package of information. In nature, functional groups are an example of that property.

Now, consider that there are artificial structures inspired on chemical organic compounds satisfying some inspired chemical rules. Then, the same characteristics of modularity, inheritance, organization and stability found in chemical organic compounds are present in the artificial structures. At last, these artificial structures will have a behavior, such that the model of artificial organic networks presented in Fig. 3.2 can be realized. In that sense, Table 3.1 presents the elements of chemical organic compounds used in artificial organic networks.

### 3.1.2 Objectives

Artificial organic networks (AONs for short) is a computational technique inspired on chemical organic compounds that models unknown engineering systems [1, 2]. In particular, this technique has two important characteristics: it is a stable algorithm for suitable conditions (see Sect. 3.2) and it represents, at least partially, the

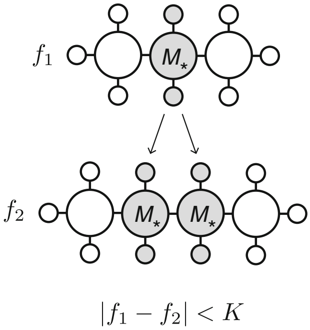


**Fig. 3.3** The metaphor of chemical organic compounds

**Table 3.1** Elements of chemical organic compounds and the meanings in artificial organic networks

| Chemical organic compounds | Meaning in artificial organic networks |
|---|---|
| Atoms | Basic structural units. Parameters of properties |
| Molecules | Basic units of information |
| Compounds | Complex units of information made of molecular units |
| Mixtures | Combination of compounds |
| Chemical bonds | Rules of interaction in atomic and molecular units |
| Stoichiometric coefficients | Definite ratios in mixtures. Weights of compounds |
| Chemical balance interaction | Solution to definite ratios of mixtures |
| Enthalpy | Energy of molecules or compounds |
| Environment | Input or exciting signals |
| Chemical properties | Output signals. Behaviors |
| Chemical organic compound | Structure and behavior of the obtained model |

internal mechanisms of unknown engineering systems. In addition, obtained models of artificial organic networks also hold chemical characteristics listed in Chap. 2.

To reach the goal, the artificial organic network technique defines components and their interactions, mathematically. It also provides the levels of energy expected in components in order to follow chemical rules.

## 3.2 Artificial Organic Compounds

The artificial organic networks technique defines four components and two interactions inspired on chemical organic compounds. In particular, the topology of components in AONs is based on the Lewis dot structures; thus, graph theory is the simplest way to model it (refer to Appendix A).

### *3.2.1 Components*

An artificial organic network is a set of graphs. Each graph represents a molecule with atoms as vertices and chemical bonds as edges. In addition, these molecules interact via the so-called chemical balance interaction, forming a mixture of compounds. Thus, in the general case, an artificial organic network is a mixture of compounds. It can be identified four components of AONs: atomic units, molecular units, compounds, and mixtures.

#### 3.2.1.1 Atomic Units

The simplest structural component of artificial organic networks is the atom. It parameterizes a molecular unit (see Sect. 3.2.1.2). In order to do that and to satisfy the ground-state principle electron configuration inside the atom with the lowest energy, the atom may be described with the number of valence electrons or the number of degrees of freedom the atom can connect to others. Moreover, in organic compounds, the octet rule supposes the ground-state principle; but in the general case, it can be considered an $\alpha$-rule, where $\alpha$ is the maximum number of valence electrons that each atom must have in the outer shell. The following definition formalizes the above description of atoms:

**Definition 3.1** ***(atomic unit)*** *Let $A^{(\alpha)}$ be a set of elements $a_i$ and $e_i$ a positive integer. Moreover, let $\alpha$ be a positive integer related to A. An element $a_i$ is said to be an atom if it has a fixed value $e_i$ denoting the* number of valence electrons *satisfying $\forall i, e_i \leq \frac{\alpha}{2}$, where $\alpha$ stands for the* full number of valence electrons. *Then, $A^{(\alpha)}$ is said to be a set of atom units. Let also $a_i, a_j \in A^{(\alpha)}$ be two atomic units and $e_i, e_j$ be the valence *electrons of those. If $e_i = e_j$ then $a_i$ and $a_j$ are called* similar atoms *denoted by $a_i = a_j$. If $e_i \neq e_j$ then $a_i$* and *$a_j$ are called* different atoms *denoted by $a_i \neq a_j$.*

It is remarkable to say that the valence electrons $e_i$ is the actual number of electrons that an atom $a_i$ has in its outer shell, and $\alpha$ stands for the total number of electrons shared in $a_i$, so that the $\alpha$-rule holds. Moreover, the maximum number of pairs of electrons shared in the outer shell of $a_i$ is equal to $\alpha/2$.

*Example 3.1* Represent a hydrogen atom $H$ and a carbon atom $C$ using the above definition. Consider $\alpha = 8$.

**Solution 3.1** Let $a_1 = H$ and $a_2 = C$ be two atomic units in the set $A^{(8)}$. Since, a hydrogen atom has one valence electron and the carbon atom has four valence electrons; then, $e_1 = 1$ and $e_2 = 4$.

*Example 3.2* Represent a set of atoms that can share at most 12 valence electrons. Is the atom $a_1$ with $e_1 = 4$ be part of that set? Is the atom $a_2$ with $e_2 = 8$ be part of it?

**Solution 3.2** The set of atoms with these characteristics is represented as $A^{(12)}$. In fact, $a_1 \in A^{(12)}$ because $e_1 < \frac{12}{2}$; but, $a_2 \notin A^{(12)}$ because $e_2 > \frac{12}{2}$.

In addition, atomic units also have associated the number of degrees of freedom they can connect to others, as follows:

**Definition 3.2** ***(degree of atomic units)*** *Let $a_i \in A^{(\alpha)}$ be an atom with a number of valence electrons $e_i$. Let $d_i, f_i$ be two positive integers with $d_i, f_i \leq \alpha$. Then, $d_i$ is the* degree of an atomic unit *$a_i$ defined as the number of valence electrons shared with other atoms, and $f_i$ is the* number of free valence electrons *in atom $a_i$ both of them holding: $e_i = d_i + f_i$.*

Figure 3.4 shows an atomic unit. Notice that $a_i$ is the indentifier and $e_i, d_i, f_i$ are values representing the number of valence electrons, the degree of the atom and the free valence electrons, respectively. Actually, the latter values are completely defined by the notion of the generalized octet-rule realized on the full number of valence electrons $\alpha$. In that sense, an atomic unit is completely defined, as expected from chemical organic compounds theory.

Moreover, atomic units are also considered parameters of artificial organic networks, as described later in this chapter. In that sense, the following definition holds:

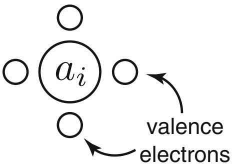


**Fig. 3.4** Example of an atomic unit

**Definition 3.3** ***(atomic value)*** *Let $a_i$ be an atom unit and $v_{a_i}$ a complex value. Then, $v_{a_i}$ is said to be the* value of atom $a_i$.

Throughout this book, the values of atoms $v_{a_i} \in \mathbb{C}$ will be written as the identifier of atoms $a_i$ indistinctively, except when a clear difference between them be required.

### 3.2.1.2 Molecular Units

In artificial organic networks, the basic unit with information is the molecule. It is defined with a structure and a behavior. In particular, the structure is associated with an arrangement of atoms and the behavior is a response of some input signals. Then, the following statements hold:

**Definition 3.4** ***(structure of a molecular unit)*** *Let $M = (A^{(\alpha)}, B)$ be a graph and $A^{(\alpha)}$ be a set of $n \geq 2$ atoms (vertices) connected with a set of covalent bonds $B$ (edges). Also, let $d_i$ and $e_i$ be the degree and the number of valence electrons of atomic atoms $a_i \in A^{(\alpha)}$, respectively. Then, $M$ is said to be a* molecule *if for all $d_i$, $d_i \leq e_i$. Moreover, if for each $d_i$, $d_i = e_i$ then $M$ is said to be a* stable molecule. *If for any $d_i$, $d_i < e_i$ then $M$ is said to be an* unstable molecule. *Those are the only states of a molecular unit $M$.*

**Definition 3.5** ***(behavior of a molecular unit)*** *Let $M$ be a molecule and $X$ be the set of inputs that excite the molecule $M$. Also, let $\varphi$ be a function such that $\varphi : X \to \mathbb{R}$. Then, $\varphi$ is referred to the* behavior of the molecule $M$ due to $X$, *if there exists an atom $a_c \in A^{(\alpha)}$ of $M$ with degree $d_c$ and the following holds:*

1. $\varphi = \varphi(X, d_c)$.
2. $\varphi$ converges in the input domain $X$.

*Example 3.3* Let $M$ be a molecule defined as $M = (\{a_1, a_{2,1}, a_{2,2}, a_{2,3}\}, \{b_{12}, b_{12}, b_{12}\})$ with number of valence electrons $\{e_1 = 3, e_{2,i} = 1\}, \forall i = 1, 2, 3$. Draw $M$ assuming that covalent bonds $b_{ij}$ means that atom $a_i$ is connected with atom $a_j$.

**Solution 3.3** Figure 3.5 shows molecule $M$. Notice that covalent bonds are undirected edges in the graph.

*Example 3.4* Consider the above molecule $M$ in Example 3.3. Propose a molecular behavior $\varphi(x, d_1)$ for the input signal $x \in X$. Which one is the atom $a_c$?

**Solution 3.4** One possible molecular behavior may be written as (3.1) with atom $a_c = a_1$ since the degree of atom in the molecular behavior depends on $d_1$.

$$\varphi(x, d_1) = \sum_{i=1}^{d_1} a_{2,i} \cdot x^i \tag{3.1}$$

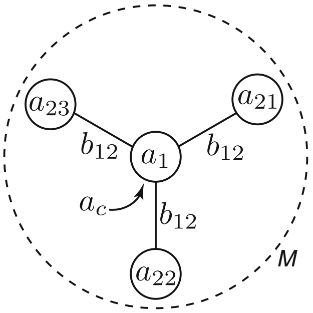


**Fig. 3.5** Graph of molecule $M$ in Example 3.3

Notice that $\varphi$ increases its meaning while the degree of atom $d_1$ increases.

It is remarkable to say that stable molecules are the preferred ones more than unstable molecules because a stable molecule means minimum of energy due to the ground-state principle of atomic units. But consider that only unstable molecules might be connecting to others in order to increase the complexity of its behavior. Moreover, Definitions 3.4 and 3.5 specify that molecules encapsulate information via the $\varphi$-function, but they are constrained to interact with other molecules by the fact of the structure in graphs (Definitions 3.1 and 3.2). Also, consider that the $\varphi$-function in Definition 3.5 is inspired in order to follow chemical rules. For example, property 1 assures that the function depends on the number of valence electrons shared with the atom $a_c$ as shown in Fig. 3.5. As noticed, if the number of atoms associated to $a_c$ changes, the $\varphi$-function needs to have a different behavior. For example, in Fig. 3.5, the addition of $a_{2,i}$ atoms increases the meaning of (3.1). For instance, consider (3.1) with $d_1 = 1$, then $\varphi$ is a linear equation with slope equal to the value of $a_{2,1}$ while if $d_1 = 3$, $\varphi$ is a cubic equation with coefficients $a_{2,1}$, $a_{2,2}$, $a_{2,3}$. Thus, atomic units can also be considered as properties of molecules as described in Sect. 3.2.1.1.

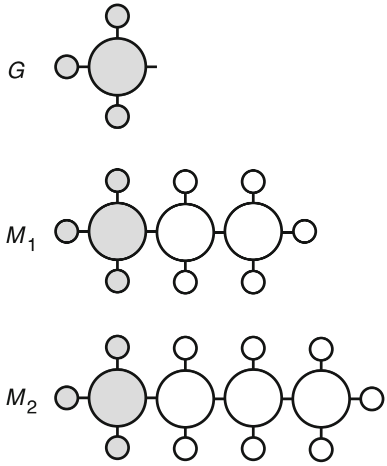
Moreover, property 2 of Definition 3.5 is modeling stability of chemical structures in a mathematical fashion. Here, the behavior of molecule is inspired on reaching a steady-state of the molecule; then, in mathematics, convergence is one way to do that. Again, consider the $\varphi$-function in (3.1). This is a power series with a radius of convergence $R$ such that $\|x\| < R$. In this case, it is easy to see that the interval of convergence of $\varphi$-function is $\|x\| < 1$.

On the other hand, two kinds of molecules are defined in artificial organic networks: functional groups and primitive molecules.

**Definition 3.6** ***(functional group)*** *Let $M$ be a molecule and $G$ be a subgraph of $M$, denoted by $G \subseteq M$. Then, $G$ is a* functional group *if $G$ is an unstable molecule.*

It is clear by Definition 3.6 that a functional group can be any subgraph of any molecule. However, consider the inverse process: defining a functional group $G$, a set of molecules $\Omega = \{M_1, ..., M_k\}$ can be spanned from $G$. Then, all molecules in $\Omega$ will have the same primary structure defined on $G$, as shown in Fig. 3.6. This notion

**Fig. 3.6** Example of a functional group



of spanning will be useful when formalizing artificial organic networks. In addition, a functional group can act as kernel of molecules (and later of compounds).

Accordingly to the ground-state principle and bonds in chemistry, molecules interact each other in order to minimize the energy in structure, reaching a steady state of energy. This assumption requires that molecules be unstable. Otherwise, molecules are in steady state and only chemical reactions can perform interaction among molecules. Thus, using the criterion of unstable molecules, primitive molecules are introduced in Definition 3.7.

**Definition 3.7** ***(primitive molecule)*** *Let G be a functional group and* $M = (A^{(\alpha)}, B)$ *be an unstable molecule spanned from G. Then, M is said to be a* primitive molecule*, and it satisfies the following properties:*

1. *M is unique.*
2. *For each* $a_i \in A^{(\alpha)}$*, there is at least any* $d_i$, $d_i < e_i$.
3. *B is a set of polar covalent bonds.*

It is remarkable to say that primitive molecules can only be made if all of the relationships among atomic units are polar covalent bonds (see Sect. 3.2.2) because it refers to that there are no relationships between similar atoms inside primitive molecules. In Fig. 3.7 is shown all possible primitive molecules spanned the given functional group *G*.

Finally, artificial organic networks makes a distinction between simple and complex molecules. For instance, consider a simple molecule to be any molecule only made of polar covalent bonds, e.g. functional groups, primitive molecules, or any other molecule with an atom $a_c$ and other atoms different from it and attached directly to it, like molecule in Fig. 3.5. If molecules are made of polar and nonpolar covalent bonds (refer to Sect. 3.2.2), then these molecules are complex. In order to distinguish

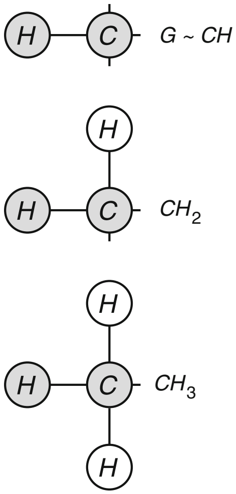


**Fig. 3.7** Example of primitive molecules spanned from $G$

both simple and complex molecules in artificial organic networks, the first ones are simply named molecular units, while the second ones are named compounds.

#### 3.2.1.3 Compounds

In artificial organic networks, a compound unit is a complex molecule made of two or more primitive molecules which they interact together in order to increase the complexity of its behavior. Since compounds are molecules, they have an structure and a behavior, as described next:

**Definition 3.8** ***(structure of a compound)*** *Let $\Omega = \{M_1, ..., M_k\}$ be a set of primitive molecules spanned from a functional group G. Also, let $C = (\Omega, B_N) \supseteq G$ be a molecule consisting of primitive molecules $\Omega$ linked with a set of nonpolar covalent bonds $B_N$. If for $|B_N|$-pairs*[1] *of molecules $(M_i, M_j) \in \Omega^2$ with $i \neq j$, there exists a pair of similar atomic units $(a_i, a_j)$, such that, $b_n^{ij} \in B_N$; then, C is a* compound.*Also, let $D_i$ be the set of the degree of atoms and $E_i$ be the set of the number of valence electrons in each $M_i \in \Omega$.* If $\sum_i D_i = \sum_i E_i$, *then C is said to be a* stable compound. *Otherwise, C is an* unstable compound.

**Definition 3.9** ***(behavior of a compound)*** *Let C be a compound made of a set of molecules $\Omega = \{M_1, ..., M_k\}$ with molecular behaviors $\varphi_1, ..., \varphi_k$. Also, let X be the set of inputs that excite the compound C. Then, $\psi$ is the* behavior of the compound $C$ due to $X$, *such that,* $\psi : \varphi_1 \times \cdots \times \varphi_k \to \mathbb{R}$.

[1] $|(\cdot)|$ stands for the cardinality of the set $(\cdot)$.

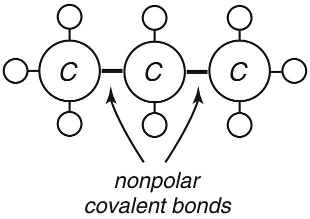


**Fig. 3.8** Example of a compound

As notice, compounds are based on nonpolar covalent bonds as shown in Fig. 3.8. Roughly speaking, a compound is necessarily made of at least two similar atoms joined together. Additionally, compounds map from the set of behaviors of primitive molecules to a real value giving a more complex behavior than simple molecules. This is important when different information needs to be crossover. In practice, compounds might be nonlinear relationships among molecular behaviors.

Finally, stable compounds are more preferable than unstable ones. In order to do this, primitive molecules or atomic units have to be used.

#### 3.2.1.4 Mixtures

In nature, when two or more molecules are mixed together, the resultant mixture contains more information. For instance, it assumes that any number of molecules can interact without sharing electrons, mixing them up in definite ratios. In such cases, artificial organic networks defines the chemical balance interaction, useful to find the optimal ratios of molecules to get the minimum loss energy in the whole structure. Following, mixtures of molecules and the chemical balance interaction of AONs are introduced.

**Definition 3.10** ***(mixture of molecules)*** *Let $\Gamma = \{M_1, ..., M_k\}$ be a set of molecules with a set of behaviors $\Phi = \{\varphi_1, ..., \varphi_k\}$. Then, the* mixture of molecules *$S$ is a linear combination of behaviors of molecules in $\Phi$ such that there exists a set of coefficients $\Lambda = \{\alpha_1, ..., \alpha_k\}$ of real values, called the* stoichiometric coefficients. *Hence,*

$$S(X) = \sum_{i=1}^{k} \alpha_i \varphi_i(X) \tag{3.2}$$

*Moreover, $\Phi$ is the* basis of the mixture of molecules, *$\Gamma$ is the structure of the mixture of molecules, and $S(X)$ is the behavior of the mixture of molecules.*

In general, the notion of mixtures can be expanded to a mixture of compounds allowing the opportunity to represent segmented, understandable information and behaviors. It follows from the fact that compounds are defined as molecules. Let $\Gamma = \{C_1, ..., C_k\}$ be the set of compounds with behaviors $\Phi = \{\Psi_1, ..., \Psi_k\}$. Then,

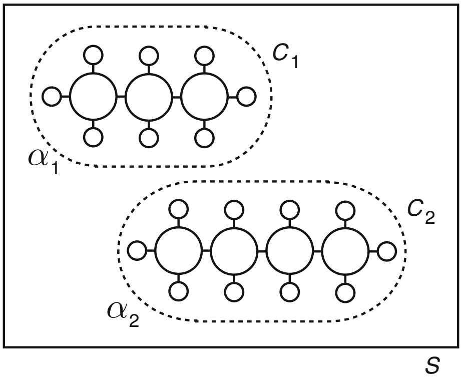


**Fig. 3.9** Example of mixtures

$S$ is the mixture of compounds if $\Gamma$ is its structure, $\Phi$ is its basis and $S(X)$ is its behavior such that (3.2) holds.

Figure 3.9 shows an example of a mixture of molecules and a mixture of compounds.

## *3.2.2 Interactions*

The above section introduces all components in artificial organic networks. In this section, two interactions are defined: covalent bonds and the chemical balance. The first interaction refers to the relational rules that allow atoms to join together, forming molecules and compounds. The second interaction refers to the way that molecules and compounds relate among them without sharing electrons.

### 3.2.2.1 Covalent Bonds

Atomic interaction is very important in artificial organic networks because it defines the way that they link each other. In chemical organic compounds, this interaction is known as a chemical covalent bonding. Artificial organic networks defines covalent bonding as following:

**Definition 3.11** ***(covalent bond)*** *Let* $a_i, a_j \in A^{(\alpha)}$ *be two atomic units. Also, let* $B$ *be a set of elements* $b_k^{ij}$. *If* $b_k^{ij}$ *links atomic units* $a_i$ *and* $a_j$ *then it is called a* covalent bond.*Hence,* $B$ *is called the* set of covalent bonds. *In particular, if* $a_i = a_j$ *then* $b_k^{ij}$ *is called a* nonpolar covalent bond. *Similarly, if* $a_i \neq a_j$ *then* $b_k^{ij}$ *is called a* polar covalent bond.

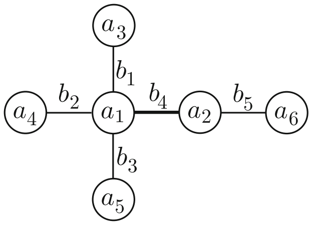


**Fig. 3.10** Structure of molecule $M$ in Example 3.5 explaining covalent bonds

*Example 3.5* Suppose that there exist two atoms $a_1$, $a_2$ and four atoms $a_3$, $a_4$, $a_5$, $a_6$ in $A^{(8)}$ with number of valence electrons $e_{1,2} = 4$, $e_{3,4,5,6} = 1$. Also, consider that there is a molecule $M$ made of these atoms with covalent bonds $B = \{b_1^{13}, b_2^{14}, b_3^{15}, b_4^{12}, b_5^{26}\}$. Determine the shape of $M$ and classify covalent bonds into polar and nonpolar bonds.

**Solution 3.5** Using Definition 3.11, the structure of molecule $M$ is absolutely determined, as shown in Fig. 3.10. From this figure, notice that $a_1$ and $a_2$ are two similar atoms linked with $b_4^{12}$. Thus, from Definition 3.11, $b_4^{12}$ is a nonpolar covalent bond, and the set $B \backslash \{b_4^{12}\}$ contains polar covalent bonds because the atoms joined by them are different. Remember that similarity in atoms are based on the number of valence electrons $e_i$.

As discussed in Sect. 3.2.1.2, the behavior of molecules is built using the combination of atoms according to a central atom $a_c$. In such that case, polar covalent bonds are used. However, compounds use nonpolar covalent bonds to define their behaviors in terms of mapping simple molecular behaviors. Focusing on that mapping, nonpolar covalent bonds play an importat role. Thus, artificial organic networks makes an specification of them. In fact, nonpolar covalent bonds have some properties that promote a kind of behavior on them, as stated next:

**Definition 3.12** ***(properties of nonpolar covalent bonds)*** *Let $b_k^{ij} \in B$ be a nonpolar covalent bond of a pair of atoms $a_i, a_j \in A^{(\alpha)}$. Also, let $\Delta = \langle \delta_1, ..., \delta_n \rangle$ be an n-tuple of interesting properties of $b_k^{ij}$. Then, $b_k^{ij}$ is said to be a* nonpolar covalent bond characterized by $\Delta$. *Also, let $\pi$ be a mapping such that $\pi : \Delta \to \mathbb{R}$. Then, $\pi$ is referred to the* behavior of the nonpolar covalent bond $b_k^{ij}$.

In fact, Definition 3.12 does not determine any particular property $\Delta$; but it would be interesting to get inspiration from some properties of chemical covalent bonds as reported in Chap. 2. For instance, the length, the order and the minimum energy of bonds might be proposed, as explained later in Chap. 4.

#### 3.2.2.2 Chemical Balance Interaction

On the other hand, when molecules and compounds are completely stable, they cannot interact each other unless a chemical reaction carries out. In that sense, a chemical balance interaction can be used instead. In artificial organic networks, the chemical balance interaction is defined as follows:

**Definition 3.13** ***(chemical balance interaction)*** *Let X be the set of inputs that excites the set of molecules $\Gamma = \{M_1, ..., M_k\}$ with behaviors $\Phi = \{\varphi_1, ..., \varphi_k\}$. Let $S(X)$ be the response of the mixture of molecules S, such that,*

$$S(X) = \sum_{i=1}^{k} \alpha_i \varphi_i(X) \tag{3.3}$$

*The* chemical balance interaction *is the solution to the problem of finding the set of stoichiometric coefficients $\Lambda = \{\alpha_1, ..., \alpha_k\}$ of the mixture of molecules S.*

*Example 3.6* Let $M_1$ and $M_2$ be two molecules in the set of molecules $\Gamma$ of a mixture $S$ with molecular behaviors $\varphi_1$ and $\varphi_2$, respectively. Also, let $\Lambda = \{1.0, -0.5\}$ be the set of stoichiometric coefficients of $S$. Obtain the response of $S$ with respect to an input signal $x$, if $\varphi_1$ and $\varphi_2$ are defined as (3.4) and (3.5). In addition, determine the response of $S$ in the input domain $x \in [-1, 1]$.

$$\varphi_1(x) = 5x^2 + 1 \tag{3.4}$$

$$\varphi_2(x) = 4x^2 - 6x + 2 \tag{3.5}$$

**Solution 3.6** Using (3.3), the response of $S$ can be expressed as (3.6):

$$S(x) = \alpha_1\varphi_1 + \alpha_2\varphi_2 \tag{3.6}$$

Substituting (3.4) and (3.5) with $\Lambda$ in (3.6), the response of $S$ is finally written as (3.7).

$$S(x) = 3x^2 + 3x \tag{3.7}$$

The response $S(x)$ for all $x \in [-1, 1]$ is shown in Fig. 3.11.

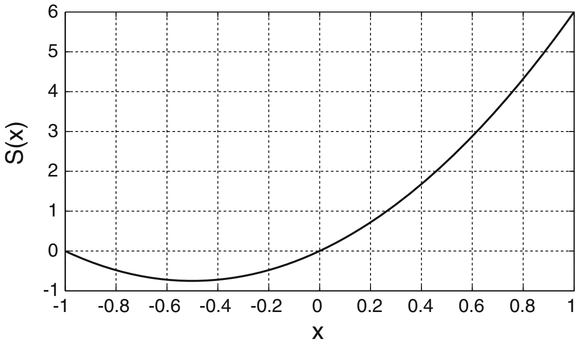


**Fig. 3.11** Response of the mixture of molecules $S$ of Example 3.6.

## 3.3 Networks of Artificial Organic Compounds

As described above, artificial organic networks technique inherits characteristics of chemical organic compounds. In this section, these characteristics are discussed in terms of the structure, behavior and mixtures of artificial organic compounds.

### *3.3.1 The Structure*

The structure of artificial organic networks is organized from basic units to complex units. As discussed later in Sect. 3.4.1, the structure of AONs is obtained from a chemically inspired strategy that will assure self-organizing artificial organic compounds. If neccesary, this organization in structures might be found using optimization algorithms via energy minimization.[2]

Then, the idea behind organizing structures in AONs refers to encapsulation and inheritance of information. For instance, molecular topologies can be inherited to other artificial organic compounds getting similar structure and behavior. In that sense, modular units (or sub-models) can also be obtained when AONs are used for modeling systems.

### *3.3.2 The Behavior*

In terms of the behavior of artificial organic networks, molecules encapsulate nonlinear relationships among input signals, but also are parametrized by atomic values. For instance, consider any molecule with instanced atomic values, such that they can be used as metadata; then, artificial organic networks are self-organizing topologies with classified data for posterior analysis of the model, giving to AONs a partial understanding of the modeled system.

On the other hand, artificial organic compounds can also assimilate nonlinear relationships among molecules. This characteristic offers encapsulation of complex behaviors due to input signals of the given system.

### *3.3.3 Mixtures of Compounds*

At last, mixtures can be used to combine linearly complex behaviors in definite ratios for capturing correlations among properties assimilated by molecules and compounds. It is remarkable to say that mixtures can also be used for combining

[2] In Chap. 4, this statement is proved and a search algorithm based on optimization processes is presented.

compounds from different systems in order to get hybrid models. For example, image processing systems can used artificial compounds as different filters that act over another compounds modeling images, via mixtures, getting processed images.

## 3.4 The Technique of Artificial Organic Networks

Now that components and interactions have been defined, the artificial organic networks technique can be described from three different but complementary perspectives. Firstly, a chemically inspired rule of levels of energy in artificial organic compounds is introduced in order to determine how atoms, molecules, compounds and mixtures will be organized. Then, the mathematical definition of artificial organic networks is presented. Finally, all aspects described earlier are summarized into the model of artificial organic networks.

### *3.4.1 Levels of Energy in Components*

Artificial organic networks technique considers energy minimization as the central observation. For instance, take a look at the objective of AONs: modeling systems. In that way, information like input and output signals are required. From signal theory, information in systems might be seen as energy, i.e. energy signals. Thus, the structure of artificial organic networks 'grows' from this information; but chemically, it has to be minimized. Components and interactions are prepared to do that; however, they need a rule for this process. It is summarized in the three-level energy model as follows.

It identifies three levels of energy around the structure of artificial organic networks in which the energy is supposed to be minimized. The information of the system allows organizing the structure of artificial organic networks. First, atoms interact among them to make molecules. Then, molecules interact to make compounds. And finally, compounds interact making mixtures. In a nutshell, the three-level energy rule is presented next:

**1st level:** (***ground-state electron configuration***) Information tends to be packaged in stable molecules using atomic units, i.e. as parameters, with free valence electrons. Polar covalent bonds are required at this level of energy minimization.

**2nd level:** (***composite units***) Information not covered in stable molecules tends to be packaged using a set of molecules, i.e. primitive molecules, to create compounds. Actually, nonpolar covalent bonds are required at this level of energy minimization.

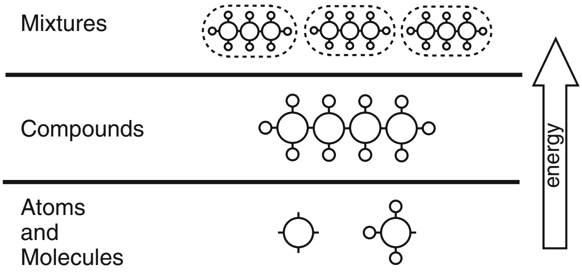


**Fig. 3.12** The three-level energy rule of artificial organic networks

**3rd level:** (***mixtures***) Complex information tends to be packaged as mixtures of molecules and compounds. At this level, chemical balance interaction regulates energy in structures.

Figure 3.12 shows this energy rule of artificial organic networks. Notice that this rule allows components of AONs interact among them, as chemical organic compounds do in nature. Furthermore, this rule implicitly organizes the structure of the artificial organic network because simple information is stored in primitive molecules, and complex information can either be modeled as compounds or mixtures.

Additionally, a mathematical reference and components of AONs related to each level of energy are also depicted in Fig. 3.12. It is remarkable to say that in this rule, lower levels guarantee energy minimization more than upper levels. In practice, this energy rule has to be implemented in an algorithm, let say $f$.

### 3.4.2 Formal Definition of Artificial Organic Networks

Mathematically, the notion of artificial organic networks is formalized into the following definition:

**Definition 3.14** ***(artificial organic networks)*** *Let $\varGamma_f$ be the structure of a mixture $S$ built by algorithm $f$, with basis $\varPhi$, a set of stoichiometric coefficients $\varLambda = \{\alpha_1, .., \alpha_m\}$ and $S(X)$ be the behavior of a mixture $S$ due to a set of inputs $X$. Then, an* artificial organic network $AON = \langle \varGamma_f, \varPhi, \varLambda, X \rangle$ *is a quadruple consisting of the structure of a mixture $S$, a basis, a set of stoichiometric coefficients and a set of inputs, that holds the following properties:*

1. *There exists a finite set $A^{(\alpha)}$ of atomic units spanning $\varGamma_f$.*
2. *There is a unique functional group $G = (A^{(\alpha)}, B_P)$ with polar covalent bonds $B_P$.*
3. *Each molecule $M_k^{(i)} \in \varOmega^{(i)}$* in *$C_i \in \varGamma_f$ is spanned from $G$ and each behavior of molecule $\varphi_k^{(i)}$ represents a single unit of information.*

4. *Each compound $C_i \in \Gamma_f$ is spanned from G and each behavior of compound $\psi_i$ is a function of the form $\psi_i : \varphi_1^{(i)} \times \cdots \times \varphi_k^{(i)} \times \cdots \times \varphi_r^{(i)} \to \mathbb{R}$, where r is the cardinality of $\Omega^{(i)}$, representing a composite unit of information.*
5. $f$ is an algorithm based on the three-level energy rule.
6. *$S(X)$ is the behavior of AON due to $X$.*

The above statement represents the formal definition of artificial organic networks. In fact, all properties assume that molecules and compounds are made of the same primary structure $G$ and they can be used for representing information in a compact form (molecules) or in composite form (compounds). According to the three-level energy rule, $\Gamma_f$ is defined with molecules as the first compact units with information. Compounds are the next level of packing information, and finally, the mixture of compounds is the last level of packing information. This notion of packaging or encapsulation of information reveals some understandings on how artificial organic networks might be used for partial interpretation of modeled systems. Figure 3.13 shows a simple artificial organic network.

### 3.4.3 Model of Artificial Organic Networks

In order to use artificial organic networks as a model of any given system, two steps are defined: the training process and the inference process. In the first case, the training process refers to build the AON-structure based on atoms, molecules, compounds, mixtures and interactions, and then to fix all parameter values in the structure capturing all relevant information from the given system, as described in Sect. 3.5.

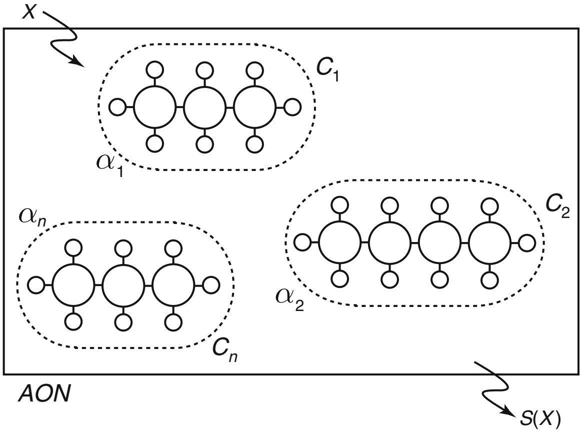


**Fig. 3.13** A simple artificial organic network

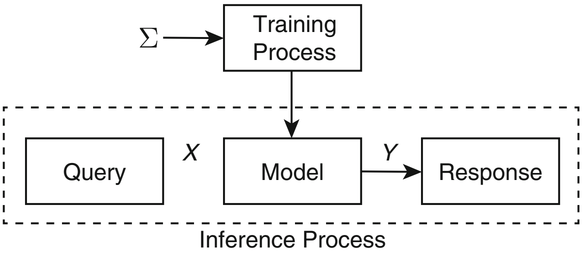


**Fig. 3.14** The training and inference processes of artificial organic networks technique

The second process refers to actually use the AON-structure, obtained in the first step, as an inference system. In that sense, the AON-structure can return an output value based on an input value of the system. Figure 3.14 shows the two steps of the artificial organic network model. Notice that once the AON-structure is found by the training process, it can be used separately as a model of the given system.

## 3.5 Implementation Issues

Formally, the artificial organic networks technique faces two open problems when it has to be used as an inference system or as a modeling system. These problems are introduced following.

### 3.5.1 The Search Topological Parameters Problem

Suppose that there is any given system $\Sigma = (X, Y)$ with a set of input signals $X$ and a set of output signals $Y$. Also, let suppose that there exists an artificial organic network *AON* with structure $\Gamma_f$ and basis $\Phi$ that has to model the system $\Sigma$. However, all parameters in $\Gamma_f$ has to be found in order to use *AON* as a model of $\Sigma$. In other words, a topology of molecules and mixtures are already fixed, but atomic parameters and sotichiometric coefficients are not set. Then, any process to find these values is necessary. Equivalently, this problem can be stated as Definition 3.15.

**Definition 3.15** ***(search topological parameters problem)*** *Let $\Sigma$ be any given system with input signals $X$ and output signals $Y$, and let $AON = \langle \Gamma_f, \Phi, \Lambda, X \rangle$ be an artificial organic network as a model of $\Sigma$. Also, let $\varphi_k^{(i)}$ be the behavior of molecule $M_k^{(i)} \in \Omega^{(i)}$, for all compounds $C_i \in \Gamma_f$, as a parametric function $\varphi_k^{(i)} = \varphi_k^{(i)}(\theta_{1k}^{(i)}, ..., \theta_{lk}^{(i)})$ with atomic parameters $\theta_{1k}^{(i)}, ..., \theta_{lk}^{(i)}$. Also, let $S(X)$ be the behavior of AON due to X, such that,*

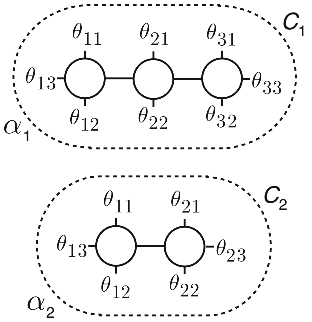


**Fig. 3.15** The search topological parameters problem (STPP) refers to find all parameters in the AON-structure

$$S(X) = \sum_{i=1}^{m} \alpha_i \varphi_k^{(i)}(\theta_{1k}^{(i)}, ..., \theta_{lk}^{(i)}, X) \tag{3.8}$$

*The* search topological parameters problem *(or* STPP *for short) refers to find the set of stoichiometric coefficients $\Lambda = \{\alpha_1, ..., \alpha_m\}$ and the collection of atomic parameters $\theta_{1k}^{(i)}, ..., \theta_{lk}^{(i)}$ for all $i = 1, ..., m$, such that $S(X)$ be equivalent to $Y$.*

As notice, the search topological parameters problem indicates that has to be a procedure to find all parameter values in a prior structure of artificial organic networks, as shown in Fig. 3.15. In this case, Definition 3.15 does not define how to obtain $\Gamma_f$.

## 3.5.2 The Build Topological Structure Problem

Let suppose that there is the same system $\Sigma$ and an artificial organic network *AON* has to model it. But now, *AON* does not have any structure $\Gamma_f$. Then, there exists a problem to find the structure. Mathematically, it can be stated as follows:

**Definition 3.16** ***(build topological structure problem)*** *Let $\Sigma$ be any given system with input signals $X$ and output signals $Y$, and let $AON = \langle \Gamma_f, \Phi, \Lambda, X \rangle$ be an artificial organic network with behavior $S(X)$ due to $X$ as a model of $\Sigma$. Then, the* build topological structure problem *(or* BTSP *for short) refers to find any $\Gamma_f \neq \emptyset$, such that, $S(X)$ be equivalent to $Y$ and AON holds for the input set $X$.*

The build topological structure problem only refers to find the structure of the artificial organic network as shown in Fig. 3.16. In practice, this problem has to be merged with the search topological parameters problem in order to find the whole artificial organic network *AON* that models $\Sigma$.

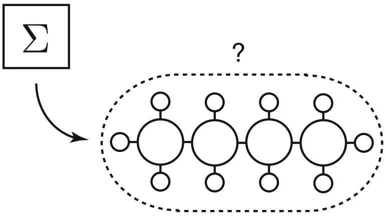


**Fig. 3.16** The build topological structure problem (BTSP) refers to find the structure of the AON

### 3.5.3 Artificial Organic Networks Based Algorithms

In summary, the artificial organic networks technique is inspired on chemical organic compounds. Actually, the technique defines components, interactions, energy minimization rules and the open problems in order to design practical algorithms for modeling systems. Three kinds of algorithms based on artificial organic networks may be identified:

**Chemically inspired algorithms** They find inspiration in organic compounds to define their functional groups and molecular structures in order to obtain certain interesting characteristics of the chemical organic compounds.

**Artificial basis algorithms** They define specific functional groups and set of atoms independently to chemical organic compounds.

**Hybrid algorithms** They mix artificial organic network structures coming from both chemically inspired and artificial basis algorithms.

In particular, Chap. 4 presents the first artificial organic network based algorithm named Artificial Hydrocarbon Networks that falls into the chemically inspired category in order to exploit some characteristics of chemical hydrocarbons.

## References

1. Ponce H, Ponce P (2011) Artificial organic networks. In: Proceedings of IEEE conference on electronics, robotics, and automotive mechanics, Cuernavaca, Mexico, pp 29–34
2. Ponce H, Ponce P (2012) Artificial hydrocarbon networks: a new algorithm bio-inspired on organic chemistry. Int J Art Intell Comput Res 4(1):39–51

# Chapter 4
# Artificial Hydrocarbon Networks

Hydrocarbons, chemical organic compounds based on hydrogen and carbon atoms, are the most stable compounds in nature. Actually, the efectiveness of stability in hydrocarbons is the electronegativity property between carbon atoms allowing strong covalent bonds. For instance, consider an artificial organic network that uses only two atoms with high stability in its structure like hydrocarbons; then, any given system may be modeled by using hydrocarbon inspired networks. In that sense, this chapter introduces artificial hydrocarbon networks.

The first section introduces an overview of the artificial hydrocarbon networks algorithm from its inspiration to its objectives. Then, this chapter presents the mathematical definition of components and interactions occupied in the approach, as well as the formal definition of artificial hydrocarbon networks. In addition, it introduces and formulates the basic algorithm of artificial hydrocarbon networks. In following chapters, improvements to this algorithm are also presented. After that, two important theorems related to the metrics of the basic algorithm are introduced and proved. Finally, this chapter discusses implementability of the algorithm.

## 4.1 Introduction to Artificial Hydrocarbon Networks

Hydrocarbons are simple and stable, chemical structures that are very attractive to perform and implement artificial organic networks in an algorithm named artificial hydrocarbon networks. In order to understand the approach, this section presents an overview of chemical hydrocarbon compounds, used as inspiration, and the scope of the algorithm.

### *4.1.1 Chemical Inspiration*

In nature, hydrocarbons are the simplest and the most stable chemical organic compounds. These two properties may hydrocarbons be used as artificial organic networks because they are made of hydrogen and carbon atoms, delimiting the definition

H. Ponce-Espinosa et al., *Artificial Organic Networks*,
Studies in Computational Intelligence 521, DOI: 10.1007/978-3-319-02472-1_4,

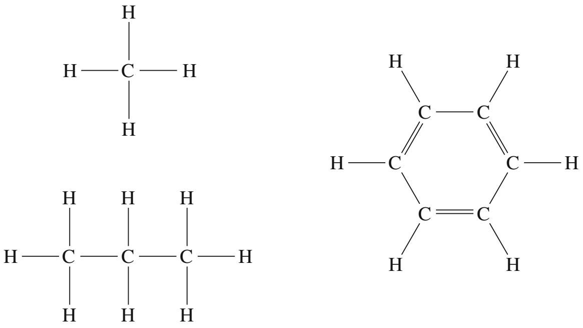


Fig. 4.1 Examples of different hydrocarbons due to the carbon property

**Table 4.1** Bond enthalpies present in hydrocarbons in kJ/mol

| Chemical bond | Enthalpy (kJ/mol) |
|---|---|
| $C - C$ | 350 |
| $C - H$ | 415 |
| $C = C$ | 611 |
| $C \equiv C$ | 837 |

of atomic units in AONs [1]. Also, stability in hydrocarbons is highly important to the artificial organic network technique.

Up to date, more than 20 million different chemical hydrocarbon compounds have been identified and catalogued. This versatility is due to the property of carbon atoms to form chains and rings of them, as shown in Fig. 4.1. That is one of the reasons hydrocarbons inspire the algorithm introduced in this chapter: combinations of two different atoms have the potential to form infinite number of different chemical molecules. To the approach, this potentiality is translated to have many compounds with different behaviors formed by two atomic units.

Moreover, the property of carbon chains is chemically explained via bond energy of carbon elements [2, 3]. For instance, consider only the chemical bonds present in hydrocarbons. Actually, the energy of these particular bonds are high in comparison to the energy of chemical bonds between other elements, giving the stability property to hydrocarbons. In other words, bond enthalpies represent the minimum energy that has to be applied to break them. Table 4.1 summarizes the enthalpies of chemical bonds present in hydrocarbons. Thus, stability is another reason to use hydrocarbons as inspiration of the algorithm introduced.

### 4.1.2 Objectives and Scope

Artificial hydrocarbon networks (AHNs for short) is a computational algorithm inspired on chemical hydrocarbon compounds and based on artificial organic networks (discussed in Chap. 3). The objective of this mathematical model is to constrain an artificial organic network to conditions of stability and simplicity shown by hydrocarbons, in order to allow them to model unknown engineering systems and to partially understand unknown information inside systems, e.g. as gray boxes. Moreover, artificial hydrocarbon networks algorithm solves the search topological parameters (STPP) and the build topological structure (BTSP) problems of artificial organic networks discussed in Sect. 1.5.

In terms of machine learning, artificial hydrocarbon networks is a supervised learning algorithm because it needs the inputs and the outputs of the unknown system to model.

On the other hand, the scope of artificial hydrocarbon networks is limited to modeling causal and continuous physical systems. Examples of some engineering systems that can be modeled and inferred by AHNs can be audio signals, environmental variables, voltage and current signals, and so forth.

## 4.2 Basics of Artificial Hydrocarbon Networks

Artificial hydrocarbon networks algorithm is based on the framework of artificial organic networks. In that way, the following section introduces all components and interactions instantiated from the AONs technique, arising on mathematical and algorithmic formulations to derive the basic AHN-algorithm. In the first part, atomic units, molecular units, compounds and mixtures are introduced. Then, covalent bonding and chemical balance interactions are presented. After that, the formal definition of AHNs is described and the chemical rule of artificial compound formation is discussed. At last, the basic algorithm of artificial hydrocarbon networks is introduced.

It is remarkable to say that artificial hydrocarbon networks is inspired on the way chemical hydrocarbon compounds are formed. In that sense, some chemical elements are adopted like atoms, their valence electrons, and the chemical rule of nonpolar covalent bonding formation in hydrocarbons.

### *4.2.1 Components*

The artificial hydrocarbon network algorithm defines four components to produce a set of artificial compounds: atomic units, molecular units, compounds and mixtures. Recall that these definitions were presented and discussed when artificial organic networks technique was introduced. Refer to Chap. 3, if necessary.

#### 4.2.1.1 Atomic Units

The AHN-algorithm defines two different atomic units. Inspired on chemical hydrocarbon compounds, these two atomic units are the hydrogen atom and the carbon atom. In addition, the valence electrons of both atomic units are chemically based, as the following definition:

**Definition 4.1** (***atomic units***) *Let $H$ and $C$ be two different atomic units with valence electrons $e_H$ and $e_C$, respectively, in the set of atomic units $A^{(8)}$. Then, $H$ is called* hydrogen atom *with $e_H = 1$ and $C$ is called* carbon atom *with $e_C = 4$. Also, $A^{(8)}$ is called the* set of atom units of artificial hydrocarbon networks.

Notice that the set of atom units of artificial hydrocarbon networks claims for the octet rule as defining $\alpha = 8$. It is important because this condition will constrain the structural forming of artificial hydrocarbon compounds.

**Definition 4.2** (***atomic values***) *Let $v_H$ and $v_C$ two complex numbers $v_H, v_C \in \mathbb{C}$. Then, $v_H$ and $v_C$ are* atomic values of hydrogen and carbon atoms*, respectively.*

### 4.2.1.2 Molecular Units

The interaction among hydrogen and carbon atoms is important to build artificial hydrocarbon compounds. Definition 4.3 introduces the notion of a particular interaction of atoms in AHNs denominated $CH$-molecules.

**Definition 4.3** (***CH-molecules***) *Let $H_i$ be a hydrogen atom with atomic value $v_{H_i}$, and $C$ be a carbon atom with atomic value $v_C$. Also, let $M = (A^{(8)}, B_P)$ be a molecular unit formed with a set of atom units $a_k \in A^{(8)}$ and a set of polar covalent bonds $b^{pq} \in B_P$,* for all *$a_p \neq a_q$. Then, $M$ is said to be a $CH$*-molecule *if $H_i, C \in A^{(8)}$ and $b^{CH_i} \in B_P$, and there exists a molecular behavior $\varphi$ around some atom $a_c \in A^{(8)}$ due to some input $X$, such that, $\varphi = \varphi(X, d_c, v_{H_i})$. Moreover, $a_c = C$, and then $\varphi = v_C$.*

In other words, $CH$-molecules are built with one carbon atom and up to four hydrogen atoms. Actually, any $CH$-molecule has the carbon atom as the central one of $M$ with structure $M = (\{C, H_1, \dots, H_d\}, \{b^{CH_1}, \dots, b^{CH_d}\})$, where, $d$ is the degree of freedom of the carbon atom, such that, $0 < d \leq e_C$. Figure 4.2 shows

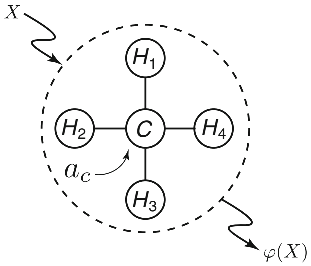


**Fig. 4.2** Structure of a $CH$-molecule

a $CH$-molecule. Through this book, $CH$-molecules with defined degree of freedom $d$ will be referred as $CH_d$.

In fact, the AHN-algorithm proposes the carbon atom as the central one because chemical hydrocarbon compounds change their behavior by adding or subtracting carbon atoms. Thus, carbon atoms have more complex functionality than hydrogen atoms.

It is remarkable to say that $CH$-molecules have at least a carbon and a hydrogen atom because of the definition of a molecular unit.

Once the structure of a $CH$-molecule is defined, the behavior $\varphi$ of that molecule has to be introduced. In the following propositions of $CH$-molecular behaviors, the set of input signals is restricted to a single input $x$.

**Proposition 4.1** **(*first model of CH-molecules*)** *Let $M$ be a $CH$-molecule with $a_c = C$. Also, let $\varphi$ be the behavior of molecule $M$ due to an input signal $x$ with $|x| < 1$. Then, the behavior $\varphi$ holds:*

$$v_{H_i} = h_i, \quad h_i \in \mathbb{C} \tag{4.1}$$

$$\varphi(x) = \sum_{i=1}^{d \le e_C} h_i \cdot x^i \tag{4.2}$$

*where, $d$ is the degree of freedom of the $C$ atom, $h_i$ are the valued constants of $H_i$ atoms, and $e_C$ is the number of valence electrons of $C$.*

The above Proposition 4.1 conditions the behavior of molecules to be in a polynomial form. In fact, the functions of molecules are expressed in a truncated power series fashion in order to perform an outer shell of atoms like in nature. Notice that, the radius of convergence in the open interval $|x| < 1$ guarantees a mathematical stability of $CH$-molecules.

*Remark 4.1* Consider the expanded form of $\varphi$ with $d = 4$ is defined as: $\varphi(x) = h_1 \cdot x + h_2 \cdot x^2 + h_3 \cdot x^3 + h_4 \cdot x^4$. Be careful in the order of the polynomial function because the values of hydrogen atoms are not interchangeable.

*Example 4.1* Calculate the behavior of the $CH_3$ molecule due to $x = 0.4$, if the values of its hydrogen atoms are: $h_1 = 2$, $h_2 = 6$, $h_3 = -5$. What is the value of the carbon atom at $x = 0.4$?

**Solution 4.1** Using (4.2),

$$\varphi(x) = (2)(0.4) + (6)(0.4)^2 + (-5)(0.4)^3 = 0.80 + 0.96 - 0.32 = 1.44$$

Then, the value of the behavior of $CH_3$ is 1.44. Actually, the value of $C$ at $x = 0.4$ is $v_C = 1.44$ because $v_C = \varphi(x)$.

*Example 4.2* Calculate the behavior of the $CH_3$ molecule due to $x = 0.4$, if the values of its hydrogen atoms are: $h_1 = 2, h_2 = -5, h_3 = 6$. Is this molecule the same as the molecule depicted in Example 4.1?

**Solution 4.2** Using (4.2),

$$\varphi(x) = (2)(0.4) + (-5)(0.4)^2 + (6)(0.4)^3 = 0.800 - 0.800 + 0.384 = 0.384$$

Then, the value of the behavior of $CH_3$ is 0.384. In fact, this molecule is not the same as the $CH_3$ in Example 4.1 because the order of the hydrogen atoms are not the same. It can be demonstrated comparing the results from the two molecules: $1.44 \neq 0.384$ at $x = 0.4$.

In order to compute behaviors of $CH$-molecules and using Remark 4.1, a convention for linking hydrogen atoms to the carbon atom is proposed. This is known as the counterclockwise convention and it states that hydrogen atoms are linked to the carbon atom starting from the top of the molecule and going on a counterclockwise direction. Figure 4.3 shows this convention.

Remark 4.1 shows that Proposition 4.1 has a disadvantage in the ordering of hydrogen atoms linked to the carbon atom in $CH$-molecules. In addition, in this particular arrangement hydrogen atoms do not offer any understanding in the meaning of these artificial hydrocarbon molecules. Thus, the model in Proposition 4.2 improves $CH$-molecules.

**Proposition 4.2** (***second model of*** $CH$***-molecules***) *Let $M$ be a $CH$-molecule with $a_c = C$. Also, let $\varphi$ be the behavior of molecule $M$ due to an input signal $x$ with $|x| < 1$. Then, the behavior $\varphi$ holds:*

$$v_{H_i} = h_i, \quad h_i \in \mathbb{C} \tag{4.3}$$

$$\varphi(x) = \prod_{i=1}^{d \leq e_C} (x - h_i) \tag{4.4}$$

*where, $d$ is the degree of freedom of the $C$ atom, $h_i$ are the valued constants of $H_i$ atoms, and $e_C$ is the number of valence electrons of $C$.*

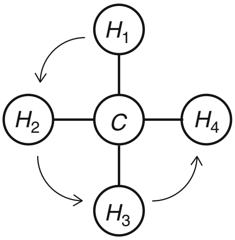


**Fig. 4.3** Counterclockwise convention of hydrogen atoms in $CH$-molecules

Notice that behaviors of hydrogen atoms associated to the carbon molecule represent the roots of the $CH$-molecule behaviors, increasing the meaning of these values. Additionally, the order of hydrogen atoms is unimportant since the commutative law of multiplication (in contrast to geometry interpretation of molecules in organic chemistry). Thus, artificial hydrocarbon networks algorithm uses the factored form of Proposition 4.2. In addition, that enhancement allows a more flexible structure of artificial hydrocarbon molecules in which hydrogen atoms can be associated freely without any order. Thus, there is no importance or priority in selecting any of these values, against to the previous model (Proposition 4.1). Other remarkable characteristics are that $\varphi$ is also normalized and hydrogen values remain unchangeable.

*Example 4.3* Calculate the behavior of the $CH_3$ molecule due to $x = 0.4$, if the values of its hydrogen atoms are: $h_1 = 0.2, h_2 = 0.6, h_3 = -0.5$. Use Proposition 4.2.

**Solution 4.3** Using (4.4),

$$\varphi(x) = (0.4 - 0.2)(0.4 - 0.6)(0.4 + 0.5) = (0.2)(-0.2)(0.9) = -0.036$$

Then, the value of the behavior of $CH_3$ is $-0.036$.

*Example 4.4* Calculate the behavior of the $CH_3$ molecule due to $x = 0.4$, if the values of its hydrogen atoms are: $h_1 = 0.2$, $h_2 = -0.5$, $h_3 = 0.6$. Is this molecule the same as the molecule depicted in Example 4.3?

**Solution 4.4** Using (4.4),

$$\varphi(x) = (0.4 - 0.2)(0.4 + 0.5)(0.4 - 0.6) = (0.2)(0.9)(-0.2) = -0.036$$

Then, the value of the behavior of $CH_3$ is $-0.036$. This molecule is the same as the molecule depicted in Example 4.3 because $-0.036 = -0.036$.

Furthermore, artificial hydrocarbon networks algorithm spans $CH$-molecules to form artificial hydrocarbon compounds. In that way, the functional group of artificial hydrocarbon networks is defined as follows.

**Definition 4.4** (***functional group***) *Let $H$ and $C$ be a hydrogen atom and a carbon atom, respectively, joined together by a polar covalent bond $b^{CH}$. Also, let $G$ be a molecule. Then, $G$ is said to be a* functional group of $CH$-molecules *if $G = (\{C, H\}, \{b^{CH}\})$.*

In Fig. 4.4 is shown the functional group of $CH$-molecules. In fact, because $G$ is equal to the molecule $CH$, $G$ is also known as the $CH$ functional group.

Definitions 4.3 and 4.4 can be used to derive $CH$-primitive molecules, unstable artificial hydrocarbon molecules that will be used to form complex molecules, so-called artificial hydrocarbon compounds. Thus, $CH$-primitive molecules are introduced in the following Lemma 4.1.

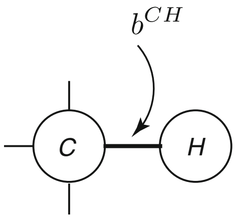


**Fig. 4.4** Functional group of *CH*-molecules

**Lemma 4.1** (***CH*-primitive molecules**) Let $G$ be a $CH$ functional group, and let $CH, CH_2, CH_3$ be three different $CH$-molecules spanned from $G$ with behaviors $\varphi_{CH}, \varphi_{CH_2}, \varphi_{CH_3}$ due to the set of inputs $X$. Then, there are at most three *CH-primitive molecules. No other molecules more than* $CH, CH_2, CH_3$ *are CH-primitive molecules.*

*Proof* Let $M$ be a primitive molecule as Definition 3.7. By induction, using Definition 4.3, $M$ can be $CH$, $CH_2$, $CH_3$ and $CH_4$, no other $CH$-molecules can be depicted because the degree of freedom $d$ is delimited to the interval $0 < d \leq e_C = 4$. But, $CH_4$ is a stable molecule, thus $CH_4$ is not a primitive molecule. Then, the only $CH$-primitive molecules are $CH$, $CH_2$, $CH_3$. □

#### 4.2.1.3 Compounds

Artificial hydrocarbon networks algorithm build complex molecules called artificial hydrocarbon compounds. In fact, those are made of $CH$-primitive molecules, as follows:

**Definition 4.5** (***hydrocarbon compounds***) *Let* $C = (\Omega, B_N)$ *be a compound consisting of primitive molecules* $\Omega$ *linked with a set of nonpolar covalent bonds* $B_N$*. Then,* $C$ *is said to be a* hydrocarbon compound *if* $\Omega = \{M_1, \ldots, M_k\}$ *is a set of CH-primitive molecules, spanned from the CH functional group, with molecular behaviors* $\varphi_1, \ldots, \varphi_k$*. Moreover,* $\psi$ *is the* behavior of $C$ due to any set of input signals $X$*, such that,* $\psi : \varphi_1 \times \cdots \times \varphi_k \to \mathbb{R}$.

The behavior of artificial hydrocarbon compounds $\psi$ is a mapping from a composite function of $CH$-molecular behaviors to a real value. In Sect. 4.2.2, different mappings are proposed.

#### 4.2.1.4 Mixtures

At last, artificial hydrocarbon networks algorithm uses the same definition of mixtures as described in artificial organic networks (see Definition 3.10). In that sense, AHN-algorithm has mixtures of hydrocarbon compounds with basis $\Phi = \{\psi_1, \ldots, \psi_k\}$. Figure 4.5 shows a mixture of artificial hydrocarbon compounds.

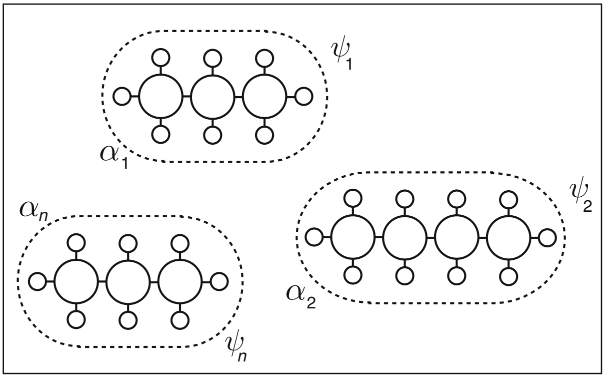


**Fig. 4.5** Example of a mixture of artificial hydrocarbon compounds

## 4.2.2 Interactions

In this section, an interaction among $CH$-primitive molecules is presented in order to determine the mechanisms of how artificial hydrocarbon compounds are built. Then, an interaction among these artificial hydrocrbon compounds is defined to mix them up. In practice, these two different interactions are implemented to perform the artificial hydrocarbon networks algorithm.

### 4.2.2.1 Nonpolar Covalent Bonds

As detailed in Chap. 3, covalent bonds are the interactions among atomic units. Actually, polar covalent bonds are present between two different atomic units. Thus, they are trivial in artificial hydrocarbon networks as noticed in Remark 4.2.

*Remark 4.1 Let $B_{CH}$ all possible combinatorial interactions between two atomic units in artificial hydrocarbon networks. Using Definition 4.1, these interactions are the covalent bonds $b^{CC}, b^{CH}, b^{HH} \in B_{CH}$. Moreover, using Definition 3.11, the only polar covalent bond in the set of all possible interactions $B_{CH}$ is $b^{CH}$. However, $b^{CH}$ defines $CH$-molecules as stated in Definition 4.3. Thus, $b^{CH}$ bonds do not transmit (map) information from atom to atom. Concluding, $b^{CH}$ polar covalent bonds are trivial in artificial hydrocarbon networks.*

In addition, the nonpolar covalent bond $b^{HH}$ is not possible because a molecule of two hydrogen atoms is not defined in artificial hydrocarbon networks. In other words, only molecules with one carbon atom and at least one hydrogen atom are totally defined (see Definition 4.3). At last, $b^{CC}$ nonpolar covalent bonds are the only ones that have behaviors associated, as introduced in Proposition 4.3.

**Proposition 4.3** (***first model of nonpolar covalent bonds***) *Let $b_k^{ij} \in B$ be a nonpolar covalent bond of a pair of molecules $M_i, M_j$ with behaviors $\varphi_i, \varphi_j$ due to an input $x$.*

*Let* $\Delta = \langle \delta_1, \ldots, \delta_5 \rangle$ *be a tuple of properties of* $b_k^{ij}$*. Then, the* behavior of nonpolar covalent bonds $\pi$ *holds:*

$$\pi(\varphi_i, \varphi_j, x) = \frac{w_i \varphi_i(x) + w_j \varphi_j(x)}{w_i + w_j} \tag{4.5}$$

*With,*

$$w_i(x) = \max\left(1 - \frac{\delta_1}{\delta_3} \left|\delta_2 - x\right|, 0\right) \tag{4.6}$$

$$w_j(x) = \max\left(1 - \frac{\delta_1}{\delta_5} \left|\delta_4 - x\right|, 0\right) \tag{4.7}$$

*where,* $\delta_1 \in \{1, 2, 3\}$ *is called the* order of bond*;* $\delta_2, \delta_4 \in \mathbb{R}$ *represents the* center of molecules $M_i, M_j$*, respectively; and* $\delta_3, \delta_5 \in \mathbb{R}$ *represents the* spread of molecules $M_i, M_j$*, respectively. Moreover, the behavior* $\Psi$ *of an artificial hydrocarbon compound* $C$ *consisting of molecules* $M_i, M_j$ *is equal to be behavior of the nonpolar covalent bond, such that,* $\Psi = \pi(\varphi_i, \varphi_j, x)$.

As noted, this model $\pi$ takes into account the center of molecules involved in the bond. Then, the centroid of molecules is calculated to perform the behavior of bonds. In fact, the centroid of molecules is derived from the weighted sum of molecules using the values $w_i, w_j$. In fact, those are triangular functions that smooth the molecular behaviors $\varphi_i, \varphi_j$.

*Example 4.5* Determine the behavior of compound $C$ in the input domain $x \in (-1, 1)$, if it is made of two $CH$-primitive molecules as Fig. 4.6 (hydrogen atoms represent the roots of molecules). Use the first model of nonpolar covalent bonds with properties $\Delta = \langle 1, -0.5, 2.0, 0.5, 2.0 \rangle$.

**Solution 4.5** Using Proposition 4.2, the molecular behaviors in $x \in (-1, 1)$ are shown in Fig. 4.7. In addition, Fig. 4.7 shows the weight values of the first model of nonpolar covalent bonds. Figure 4.8 shows the behavior of compound $C$. Remember that $\Psi = \pi(\varphi_i, \varphi_j, x)$.

The above Proposition 4.3 depends on several properties of nonpolar covalent bonds. In order to simplify it, the second model of nonpolar covalent bonds is introduced following. Notice that the behavior of these bonds is proposed to be

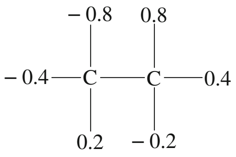


**Fig. 4.6** Artificial hydrocarbon compound $C$ of Example 4.5

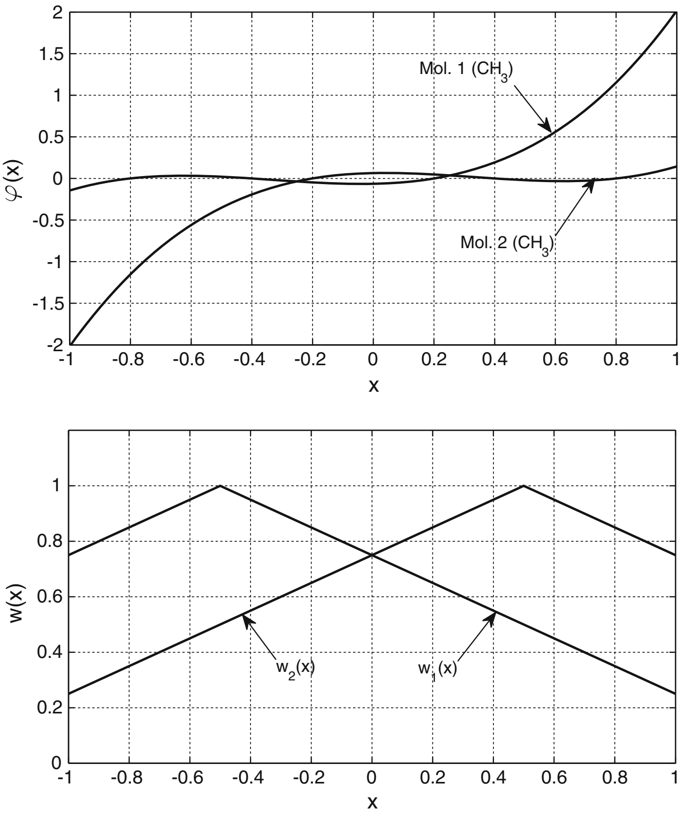


**Fig. 4.7** Molecular behaviors and weight values of Example 4.5

a nonlinear mapping, such that, the composite of molecular behaviors can capture the nonlinearities of them.

**Proposition 4.4** (***second model of nonpolar covalent bonds***) *Let $b_k^{ij} \in B$ be a nonpolar covalent bond of a pair of molecules $M_i, M_j$ with behaviors $\varphi_i, \varphi_j$ due to an input $x$. Then, the* behavior of nonpolar covalent bonds $\pi$ *holds:*

$$\pi(\varphi_i, \varphi_j, x) = \exp\left\{\varphi_i(x) + \varphi_j(x)\right\} \tag{4.8}$$

*Where,* $\exp(\cdot)$ *stands for the exponential function. Moreover, the behavior $\Psi$ of an artificial hydrocarbon compound $C$ consisting of molecules $M_i, M_j$ is equal to be behavior of the nonpolar covalent bond, such that, $\Psi = \pi(\varphi_i, \varphi_j, x)$.*

In contrast with the first model stated in Proposition 4.3, the behavior of bond $\pi$ in (4.8) is not characterized with $\Delta$ properties. This means that the order of bond or any other property is not taking into account to calculate $\pi$.

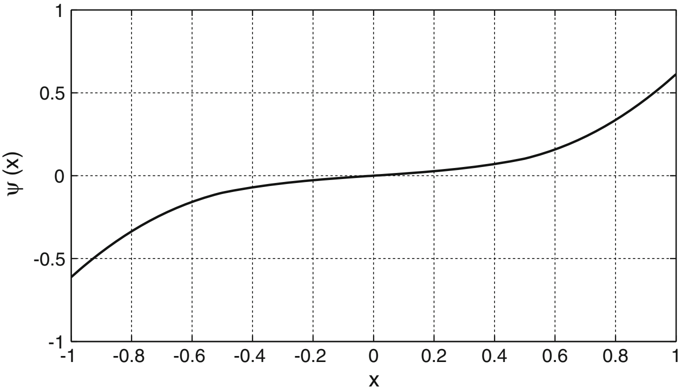


**Fig. 4.8** Behavior of compound $C$ of Example 4.5

However, in both cases, Propositions 4.3 and 4.4 does not follow any energy stabilization. In that sense, Proposition 4.5 defines the behavior of nonpolar covalent bonds using an energy function coming from the model of covalent bonds (refer to Sect. 2.2.1).

**Proposition 4.5** (***third model of nonpolar covalent bonds***) *Let $b_k^{ij} \in B$ be a nonpolar covalent bond of a pair of molecules $M_i, M_j$ with behaviors $\varphi_i, \varphi_j$ due to an input $x$. Let $\Delta = \langle \delta_1, \delta_2, \delta_3 \rangle$ be a tuple of properties of $b_k^{ij}$. Then, the* behavior of nonpolar covalent bonds $\pi$ *holds:*

$$\pi(\varphi_i, \varphi_j, x) = \delta_3 \left[1 - (\delta_1 \delta_2)^2\right] \exp\left\{-\frac{1}{2}(\delta_1 \delta_2)^2\right\}, \quad \pi \geq 0 \tag{4.9}$$

*With,*

$$\delta_2 : \theta_i \times \theta_j \to \mathbb{R} \tag{4.10}$$

*where, $\delta_1 \in \{1, 2, 3\}$ is called the* order of bond*; $\delta_2 \in \mathbb{R}$ with $\delta_2 \geq 0$ represents the* length of bond *and it is a metric on the parameters $\theta_i, \theta_j$ that characterize $\varphi_i, \varphi_j$; and, $\delta_3 \geq 0$ represents the* minimum energy of bond*. Moreover, the behavior $\Psi$ of an artificial hydrocarbon compound $C$ consisting of molecules $M_i, M_j$ is equal to the composite molecular behaviors, such that, $\Psi : \varphi_i(\theta_i) \times \varphi_j(\theta_j) \to \mathbb{R}$.*

Actually, (4.9) models (2.2). For instance, consider two $CH_3$ molecules with a minimum energy of bond $\delta_3 = 5$. Then, $\delta_1 = 1$ because these molecules are joined together with a simple bond. Figure 4.9 shows the behavior $\pi$ in an intermolecular distance domain $r \in [0, 10]$. Notice that the optimum parameter of $\delta_2 = \sqrt{3}$.

In that sense, (4.9) can be seen as an objective function for minimizing the behavior $\pi$ in terms of the intermolecular distance $r$. In other words, the optimum value of

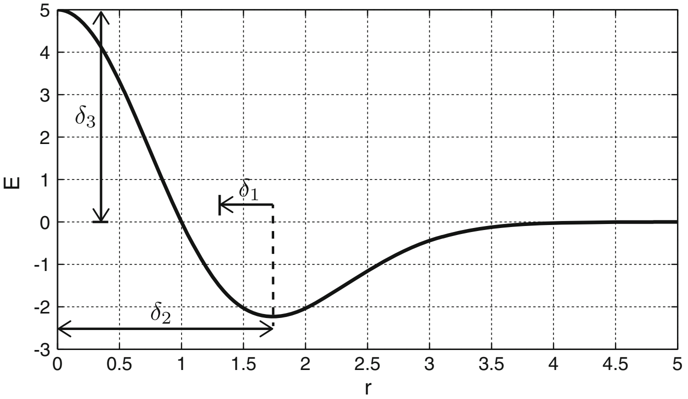


**Fig. 4.9** Behavior of a nonpolar covalent bond with $\Delta = \langle 1, \sqrt{3}, 5 \rangle$

the intermolecular distance $\frac{\partial \pi}{\partial r} = \delta_2$ represents the optimum distance at which two $CH$-molecules have to be separated.

#### 4.2.2.2 Chemical Balance Interaction

Artificial hydrocarbon networks algorithm uses the same definition of chemical balance interaction as described in artificial organic networks (see Definition 3.13). In that sense, the chemical balance interaction in AHN-algorithm finds the set of stoichiometric coefficients $\Lambda = \{\alpha_1, \ldots, \alpha_k\}$ of the mixture of artificial hydrocarbon compounds $\Gamma = \{C_1, \ldots, C_k\}$ with basis $\Phi = \{\psi_1, \ldots, \psi_k\}$.

### *4.2.3 The Algorithm*

Once the components and the interactions of artificial hydrocarbon networks algorithm are introduced, a proposition referent to the chemical rule in forming of compounds is necessary to finally state the mathematical formulation of artificial hydrocarbon networks.

#### 4.2.3.1 Chemical Rule in Forming of Compounds

As described in Sect. 3.4.1, the three-level energy model is required to propose the chemical rule that will be useful to form artificial hydrocarbon compounds. In fact, $CH$-primitive molecules have to follow that rule in order to determine how they bond.

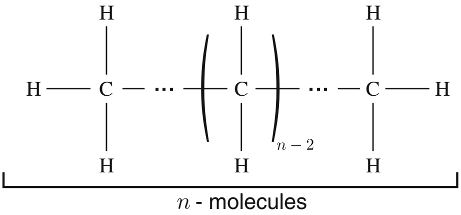


**Fig. 4.10** An artificial hydrocarbon compound presented as a saturated linear chain with $n \geq 2$

**Proposition 4.6** (***first chemical rule***) *Let $f$ be an algorithm and $n \geq 2$ be an integer number. Also, let $C = (\Omega, B_N)$ be a compound made of $CH$-primitive molecules $\Omega = \{M_1, \ldots, M_n\}$ linked with a set of nonpolar covalent bonds $B_N$. Then, $f$ returns the structure of $C$ as a* saturared linear chain*, such that, $M_1 = M_n = CH_3$ and $M_i = CH_2$ for all $i = 1, \ldots, n-2$.*

Notice that in Proposition 4.6, the algorithm $f$ that forms an artificial hydrocarbon compound $C = (\Omega, B_N)$ returns a linear chain of $CH$-primitive molecules with simple nonpolar covalent bonds as shown in Fig. 4.10. Actually, this linear chain is the easiest artificial hydrocarbon compound that can be formed. However, the order of bond does not vary, and in some cases this arbitrarity is not the optimal way to form compounds.

In order to derive a forming rule closer to chemical hydrocarbon compounds, basic observations to the trend of nonpolar covalent bonds in carbon-carbon bonds was realized. Tendencies of bonds in pairs of carbon atoms indicate that simple bonds are more frequent than double bonds, and the latter are more frequent than triple bonds. It follows from the fact that triple bonds need more energy (837 kJ/mol) than double bonds (611 kJ/mol) to be formed/broken; in the same way, double bonds have higher energy than simple bonds (350 kJ/mol) to be formed/broken, as summarized in Table 4.1. At last, the priority of occurrence of nonpolar covalent bonds in artificial hydrocarbon compounds can be written as (4.11); where, $b_i \succ b_j$ stands for the operation $b_i$ inhibits $b_j$, and $b_1, b_2, b_3$ represent simple, double and triple bonds, respectively.

$$b_1 \succ b_2 \succ b_3 \tag{4.11}$$

Then, consider any given system $\Sigma$ with an input signal $x$ and an output signal $y$. Also, consider that the system is equally splitted into $n$ different partitions $\Sigma_i$ for all $i = 1, \ldots, n$, such that, $\Sigma = \bigcup_i \Sigma_i$. Then, it is possible to capture the behavior of system $\Sigma_i$ with a $CH$-primitive molecule via its molecular behavior $\varphi_i$, as depicted in Fig. 4.11. In that sense, a proper algorithm to find the best $i$-th $CH$-primitive molecule to model $\Sigma_i$ is required.

Since there is no other information than input $x$ and output $y$ signals of system $\Sigma$, the approximation procedure using artificial hydrocarbon networks results difficult to realize. Thus, implicit information needs to be extracted from $\Sigma$, i.e. the energy of system. Based on signal theory, the energy $E$ of signal $y = \Sigma(x)$ in the interval $x \in [a, b]$ is calculated using (4.12); where, $\|g\|$ stands for the norm of $g$:

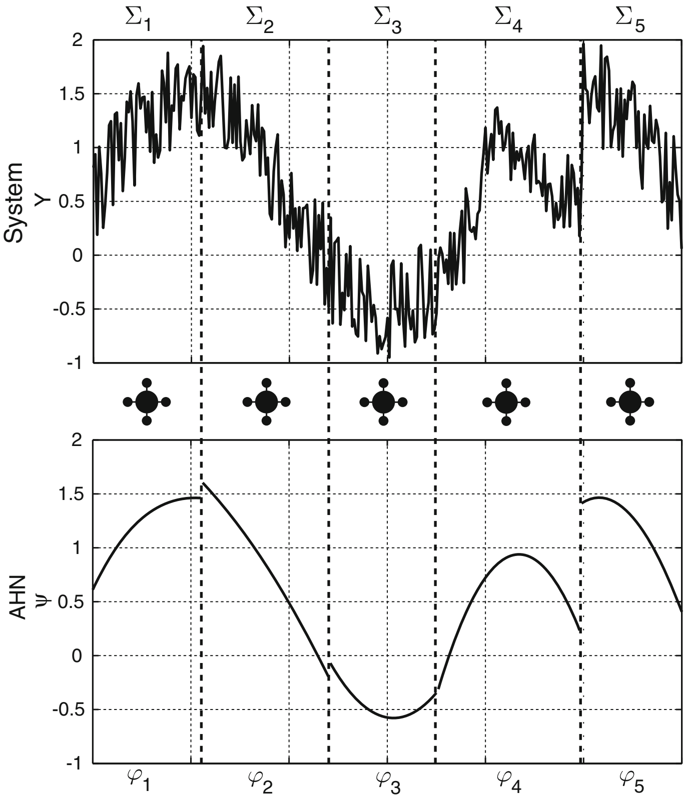


**Fig. 4.11** Example of a given system $\Sigma$ modeled with $n$ $CH$-primitive molecules

$$E(y) = \int_{a}^{b} \|y\|^2 dx, \quad x \in [a, b] \tag{4.12}$$

Thus using (4.12), the energy $E_i$ of a partition $\Sigma_i$ is expressed as (4.13); where, $x_i$ is the input and $y_i$ is the output signals of $\Sigma_i$ in the interval $x_i \in [L_1, L_2]$.

$$E_i = E(y_i) = \int_{L_1}^{L_2} \|y_i\|^2 dx, \quad x_i \in [L_1, L_2] \tag{4.13}$$

In fact, energy $E_i$ is used for selecting the $i$-th $CH$-primitive molecule in artificial hydrocarbon compounds, as explained below. Actually, the notion of energy can be used because behaviors of both molecules and nonpolar covalent bonds might be treated as energy signals.

**Table 4.2** Relationship among the order of bond $b_k$, the bond energy $E_k$ and the type of $CH$-primitive molecules

| Order of bond $b_k$ | Bond energy $E_k$ | Type of $CH$-molecules |
|---|---|---|
| 1 | 152 | $CH_3 - CH_3$ |
| 2 | 132 | $CH_2 = CH_2$ |
| 3 | 48 | $CH \equiv CH$ |

In that way, an experiment was run in order to prove a significant relationship between the energy $E_i$ and the $i$-th primitive molecule that best models $\Sigma_i$. That experiment is described in Appendix B in which the relationship energy-molecule is proved. In a nutshell, the experiment reveals that two $CH$-primitive molecules, let say $M_1$ and $M_2$, that models a given system $\Sigma$ form a nonpolar covalent bond $b_k$ with order $k$, if the energy of the system $E$ calculated as (4.12) is close to the bond energy $E_k$ as depicted in Table 4.2.

Thus, the two $CH$-primitive molecules $M_1, M_2$ can be selected by comparing the energy of system $E$ with the bond energy $E_k$ and selecting the best order of bond $k^*$ as expressed in (4.14). Once done, the type of the two $CH$-primitive molecules can be looked up in Table 4.2.

$$k^* = \arg\min_k \{|E - E_k|\}, \quad \forall k \in \{1, 2, 3\} \tag{4.14}$$

In that sense, the above experiment of two molecules can be extended to form an artificial hydrocarbon compound with $n$ $CH$-primitive molecules in order to model a system $\Sigma$ splitted into $n$ partitions $\Sigma_i$ for all $i = 1, \ldots, n$. Algorithm 4.1 computes that artificial hydrocarbon compound assuming that it is a saturated linear chain. The inhibition rule (4.11) is also used. Notice that from Lemma 4.1, a $CH$-primitive molecule cannot be a single carbon atom without hydrogen atoms; therefore, triple bonds can only appear when there are artificial hydrocarbon compounds with fixed length $n = 2$, e.g. of the form $CH \equiv CH$, as reflected in Algorithm 4.1.

Concluding, the following Proposition 4.7 raises the usage of Algorithm 4.1 to be the chemical rule applied on artificial hydrocarbon networks algorithm to form the structure of artificial hydrocarbon compounds.

**Proposition 4.7** (***second chemical rule***) *Let $\Sigma = (x, y)$ be any given system with input signal $x$ and output signal $y$, and $n \geq 2$ be an integer number. In addition, let $f$ be Algorithm 4.1 with inputs $\Sigma$ and $n$. Also, let $C = (\Omega, B_N)$ be a compound made of CH-primitive molecules $\Omega = \{M_1, \ldots, M_n\}$ linked with a set of nonpolar covalent bonds $B_N$. Then, $f$ returns the structure of $C$ as a* saturared linear chain, *such that, $C$ is close related to the energy of $\Sigma$.*

*Example 4.6* Using Algorithm 4.1, build an artificial hydrocarbon compound $C$ with three $CH$-primitive molecules to capture the energy of the system $\Sigma$ depicted in Fig. 4.12.

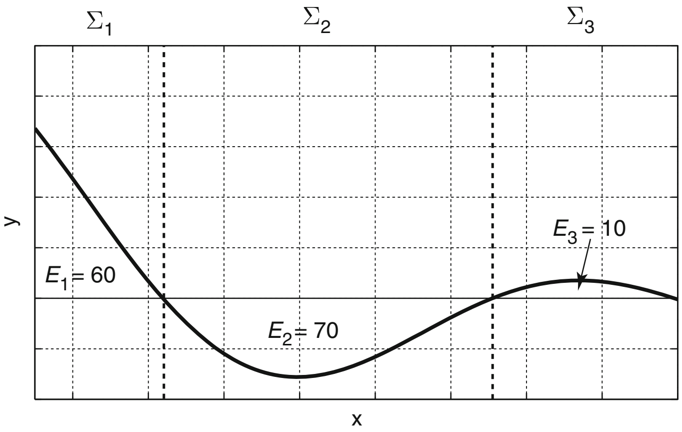


**Fig. 4.12** System $\Sigma$ split into 3 partitions, for Example 4.6

**Algorithm 4.1** CREATE-COMPOUND($\Sigma$, $n$): Compound formation based on bond energy.

**Input:** the system $\Sigma = (x, y)$ and the number of molecules $n \geq 2$.
**Output:** the structure of hydrocarbon compound $C$.

```
$k_0^* = 0$
split $\Sigma$ into $n$ equal partitions $\Sigma_i$
for $j = 1 : (n - 1)$ do
    $E_j \leftarrow$ energy of $\Sigma_j$ using (4.13)
    $E_{j+1} \leftarrow$ energy of $\Sigma_{j+1}$ using (4.13)
    $E = E_j + E_{j+1}$
    $k_j^* \leftarrow$ using (4.14) and $E$
    $M_j, M_{j+1} \leftarrow$ select two $CH$-primitive molecules using $k_j^*$ and Table 4.2
    while $(k_j^* + k_{j-1}^*) \geq 4$ do
        if $k_{j-1}^* == 1$
            $k_j^* = k_j^* - 1$
            $M_j, M_{j+1} \leftarrow$ select two $CH$-primitive molecules using $k_j^*$ and Table 4.2
        else
            $k_{j-1}^* = k_{j-1}^* - 1$
            $M_{j-1}, M_j \leftarrow$ select two $CH$-primitive molecules using $k_{j-1}^*$ and Table 4.2
        end-if
    end-while
end-for
$\Omega \leftarrow \{M_1, \ldots, M_n\}$
$B_N \leftarrow \{b_1, \ldots, b_i, \ldots, b_{n-1}\}$, $b_i$ with order $k_i^*$ for all $i = 1, \ldots, n - 1$
$C = (\Omega, B_N)$
return $C$
```

**Solution 4.6** Let $n = 3$ be the number of molecules in the compound, $k^* = (k_0^*, k_1^*, k_2^*)$ be the set of the best order of bonds in the compound, and $\Omega$ be the set of $CH$-primitive molecules forming the compound $C$. Let initialize $k^* = (0, 0, 0)$, and $\Omega = \emptyset$. Then, the system $\Sigma$ has to be split getting three partitions $\Sigma_i$ as shown in Fig. 4.12.

For $j = 1$:
$E = E_1 + E_2 = 60 + 70 = 130$.
$k_1^* = 2$. Then, $k^* = (0, 2, 0)$.
$M_1, M_2$ are $CH_2$ molecules from Table 4.2. Then, $\Omega = (CH_2, CH_2)$.
Since $k_1^* + k_0^* = 2 + 0 < 4$, this ends the first iteration.

For $j = 2$:
$E = E_2 + E_3 = 70 + 10 = 80$.
$k_2^* = 3$. Then, $k^* = (0, 2, 3)$.
$M_2, M_3$ are $CH$ molecules from Table 4.2. Then, $\Omega = (CH_2, CH, CH)$.
Since $k_2^* + k_1^* = 3 + 2 > 4$, then:
$k_1^* = 2 \neq 1$, then $k_1^* = k_1^* - 1 = 2 - 1 = 1$.
Refreshing, $k^* = (0, 1, 3)$ and $\Omega = (CH_3, CH_3, CH)$.
Since $k_2^* + k_1^* = 3 + 1 = 4$, then:
$k_1^* = 1$, then $k_2^* = k_2^* - 1 = 3 - 1 = 2$.
Refreshing, $k^* = (0, 1, 2)$ and $\Omega = (CH_3, CH, CH_2)$.
Since $k_2^* + k_1^* = 2 + 1 < 4$, this ends the second iteration.

Finally, $C = (\Omega, B_N)$ with $\Omega = (CH_3, CH, CH_2)$ and $B_N = k^* \backslash \{k_0^*\} = (1, 2)$.

Figure 4.13 shows the resultant artificial hydrocarbon compound.

As noted, Proposition 4.6 forms a saturated, linear artificial hydrocarbon compound without taking into account the system to be modeled. To improve it, Proposition 4.7 uses Algorithm 4.1 to link the energy of the system to be modeled with the structure of an artificial hydrocarbon compound. However, in both cases, these propositions generate a compound; but the system might be too complex that one hydrocarbon compound could not be sufficient to model it. Then, another algorithm has to be designed in order to form mixtures of artificial hydrocarbon compounds, reaching the third level in the three-level energy model described in the artificial organic networks technique.

Therefore, consider a system $\Sigma = (x, y)$ with input $x$ and output $y$. Let also consider an artificial hydrocarbon compound $C = (\Omega, B_N)$ with $n$ $CH$-primitive molecules with behaviors $\varphi_i$ for all $i = 1, \ldots, n$. Finally, let $\psi$ be the behavior of $C$, such that, $\psi = \psi(\varphi_1, \ldots, \varphi_n, x)$. Then, the approximation error $e_C$ of compound $C$

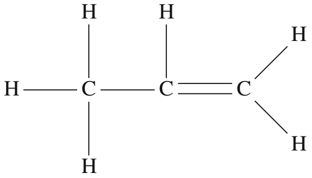


**Fig. 4.13** Artificial hydrocarbon compound capturing the energy of $\Sigma$ using Algorithm 4.1

between the output $y$ and the its behavior $\psi$ can be computed as (4.15); where, $\|g\|$ stands for the norm of $g$.

$$e_C = \|y - \psi\| \tag{4.15}$$

On the other hand, suppose that there exists a small positive constant $\varepsilon$. If $e_C > \varepsilon$, then a new artificial hydrocarbon compound $C$ of $n$ $CH$-primitive molecules has to be formed; but now, this $C$ will model the residue $e_C$ of the system. This process iterates $c$ times until $e_C \leq \varepsilon$ or $c$ reaches a maximum number of compounds $c_{max}$. At the end, there will be $c$ compounds with $n$ $CH$-primitive molecules each. The procedure is summarized in Algorithm 4.2 and Proposition 4.8 states its application.

**Algorithm 4.2** CREATE-MIXTURE($\Sigma$, $n$, $c_{max}$, $\varepsilon$): Mixture formation based on the approximation error.

**Input:** the system $\Sigma = (x, y)$, the number of molecules $n \geq 2$, the maximum number of compounds $c_{max} > 0$ and the small tolerance value $\varepsilon > 0$.
**Output:** the structure of mixture $\Gamma$.

```
$i = 0$
$R = \Sigma$
while ($\|R\| > \varepsilon$) and ($i < c_{max}$) do
    $i = i + 1$
    $C_i$ =CREATE-COMPOUND($R$, $n$)
    $R \leftarrow$ using (4.15)
end-while
$\Gamma \leftarrow \{C_1, \ldots, C_i\}$
return $\Gamma$
```

**Proposition 4.8** (***forming of mixtures***) *Let $\Sigma = (x, y)$ be any given system with input signal $x$ and output signal $y$, $n \geq 2$ and $c_{max}$ be two positive integer numbers, and $\varepsilon$ be a small positive integer. In addition, let $f$ be Algorithm 4.2 with inputs $\Sigma$,$n$, $c_{max}$ and $\varepsilon$. Also, let $\Gamma = \{C_1, \ldots, C_c\}$ be a mixture of $c$ artificial hydrocarbon compounds $C_i$ for all $i = 1, \ldots, c$. Then, $f$ returns the structure of $\Gamma$ as a* mixture of artificial hydrocarbon compounds *that is close related to the energy of $\Sigma$. Moreover, $f$ satisfies the three-level model energy.*

*Example 4.7* Using Algorithm 4.2, build a mixture of compounds $\Gamma$ with three $CH$-primitive molecules per compound and a maximum of 2 compounds, to capture the energy of the system $\Sigma = \frac{2}{9}x^5 + \frac{1}{9}x^3 - \frac{2}{3}$ for all $x \in [-1, 1]$ as depicted in Fig. 4.14. Consider a tolerance value of $\varepsilon = 0.001$.

**Solution 4.7** Let $n = 3$ be the number of molecules per compound and $c_{max} = 2$ be the maximum number of compounds. Also, let $\Gamma$ be the set of compounds $C_i$. Let initialize the residue $R = \Sigma$. Then, the residue $\Sigma$ has to be split getting three partitions $\Sigma_i$ as shown in Fig. 4.14.

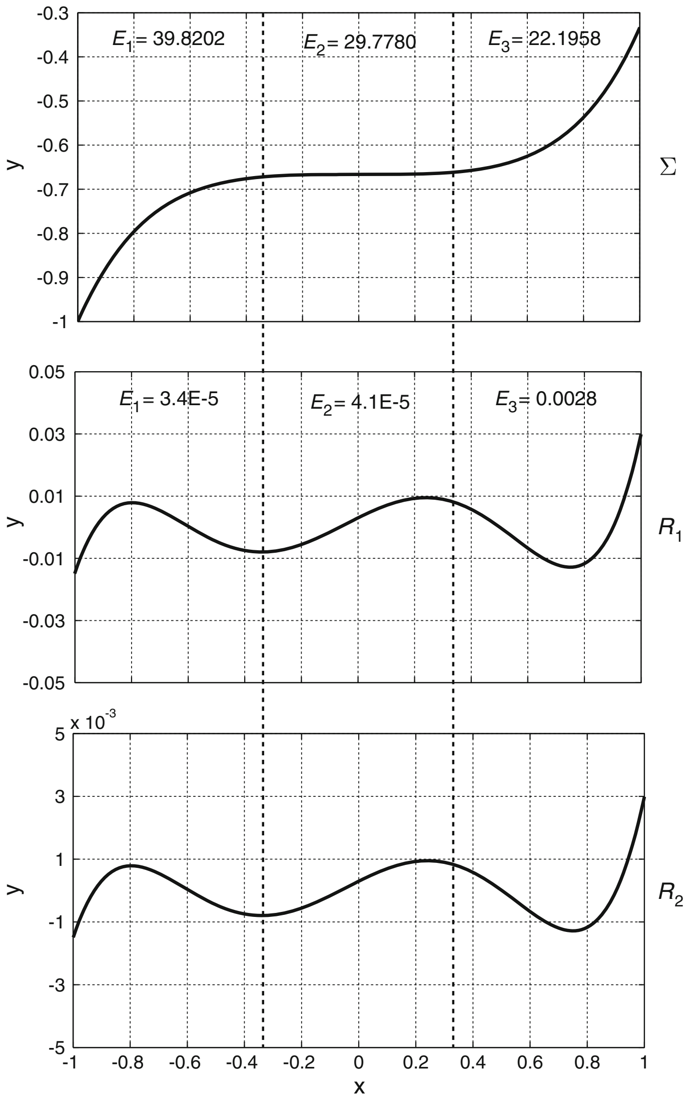


**Fig. 4.14** System $\Sigma$ split into 3 partitions with two residues $R$, for Example 4.7

For $i = 0$:

$i = 1$.

$C_1 = (\{CH_3, CH, CH_2\}, \{1, 2\})$ using Algorithm 4.1.

$R = \|R - C_1\|$ is shown in Fig. 4.14; obtained later from Proposition 4.9.

$\|R\| = 0.0029$ using the L2-norm of $R$.

$\|R\| > \varepsilon$ and $i < c_{max}$.

```
For $i = 1$:
    $i = 2$.
    $C_2 = (\{CH_3, CH, CH_2\}, \{1, 2\})$ using Algorithm 4.1.
    $R = \|R - C_2\|$ is shown in Fig. 4.14; obtained later from Proposition 4.9.
    $\|R\| = 0.0029$ using the L2-norm of $R$.
    $\|R\| > \varepsilon$ and $i = c_{max}$, this ends the loop.
```

Finally, $\Gamma = (C_1, C_2)$. Figure 4.15 shows the resultant mixture of compounds.

#### 4.2.3.2 Basic Algorithm of Artificial Hydrocarbon Networks

Notice that (4.15) requires a valued-function $\psi$. Moreover, $\psi$ depends on $CH$-primitive molecular behaviors $\varphi_i$ for all $i = 1, \ldots, n$. For instance, consider (4.4) in Proposition 4.2 to define molecular behaviors $\varphi_i$. Note that (4.4) is defined via hydrogen-atom values of the form as (4.3). Then, suppose that $\psi$ is defined as 4.16,

$$\psi(\varphi_1, \ldots, \varphi_n, x) = \begin{cases} \varphi_1(x) & L_0 \le x < L_1 \\ \cdots & \cdots \\ \varphi_n(x) & L_{n-1} \le x \le L_n \end{cases} \tag{4.16}$$

where, the set of $L_j$ for all $j = 0, \ldots, n$ represents the bounds of the input domain in molecular behaviors with lower bound $L_0 = a$ and upper bound $L_n = b$ in the overall input domain $x \in [a, b]$. Particularly to Algorithm 4.1, these bounds are set by partitions $\Sigma_i$, such that, they follow (4.17).

$$L_j - L_{j-1} = \frac{b-a}{n}, \quad j = 1, \ldots, n \tag{4.17}$$

At last, it should be an algorithm for finding the collection of hydrogen values that parameterizes all $\varphi_i$ functions and obtain a valued-function $\psi$ due to input $x$, as follows.

Suppose that each molecular behavior $\varphi_i = \varphi_i\,(h_{i1}, \ldots, h_{id}, x)$, where the set of $h_{im}$ is the set of hydrogen values associated to the $i$-th $CH$-primitive molecule and $d$ is the degree of freedom of that molecule, models a partition $\Sigma_i = (x_i, y_i)$ of the overall system $\Sigma$. In order to do that, an optimization process is proposed to find the values of hydrogen atoms $h_{im}$ using the objective function $E_i$ in (4.18); where, $x_{ik} \in x_i$ is the $k$-th sample of $x_i$ for all $k = 1, \ldots, q_i$; $y_{ik} \in y_i$ is the $k$-th sample of $y_i$ for all $k = 1, \ldots, q_i$; and $q_i$ is the number of samples in the partition $\Sigma_i$.

$$E_i = \frac{1}{2} \sum_{k=1}^{q_i} (y_{ik} - \varphi_i(x_{ik}))^2 \tag{4.18}$$

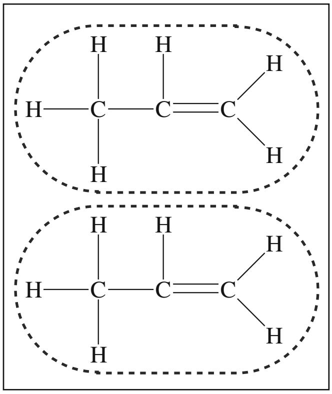


**Fig. 4.15** Mixture of compounds capturing the energy of $\Sigma$ using Algorithm 4.1

Thus, using the least squares estimates method (LSE) with objective function (4.18) and $\varphi_i = \varphi_i\,(h_{i1}, \ldots, h_{id}, x)$, the set of all $h_{im}$ is obtained. In that sense, $\psi$ is now a valued-function with parameters $h_{im}$, and (4.15) can be computed.

On the other hand, Algorithm 4.2 does not calculate the behavior $S(x)$ of the final mixture $\Gamma$. Since the structure $\Gamma$ has to model $\Sigma$, (4.19) has to be minimized; where, $E_\Sigma$ stands for the objective function between the system and the model; $x_k \in x$ is the $k$-th sample of $x$ for all $k = 1, \ldots, q$; $y_k \in y$ is the $k$-th sample of $y$ for all $k = 1, \ldots, q$; $q$ is the total number of samples in $\Sigma$; $\psi_i$ is the $i$-th compound behavior of $C_i$ in the mixture $\Gamma$ of the form as (4.16); $\alpha_i$ is the $i$-th stoichiometric coefficient $\alpha_i \in \Lambda$; and $c$ is the number of compounds in the mixture.

$$E_\Sigma = \frac{1}{2}\sum_{k=1}^{q}\left(y_k - \sum_{i=1}^{c}\alpha_i\psi_i(x_k)\right)^2 \tag{4.19}$$

Again, the least squares estimates method with objective function (4.19) can be applied in order to obtain the set of stoichiometric coefficients $\Lambda$.

In addition, Algorithm 4.2 can be modified to determine both the structure and the behavior of the mixture of artificial hydrocarbon compounds as depicted in Algorithm 4.3. Actually, this algorithm builds a simple artificial hydrocarbon network as stated in Proposition 4.9.

**Proposition 4.9** (***basic AHN-algorithm***) *Let $\Sigma = (x, y)$ be any given system with input signal $x$ and output signal $y$, $n \geq 2$ and $c_{max}$ be two positive integer numbers, and $\varepsilon$ be a small positive integer. In addition, let $f$ be Algorithm 4.3 with inputs $\Sigma, n, c_{max}$ and $\varepsilon$. Also, let $\Gamma = \{C_1, \ldots, C_c\}$ be a mixture of $c$ artificial hydrocarbon compounds $C_i$ for all $i = 1, \ldots, c$, let $H$ be the set of hydrogen values in molecular behaviors of each $C_i$, and let $\Lambda$ be the set of stoichiometric coefficients that weights the mixture of compounds $\Gamma$. Then, $f$ satisfies the three-level model energy and returns the structure of $\Gamma$ as a mixture of artificial hydrocarbon compounds that is*

**Algorithm 4.3** SIMPLE-AHN($\Sigma, n, c_{max}, \varepsilon$): Basic algorithm of artificial hydrocarbon networks.

**Input:** the system $\Sigma = (x, y)$, the number of molecules $n \geq 2$, the maximum number of compounds $c_{max} > 0$ and the small tolerance value $\varepsilon > 0$.

**Output:** the structure of mixture $\Gamma$, the set of hydrogen values $H$, and the set of stoichiometric coefficients $\Lambda$.

```
$i = 0$
$R = \Sigma$
while ($R > \varepsilon$) and ($i < c_{max}$) do
    $i = i + 1$
    $C_i$ =CREATE-COMPOUND($R, n$)
    $H_i = \{h_{i1}, \ldots, h_{id}\}$ minimizing (4.18) using LSE and information of $C_i$
    $R \leftarrow$ using (4.15)
end-while
$\Gamma \leftarrow \{C_1, \ldots, C_i\}$
$H \leftarrow \{H_1, \ldots, H_i\}$
$\Lambda \leftarrow$ minimizing (4.19) using LSE.
return $\Gamma, H$ and $\Lambda$
```

*close related to the energy of $\Sigma$ and completely determines the parameters of $\Gamma$ in $H$ and $\Lambda$. Moreover, $f$ approximates $\Sigma$.*

*Example 4.8* Using Algorithm 4.3, build a mixture of compounds $\Gamma$ with three $CH$-primitive molecules per compound and a maximum of 2 compounds, to capture the energy of the system $\Sigma = \frac{2}{9}x^5 + \frac{1}{9}x^3 - \frac{2}{3}$ for all $x \in [-1, 1]$ as depicted in Fig. 4.14. Consider a tolerance value of $\varepsilon = 0.001$.

**Solution 4.8** Let $n = 3$ be the number of molecules per compound and $c_{max} = 2$ be the maximum number of compounds. Also, let $\Gamma$ be the set of compounds $C_i$. Let initialize the residue $R = \Sigma$. Then, the residue $\Sigma$ has to be split getting three partitions $\Sigma_i$ as shown in Fig. 4.14.

For $i = 0$:

$i = 1$.

$C_1 = (\{CH_3, CH, CH_2\}, \{1, 2\})$ using Algorithm 4.1.

$$\psi_1(x) = \begin{cases} \varphi_1(x) = (x - h_{11})(x - h_{12})(x - h_{13}) & -1 \leq x < -\frac{1}{3} \\ \varphi_2(x) = (x - h_{21}) & -\frac{1}{3} \leq x < \frac{1}{3} \\ \varphi_3(x) = (x - h_{31})(x - h_{32}) & \frac{1}{3} \leq x \leq 1 \end{cases}$$ using (4.16) and $C_1$.

$H_1 = \{0.42, -0.79 + 0.76i, -0.79 - 0.76i, 76.84, 1.26, -0.39\}$ using $\psi_1$ with least squares estimates.

$R = \|R - \psi_1\|$ is shown in Fig. 4.14.

$\|R\| = 0.0029$ using the L2-norm of $R$.

$\|R\| > \varepsilon$ and $i < c_{max}$.

For $i = 1$:

$i = 2$.

$C_2 = (\{CH_3, CH, CH_2\}, \{1, 2\})$ using Algorithm 4.1.

$$\psi_2(x) = \begin{cases} \varphi_1(x) = (x - h_{11})(x - h_{12})(x - h_{13}) & -1 \leq x < -\frac{1}{3} \\ \varphi_2(x) = (x - h_{21}) & -\frac{1}{3} \leq x < \frac{1}{3} \\ \varphi_3(x) = (x - h_{31})(x - h_{32}) & \frac{1}{3} \leq x \leq 1 \end{cases}$$ using (4.16) and $C_2$.

$H_2 = \{-0.31, -0.78 + 0.16i, -0.78 - 0.16i, 0.14, 0.95, 0.50\}$ using $\psi_2$ with least squares estimates.

$R = \|R - \psi_2\|$ is shown in Fig. 4.14.

$\|R\| = 0.0029$ using the L2-norm of $R$.

$\|R\| > \varepsilon$ and $i = c_{max}$, this ends the loop.

Finally:
$\Gamma = (C_1, C_2)$
$H = (H_1, H_2)$
$\Lambda = (1.0, -0.99)$ using $S(x) = \alpha_1\psi_1 + \alpha_2\psi_2$ with least squares estimates.

Figure 4.16 shows the resultant mixture of compounds.

As noted so far, Algorithm 4.3 sets the input domain of $CH$-primitive molecular behaviors $\varphi_i$ constrained to (4.17). In other words, $CH$-primitive molecules in hydrocarbon compounds are distributed equally over the input domain $x \in [a, b]$. In fact, this assumption is not optimal because some molecules could act over bigger regions while others are preferred to act in smaller regions. To solve this problem, Proposition 4.5 is used for finding the optimal values of length $r_j$ between molecules.

For instance, consider that an artificial hydrocarbon compound has two $CH$-primitive molecules $M_1$ and $M_2$ with molecular behaviors $\varphi_1$ and $\varphi_2$, respectively. Then, this compound will model a system $\Sigma = (x, y)$ with partitions $\Sigma_1(x_1, y_1)$ and $\Sigma_2 = (x_2, y_2)$ splitted using an arbitrary value of length $r \in x$. Also, suppose that the objective function (4.18) is applied to obtain the hydrogen values associated to behaviors $\varphi_1, \varphi_2$ using $\Sigma_1$, $\Sigma_2$, resulting in two error function values $E_1$ and $E_2$.

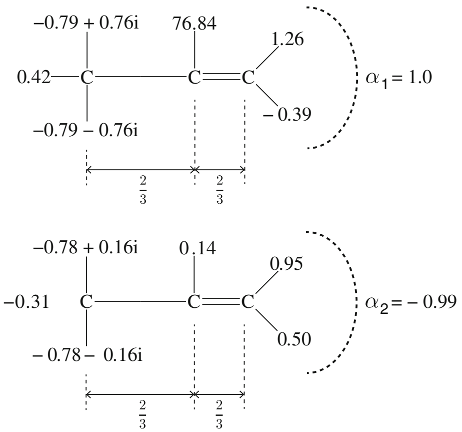


**Fig. 4.16** Mixture of compounds capturing the energy of $\Sigma$ using Algorithm 4.3

In that sense, the problem is to find the optimal value of $r$, such that, $E_1$ and $E_2$ be minimized and $E_1 = E_2$.

Let $\pi$ be the behavior of the nonpolar covalent bond $b^{12}$, between molecules $M_1, M_2$, as described in (4.9). Then, $\pi$ can be used as the objetive function to find the optimal value of $r$ guiding the process in the search space. Thus, proposing property $\delta_2 = \frac{r}{E_m}$, $\pi$ is expressed as (4.20); where, $E_r$ is the objective function to be minimized, $\delta_1$ is selected depending on the order of bond between $M_1, M_2$, $r$ is the optimal intermolecular distance (length of bond), and $E_m = E_1 - E_2$ is the error between the two error values $E_1, E_2$. It is remarkable to say that $\delta_2$ measures the length of bond $r$ related to the quantity $E_m$. Then, the larger the quantity $E_m$ is, the shorter the length of bond needs to be.

$$E_r = \pi = \delta_3 \left[ 1 - \left( \frac{\delta_1 r}{E_m} \right)^2 \right] \exp \left\{ -\frac{1}{2} \left( \frac{\delta_1 r}{E_m} \right)^2 \right\}, \quad \delta_1 \in \{1, 2, 3\} \tag{4.20}$$

Since (4.20) depends on the error functions $E_1, E_2$ calculated from the current partitions due to $r$, and $r$ is updated from (4.20), the gradient descent method is used as an strategy to optimize the length of bond $r$. There is the updating rule of $r$ in (4.21); where, $r_{t+1}$ is the future value of $r$, $r_t$ is the current value of $r$, and $\Delta r_t$ is the current change in length of bond $r$ defined as (4.22) with step size $0 < \eta < 1$, and current error function values $E_{1t}, E_{2t}$.

$$r_{t+1} = r_t + \Delta r_t \tag{4.21}$$

$$\Delta r_t = -\eta \left( E_{1t} - E_{2t} \right) \tag{4.22}$$

In the general case where there are $n$ $CH$-primitive molecules, there are $n - 1$ nonpolar covalent bonds of the form $b^{jk}$ between two molecules $M_j, M_k$ with $j \neq k$. Thus, for each nonpolar covalent bond, there is a length of bond $r^{jk}$ associated to it. In particular, each $r^{jk}$ is found by applying (4.21). Interestingly, these lengths of bonds follow the arrangement of bounds (4.23) in the overall input domain $x \in [a, b]$. However, notice that $r^{n-1,n} = b$; thus, it can be set initially.

$$\left\{ \left[ a, r^{12} \right], \left[ r^{12}, r^{23} \right], \left[ r^{23}, r^{34} \right], ..., \left[ r^{n-2,n-1}, b \right] \right\} \tag{4.23}$$

The above discussion is summarized in Algorithm 4.4 which obtains the optimal values of $r^{jk}$ and generating the optimal artificial hydrocarbon compound that uses these bounds.

*Example 4.9* Using Algorithm 4.4, optimize the artificial hydrocarbon compound $C = (\{CH_3, CH_2, CH_3\}, \{1, 1\})$ that captures the energy of the system $\Sigma = \sin(\pi x + 0.75)$ for all $x \in [0, 2]$. Consider a step size of $\eta = 0.1$.

**Solution 4.9** Let $n = 3$ be the number of molecules in $C$. Also, let $a$ and $b$ the lower and upper bounds in the input domain, such that, $a = 0$ and $b = 2$. Then,

**Algorithm 4.4** OPTIMUM-COMPOUND($\Sigma, C, n, \eta$): Optimal artificial hydrocarbon compound.

**Input:** the system $\Sigma = (x, y)$, the compound $C$ to be optimized, the number of molecules $n \geq 2$ and the learning rate $0 < \eta < 1$.
**Output:** the set of hydrogen values $H$ and the set of intermolecular distances $\Pi$.

```
determine the initial values of $r^{jk} \in \Pi$ using (4.17)
while stop condition is not reached do
    split $\Sigma$ into $n$ partitions $\Sigma_i$ using $r^{jk}$
    for each $\Sigma_i$ do
        $H_i = \{h_{i1}, \ldots, h_{id}\}$ minimizing (4.18) using LSE and information of $C$
        $E_i \leftarrow$ error function value using (4.18)
    end-for
    update all $r^{jk} \in \Pi$ applying rule (4.21) and (4.22) with step size $\eta$
end-while
$H \leftarrow \{H_1, \ldots, H_n\}$
return $H$ and $\Pi$
```

Algorithm 4.4 requires to initialize the intermolecular distances $r^{jk} \in \Pi$ such that (4.17) holds, as written in (4.24). Moreover, the bounds of molecules $L$ are computed as (4.25), giving $L = (0.0, 0.66, 1.33, 2.0)$.

$$r_i = \frac{b-a}{n} = \frac{2}{3}, \quad \forall i = 1, \ldots, n-1 \tag{4.24}$$

$$\begin{cases} L_0 = a \\ L_k = r_{k-1} + L_{k-1}, \qquad \forall k = 1, \ldots, n-1 \\ L_n = b \end{cases} \tag{4.25}$$

Since the following process is iterative, consider the first iteration at $t = 0$:

Split $\Sigma$ in $n = 3$ partitions using bounds $L$ as (4.26).

$$\Sigma_i(x_k, y_k), \quad \forall k \; L_{i-1} \leq x_k \leq L_i, \quad \forall i = 1, \ldots, n \tag{4.26}$$

Figure 4.17 (first) shows the partitions at $t = 0$.

For $\Sigma_1$:
$H_1 = \{-5.47, 1.34, 0.22\}$ using molecule $CH_3$ with LSE.
$E_1 = 0.0083$ using (4.18).
For $\Sigma_2$:
$H_2 = \{1.32, 0.53\}$ using molecule $CH_2$ with LSE.
$E_2 = 0.2694$ using (4.18).
For $\Sigma_3$:
$H_3 = \{1.06, 0.57, 0.39\}$ using molecule $CH_3$ with LSE.
$E_3 = 0.0008$ using (4.18).
Update intermolecular distances using (4.21) and (4.22):

$$
\begin{aligned}
&r_1 = r_1 - \eta\,(E_1 - E_2) = \tfrac{2}{3} - 0.1\,(0.0083 - 0.2694) = 0.6928\\
&r_2 = r_2 - \eta\,(E_2 - E_3) = \tfrac{2}{3} - 0.1\,(0.2694 - 0.0008) = 0.6398\\
&L = (0.0, 0.6928, 1.3326, 2.0)
\end{aligned}
$$

Figure 4.17 (second) shows the partition at $t = 1$, and the following frames in Fig. 4.17 represent the evolution of compound $C$ until a stop criterion is reached, e.g. a maximum number of iterations.

Finally:
$H = (H_1, H_2, H_3)$
$\Pi = (r_1, r_2)$

To this end, Algorithm 4.4 is applied into Algorithm 4.3 obtaining the modified version of the basic artificial hydrocarbon network algorithm (Algorithm 4.5) as stated in Proposition 1.1.

**Algorithm 4.5** SIMPLE-AHN($\Sigma$, $n$, $c_{max}$, $\varepsilon$, $\eta$): Modified algorithm of artificial hydrocarbon networks.

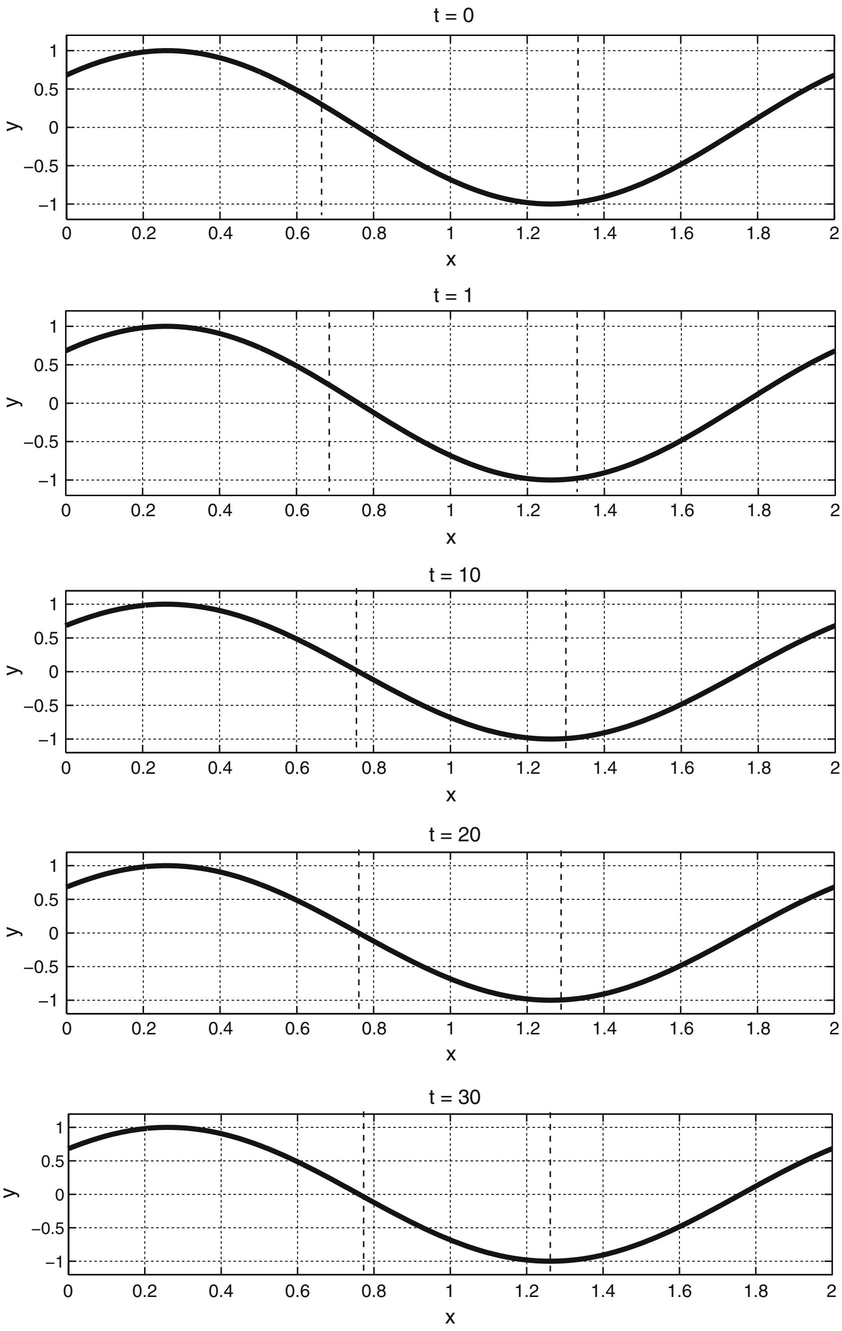
**Input:** the system $\Sigma = (x, y)$, the number of molecules $n \geq 2$, the maximum number of compounds $c_{max} > 0$, the small tolerance value $\varepsilon > 0$ and the learning rate $\eta$.
**Output:** the structure of mixture $\Gamma$, the set of hydrogen values $H$, the set of stoichiometric coefficients $\Lambda$ and the set of intermolecular distances $\Pi$.

```
$i = 0$
$R = \Sigma$
**while** $(R > \varepsilon)$ and $(i < c_{max})$ **do**
    $i = i + 1$
    $C_i$ =CREATE-COMPOUND($R$, $n$)
    $[H_i, \Pi_i]$ =OPTIMUM-COMPOUND($R$, $C_i$, $n$, $\eta$)
    $R \leftarrow$ using (4.15)
**end-while**
$\Gamma \leftarrow \{C_1, \ldots, C_i\}$
$H \leftarrow \{H_1, \ldots, H_i\}$
$\Lambda \leftarrow$ minimizing (4.19) using LSE.
$\Pi = \{\Pi_1, \ldots, \Pi_i\}$
**return** $\Gamma, H, \Lambda$ and $\Pi$
```

**Proposition 4.10** (***modified AHN-algorithm***) *Let $\Sigma$ be any given system. Also, let $f$ be Algorithm 4.5 with proper inputs. Then, $f$ satisfies the three-level model energy and returns an artificial hydrocarbon network with* structure *close related to the energy of $\Sigma$ and* behavior *totally parameterized, such that, it approximates $\Sigma$.*

It is important to notice that intermolecular distances define the behavior of molecular behaviors. Thus, in notation of (4.16), the intermolecular distances in (4.23) are expressed as (4.27). Recalling that $L_0 = a$ and $L_n = b$ in the input domain $x \in [a, b]$.

$$r_i = r^{i,i+1}, \quad i = 1, \ldots, n-1 \tag{4.27}$$



t = 0

t = 1

t = 10

t = 20

t = 30

**Fig. 4.17** Evolution of the optimum compound using Algorithm 4.4: (*first*) at iteration $t = 0$, (*second*) at iteration $t = 1$, (*third*) at iteration $t = 10$, (*forth*) at iteration $t = 20$, (*fifth*) at iteration $t = 30$. *Dashed lines* represent bounds of molecules

*Example 4.10* Using Algorithm 4.5, build an artificial hydrocarbon network with $n = 3$ and $c_{max} = 1$ of the system $\Sigma = \sin(\pi x + 0.75)$ for all $x \in [0, 2]$. Consider a step size of $\eta = 0.1$ and a tolerance value of $\varepsilon = 0.1$.

**Solution 4.10** Let $n = 3$ be the number of molecules in $C$. Also, initialize $i = 0$ and the residue $R = \Sigma$.

For $i = 0$:

$i = 1$.

$C_1 = (\{CH_3, CH_2, CH_3\}, \{1, 1\})$ using Algorithm 4.1.

$H_1 = \{-5.91, 1.31, 0.36, 1.30, 0.54, 1.08, 0.57, 0.38\}$ using Algorithm 4.4.

$\Pi_1 = (0.7809, 0.4793)$ using Algorithm 4.4.

$\|R\| = 1.0231$ using the L2-norm of $R$.

$\|R\| > \varepsilon$ and $i = c_{max}$, this ends the loop.

Finally,

$\Gamma = (C_1)$

$H = (H_1)$

$\Lambda = (1.0)$ because there is one compound.

$\Pi = (\Pi_1)$

Actually, the resultant response of the artificial hydrocarbon network is the one depicted at last in Fig. 4.17.

Previous algorithms were implemented in the Artificial Organic Networks Toolkit using LabVIEW™ (see Appendix D). Please, refer to Appendix E for some examples of this toolkit.

### *4.2.4 Mathematical Formulation*

Once components, interactions and different chemical rules were defined and proposed, the model of artificial hydrocarbon networks is formally introduced in Definition 4.6:

**Definition 4.6** (***artificial hydrocarbon networks***) *Let $AHN = \langle \Gamma_f, \Phi, \Lambda, X\rangle$ be an artificial organic network made of a mixture $S$, and $A^{(8)}$ be a set of hydrogen and carbon atomic units that exists in AHN. Also, let $G$ be the unique functional group of AHN. Then, AHN is an* artificial hydrocarbon network *and $G$ is called the CH* functional group. *Moreover, AHN has the following properties:*

1. *$\Gamma_f$ is a mixture of artificial hydrocarbon compounds with basis $\Phi$ and a set of stoichiometric coefficients $\Lambda$.*
2. *$\Gamma_f$ and $S$ are obtained using Algorithm 4.5 as the three-level energy rule $f$.*
3. *$S(X)$ is the behavior of AHN due to $X$.*

## 4.3 Metrics of Artificial Hydrocarbon Networks

In this section, two metrics of the modified artificial hydrocarbon networks algorithm are computed in order to determine the computational time complexity and the stability of Algorithm 4.5. For further information about computational complexity and stability analysis theory, refer to Sect. 1.4.

### 4.3.1 Computational Complexity

Consider any given system of the form $\Sigma = (x, y)$ with an input signal $x$ and an output signal $y$. Also, consider $q$ training samples of $\Sigma$ in order to get a model *AHN* using Algorithm 4.5 with $c$ compounds and $n$ molecules per compound. Assume that a minimum error $\varepsilon > 0$ satisfies the stop criterion for finding the best intermolecular distances $r^{jk} \in \Pi$. Then, Theorem 4.1 holds for Algorithm 4.5.

**Theorem 4.1** (***time complexity***) *Algorithm 4.5 has time complexity $O(cnq \ln \frac{1}{\varepsilon})$ with $q \geq n \geq c \geq 2$ and a small value $\varepsilon > 0$.*

*Proof* Consider Algorithm 4.5. Time complexities are measured using the worst case assumption.

*CREATE-COMPOUND:* Consider Algorithm 4.1.

The first instruction is an assignment, thus it has time complexity $O(1)$.

The second step splits signal $\Sigma$ into $n$ partitions $\Sigma_k$ for $k = 1, \ldots, n$, using the set of intermolecular distances $\{r^{j,j+1}\}$ with $j = 0, \ldots, n-1$, and $r^{01} = a$ and $r^{n,n+1} = b$; where, $x \in [a, b]$. Assume a simple algorithm for splitting the signal using the set of rules of the form as (4.28), for all $i = 1, \ldots, q$.

$$R_\alpha : \text{ if } r^{\alpha-1,\alpha} \leq x_i < r^{\alpha,\alpha+1}, \text{ then } (x_i, y_i) \in \Sigma_\alpha, \text{ with } \alpha = 1, \ldots, n \tag{4.28}$$

Then, each rule $R_\alpha$ in (4.28) requires 4 operations (i.e. 2 comparisons, one AND-operation and one assignment). Since there are $n$ rules in (4.28), and assuming the worst case when the $q$ samples are in the last rule $R_n$, then the time complexity is $O(4nq)$.

The third step considers a for-loop. First, the measurement of energy in a partition is required. Suppose that $\Sigma_k$ has $q_k$ samples. Using (4.13), the energy of the partition can be obtained from (4.29); where, $y_k$ represents the $k$-th sample of output signal in $\Sigma_k$. Notice that this representation is valid because the integral of (4.13) depends on the number of $q_k$ samples in $\Sigma_k$.

$$E_k = \sum_{i=1}^{q_k} {y_k}^2 \tag{4.29}$$

Clearly, (4.29) needs $q_k - 1$ additions and $q_k$ multiplications. Thus, the time complexity for computing the energy in each partition is $O(q_k)$. Since, there are two measures of energy, one sum of those, three comparisons and a decision from (4.14), and a selection of two $CH$-primitive molecules in a lookup table, the time complexity until now is $O(q_k + q_k + 1 + 4 + 1) \sim O(q_k)$.

Then, a while-loop encloses a decision procedure. At most, the while-loop enters three times, only if two subsequent bonds with orders $k_j^*, k_{j+1}^*$ are equal to 3. In addition, the if-else decision has order $O(3) \sim O(1)$ because the total operations are: one for the decision, one for the subtraction and one for the selection in a lookup table. At last, the while-loop has time complexity $O(9) \sim O(1)$.

At the end, the for-loop has time complexity $\sum_{k=1}^{n-1} (O(q_k) + O(1)) \sim O(q + n)$, since $q = \sum_{k=1}^{n} q_k$.

The overall time complexity of Algorithm 4.1 is $O(1 + 4nq + q + n + 1) \sim O(4nq + n + q + 2)$ assuming that the returning operation is one instruction.

*OPTIMUM-COMPOUND*: Consider Algorithm 4.4.

The first step requires a split procedure. From the above discussion, splitting the signal using (4.28) has time complexity $O(4nq)$.

The second step has a while-loop. Starting with the inner instructions, the first step is again a splitting function with time complexity $O(4nq)$. The second step has a for-loop from $k = 1, \ldots, n$. Firstly, it computes the parameters $H_{is}$ with $s = 1, \ldots, d$ ($d$ represents the maximum number of hydrogen atoms associated to the carbon atom) using least squares estimates. In fact, this method has time complexity $O(C^2N)$ [4]; where $C$ is the number of features and $N$ is the number of training samples. In terms of Algorithm 4.4, the time complexity of this step is $O(d^2q_k)$ and it can be approximated to $O(q_k)$ because $d = \{1, 2, 3\}$ is restricted to the number of hydrogen atoms in the $CH$-primitive molecules, and $d \ll q_k$. Then, the for-loop also computes the error between the partition $\Sigma_k$ and the structure of $CH$-primitive molecules using (4.18). It can be seen that this computation requires $q_k$ subtractions, $q_k - 1$ additions and $q_k + 1$ multiplications. Then, the time complexity is $O(3q_k) \sim O(q_k)$. At last, the loop runs $n$ times for the $n$ partitions of $\Sigma$. Thus, the upper bound complexity of the for-loop in Algorithm 4.4, is $\sum_{k=1}^{n} (O(q_k)) \sim O(q)$, since $q = \sum_{k=1}^{n} q_k$. Finally, the third step of the while-loop updates the set of intermolecular distances using (4.21) and (4.22). Clearly, they need three operations (i.e. two subtractions and one multiplication) for the $n - 1$ intermolecular distances not including the extremes of the interval $r^{01}$ and $r^{n,n+1}$. Thus, the time complexity for the last step is $O(3n - 3) \sim O(n)$. In that sense, the overall time complexity of one iteration in the while-loop is $O(4nq + q + n) \sim O(nq + q + n) \sim O(nq)$ because $O(nq) > O(q) \geq O(n)$ if $q \geq n \geq 2$.

In order to determine the number of cycles in the while-loop, the following discussion uses the notion of the analytical complexity in descent methods as (4.21) and (4.22). For instance, consider the optimization problem (4.30),

$$\min_{r \in \mathbb{R}^m} f(r) \tag{4.30}$$

Where, $f(r)$ is the objective function to find the optimal values of intermolecular distances $\{r^{j,j+1}\}^*$ for $j = 0, \ldots, n$. Moreover, consider a relaxation of (4.30) using a sequence of values $\{f(r)\}$, such that, $f(r_{t+1}) \leq f(r_t)$ for all $t = 0, 1, \ldots$. This problem can be solved applying a descent method, if the following properties hold [5]:

- $f(r)$ is bounded below.
- $f(r)$ has a first derivative $f'(r)$ continuos in $\mathbb{R}^m$.
- $f'(r)$ is Lipschitz, such that, $\|f'(r_{t+1}) - f'(r_t)\| \leq L\|r_{t+1} - r_t\|$ with some positive value $L$, and $\|\cdot\|$ standing for the Euclidean norm (4.31); where, $x = (x_1, \ldots, x_n)$ is a vector in $\mathbb{R}^n$.

$$\|x\| = \sqrt{x_1^2 + \cdots + x_n^2} \tag{4.31}$$

If this is the case, the values of the set $\{r\}^*$ are reached with an accuracy $\varepsilon > 0$ at time complexity $O(\ln \frac{1}{\varepsilon})$, using a gradient descent method [5]. Notice that the steepest descent method (4.21) and (4.22) is valid for solving (4.30) because the gradient descent method coincides with the steepest descent method when $\|\cdot\|$ stands for the Euclidean norm [6].

In order to apply (4.21) and correctly, let $f(r)$ be a function $E_f(r) = E_u - E_v$ representing the difference of the energy of any molecule $u$ and the energy of any molecule $v$ with intermolecular distance $r^{uv}$ as in (4.22). It is enough to prove that $E_f(r)$ has the above properties.

By definition of (4.18), energies $E_u$ and $E_v$ are bounded below to $E_u = E_v = 0$; thus, $E_f(r)$ is bounded below at $E_f(r) = 0$, proving the first property. On the other hand, it is easy to see that $E_f'(r) \simeq \Delta E_f(r_t) = E_f(r_{t+1}) - E_f(r_t)$ and because $E_f(r_k) \in \mathbb{R}^m, \forall k = 0, 1, \ldots,$ then $\Delta E_f(r) \in \mathbb{R}^m$, proving the second property. Finally, to prove the third property, $E_f(r)$ needs to have a second derivative $E_f''(r)$, such that, $\|E_f''(r)\| \leq L, \forall r \in \mathbb{R}^m$. It is easy to see that $E_f''(r) \simeq \Delta^2 E_f(r_t)$, then $\Delta^2 E_f(r_t) = E_f(r_{t+2}) - 2E_f(r_{t+1}) + E_f(r_t)$ is properly defined. Moreover, $\|\Delta^2 E_f(r_t)\| \leq L$. Thus, it proves the third property.

At last, the while-loop in Algorithm 4.4 has time complexity $O(nq \ln \frac{1}{\varepsilon})$.

Then, Algorithm 4.4 has time complexity $O(4nq + nq \ln \frac{1}{\varepsilon} + 1) \sim O(nq + nq \ln \frac{1}{\varepsilon})$ assuming that the returning operation is one instruction.

*SIMPLE-AHN*: Consider Algorithm 4.5.

This algorithm has the first two assignments with time complexity $O(2) \sim O(1)$. Then, the while-loop has at most $c$ compounds. Inside, the first assignment has time complexity $O(1)$, the CREATE-COMPOUND algorithm has time complexity $O(4nq + n + q + 2)$, the OPTIMUM-COMPOUND algorithm has $O(nq + nq \ln \frac{1}{\varepsilon})$, and the residue of (4.15) has time complexity $O(2q - 1) \sim O(q)$. At last, the while-loop has time complexity $O(c(1 + 4nq + n + q + 2 + nq + nq \ln \frac{1}{\varepsilon} + q)) \sim$

$O(cnq+cn+cq+cnq\ln\frac{1}{\varepsilon}+1)$. Then, computing the set of stoichiometric coefficients implies the least squares method. Using the above discussion, the time complexity of this procedure is $O(c^2q)$. The step of returning parameters might be considered constant, such that, $O(1)$.

In the worst case, the upper bound of time complexity in Algorithm 4.5 is calculated based on (4.32):

$$O\left(cn+cnq+cq+cnq\ln\frac{1}{\varepsilon}+c^2q+3\right) \tag{4.32}$$

It is easy to show that asymptothically $O(1) < O(cn) \leq O(cnq) \leq O\left(cnq\ln\frac{1}{\varepsilon}\right)$. Then, (4.32) can be expressed as (4.33):

$$O\left(cnq\ln\frac{1}{\varepsilon}+c^2q\right) \tag{4.33}$$

If transformation $x = cq$ is applied, then (4.33) is rewritten as $O\left(xn\ln\frac{1}{\varepsilon}+xc\right)$. Also, if $n \geq c \geq 2$, then $n\ln\frac{1}{\varepsilon} \geq c$. At last, $O\left(xn\ln\frac{1}{\varepsilon}\right) \geq O\left(xc\right)$, thus (4.33) is asymptothically equal to (4.34) with $q \geq n \geq c \geq 2$ and a small value $\varepsilon > 0$:

$$O\left(cnq\ln\frac{1}{\varepsilon}\right) \tag{4.34}$$

So, it is shown that Algorithm 4.5 has an upper bound time complexity as (4.34) with $q \geq n \geq c \geq 2$ and a small value $\varepsilon > 0$. □

### 4.3.2 Stability

The efficiency in accuracy of learning algorithms is desirable when proposing new learning methodologies. In fact, stability analysis of algorithms gives an interesting measurement in terms of efficiency. Particularly, the analysis of stability of Algorithm 4.5 is presented, using the sensivity analysis developed by Bousquet and Eliseeff [7] in which it aims to determine how much a variation in the input of the algorithm modifies the behavior of its output [7].

Consider any given system of the form $\Sigma = (x, y)$ with an input signal $x$ and an output signal $y$. Also, consider $q$ training samples of $\Sigma$ in order to get a model *AHN* using Algorithm 4.5 with $c$ compounds and $n$ molecules per compound. Also, let $L$ be a loss function such that it measures accuracy in Algorithm 4.5 between the model *AHN* in $x$ and the output value $y$, expressed as (4.35).

$$L(AHN(x), y) = (AHN(x) - y)^2 \tag{4.35}$$

Moreover, the loss function $L$ is said to be $\sigma$-admissible if $L$ is convex with respect to its first argument $z \in \mathbb{R}$ and (4.36) holds:

$$\forall z_1, z_2 \in \mathbb{R}, \quad \forall y \in Y, \quad |L(z_1, y) - L(z_2, y)| \leq \sigma |z_1 - z_2| \tag{4.36}$$

Then, Algorithm 4.5 has uniform stability $\beta$ with respect to the loss function $L$ if (4.37) holds; where, $L(AHN_{\Sigma}, .)$ represents the loss function of all samples in $\Sigma$ and $L(AHN_{\Sigma^{\setminus i}}, .)$ represents the loss function of all samples in $\Sigma$ without the $i$-th sample.

$$\forall \Sigma, \forall i \in \{1, \ldots, q\}, \quad \|L(AHN_{\Sigma}, .) - L(AHN_{\Sigma^{\setminus i}}, .)\|_{\infty} \leq \beta \tag{4.37}$$

Thus, (4.37) assures that the maximum difference between the cost of some input data set to Algorithm 4.5 and the cost of the same input data set with one variation, is less or equal to an upper bound $\beta$. At last, $AHN$ will be stable if $\beta$ decreases proportionally to $\frac{1}{q}$.

In order to define the response of algorithm $AHN$, a reproducing kernel Hilbert space (RKHS) $F$ is introduced. The fundamental property of a RKHS is written as (4.38); where, there is a unique kernel $k$ in the Hilbert space $H$, such that, $\forall x, y \in H, k_x(y) = k(y, x)$ is a function in the Hilbert space, and $f \in H$ is a set of functions in a convex subset of a linear space $F$.

$$\forall f \in F, \quad \forall x, \quad f(x) = \langle f, k(x, .)\rangle \tag{4.38}$$

Thus, the following Theorem 4.2 holds for stability in artificial hydrocarbon networks in which it assures that Algorithm 4.5 is spanned by the optimum set of functions $f$ that minimizes the loss function $L$, and by a unique kernel $k$.

**Theorem 4.2** (***uniform stability***) *Let $k$ be a bounded kernel such that $k(x, x) \leq \kappa^2$ and $Y \in [a, b]$ be the output vector of AHN bounded in the range $B = b - a$. Then, Algorithm 4.5 has uniform stability $\beta$ with respect to the loss function L defined in (4.35) that is $2B$-admissible if AHN is defined by*

$$AHN_{\Sigma} = \arg\min_{f \in F} \frac{1}{q} \sum_{i=1}^{q} L(f, y_i) + \lambda \|C\|_k^2, \quad \forall y_i \in Y \tag{4.39}$$

*Moreover,*

$$\beta \leq \frac{2\kappa^2 B^2}{\lambda \min\{q_{ij}\}}, \quad \forall i \in \{1, \ldots, n\}, \quad j \in \{1, \ldots, c\} \textit{ with } q = \frac{1}{c} \sum_{j=1}^{c} \sum_{i=1}^{n} q_{ij} \tag{4.40}$$

*Proof* The definition of Algorithm 4.5 in terms of (4.39) comes directly from the theorem related to the definition of algorithms in Hilbert spaces as described by Bousquet and Eliseeff [7]. In fact, the uniform stability $\beta$ of any given algorithm $A_S$

in the Hilbert space reproduced by the kernel $k$ is like (4.41):

$$\beta \leq \frac{\sigma^2 \kappa^2}{2\lambda m} \tag{4.41}$$

Where, $\sigma$ refers to the admissibility of loss function $L$ associated to $A_S$, $\kappa$ is the upper bound of the kernel $k$ such that $\forall x$, $k(x, x) \leq \kappa^2 < \infty$, the parameter $\lambda$ is the penalty in the norm of the basis $N(f)$, $\forall f \in F$ such that $N(f) = \|f\|_k^2 \leq \frac{B}{\lambda}$ with some positive value of $B$, and $m$ is the number of training samples.

In order to prove (4.40), consider that the uniform stability $\beta$ of least squares estimates is equal to (4.41) with $\sigma = 2B$ (as described by Bousquet and Eliseeff [7]) like in (4.42):

$$\beta \leq \frac{2\kappa^2 B^2}{\lambda m} \tag{4.42}$$

Actually, Algorithm 4.5 uses least squares estimates in each of the $n$ molecules to form a compound. Thus, let $q_i$ be the number of samples in the $i$-th molecule, such that $q = \sum_{i=1}^{n} q_i$. Then, using (4.42), each molecule provides a maximum risk bounded by (4.43).

$$\beta_i \leq \frac{2\kappa^2 B^2}{\lambda q_i}, \quad \forall i \in \{1, \ldots, n\} \tag{4.43}$$

Notice that from definition, $B = b - a$ remains constant for all molecules. Thus, the overall uniform stability $\beta_C$ of a compound is given by the maximum of all uniform stabilities $\{\beta_i\}$ as depicted in (4.44).

$$\beta_C = \sum_{i=1}^{n} \beta_i \leq \frac{2\kappa^2 B^2}{\lambda} \max\left(\frac{1}{q_i}\right), \quad \forall i \in \{1, \ldots, n\} \tag{4.44}$$

Moreover, Algorithm 4.5 uses $c$ compounds. Thus, the total uniform stability $\beta_T$ of *AHN* is given by the maximum of all uniform stabilities $\{\beta_C^{(j)}\}$ of compounds like (4.45); where $q_{ij}$ is the number of samples in the $i$-th molecule of the $j$-th compound such that $cq = \sum_{j=1}^{c} \sum_{i=1}^{n} q_{ij}$.

$$\beta_T = \sum_{j=1}^{c} \beta_C^{(j)} \leq \frac{2\kappa^2 B^2}{\lambda} \max\left(\frac{1}{q_{ij}}\right), \quad \forall i \in \{1, \ldots, n\}, \quad j \in \{1, \ldots, c\} \tag{4.45}$$

Then, (4.45) proves the uniform stability of Algorithm 4.5 depicted in (4.40). □

Notice that Theorem 4.2 provides the upper bound of uniform stability. In fact, (4.40) gives the notion that increasing the number of samples per molecule per compound $q_{ij}$ gives a better stability of Algorithm 4.5. Finally, it is remarkable to say that

Theorem 4.2 guarantees stability of Algorithm 4.5 only for approximation problems, since artificial hydrocarbon networks is mainly developed for these problems.

## 4.4 Artificial Hydrocarbon Networks Practical Features

In this section, some characteristics of artificial hydrocarbon networks are highlighted in order to exploit them, but also to understand them. In practice, artificial hydrocarbon networks approximate given systems; araising in different applications like inference, clustering, classification, pattern recognition, and even optimization.

### *4.4.1 Partial Knowledge Representation*

One of the objectives of artificial organic networks and in particular for artificial hydrocarbon networks is the idea to extract information from the modeled system to provide metadata of it and in consequence to better understand it. For instance, the following five different levels of information are given when the training process of AHNs finishes: molecular clustering, clustering of compounds, stoichiometric coefficients, hydrogen coefficients, and intermolecular distances.

In order to have a visual representation of artificial hydrocarbon networks, consider the adoption of the structural formulas in chemistry. Figure 4.18 shows an example of a structural formula applied to AHNs. Notice that there are two compounds, each one showing its associated stoichiometric coefficient at the bottom. In

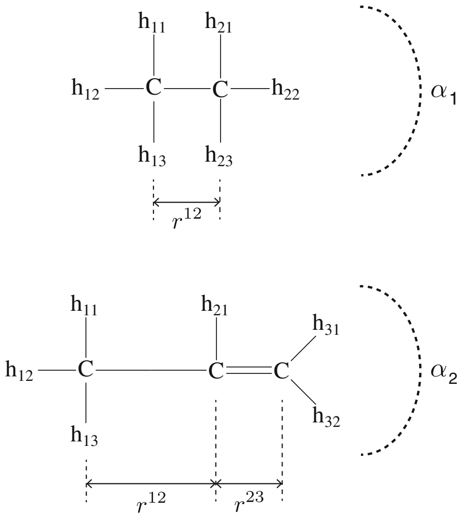


**Fig. 4.18** Example of an structural formula applied to visual representations of artificial hydrocarbon networks

addition, each compound is made of different $CH$-primitive molecules. Each $CH$-molecule is identified by the carbon atom $C$. Then, the number of carbon atoms in the compound refers to the number of $CH$-primitive molecules associated to the compound. Also, each molecule has a finite number of hydrogen atoms $h_{ij}$ (the $j$-th hydrogen atom of the $i$-th molecule in the compound) representing instances of the specific trained AHN. Finally, the values show above the stoichiometric coefficients represent intermolecular distances in compounds, e.g. the length between carbon atoms.

#### 4.4.1.1 Molecular Clustering

Artificial hydrocarbon networks organize molecules in order to capture clusters of information. In fact, these clusters are grouped by similar characteritics. For example, Algorithm 4.5 implicitly determines the criterion of clustering similar points. One of those is approximated by one molecule; thus, the number of molecules in the AHN-structure corresponds to the number of clusters generated (see Fig. 4.18). In Algorithm 4.5, finding the optimum intermolecular distances assures the minimum error (4.18) per molecule that it geometrically represents similar points. In a later chapter, the optimal number of molecules is treated.

#### 4.4.1.2 Clustering of Compounds

In the essence of molecular clustering, the number of compounds refers to the number of components that will form a basis of the final approximation. Consider, for example, the behavior of the network $S(x)$ be a vector that is spanned by a linear combination of behaviors of compounds $\psi_i \in \Phi$, that it is the basis of $S(x)$. With a slight modification of Algorithm 4.5, each compound should extract particular information. For instance, consider the decomposition of an audio signal. In that sense, artificial hydrocarbon networks might be used for representing the fundamental signal and the armonics by compounds.

#### 4.4.1.3 Stoichiometric Coefficients

Since the set of compounds is the basis of the AHN approximation, then stoichiometric coefficients are the coordinates of the behavior of the AHN. In fact, each coefficient can be interpreted as the total weight each compound is supporting to the final approximation of the model. This is easy to check because finding the stoichiometric coefficients corresponds to solve a multilinear regression.

#### 4.4.1.4 Hydrogen Parameters

Indeed, hydrogen parameters are highly important in the interpretation of knowledge captured by artificial hydrocarbon networks. It is easy to check from Proposition 4.2 that hydrogen parameters are the roots (or zeros) of the approximated molecular behavior. Moreover, since hydrogen parameters are complex numbers, it is evident that real-valued hydrogen parameters refers to molecular behaviors that actually cross in $y_a = 0$, $y_a \in y$. In addition, from the fundamental theorem of algebra, the number of real-valued hydrogen parameters represents the number of times approximated output $y$ crosses by zero. For example, in control theory, the number of poles (roots in the denominator) in a transfer function is closely related to the stability of the physical system. In that sense, the transfer function refers to the model obtained from any physical system. Moreover, if AHNs are applied for modeling the given system and applying the proper transformation in order to capture the frequency domain of the system, hydrogen parameters would be very useful in the analysis of the given system. Another geometrical interpretation of hydrogen parameters is that since $h \in \mathbb{C}$ can be decomposed into a rectangular representation $h = \sigma + \omega i$, $\forall \sigma$, $\omega \in \mathbb{R}$, the real part $\sigma$ gives an insight about when the molecular behavior changes its concavity but not necessarily crosses in $y_a = 0$.

On the other hand, if hydrogen parameters are read using the intermolecular distances as the interval domain of the molecular behavior; then, hydrogen parameters inside that interval will say how far or close are the similar points $(x, y)$ clustered by the molecular behavior.

In addition, all data captured by any molecular behavior might be reproduced using hydrogen parameters and Propositions 4.1 or 4.2. This direct application of hydrogen parameters can be used for learning purposes when the AHN-structure turns to an inference system.

#### 4.4.1.5 Intermolecular Distances

Other important parameters are intermolecular distances $r^{ij}$ that measures the optimal length between two molecules $i$ and $j$. In fact, these distances provide the interval of action in molecular behaviors. Since these distances implicitly measures similarity in points $(x, y)$, those can be used for clustering. It is remarkable to say that unlikeness in clusters might refer to possible discontinuities or other interesting points in the behavior of the modeled system. At last, intermolecular distances can capture these interesting points.

### *4.4.2 Practical Issues in Partial Knowledge Extraction*

As described above, partial knowledge extraction is obtained from different instances of the artificial hydrocarbon network. Some characteristics were highlighted, but those can be used practically when considering AHNs as a learning algorithm for modeling problems.

For instance, the intermolecular distances can be extracted from the AHN-structure and then use them for another processing, e.g. extract disimilar information or analyzing critical points in the system.

Since these intermolecular distances and molecules perform clustering while modeling any given system, these molecular units might be used for classification. Again, as said above, one direct application of hydrogen parameters is generalization of data in an inference system, i.e. learning procedures with AHNs.

Notice that molecular units and compounds encapsulate information as characteristics of systems. Thus, one previous training molecule or compound can be used in future systems when similar characteristics are expected. Hence, molecules and compounds inherit information, and reusability might be considered.

## References

1. Ponce H, Ponce P (2012) Artificial hydrocarbon networks: a new algorithm bio-inspired on organic chemistry. Int J Artif Intell Comput Res 4(1):39–51
2. Ponce H, Ponce P, Molina A (2013) A new training algorithm for artificial hydrocarbon networks using an energy model of covalent bonds. In: Proceedings of the 7th IFAC conference on manufacturing modelling, managament, and control. International federation of automatic control, Saint Petersburg, Russia, pp 602–608
3. Ponce H, Ponce P, Molina A (2012) A novel adaptive filtering for audio siganls using artificial hydrocarbon networks. In: Proceedings of IEEE 9th international conference on electrical engineering, computing science and automatic control. Mexico, Mexico, pp 277–282
4. Schoukens J, Rolain Y, Gustafsson F, Pintelon R (1998) Fast calculation of least-squares estimates for system identification. In: Proceedings of the 37th IEEE conference on decision and control, vol 3. Tampa, Florida, pp 3408–3410
5. Nesterov Y (2004) Introductory lectures on convex optimization. Kluwer Academic Publishers, Dordrecht
6. Boyd S, Vandenberghe L (2009) Convex optimization. Cambridge University Press, Cambridge
7. Bousquet O, Eliseeff A (2002) Stability and generalization. J Mach Learn Res 2:499–526

# Chapter 5
# Enhancements of Artificial Hydrocarbon Networks

Artificial hydrocarbon networks (AHN) algorithm builds and trains a model for any given system. However, that model considers a single-input-and-single-output (SISO) system and a fixed number of molecules. These assumptions limit the performance of the obtained model. For example, systems that are not SISO cannot be handled easy with artificial hydrocarbon networks, or the number of molecules is difficult to determine, except from experience tuning or trail-and-error.

Thus, this chapter presents three enhancements of the AHN-algorithm: an optimization in the number of molecules, an extension to model multiple-inputs-and-multiple-outputs (MIMO) systems, and a generalization to create recurrent carbon-based networks inspired on chemical aromatic compounds.

## 5.1 Optimization of the Number of Molecules

As noted in simple and modified artificial hydrocarbon networks algorithm, the topology depends on the number of molecules $n$ and the number of compounds $c$. However, these values has to be selected from previous experience or subjected to the system $\Sigma$. In this section, an improvement of the number of molecules is presented. Functionally, it optimizes the structure of artificial hydrocarbon networks to the minimum artificial hydrocarbon compound that approximates $\Sigma$ with some accuracy $\varepsilon > 0$.

### *5.1.1 The Hess' Law*

In chemistry, enthalpy $\Delta H^\circ$ refers to the energy transfer in a thermodynamic system (see Sect. 2.4.2). In particular, enthalpy is the measure of heating interchange in a chemical reaction at constant pressure. Closely to molecules, it is the energy required to form or break a chemical bond. Thus, enthalpy is the chemical energy

H. Ponce-Espinosa et al., *Artificial Organic Networks*,
Studies in Computational Intelligence 521, DOI: 10.1007/978-3-319-02472-1_5,

in a reaction at constant pressure. In order to compute the enthalpy $\Delta H^{\circ}_{reaction}$ in a chemical reaction, it is helpful to understand the Hess's law. Roughly, it states that the enthalpy of the reaction is independent of the pathway between the initial and final states [1], as shown in Fig. 5.1. The process begins with an initial set of molecules that chemically react forming new compounds. Those react again and again several times, e.g. a chain of chemical reactions or paths, until the process stops with a final set of molecules. Then, the Hess' law assures that the initial set of molecules (reactants) has enthalpy equal to the enthalpy of the set of final molecules (products).

Let $\Omega_{reactans}$ be the set of initial molecules called reactants, and $\Omega_{products}$ be the set of final molecules called products. Then, a chemical reaction is a process to transform $\Omega_{reactants}$ in $\Omega_{products}$ and is expressed as (5.1) with enthalpy of reaction $\Delta H^{\circ}_{reaction}$

$$\Omega_{reactants} \rightarrow \Omega_{products} \tag{5.1}$$

Moreover, each molecule $M_i \in \Omega_{reactants}$ has enthalpy $\Delta H^{\circ}_i$ in the set of reactant enthalpies $\Delta H^{\circ}_{reactants}$ and each molecule $M_j \in \Omega_{products}$ has enthalpy $\Delta H^{\circ}_j$ in the set of product enthalpies $\Delta H^{\circ}_{products}$. Then, the enthalpy of reaction $\Delta H^{\circ}_{reaction}$ is equal to (5.2):

$$\Delta H^{\circ}_{reaction} = \sum \Delta H^{\circ}_{products} - \sum \Delta H^{\circ}_{reactants} \tag{5.2}$$

### 5.1.2 Boiling and Melting Points in Hydrocarbons

In organic chemistry, there is known that hydrocarbon compounds have higher boiling and melting points while the number of carbon atoms increases [2]. Figure 5.2 shows that relationship. Notice that the boiling point in hydrocarbons increases monotonically while the melting point increases between two bounds when the number of carbon atoms is small. In both cases, this tendency occurs similarly in alkanes, alkenes

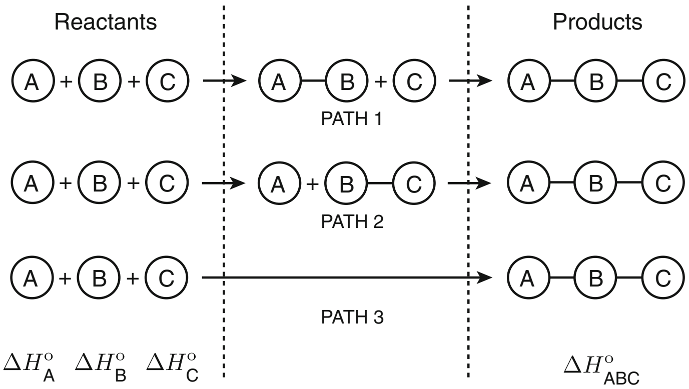


**Fig. 5.1** Enthalpy of reaction a in chain of chemical reactions

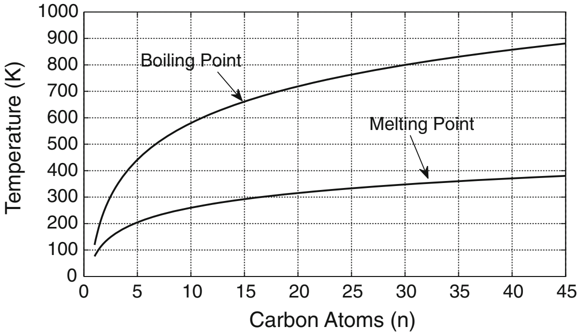


**Fig. 5.2** Estimation of the boiling point ($K$) with respect to the number of carbon atoms ($n$) in hydrocarbon compounds

and alkynes. In addition, branched hydrocarbon compounds contribute to decreasing boiling and melting points. However, from the point of view of artificial hydrocarbon networks, branching is out of scope. At last, only boiling points will be considering next because at these temperatures chemical bondings are subjected to be formed or broken.

### 5.1.3 Enthalpy in Artificial Hydrocarbon Networks

In fact, enthalpy $\Delta H^\circ$ is proportionally to the temperature of a given compound, as written in (5.3); where, $\Delta C_p$ is the change of specific heat, $T$ is the temperature of the compound in $K$ and $\Delta H^\circ_{0K}$ is the enthalpy of the compound at temperature $T_0 = 0K$.

$$\Delta H^\circ = \Delta C_p T + \Delta H^\circ_{0K} \tag{5.3}$$

Thus, if a molecule $M$ is subjected to a chemical reaction; the enthalpy of reaction $\Delta H^\circ_M$ in $M$ can be derived from (5.3) with temperature $T$ equal to the boiling point $T_b$, as expressed in (5.4).

$$\Delta H^\circ_M = \Delta C_p T_b + \Delta H^\circ_{0K} \tag{5.4}$$

From (5.4) and the discussion of boiling points in Sect. 5.1.2, the following observations hold:

- The greater the boiling point $T_b$ is in a molecules $M$, the larger the number of carbon atoms $n$ is present in $M$.
- The greater the boiling point $T_b$ is in a molecule $M$, the larger the value of enthalpy of reaction $\Delta H^\circ_M$ is in $M$.

Concluding, the larger the number of carbon atoms $n$ is in a molecule $M$, the larger the value of enthalpy of reaction $\Delta H^\circ_M$ is in $M$, and vice versa. Actually, this conclusion is

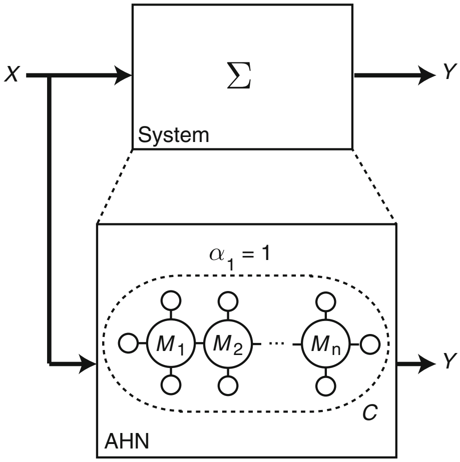


**Fig. 5.3** Structure of the artificial hydrocarbon network *AHN* that models $\Sigma$

used for optimizing the number of *CH*-primitive molecules in artificial hydrocarbon compounds.

Let $\Sigma$ be any given system and *AHN* be an artificial hydrocarbon network that models $\Sigma$. Also, let $c = 1$ be the number of compounds in *AHN*. Then, the artificial hydrocarbon network *AHN* only have one artificial hydrocarbon compound $C$ made of $n \geq 2$ *CH*-primitive molecules; thus, the stoichiometric coefficient is $\alpha_1 = 1$, as shown in Fig. 5.3. The objective is to find the optimal number of molecules $n^*$ such that *AHN* models $\Sigma$ closely.

Let $\Delta H^\circ_\Sigma$ be the enthalpy of the system, i.e. the energy $E$ of the system calculated as (4.12). In addition, let $\Delta H^\circ_M$ be the enthalpy of the compound $C$ in *AHN*. Also, suppose that $\Gamma_p$ is the set of all possible optimal compounds of the form as (5.5); where $\Gamma$ is the set of all compounds in an artificial hydrocarbon network.

$$\Gamma_p = \left\{ C_i \in \Gamma | \Delta H^\circ_{C_i} = \Delta H^\circ_M \right\} \tag{5.5}$$

Then, the compound $C$ that models $\Sigma$ must have an enthalpy equal to the enthalpy of the system like (5.6):

$$\Delta H^\circ_M = \Delta H^\circ_\Sigma \tag{5.6}$$

Since, a *CH*-primitive molecule has exactly one carbon atom, then the number of *CH*-primitive molecules $n_i$ is equal to the number of carbon atoms in a compound $C_i \in \Gamma_p$. In addition, it is easy to check that no matter how compounds in $\Gamma_p$ were formed because of the Hess' law, these compounds have the same value of enthalpy as the system has, if there is no loss of energy in the chemical reaction, i.e. $\Delta H^\circ_{reaction} = 0$. In that sense, the enthalpy of a compound $C_i \in \Gamma_p$ is equal to the sum of enthalpies of the $n_i$ *CH*-primitive molecules that conforms $C_i$ as (5.7); where, $\Delta H^\circ_k$ represents the enthalpy of the $k$-th *CH*-primitive molecule in $C_i$.

$$\sum_{k=1}^{n_i} \Delta H_k^\circ = \Delta H_{C_i}^\circ \tag{5.7}$$

Notice that the enthalpy of a compound $C_i \in \Gamma_p$ in (5.7) depends on the number of molecules $n_i$ and the enthalpies $\Delta H_k^\circ$. However, the number of molecules does not have an upper bound, i.e. $n_i \geq 2$ and $n_i \in [2, \infty)$. Additionally, enthalpies $\Delta H_k^\circ$ vary continuously in the interval $[0, \infty)$. Thus, find the optimum compound $C$ in $\Gamma_p$ is an $NP$-problem. Then, a chemical rule based on the enthalpy of hydrocarbon compounds is designed as follows.

Suppose that enthalpies $\Delta H_k^\circ$ are fixed, then (5.6) is difficult to reach varying only the number of molecules $n_i$ arising to $\Delta H_{C_i}^\circ \leq \Delta H_\Sigma^\circ$. Thus, there exists a positive number $m$ such that (5.8) holds:

$$m \Delta H_{C_i}^\circ = \Delta H_\Sigma^\circ \tag{5.8}$$

From (5.8) and using the conclusion about the relationship between the number of carbon atoms and the enthalpy of a molecule ($n_i \propto \Delta H_{C_i}^\circ$), the heuristic (5.9) is proposed:

$$m = \frac{\Delta H_\Sigma^\circ}{\Delta H_{C_i}^\circ} \propto \frac{\Delta H_\Sigma^\circ}{n_i} \tag{5.9}$$

Finally, (5.9) is useful to announce the following rules, assuming that the initial value of $n_i = 2$:

- If $m > 1$, then increase $n_1$ by one, and recalculate $m$.
- If $m \leq 1$, then remain $n_i$ unchanged, and stop the process.

**Algorithm 5.1** ENTHALPY-RULE($\Sigma$, $C$, $n$): Updating the number of molecules in compound $C$ using an enthalpy-based rule.

**Input:** the system $\Sigma = (x, y)$, the current compound $C$ and the current number of molecules $n$ in $C$.

**Output:** the updated number of molecules $n$ in $C$ and value $m$.

```
$\Delta H_\Sigma^\circ \leftarrow$ using (4.12)
$\Delta H_C^\circ \leftarrow$ using (4.12)
$m \leftarrow$ using $\Delta H_\Sigma^\circ$ and $\Delta H_C^\circ$ in (5.9)
if $m > 1$
    $n = n + 1$
end-if
return $n$ and $m$
```

At last, the optimum number of $CH$-primitive molecules $n^*$ of the optimum compound $C \in \Gamma_p$ is equal to the unchanged value $n_i$ coming from the above rules. To this end, Algorithm 5.1 presents the procedure to optimize the number of molecules in a compound and Proposition 5.1 formalizes it.

**Proposition 5.1** (***enthalpy rule***) *Let $\Sigma$ be any given system, $C$ be a compound of an artificial hydrocarbon network that models $\Sigma$ and $n$ be the number of molecules in $C$. Then, using Algorithm 5.9 with inputs $\Sigma$, $C$ and current $n$, the number of molecules in $C$ will be updated such that $C$ will tend to be the optimum compound with the minimum number of molecules $n$, if and only if the initial value of $n = 2$.*

*Example 5.1* Consider that any given system $\Sigma$ has energy $E = 300$ and it has to be modeled with an artificial hydrocarbon network of one compound $C$. Table 5.1 shows the enthalpy values of the compound $C$ with respect to different number of molecules $n$. Using the enthalpy-rule of Algorithm 5.1, determine the optimum number of molecules in $C$.

**Solution 5.1** Initialize the number of molecules $n = 2$, and the enthalpy of system using the energy signal $E$ such that $\Delta H^{\circ}_{\Sigma} = 300$. Then, compute the heuristic rule (5.9):

$$m(n = 2) = \frac{300}{170} = 1.76$$

Since, $m > 1$, then $n = 3$. Recalculating:

$$m(n = 3) = \frac{300}{170} = 1.43$$

Since, $m > 1$, then $n = 4$. Recalculating:

$$m(n = 4) = \frac{300}{230} = 1.30$$

Since, $m > 1$, then $n = 5$. Recalculating:

$$m(n = 5) = \frac{300}{297} = 1.01$$

Since, $m > 1$, then $n = 6$. Recalculating:

**Table 5.1** Different enthalpy values of $C$ with respect to the number of molecules $n$ of Example 5.1

| $n$ | $\Delta H^{\circ}_{C}$ |
|---|---|
| 2 | 170 |
| 3 | 210 |
| 4 | 230 |
| 5 | 297 |
| 6 | 299 |
| 7 | 300 |
| 8 | 300 |

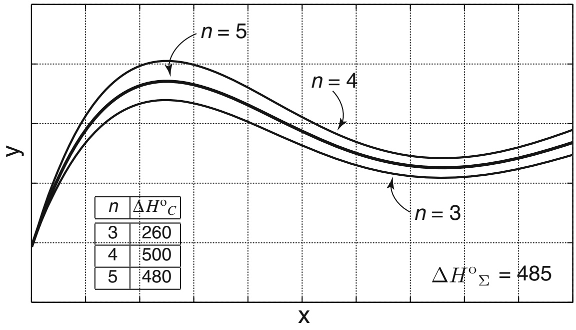


**Fig. 5.4** Example of a non-monotonic increasing function of enthalpy with respect to the number of molecules in a compound of AHNs

$$m(n = 6) = \frac{300}{299} = 1.01$$

Since, $m > 1$, then $n = 7$. Recalculating:

$$m(n = 3) = \frac{300}{300} = 1.00$$

Since, $m = 1$; then, stop the process and return the optimal value $n = 7$. Notice that in this example, it is easy to know that the optimum number of molecules is $n = 7$ by examining Table 5.1 directly because at this value, the enthalpy of the compound $\Delta H_C^\circ$ is equal to the energy of the system $E$. However, in real life applications, that kind of tables is unknown. Then, Algorithm 5.1 is useful for optimizing the number of molecules in a compound. Also note that the enthalpy is a monotonic increasing function with respect to the number of molecules. But, this is not the general case in artificial hydrocarbon networks because in some cases the approximation of molecules may compute higher values of enthalpies than the expected ones, as shown in Fig. 5.4.

Using Proposition 5.1, the artificial hydrocarbon networks algorithm depicted in Algorithm 4.5 can be modified in order to optimize the number of molecules in a compound, as shown in Algorithm 5.2.

## 5.2 Extension to the Multidimensional Case

Learning algorithms that deal with several attribute and target variables are highly important in machine learning. Thus, an extension of artificial hydrocarbon networks to the multidimensional case is presented in this section.

**Algorithm 5.2** OPTIMIZED-AHN($\varSigma, n_{max}, c_{max}, \varepsilon, \eta$): Modified algorithm of artificial hydrocarbon networks with optimum number of molecules.

**Input:** the system $\varSigma = (x, y)$, the maximum number of molecules $n_{max} \geq 2$, the maximum number of compounds $c_{max} > 0$, the small tolerance value $\varepsilon > 0$ and the learning rate $\eta$.
**Output:** the structure of mixture $\varGamma$, the set of hydrogen values $H$, the set of stoichiometric coefficients $\varLambda$ and the set of intermolecular distances $\varPi$.

```
$i = 0$
$R = \varSigma$
while ($R > \varepsilon$) and ($i < c_{max}$) do
    $i = i + 1$
    $n = 2$
    $m \to \infty$
    while $m > 1$ and $n \leq n_{max}$ do
        $C_i$ =CREATE-COMPOUND($R, n$)
        [$H_i, \varPi_i$] =OPTIMUM-COMPOUND($R, C_i, n, \eta$)
        [$n, m$] =ENTHALPY-RULE($R, C_i, n$)
    end-while
    $R \leftarrow$ using (4.15)
end-while
$\varGamma \leftarrow \{C_1, ..., C_i\}$
$H \leftarrow \{H_1, ..., H_i\}$
$\varLambda \leftarrow$ minimizing (4.19) using LSE.
$\varPi = \{\varPi_1, ..., \varPi_i\}$
return $\varGamma, H, \varLambda$ and $\varPi$
```

### 5.2.1 Components and Interactions

Useful $CH$-molecules, artificial hydrocarbon compounds, mixtures and nonpolar covalent bonds propositions for multidimensional purposes are introduced. Recall that $H$ and $C$ represent a hydrogen atom and a carbon atom, respectively. Refer to Sect. 4.2 for a detailed information about definitions of components and interactions in artificial hydrocarbon networks.

**Proposition 5.2** (***first model of multidimensional*** $CH$***-molecules***) *Let $M$ be a $CH$-molecule with $a_c = C$. Also, let $\varphi$ be the behavior of molecule $M$ due to an input signal $\mathbf{x} = (x_1, ..., x_n)$ in $\mathbb{R}^n$ with norm $\|\mathbf{x}\| < 1$. Then, the behavior $\varphi = (\varphi_1, ..., \varphi_j, ..., \varphi_m)$ in $\mathbb{R}^m$ holds:*

$$\mathbf{v}_{H_i} = \mathbf{h}_i \ , \ \mathbf{h}_i \in \mathbb{C}^n \tag{5.10}$$

$$\varphi_j (\mathbf{x}) = \sum_{r=1}^{n} \sum_{i=1}^{d \leq e_C} h_{ir} \cdot x_r^i \tag{5.11}$$

*Where, $\mathbf{v}_{H_i}$ represents a set of hydrogen values of the $H_i$ atom, $\mathbf{h}_i = (h_{i1}, ..., h_{in})$ is the vector of constants values of $H_i$, $d$ is the degree of freedom of the $C$ atom, and $e_C$ is the number of valence electrons of $C$.*

*Example 5.2* Consider an input signal $\mathbf{x} = (x, y)$ that excites a *CH*-molecule. Determine the shape of the behavior $\varphi = \varphi_1$ using Proposition 5.2 with degree of freedom $d = 3$.

**Solution 5.2** Using (5.11), define the behavior $\varphi_1$:

$$\varphi_1(\mathbf{x}) = h_{11}x_1^1 + h_{21}x_1^2 + h_{31}x_1^3 + h_{12}x_2^1 + h_{22}x_2^2 + h_{32}x_2^3 \tag{5.12}$$

$$\varphi_1(\mathbf{x}) = h_{11}x + h_{21}x^2 + h_{31}x^3 + h_{12}y + h_{22}y^2 + h_{32}y^3 \tag{5.13}$$

Then,

$$\mathbf{h}_1 = (h_{11}, h_{12}) \tag{5.14}$$

$$\mathbf{h}_2 = (h_{21}, h_{22}) \tag{5.15}$$

$$\mathbf{h}_3 = (h_{31}, h_{32}) \tag{5.16}$$

Figure 5.5 shows the representation of this *CH*-molecule. Notice that the position of each $\mathbf{h}_i$ remains the counterclockwise convention proposed in Sect. 4.2.1.2.

*Remark 5.1* Let $\varphi_r^{1D}$ be a molecular behavior of the form as (4.2). Then, (5.11) can be rewritten as (5.17) for all $x_r \in \mathbf{x}$:

$$\varphi_j(\mathbf{x}) = \sum_{r=1}^{n} \varphi_r^{1D}(x_r) \tag{5.17}$$

Equation (5.17) allows to define the following proposition as a sum of product forms.

**Proposition 5.3** (***second model of multidimensional** CH**-molecules***) *Let $M$ be a CH-molecule with $a_c = C$. Also, let $\varphi$ be the behavior of molecule $M$ due to an input signal $\mathbf{x} = (x_1, ..., x_n)$ in $\mathbb{R}^n$ with norm $\|\mathbf{x}\| < 1$. Then, the behavior $\varphi = (\varphi_1, ..., \varphi_j, ..., \varphi_m)$ in $\mathbb{R}^m$ holds:*

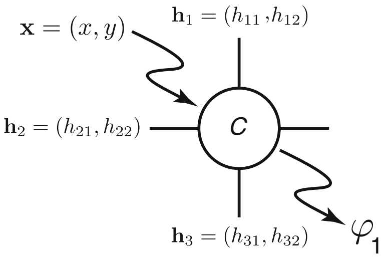

$\mathbf{x} = (x, y)$
$\mathbf{h}_1 = (h_{11}, h_{12})$
$\mathbf{h}_2 = (h_{21}, h_{22})$
C
$\mathbf{h}_3 = (h_{31}, h_{32})$
$\varphi_1$

**Fig. 5.5** Representation of the *CH*-molecule of Example 5.2 using the counterclockwise convention

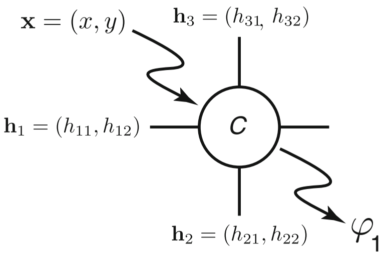

$\mathbf{x} = (x, y)$
$\mathbf{h}_3 = (h_{31}, h_{32})$
$\mathbf{h}_1 = (h_{11}, h_{12})$
C
$\mathbf{h}_2 = (h_{21}, h_{22})$
$\varphi_1$

**Fig. 5.6** Representation of the *CH*-molecule of Example 5.3 using Proposition 5.3

$$\mathbf{v}_{H_i} = \mathbf{h}_i \ , \ \mathbf{h}_i \in \mathbb{C}^n \tag{5.18}$$

$$\varphi_j (\mathbf{x}) = \sum_{r=1}^{n} \prod_{i=1}^{d \leq e_C} (x_r - h_{ir}) \tag{5.19}$$

*Where, $\mathbf{v}_{H_i}$ represents a set of hydrogen values of the $H_i$ atom, $\mathbf{h}_i = (h_{i1}, ..., h_{in})$ is the vector of constants values of $H_i$, $d$ is the degree of freedom of the C atom, and $e_C$ is the number of valence electrons of C.*

Notice that (5.19) comes from directly from (5.17), allowing to position $\mathbf{h}_i$ in any location around the carbon atom $C$, as shown in Fig. 5.6.

*Example 5.3* Consider an input signal $\mathbf{x} = (x, y)$ that excites a *CH*-molecule. Determine the shape of the behavior $\varphi = \varphi_1$ using Proposition 5.3 with degree of freedom $d = 3$.

**Solution 5.3** Using 5.19, define the behavior $\varphi_1$:

$$\varphi_1(\mathbf{x}) = (x_1 - h_{11})(x_1 - h_{21})(x_1 - h_{31}) + (x_2 - h_{12})(x_2 - h_{22})(x_2 - h_{32}) \tag{5.20}$$

$$\varphi_1(\mathbf{x}) = (x - h_{11})(x - h_{21})(x - h_{31}) + (y - h_{12})(y - h_{22})(y - h_{32}) \tag{5.21}$$

Then,

$$\mathbf{h}_1 = (h_{11}, h_{12}) \tag{5.22}$$

$$\mathbf{h}_2 = (h_{21}, h_{22}) \tag{5.23}$$

$$\mathbf{h}_3 = (h_{31}, h_{32}) \tag{5.24}$$

Figure 5.6 shows the the representation of this *CH*-molecule. Notice that it does not matter where $\mathbf{h}_i$ are located around the $C$ atom.

It is important to say that Proposition 5.2, and more exactly (5.11), is a simplification of a multivariate polymonial of degree $d$ in order to compute the least squares estimates easily between any given system $\Sigma$ in a multidimensional space and the behavior of a *CH*-molecule. Moreover, Proposition 5.3, and more exactly (5.19), extends the notion of polynomial roots. However, $\mathbf{h}_i$ does not represent a root of a multivariate polynomial; but it does represent the roots of the projection of the multivariate polynomial onto a specific dimension $r$.

**Proposition 5.4** (***multidimensional hydrocarbon compounds***) *Let* $C = (\Omega, B_N)$ *be a compound consisting of primitive molecules* $\Omega$ *linked with a set of nonpolar covalent bonds* $B_N$*. Then,* $C$ *is said to be a* multidimensional hydrocarbon compound *if* $\Omega = \{M_1, ...M_k\}$ *is a set of CH-primitive molecules, spanned from the CH functional group, with molecular behaviors* $\varphi_1, ...\varphi_k$ *each one in* $\mathbb{R}^m$*. Moreover,* $\psi = (\psi_1, ..., \psi_m)$ *in* $\mathbb{R}^m$ *is the* behavior of $C$ due to any input signal $\mathbf{x} \in \mathbb{R}^n$*, such that,* $\psi : \varphi_1 \times \cdots \times \varphi_k \to \mathbb{R}^m$*. Actually,* $\psi : \mathbb{R}^{k \times m} \to \mathbb{R}^m$*.*

**Proposition 5.5** (***model of multidimensional nonpolar covalent bonds***) *Let* $b_k^{ij} \in B$ *be a nonpolar covalent bond of a pair of molecules* $M_i, M_j$ *with behaviors* $\varphi_i, \varphi_j$ *each one in* $\mathbb{R}^m$ *due to an input* $\mathbf{x} \in \mathbb{R}^n$*. Let* $\Delta = \langle \delta_1, \delta_2, \delta_3 \rangle$ *be a tuple of properties of* $b_k^{ij}$*. Then, the* behavior of nonpolar covalent bonds $\pi$ *holds:*

$$\pi(\varphi_i, \varphi_j, \mathbf{x}) = \delta_3 \left[1 - (\delta_1 \delta_2)^2\right] \exp\left\{-\frac{1}{2}(\delta_1 \delta_2)^2\right\}, \ \pi \geq 0 \tag{5.25}$$

*With,*

$$\delta_2 : \theta_i \times \theta_j \to \mathbb{R} \tag{5.26}$$

*Where,* $\delta_1 \in \{1, 2, 3\}$ *is called the* order of bond*;* $\delta_2 \in \mathbb{R}$ *with* $\delta_2 \geq 0$ *represents the* length of bond *and it is a metric on the parameters* $\theta_i, \theta_j$ *each one in* $\mathbb{R}^p$ *that characterize* $\varphi_i, \varphi_j$*; and,* $\delta_3 \geq 0$ *represents the* minimum energy of bond*. Moreover, the behavior* $\psi$ *of an artificial hydrocarbon compound* $C$ *consisting of molecules* $M_i, M_j$ *is equal to the composite molecular behaviors, such that,* $\psi : \varphi_i(\theta_i) \times \varphi_j(\theta_j) \to \mathbb{R}^m$*.*

For instance, consider the definition of artificial hydrocarbon compounds in (4.16). Then, using Proposition 5.4, the latter can be extended to the multidimensional case as expressed in (5.27) in the input domain $\mathbf{x} \in [\mathbf{a}, \mathbf{b}]$ with $\mathbf{a}, \mathbf{b} \in \mathbb{R}^n$:

$$\psi_j\left(\varphi_{1j}, ..., \varphi_{kj}, \mathbf{x}\right) = \begin{cases} \varphi_{1j}(\mathbf{x}) & \mathbf{r}_0 \leq \mathbf{x} < \mathbf{r}_1 \\ \cdots & \cdots \\ \varphi_{kj}(\mathbf{x}) & \mathbf{r}_{k-1} \leq \mathbf{x} < \mathbf{r}_k \end{cases}, \ \forall j = 1, ..., m \tag{5.27}$$

Where, $\psi = (\psi_1, ..., \psi_j, ..., \psi_m)$ is the behavior of a compound due to an input signal $\mathbf{x} \in \mathbb{R}^n$ that depends on the set of molecular behaviors $\Phi = \{\varphi_1, ..., \varphi_u, ..., \varphi_k\}$ each one in $\mathbb{R}^m$ such that $\varphi_u = (\varphi_{u1}, ..., \varphi_{um})$, and the set of parametes $\mathbf{r}_i$ for all $i = 0, ..., k$ that characterize molecular behaviors stands for the $i$-th bound in $\mathbb{R}^n$ of the input domain in molecular behaviors with lower bound $\mathbf{r}_0 = \mathbf{a}$ and upper bound $\mathbf{r}_k = \mathbf{b}$.

**Proposition 5.6** (***mixture of multidimensional molecules***) *Let $\Gamma = \{M_1, ..., M_k\}$ be a set of molecules with a set of behaviors $\Phi = \{\varphi_1, ..., \varphi_u, ..., \varphi_k\}$ each one in $\mathbb{R}^m$ such that $\varphi_u = (\varphi_{u1}, ..., \varphi_{um})$. Then, the* mixture of multidimensional molecules $S = (S_1, ..., S_j, ..., S_m)$ *in $\mathbb{R}^m$ is a linear combination of behaviors of molecules in $\Phi$ such that there exists a set of coefficients $\Lambda = \{\alpha_1, ..., \alpha_j, ..., \alpha_m\}$ with $\alpha_j = \{\alpha_{1j}, ..., \alpha_{kj}\}$ of real values, called the* stoichiometric coefficients. *Hence,*

$$S_j(\mathbf{x}) = \sum_{i=1}^{k} \alpha_{ij}\varphi_{ij} \tag{5.28}$$

*Moreover, $\Phi$ is the* basis of the mixture of multidimensional molecules*, $\Gamma$ is the* structure of the mixture of multidimensional molecules*, and $S(\mathbf{x})$ is the* behavior of the mixture of multidimensional molecules.

*Example 5.4* Using (5.28), determine the behavior of the mixture of 3 $CH$-molecules in $\mathbb{R}^2$ if the set of behaviors of that molecules $\Phi$ is defined as (5.29) and the set of stoichiometric coefficients $\Lambda$ is defined as (5.30).

$$\Phi = \begin{pmatrix} 2.5 & 4.0 & 7.6 \\ 3.1 & 8.3 & 9.2 \end{pmatrix} \tag{5.29}$$

$$\Lambda = \begin{pmatrix} 0.5 & 0.8 \\ 0.3 & 0.2 \\ 0.1 & 0.6 \end{pmatrix} \tag{5.30}$$

**Solution 5.4** In matrix notation, the set of equations in (5.28) can be rewritten as (5.31):

$$S(\mathbf{x}) = \text{diag}\left(\Lambda^T \Phi^T\right) \tag{5.31}$$

Where, diag$(A)$ represents the vector of elements in the diagonal of a square matrix $A$ and $A^T$ represents the transpose matrix of $A$. Then, the behavior of the mixture of multidimensional molecules is easy to compute using (5.31):

$$S(\mathbf{x}) = \text{diag}\begin{pmatrix} 3.21 & 4.96 \\ 7.36 & 9.66 \end{pmatrix} = (3.21, 9.66) \tag{5.32}$$

Then, the behavior of the mixture of 3 $CH$-molecules is $S(\mathbf{x}) \in \mathbb{R}^2$.

### 5.2.2 *Multidimensional AHN-Algorithm*

Using propositions of Sect. 5.2.1, Algorithm 5.2 can be modified to implement artificial hydrocarbon networks for modeling multiple-inputs-and-multiple-outputs systems $\Sigma = (\mathbf{x}, \mathbf{y})$ with $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{y} \in \mathbb{R}^m$, as shown in Algorithm 5.3. Notice that $k$ refers to the number of molecules, $k_{max}$ is the maximum number of molcules in a compound and $\mu$ is the heuristic (5.9). In addition, several practical implementations in Algorithm 5.3 are described as follows.

**Algorithm 5.3** MULTIDIMENSIONAL-AHN($\Sigma, k_{max}, c_{max}, \varepsilon, \eta$): Algorithm of multidimensional artificial hydrocarbon networks.

**Input:** the system $\Sigma = (\mathbf{x}, \mathbf{y})$, the maximum number of molecules $k_{max} \geq 2$, the maximum number of compounds $c_{max} > 0$, the small tolerance value $\varepsilon > 0$ and the learning rate $\eta$.
**Output:** the structure of mixture $\Gamma$, the set of hydrogen values $H$, the set of stoichiometric coefficients $\Lambda$ and the set of intermolecular distances $\Pi$.

```
$i = 0$
$R = \Sigma$
while $(R > \varepsilon)$ and $(i < c_{max})$ do
    $i = i + 1$
    $k = 2$
    $\mu \to \infty$
    while $\mu > 1$ and $k \leq k_{max}$ do
        $C_i$ =CREATE-COMPOUND($R, k$)
        $[H_i, \Pi_i]$ =OPTIMUM-COMPOUND($R, C_i, k, \eta$)
        $[k, \mu]$ =ENTHALPY-RULE($R, C_i, k$)
    end-while
    $R \leftarrow$ using (4.15)
end-while
$\Gamma \leftarrow \{C_1, ..., C_i\}$
$H \leftarrow \{H_1, ..., H_i\}$
$\Lambda \leftarrow$ minimizing (4.19) using LSE.
$\Pi = \{\Pi_1, ..., \Pi_i\}$
return $\Gamma, H, \Lambda$ and $\Pi$
```

### 5.2.2.1 Creating Compounds

The multidimensional artificial hydrocarbon networks algorithm uses the CREATE-COMPOUND($R, k$) Algorithm (4.1). In particular, that algorithm has to calculate the energy of a signal partition $\Sigma_j = (\mathbf{x}_j, \mathbf{y}_j)$. In practice, this energy signal $E_j$ can be computed using (5.33); where, $[\mathbf{r}_{j-1}, \mathbf{r}_j]$ stands for the interval in $\mathbb{R}^n$ of $\Sigma_j$, $\mathbf{x}_q \in \mathbf{x}$ is an input sample, $\mathbf{y}_q \in \mathbf{y}$ is an output sample and $\|g\|$ stands for the norm of $g$:

$$E_j = \sum_q \|\mathbf{y}_q\|^2 \ , \ \forall \mathbf{x}_q \in [\mathbf{r}_{j-1}, \mathbf{r}_j] \tag{5.33}$$

#### 5.2.2.2 Optimizing Compounds

On the other hand, Algorithm 5.3 uses the OPTIMUM-COMPOUND$(R, C_i, k, \eta)$ Algorithm (4.4). First, it requires to initialize the values of intermolecular distances in $\Pi$, then (5.34) can be used to determine the initial values of intermolecular distances $\mathbf{r}^{jk} \in \Pi$, if $[\mathbf{a}, \mathbf{b}] \subseteq \mathbb{R}^n$ is the interval of the input domain.

$$\mathbf{r}_j - \mathbf{r}_{j-1} = \frac{\mathbf{b} - \mathbf{a}}{k}\ ,\ j = 1, ..., k \tag{5.34}$$

Next, Algorithm (4.4) uses the least squares estimates for finding the best values of hydrogen atoms. In that sense, the objective function $E_i$ for multivariate polynomial functions in (5.35) is used; where, $\varphi_j \in \mathbb{R}^m$ is the $j$-th molecular behavior, $q_j$ is the number of samples in the partition $\Sigma_j$, $\mathbf{x}_{jk} \in \mathbf{x}_j$ is the $j$-th sample of $\mathbf{x}_j$ for all $k = 1, ..., q_j$ and $\mathbf{y}_{jk} \in \mathbf{y}_j$ is the $j$-th sample of $\mathbf{y}_j$ for all $k = 1, ..., q_j$.

$$E_j = \frac{1}{2} \sum_{k=1}^{q_j} \left(\mathbf{y}_{jk} - \varphi_j\left(\mathbf{x}_{jk}\right)\right)^2 \tag{5.35}$$

Finally, the intermolecular distances $\mathbf{r}^{jk} \in \Pi$ have to be updated using the multidimensional rule expressed in (5.36) and (5.37); where, $\mathbf{r}_{t+1}$ is the future value of $\mathbf{r}^{jk}$, $\mathbf{r}_t$ is the current value of $\mathbf{r}^{jk}$, $\Delta r_t$ is the change in length of bond $\mathbf{r}^{jk}$, $I$ is the identity matrix of dimension $n$, diag$(A)$ is the diagonal matrix of $A$, $0 < \eta < 1$ is the step size, and $E_{1t}, E_{2t}$ are the current error functions of two adjacent molecules $M_j, M_k$ calculated as (5.35).

$$\mathbf{r}_{t+1} = \mathbf{r}_t + \text{diag}\left(\Delta r_t I\right) \tag{5.36}$$

$$\Delta r_t = -\eta\left(E_{1t} - E_{2t}\right) \tag{5.37}$$

#### 5.2.2.3 Calculating the Enthalpy Rule

Also, Algorithm 5.3 uses the ENTHALPY-RULE$(R, C_i, k)$ Algorithm (5.1) to optimize the number of molecules in a compound. In fact, it calculates the enthalpy of the system $\Delta H^\circ_\Sigma$ and also the enthalpy of compound $\Delta H^\circ_C$. This rule can be derived by applying (5.33) for both enthalpies in the interval $[\mathbf{a}, \mathbf{b}] \subseteq \mathbb{R}^n$.

*Example 5.5* Using Algorithm 5.3, build an artificial hydrocarbon network with $k_{max} = 2$ and $c_{max} = 1$ of the system (5.38). Consider a step size of $\eta = 0.01$ and a tolerance value of $\varepsilon = 0.1$.

$$y_1 = x_1 x_2\ ,\ x_1 \in [0, 2]\ ,\ x_2 \in [0, 2] \tag{5.38}$$

**Solution 5.5** Initialize $i = 0$ and the residue $R = \Sigma$.

For $i = 0$:

$i = 1$.

$k = 2$.

$\mu \rightarrow \infty$.

For$j = 0$ :

$C_1 = (\{CH_3, CH_3\}, \{1\})$ using Algorithm 4.1.

$H_1 = \{\mathbf{h}_{11}, \mathbf{h}_{12}, \mathbf{h}_{13}, \mathbf{h}_{21}, \mathbf{h}_{22}, \mathbf{h}_{23}\}$ with:

$\mathbf{h}_{11} = (1.50, 0.00)$

$\mathbf{h}_{12} = (0.44, 0.23 + 0.05i)$

$\mathbf{h}_{13} = (0.27, 0.23 - 0.05i)$

$\mathbf{h}_{21} = (0.48, 0.00)$

$\mathbf{h}_{22} = (0.16 + 0.17i, 0.11 + 0.13i)$

$\mathbf{h}_{23} = (0.16 - 0.17i, 0.11 + 0.13i)$ using Algorithm 4.4 with (5.35), and

$\Pi_1 = (0.9923, 0.9844)$ using (5.36) and (5.37).

$k = 3$ and $m = \frac{338.1953}{336.1423} = 1.0061$ using Algorithm 5.1.

$\mu > 1$ and $k > k_{max}$, this ends the loop.

$\|R\| = 18.33$ using (4.15).

$\|R\| > \varepsilon$ and $i = c_{max}$, this ends the loop.

Finally,

$\Gamma = (C_1)$

$H = (H_1)$

$\Lambda = (1.0)$ because there is one compound with one output.

$\Pi = (\Pi_1)$

Actually, the overall response of the artificial hydrocarbon network is shown in Fig. 5.7 and the AHN-structure is depicted in Fig. 5.8.

## 5.3 Recursive Networks Using Aromatic Compounds

Until now, artificial hydrocarbon networks have been structured in linear chains of $CH$-primitive molecules. However, these topologies do not present recursive properties. In that sense, a cyclic network topology is proposed to handle dynamic systems. In fact, these recursive networks are inspired on chemical aromatic compounds, i.e. arenes. Figure 5.9 shows the proposed topology. It is remarkable to say that the recursive $CH$ molecule is inspired on arenes because it is the most stable structure of cyclic compounds in nature.

As noted in Fig. 5.9, the recursive AHN-structure can be treated as another $CH$-primitive molecule as the following Proposition 5.7 describes:

**Proposition 5.7** (***recursive CH molecules***) *Let $M$ be a $CH$-molecule of the form as Fig. 5.9 with $a_c = C_1$. Also, let $\varphi$ be the behavior of molecule $M$ due to an input signal $x$ with $|x| < 1$. Then, the behavior of $\varphi$ holds:*

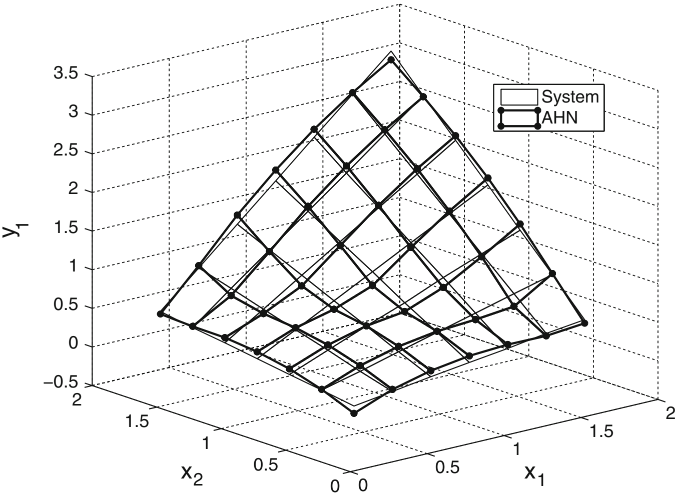


**Fig. 5.7** Model obtained from Algorithm 5.3 in Example 5.5

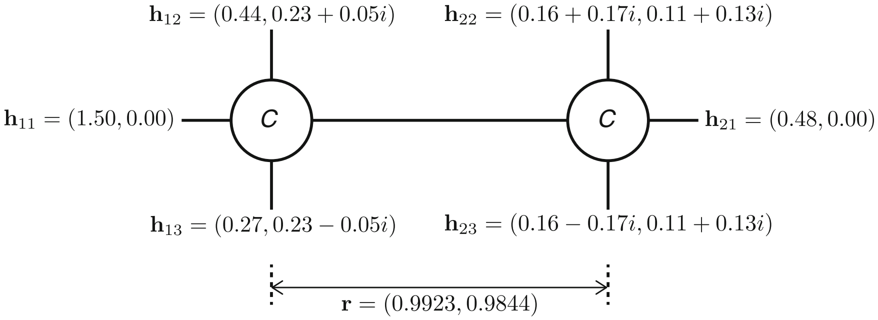

$\mathbf{h}_{12} = (0.44, 0.23 + 0.05i)$
$\mathbf{h}_{22} = (0.16 + 0.17i, 0.11 + 0.13i)$
$\mathbf{h}_{11} = (1.50, 0.00)$
C
C
$\mathbf{h}_{21} = (0.48, 0.00)$
$\mathbf{h}_{13} = (0.27, 0.23 - 0.05i)$
$\mathbf{h}_{23} = (0.16 - 0.17i, 0.11 + 0.13i)$
$\mathbf{r} = (0.9923, 0.9844)$

**Fig. 5.8** AHN-structure of Example 5.5

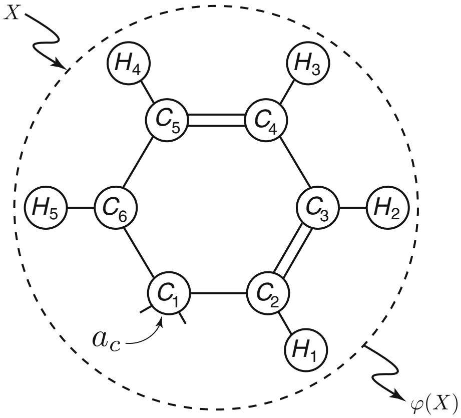
$$v_{H_i} = h_i \ , \ h_i \in \mathbb{C} \tag{5.39}$$

$$\varphi(x_n) = \sum_{i=1}^{N-1} h_i \cdot x_{n-i} \tag{5.40}$$

*Where, N is the number of carbon atoms in M, and n is the current state.*

In this case, the parameters in the recursive *CH* molecule do not mean any roots in a polynomial. In contrast, these hydrogen parameters are subjected to compute the lastest value of themodeled signal in terms of some values locally stored in carbon

**Fig. 5.9** Topology of a recursive artificial hydrocarbon network inspired on arenes



atoms. For instance, it can be seen as a difference equation with coefficients equal to hydrogen parameters.

## References

1. Ganguly J (2009) Thermodynamics in earth and planetary sciences. Springer, Berlin
2. Li H, Higashi H, Tamura K (2006) Estimation of boiling and melting points of light, heavy and complex hydrocarbons by means of a modified group vector space method. ELSEVIER Fluid Phase Equilib 239(2):213–222

# Chapter 6
# Notes on Modeling Problems Using Artificial Hydrocarbon Networks

This chapter introduces several notes on using artificial hydrocarbon networks (AHNs) for modeling problems. In particular, it discusses some aspects on modeling univariate and multivariate systems, and designing linear and nonlinear classifiers using the AHN-algorithm. In addition, few inference and clustering applications are described. Finally, a review of the most important characteristics on artificial hydrocarbon networks in real-world applications are covered like how to inherit information with molecules, how to use information of parameters in AHN-structures and how to improve the training process of artificial hydrocarbon networks implementing a catalog of artificial hydrocarbon compounds.

## 6.1 Approximation Problems

This section covers the usage of artificial hydrocarbon networks to approximate univariate and multivariate systems. Without loss of generalization, let $\Sigma$ be an unknown system represented as a set of multivariate functions $f_j$ of the form as (6.1); where, $\{x_1, ..., x_i, ..., x_n\}$ is a set of attribute variables and $\{y_1, ..., y_j, ..., y_m\}$ is a set of target variables.

$$y_j = f_j\left(x_1, ..., x_n\right) \ , \ \forall j = 1, ..., m \tag{6.1}$$

In compact notation, let $\mathbf{x} = (x_1, ..., x_n)$ be the row vector of attribute variables and $\mathbf{y} = (y_1, ..., y_m)$ be the row vector of target variables such that (6.1) can be rewritten as (6.2):

$$\mathbf{y} = F\left(\mathbf{x}\right) \tag{6.2}$$

With $F$ as a row vector of functions $f_i$ like (6.3); where, $A^T$ represents the transpose of $A$.

H. Ponce-Espinosa et al., *Artificial Organic Networks*,
Studies in Computational Intelligence 521, DOI: 10.1007/978-3-319-02472-1_6,

$$F(\mathbf{x}) = \begin{pmatrix} f_1(\mathbf{x}) \\ \cdots \\ f_m(\mathbf{x}) \end{pmatrix}^T \tag{6.3}$$

Moreover, suppose that there are $q$ samples of inputs and outputs. Each attribute variable is a column vector of samples $\mathbf{x}_i = (x_{1i}, ..., x_{qi})$ and each target variable is a column vector of samples $\mathbf{y}_j = (y_{1j}, ..., y_{qj})$. Then, let $X$ be a matrix $q \times n$ of input samples and $Y$ be a matrix $q \times m$ of output samples as (6.4) and (6.5), respectively.

$$X = \begin{pmatrix} x_{11} & \cdots & x_{1n} \\ \vdots & \ddots & \vdots \\ x_{q1} & \cdots & x_{qn} \end{pmatrix} \tag{6.4}$$

$$Y = \begin{pmatrix} y_{11} & \cdots & y_{1m} \\ \vdots & \ddots & \vdots \\ y_{q1} & \cdots & y_{qm} \end{pmatrix} \tag{6.5}$$

Let say that a system $\Sigma = (X, Y)$ is equivalent to (6.2), i.e. $\Sigma(X, Y) = F$. Then, $\Sigma$ can be represented with a model $\hat{F}$ of the form as (6.6); where, *AHN* is an artificial hydrocarbon network, $\varepsilon > 0$ is a small real number representing the maximum error and $\|g\|$ stands for the norm of $g$.

$$\left\| Y - \hat{F} \right\| = \|Y - AHN\| \leq \varepsilon \tag{6.6}$$

Following, some notes on modeling univariate and multivariate systems, i.e. univariate and multivariate set of functions $F$, with artificial hydrocarbon networks are discussed.

### *6.1.1 Approximation of Univariate Functions*

For instance, suppose that there is an unknown single-input-and-single-output system $\Sigma = (X, Y)$ with a column vector of input samples $X = \mathbf{x}_1 = (x_{11}, ..., x_{q1})$ and a column vector of output samples $Y = \mathbf{y}_1 = (y_{11}, ..., y_{q1})$. Then, (6.2) is reduced to (6.7) with an unknown function $f$.

$$y_1 = f(x_1) \tag{6.7}$$

Thus, an artificial hydrocarbon network *AHN* will be used as the model $\hat{f}$ of $f$. Using Algorithm 4.5 and $X$ and $Y$ vector samples, an artificial hydrocarbon network *AHN* is trained and then used as an inference system like (6.7).

*Example 6.1* Let $f$ be the exact behavior of $\Sigma$ as expressed in (6.8). Find a model $\hat{f}$ of $f$ using artificial hydrocarbon networks in the input domain $x \in [-1, 2)$.

$$f(x) = x^5 - 4x^2 + 1\ ,\ \ x \in [-1, 2) \tag{6.8}$$

**Solution 6.1** 3000-sample pairs $(\mathbf{x}_1, \mathbf{y}_1)$ were uniformly obtained in the interval $x \in [-1, 2)$. Then, running Algorithm 4.5 a model $\hat{f}$ is obtained using the 3000-sample pairs, a maximum number of molecules $n_{max} = 6$, a maximum number of compounds $c_{max} = 1$, a tolerance $\varepsilon = 1 \times 10^{-4}$, and a step size $\eta = 0.01$. Figure 6.1 shows a comparison between function $f$ and the model $\hat{f}$ with dashed lines representing the bounds between molecules in the AHN-structure. In particular, this model has an absolute squared error value of 1.3452.

Notice that the algorithm found 4 molecules with a compound as depicted in Fig. 6.2. As shown in the latter figure, hydrogen values represent the roots of molecular behaviors, the superscript numbers of carbon atoms represent a gain in the molecular behavior, so-called the carbon atom value $v_C$. For example, this simple AHN-model can be written mathematically as (6.9). Since there is one compound, there is one stoichiometric coefficient $\alpha_1 = 1$; thus, the final model $\hat{f}$ is expressed as (6.10).

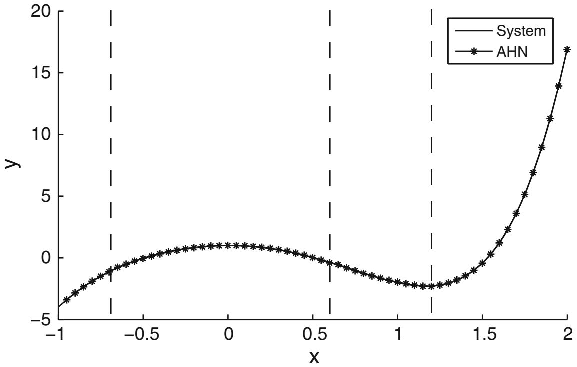


**Fig. 6.1** Comparison of function $f$ in (6.8) and the response of its AHN-model

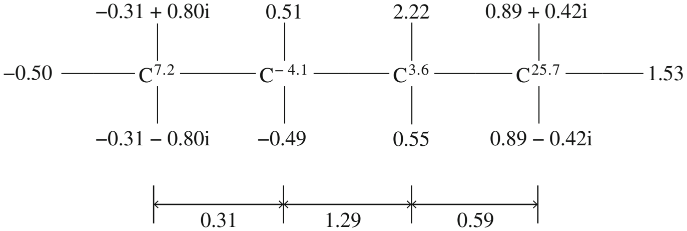


**Fig. 6.2** Artificial hydrocarbon network structure of model $\hat{f}$ of function $f$ in (6.8)

$$\psi(x) = \begin{cases} 7.17\,(x+0.50)\,(x+0.31-0.80i)\,(x+0.31+0.80i) & -1 \le x < -0.69 \\ -4.08\,(x-0.51)\,(x+0.49) & -0.69 \le x < 0.60 \\ 3.58\,(x-2.22)\,(x-0.55) & 0.60 \le x < 1.19 \\ 25.67\,(x-1.53)\,(x-0.89-0.42i)\,(x+0.89+0.42i) & 1.19 \le x < 2 \end{cases} \tag{6.9}$$

$$\hat{f}(x) = \alpha_1 \psi(x)\ ,\ \ x \in [-1, 2) \tag{6.10}$$

It is important to highlight that in Example 6.1 Algorithm 4.5 uses the second model of *CH*-molecules (Proposition 4.2) that defines a molecular behavior in a product form, and the first chemical rule (Proposition 4.6) that designs a saturated linear chain of *CH*-primitive molecules.

*Example 6.2* Let $f$ be the behavior of $\Sigma$ as expressed in (6.11); where, $g_N$ stands for a normalized random function. Find a model $\hat{f}$ of $f$ using artificial hydrocarbon networks in the input domain $x \in [-1, 2)$.

$$f(x) = x^5 - 4x^2 + 1 + 2g_N,\ \ x \in [-1, 2) \tag{6.11}$$

**Solution 6.2** 1000-sample pairs $(\mathbf{x}_1, \mathbf{y}_1)$ were uniformly obtained in the interval $x \in [-1, 2)$. Then, running Algorithm 4.5 a model $\hat{f}$ is obtained using the 1000-sample pairs, a maximum number of molecules $n_{max} = 6$, a maximum number of compounds $c_{max} = 1$, a tolerance $\varepsilon = 1 \times 10^{-4}$, and a step size $\eta = 0.001$. Figure 6.3 shows a comparison between function $f$ and the model $\hat{f}$ with dashed lines representing the bounds between molecules in the AHN-structure. In particular, this model has an absolute squared error value of 334.05 due to the noise of the system.

Notice that the algorithm found 4 molecules with a compound as depicted in Fig. 6.4. This result shows that artificial hydrocarbon networks act as filters over the data. However, the profile of molecules slightly changed in terms of intermolecular distances as obtained in Example 6.1. Again, the algorithm computes a

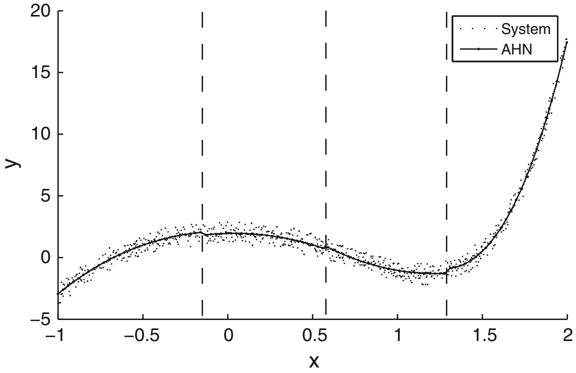


**Fig. 6.3** Comparison of function $f$ in (6.11) and the response of its AHN-model

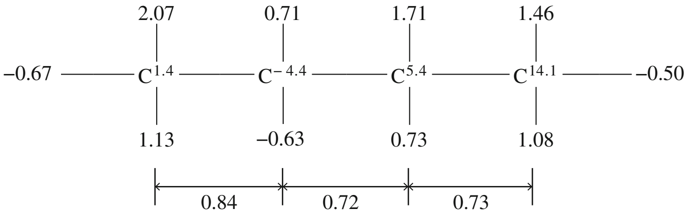


**Fig. 6.4** Artificial hydrocarbon network structure of model $\hat{f}$ of function $f$ in (6.11)

saturated linear chain of $CH$-primitive molecules with the product form based molecular behaviors.

An important property of artificial hydrocarbon networks algorithm is its ability to find discontinuities in the system using intermolecular distances, as shown in the next example.

*Example 6.3* Let $f$ be the behavior of $\Sigma$ as expressed in (6.12). Find a model $\hat{f}$ of $f$ using artificial hydrocarbon networks in the input domain $x \in [-1, 1]$.

$$f(x) = \begin{cases} \arctan(\pi x) & -1.0 \leq x < 0.1 \\ \sin(\pi x) & 0.1 \leq x < 0.9 \\ \cos(\pi x) & 0.9 \leq x \leq 1.0 \end{cases} \tag{6.12}$$

**Solution 6.3** 200-sample pairs $(\mathbf{x}_1, \mathbf{y}_1)$ were uniformly obtained in the interval $x \in [-1, 1]$. Then, running Algorithm 4.5 a model $\hat{f}$ is obtained using the 200-sample pairs, a maximum number of molecules $n_{max} = 6$, a maximum number of compounds $c_{max} = 1$, a tolerance $\varepsilon = 1 \times 10^{-4}$, and a step size $\eta = 0.01$. Figure 6.5 shows a comparison between function $f$ and the model $\hat{f}$ with dashed lines representing

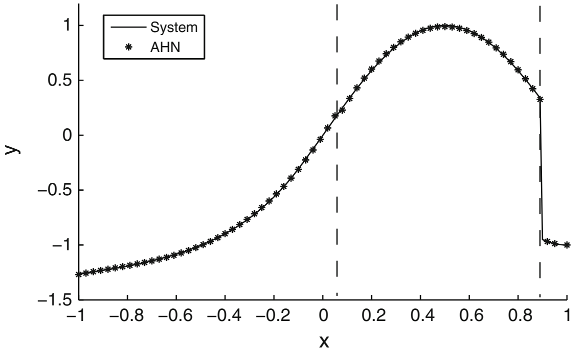


**Fig. 6.5** Comparison of function $f$ in (6.12) and the response of its AHN-model

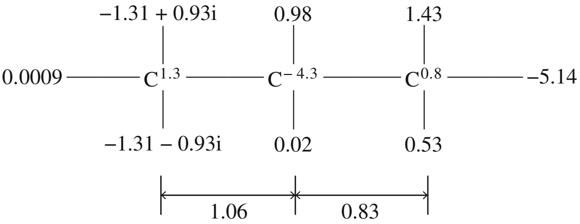


**Fig. 6.6** Artificial hydrocarbon network structure of model $\hat{f}$ of function $f$ in (6.12)

the bounds between molecules in the AHN-structure. In particular, this model has an absolute squared error value of $1.03 \times 10^{-2}$.

Three carbon atoms in the artificial hydrocarbon compound were optimally found, as shown in Fig. 6.6. Moreover, the bounds of intermolecular distances captured the discontinuities of $f$ in (6.12). For instance, the first discontinuity ocurrs at $x = 0.1$ and the first bound is located at $x_L = 0.06$. The second discontinuity occurs at $x = 0.9$ and the second bound is located at $x_L = 0.89$.

*Example 6.4* Suppose that $f$ is a periodic function representing the behavior of $\Sigma$ as written in (6.13). Find a model $\hat{f}$ of $f$ using artificial hydrocarbon networks in the input domain $x \in [0, 5]$.

$$f(x) = \sin(\pi x), \ x \in [0, 5] \tag{6.13}$$

**Solution 6.4** 500-sample pairs $(\mathbf{x}_1, \mathbf{y}_1)$ were uniformly obtained in the interval $x \in [0, 5]$. Then, running Algorithm 4.5 a model $\hat{f}$ is obtained using the 500-sample pairs, a maximum number of molecules $n_{max} = 6$, a maximum number of compounds $c_{max} = 1$, a tolerance $\varepsilon = 1 \times 10^{-4}$, and a step size $\eta = 0.001$. Figure 6.7 shows

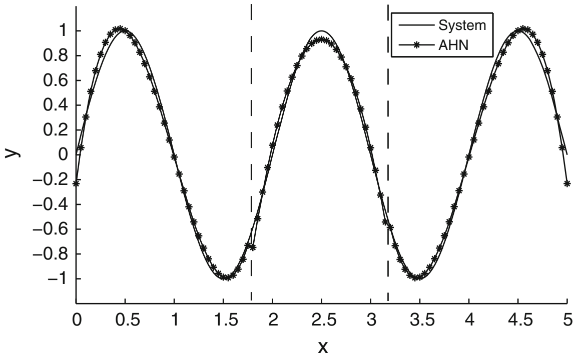


**Fig. 6.7** Comparison of function $f$ in (6.13) and the response of its AHN-model

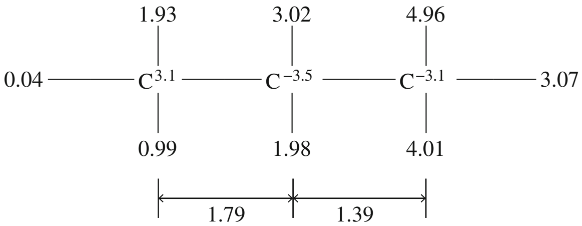


**Fig. 6.8** Artificial hydrocarbon network structure of model $\hat{f}$ of function $f$ in (6.13)

a comparison between function $f$ and the model $\hat{f}$ with dashed lines representing the bounds between molecules in the AHN-structure. In particular, this model has an absolute squared error value of 1.7978.

Notice that three $CH$-primitive molecules can model $f$ as depicted in Fig. 6.8. At last, this example demonstrates that artificial hydrocarbon networks can also model periodic functions easily.

### *6.1.2 Approximation of Multivariate Functions*

Until now, artificial hydrocarbon networks have been used for univariate systems. In this section, let suppose that there exists an unknown multiple-inputs-and-multiple-outputs system $\Sigma = (X, Y)$ expressed as (6.2). Thus, an artificial hydrocarbon network $AHN$ will be used as the model $\hat{F}$ of $F$, using Algorithm 5.3 and $X$ and $Y$ matrix samples.

*Example 6.5* Suppose that $F$ is a single function representing the behavior of $\Sigma$ as written in (6.14). Find a model $\hat{F}$ of $F$ using artificial hydrocarbon networks in the input domain $x_1 \in [0, 3]$ and $x_2 \in [0, 3]$.

$$F = y_1 = \frac{\sin(\pi x_1)}{\pi x_1} + \frac{\sin(\pi x_2)}{\pi x_2} \tag{6.14}$$

**Solution 6.5** 1000-sample tuples $(\mathbf{x}_1, \mathbf{x}_2, \mathbf{y}_1)$ were randomly obtained in the proper input space. Then, running Algorithm 5.3 a model $\hat{F}$ is obtained using the 1000-sample tuples, a maximum number of molecules $n_{max} = 6$, a maximum number of compounds $c_{max} = 1$, a tolerance $\varepsilon = 1 \times 10^{-4}$, and a step size $\eta = 0.01$. Figure 6.9 shows a comparison between function $F$ and the model $\hat{F}$. In particular, this model has an absolute squared norm error value of 503.89.

Three multidimensional $CH$-primitive molecules can model $F$ as depicted in Fig. 6.10. As noted, both surfaces $F$ and $\hat{F}$ are similar, proving that AHNs can model multivariate systems. It is remarkable to say that the way systems are splitted, i.e.

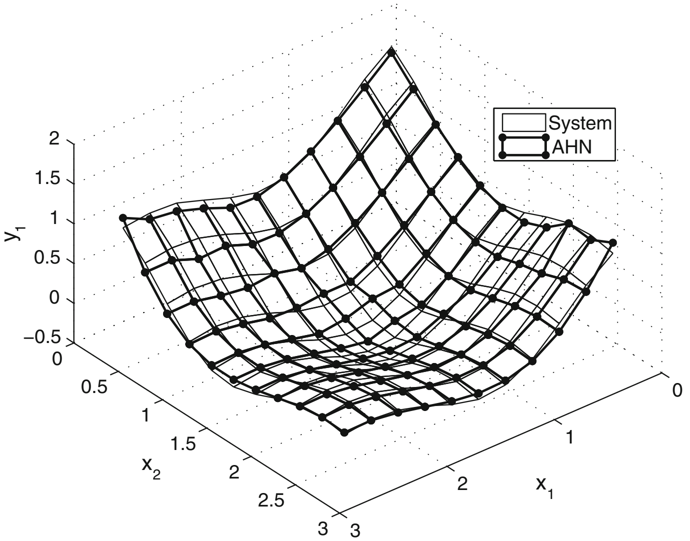


**Fig. 6.9** Comparison of function $F$ in (6.14) and the response of its AHN-model

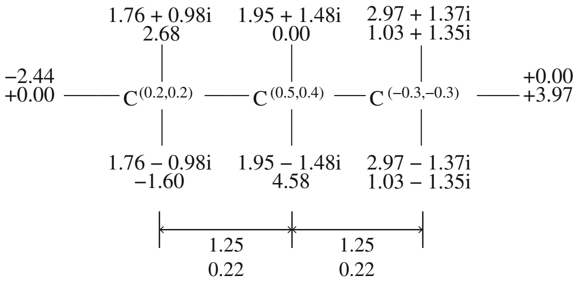


**Fig. 6.10** Artificial hydrocarbon network structure of model $\hat{F}$ of function $F$ in (6.14)

the meaning of intermolecular distances, gives different responses of how molecules capture the information.

In this particular example, bounds of molecules in 1-dimension is extended to hyperplanes of the form as (6.15); where, $L_i \in \mathbb{R}^n$ represents the $i$-th bound of molecules and $\{x_1, ..., x_n\}$ is the set of attribute variables. Then, any given point $\mathbf{x}_p = (x_{p1}, ..., x_{pn})$ is part of a molecule $M_k$ if (6.16) holds, for all positive values of $L_{(k-1)} \in \mathbb{R}^n$ and $L_k \in \mathbb{R}^n$:

$$\frac{x_1}{L_1} + \cdots + \frac{x_n}{L_n} = 1 \tag{6.15}$$

$$\frac{x_{p1}}{L_{(k-1)1}} + \cdots + \frac{x_{pn}}{L_{(k-1)n}} \leq 1 \leq \frac{x_{p1}}{L_{k1}} + \cdots + \frac{x_{pn}}{L_{kn}} \tag{6.16}$$

*Example 6.6* Suppose that $F$ is a single function representing the nonlinear behavior of $\Sigma$ like (6.17). Find a model $\hat{F}$ of $F$ using artificial hydrocarbon networks in the input domain $(x_1, x_2) \in \mathbb{R}^2$.

$$F : \begin{cases} x_1 = \cos(t) \\ x_2 = t \\ y_1 = \sin(t) \end{cases} , \; t \in [0, 15] \tag{6.17}$$

**Solution 6.6** 1500-sample tuples $(\mathbf{x}_1, \mathbf{x}_2, \mathbf{y}_1)$ were uniformly obtained in the proper input space. Then, running Algorithm 5.3 a model $\hat{F}$ is obtained using the 1500-sample tuples, a maximum number of molecules $n_{max} = 6$, a maximum number of compounds $c_{max} = 1$, a tolerance $\varepsilon = 1 \times 10^{-4}$, and a step size $\eta = 0.01$. Figure 6.11 shows a comparison between function $F$ and the model $\hat{F}$. In particular, this model has an absolute squared norm error value of $1.1 \times 10^{-2}$.

Five multidimensional $CH$-primitive molecules model $F$ as depicted in Fig. 6.12. Moreover, artificial hydrocarbon networks can capture nonlinear and coupled attribute variables in systems.

In comparison with Example 6.5, the splitting procedure defines a center of molecule $C_{M_i} \in \mathbb{R}^n$ as the $i$-th midpoint between two adjacent bounds of molecules $L_{(i-1)}, L_i$ as (6.18). In addition, a sample value $(\mathbf{x}_k, \mathbf{y}_k) \in \mathbb{R}^n \times \mathbb{R}^m$ is part of a molecule $M_i$ if (6.19) holds; where, $\Sigma_i$ is the set of observation tuples in molecule $M_i$:

$$C_{M_i} = \frac{L_{(i-1)} + L_i}{2} \tag{6.18}$$

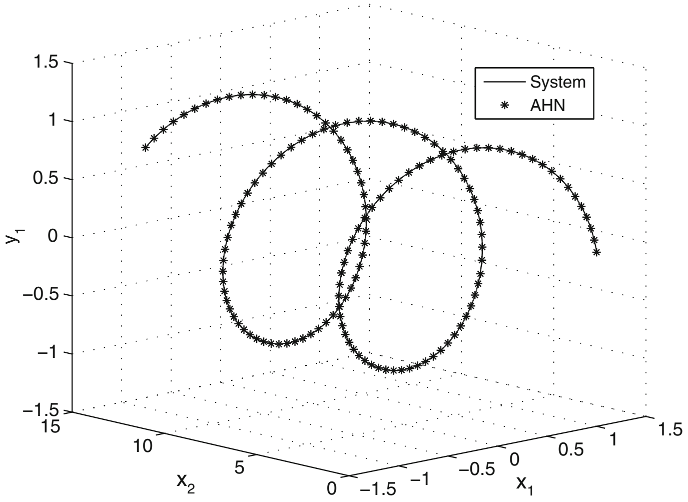


**Fig. 6.11** Comparison of function $F$ in (6.17) and the response of its AHN-model

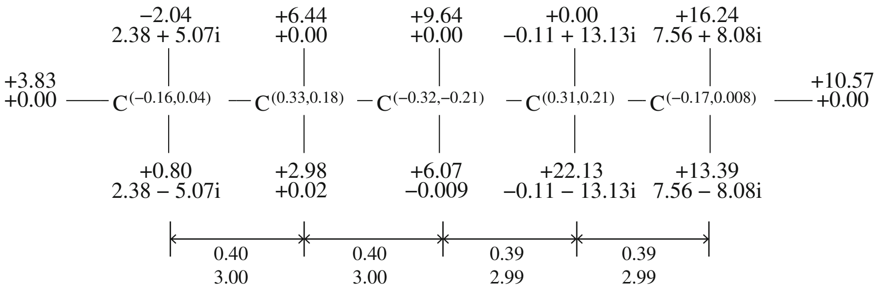


**Fig. 6.12** Artificial hydrocarbon network structure of model $\hat{F}$ of function $F$ in (6.17)

$$\Sigma_i = \left\{ (\mathbf{x}_k, \mathbf{y}_k) \in \mathbb{R}^n \times \mathbb{R}^m | i = \arg\min_j \left( \mathbf{x}_k - C_{M_j} \right),\ \forall j = 1, ..., n \right\} \tag{6.19}$$

Currently, artificial hydrocarbon networks have been used for single-input-and-single-output (SISO) and for multiple-inputs-and-single-output (MISO) systems. In the following example, a demonstration of effectiveness on predicting multiple-inputs-and-multiple-outputs (MIMO) systems with artificial hydrocarbon networks is presented.

*Example 6.7* Suppose that $F$ is a vector of functions representing the nonlinear behavior of $\Sigma$ like (6.20). Find a model $\hat{F} = \left(\hat{f}_1, \hat{f}_2\right)$ of $F$ using artificial hydrocarbon networks in the input domain $(x_1, x_2) \in \mathbb{R}^2$.

$$F = \begin{pmatrix} f_1(\mathbf{x}) = \cos(x_1)\cos(x_2) \\ f_2(\mathbf{x}) = (x_1 x_2)^2 \end{pmatrix}^T \tag{6.20}$$

**Solution 6.7** 1000-sample tuples $(\mathbf{x}_1, \mathbf{x}_2, \mathbf{y}_1, \mathbf{y}_2)$ were randomly obtained in the proper input space. Then, running Algorithm 5.3 a model $\hat{F}$ is obtained using the 1000-sample tuples, a maximum number of molecules $n_{max} = 6$, a maximum number of compounds $c_{max} = 1$, a tolerance $\varepsilon = 1 \times 10^{-4}$, and a step size $\eta = 0.01$. Figure 6.13 shows a comparison between function $f_1$ and the model $\hat{f}_1$ and Fig. 1.2 shows a comparison between function $f_2$ and the model $\hat{f}_2$. In particular, the overall model has an absolute squared norm error value of 309.94 (Fig. 6.14).

Five multidimensional $CH$-primitive molecules model $F$ as depicted in Fig. 6.15. Moreover, artificial hydrocarbon networks can capture nonlinear and coupled attribute variables in systems. Again, (6.19) was used for partitioning the system. As shown in this example, training an artificial hydrocarbon network for multivariate systems is still simple. Notice that multidimensional artificial hydrocarbon networks can be decomposed into $m$ AHN-structures; where, $m$ denotes the dimensionality of target variables. However, the structure of these $m$ artificial hydrocarbon networks are attached by multidimensional molecules; thus, it does not imply that $m$ MISO systems can be used for training $m$ different AHN-structures because in some cases, the

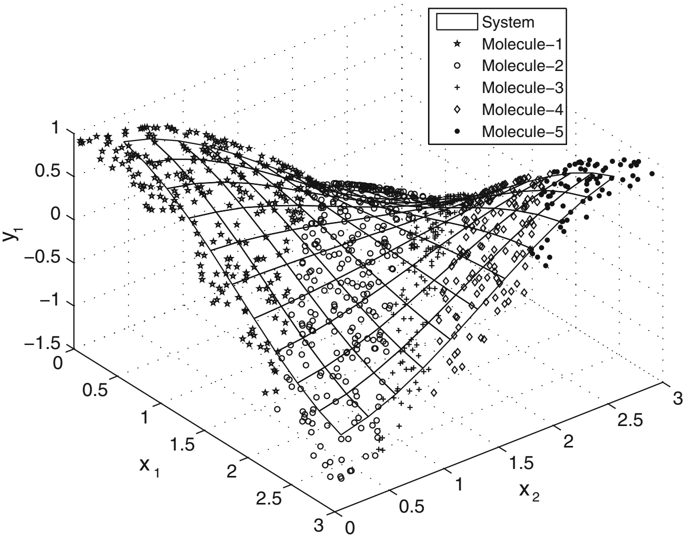


**Fig. 6.13** Comparison of function $f_1$ in (6.20) and the response of its AHN-model $\hat{f}_1$

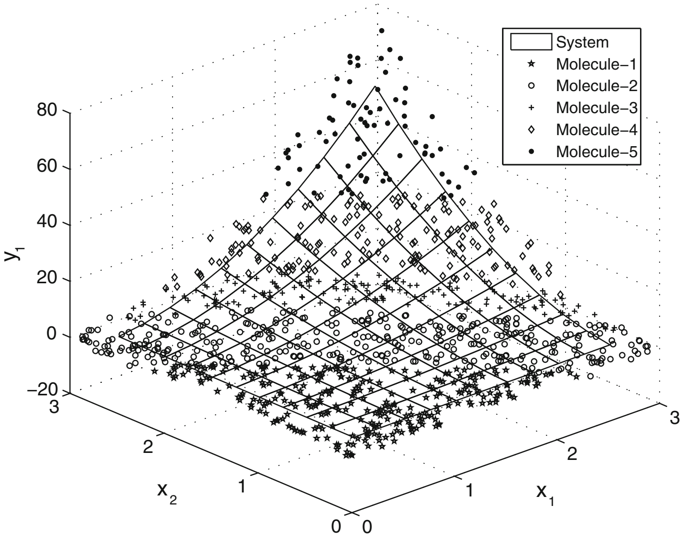


**Fig. 6.14** Comparison of function $f_2$ in (6.20) and the response of its AHN-model $\hat{f}_2$

$m$ AHN models could not have the same number of molecules and/or noncovalent bonds (i.e. order and length of bondings).

In contrast, the same example were done using artificial neural networks (ANN). In that case, a feedforward multilayer neural network was used with 6-hidden neurons with hyperbolic-tangent-sigmoid activation functions like (6.21) and 2-output

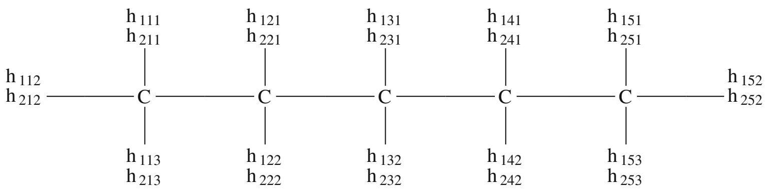


**Fig. 6.15** Artificial hydrocarbon network structure of model $\hat{F}$ of function $F$ in (6.20)

neurons with linear activation fuctions like (6.22); where, $x$ represents an input of the activation function and $e$ stands for the exponential function.

$$f(x) = \frac{2}{1+e^{-2x}} - 1 \tag{6.21}$$

$$f(x) = x \tag{6.22}$$

The backpropagation algorithm was used to train it. After 193 epochs, the response of the artificial neural network was obtained, as shown in Figs. 6.16 and 6.17 with an absolute squared norm error value of 115.75. The same attribute variables and target variables from the example fed the neural network. Notice that in both ANN and AHN models, the result is very similar. However, the weights of the neural network do not represent any information about the system, while intermolecular distances and hydrogen values give a partial understanding of the system. For example, the latter values can be used as metadata for generating other processes (refer to Chap. 7).

## 6.2 Clustering Problems

The clustering problem can be easily treated with artificial hydrocarbon networks using a target variable $\mathbf{y}_1$ as a categorical variable using for labeling attribute variables $\mathbf{x} = (x_1, ..., x_n)$. In that sense, this section introduces the design of linear and nonlinear classifiers.

### 6.2.1 Linear Classifiers

*Example 6.8* Suppose that $f$ represents a classification process of $\Sigma$ as expressed in (6.23). Find a classifier $\hat{f}$ of $f$ using artificial hydrocarbon networks in the input domain $(x_1, x_2) \in \mathbb{R}^2$.

$$f = \begin{cases} 1 & \frac{x_1}{0.7} + \frac{x_2}{0.6} < 1 \\ 0 & otherwise \end{cases} \tag{6.23}$$

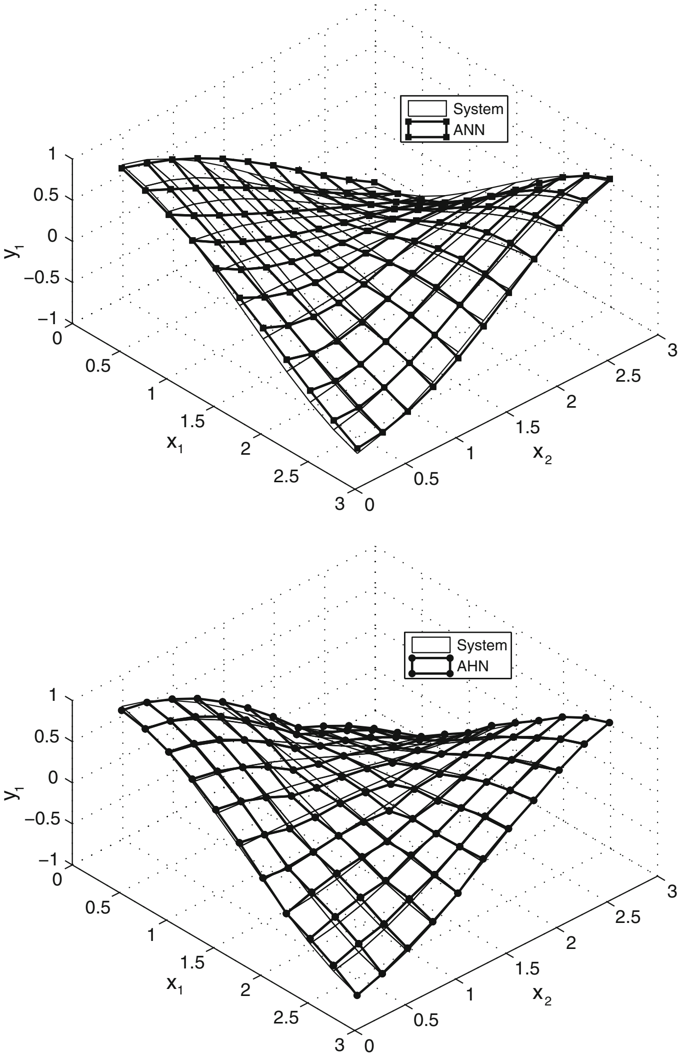


**Fig. 6.16** Comparison between the model $\hat{f}_1$ from an artificial neural network (ANN) and an artificial hydrocarbon network (AHN)

**Solution 6.8** Firstly, note that (6.23) classifies data over the line (6.24):

$$x_2 = -\frac{0.6}{0.7}x_1 + 0.6 \tag{6.24}$$

Then, a linear artificial hydrocarbon network classifier $\hat{f}$ was trained with 1000 random samples $(\mathbf{x}_1, \mathbf{x}_2, \mathbf{y}_1)$; where, $\mathbf{y}_1$ is the label of a given attribute variable, and

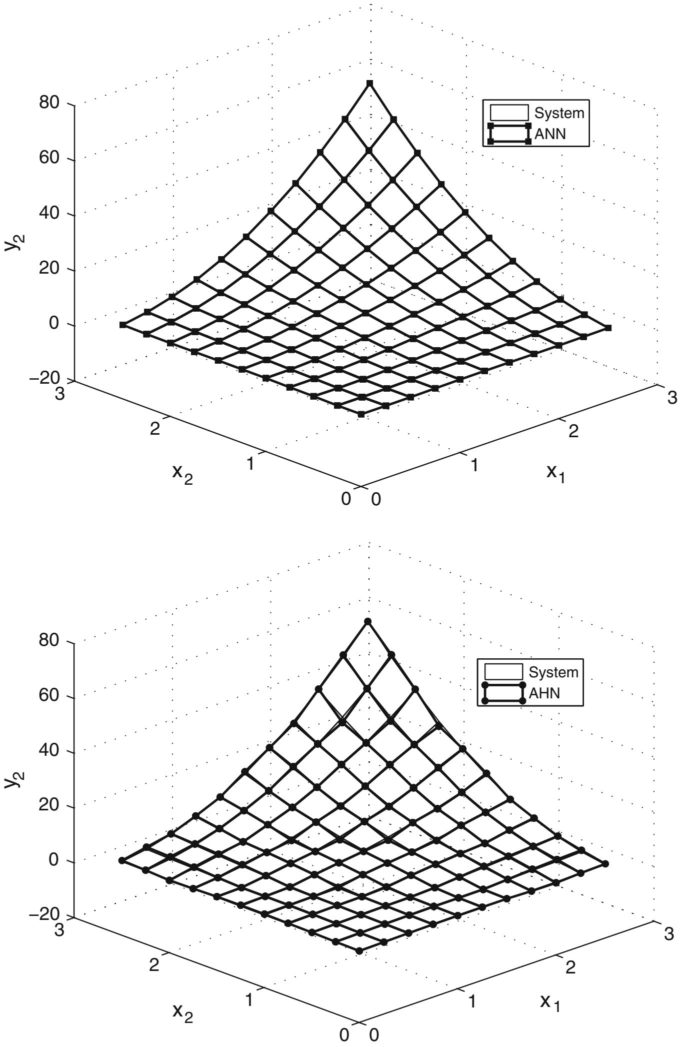


**Fig. 6.17** Comparison between the model $\hat{f}_2$ from an artificial neural network (ANN) and an artificial hydrocarbon network (AHN)

$\mathbf{x}_1$ and $\mathbf{x}_2$ are in the interval [0, 1]. Then, running Algorithm 5.3 a model $\hat{f}$ was obtained using the training data with a maximum number of molecules $n_{max} = 4$, a maximum number of compounds $c_{max} = 1$, a tolerance $\varepsilon = 1 \times 10^{-4}$, and a step size $\eta = 0.001$. Figure 6.18 shows the response of the linear AHN-classifier $\hat{f}$ and the line (6.24). In particular, the classifier has a misclassification rate of 0 %.

The linear AHN-classifier is depicted in Fig. 6.19. As noted, two *CH*-primitive molecules clusters data as (6.23). Actually, the bounds of molecules $L$ are used as

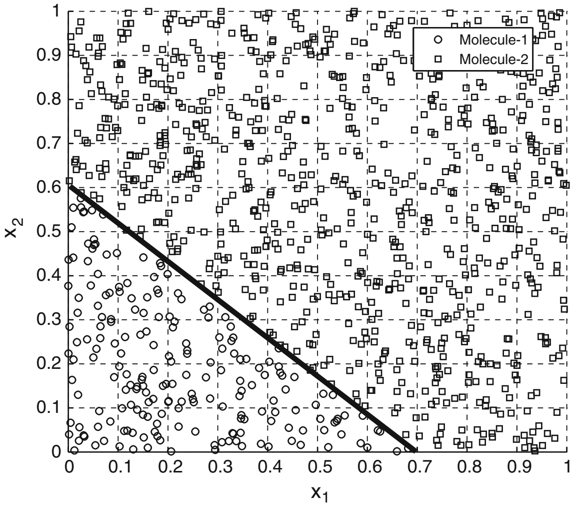
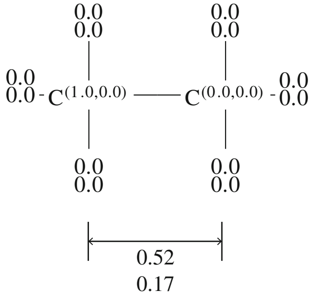


**Fig. 6.18** Response of the linear AHN-classifier $\hat{f}$ of $f$ classifier over the black line (6.24)

**Fig. 6.19** Artificial hydrocarbon network structure of linear classifier $\hat{f}$ of function $f$ in (6.23)



centers of molecules and partitions are calculated as (6.19). Notice that these centers are separated by intermolecular bonds **r**.

## 6.2.2 Nonlinear Classifiers

In order to demonstrate nonlinear artificial hydrocarbon networks classifiers, two examples are presented. The first example discusses a nonlinear classifier for a XOR classifier and the second example uses the well-known Fisher's Iris dataset.

*Example 6.9* Consider the XOR boolean operator $f$ between two attribute variables $x_1, x_2$ as written in (6.25); where, $a \cdot b$ stands for the AND operator between $a$ and $b$, and $\bar{a}$ stands for the NOT operator of $a$. Also, suppose that the logical value *false* are values in the interval [0.0, 0.3] and the logical value *true* are values in the interval [0.7, 1.0]. Then, build a nonlinear classifier $\hat{f}$ of $f$ using artificial hydrocarbon networks.

$$f = x_1 \cdot \bar{x_2} + \bar{x}_1 \cdot x_2 \tag{6.25}$$

**Solution 6.9** A nonlinear artificial hydrocarbon network classifier $\hat{f}$ was built using 200 random samples of the form $(\mathbf{x}_1, \mathbf{x}_2, \mathbf{y}_1)$; where, $\mathbf{y}_1$ is the label of the logical value $f$. Then, running Algorithm 5.3 a model $\hat{f}$ was obtained using the training data with a maximum number of molecules $n_{max} = 5$, a maximum number of compounds $c_{max} = 1$, a tolerance $\varepsilon = 1 \times 10^{-4}$, and a step size $\eta = 0.001$. Figure 6.20 shows the classification of 300 random samples as a testing set using the obtained nonlinear AHN-classifier $\hat{f}$. In particular, the classifier has a misclassification rate of 4.3 %.

The nonlinear AHN-classifier is depicted in Fig. 6.21. As noted, three *CH*-primitive molecules clusters the dataset. These molecules act over three spaces delimited by the bounds of molecules $L$. Then, the dataset is clustered using (6.16). Notice that these bounds are separated by intermolecular bonds $\mathbf{r}$.

In contrast, a self-organizing map (SOM) was trained for the same example in oder to compare it against to the AHN-model. In that way, the SOM was designed with an output layer of 20 × 20 neurons using the same 200-samples $(\mathbf{x}_1, \mathbf{x}_2, \mathbf{y}_1)$ as the input vector of the network. After 200 epochs, the response of the self-organizing map was obtained, as shown in Fig. 6.22. It is remarkable to say that Fig. 6.22 represents white regions as the logical value *false* and black regions as the logical value *true*; thus these regions are not the current positions of the 400 neurons. Moreover, the classification process obtained a misclassification rate of 9.5 %.

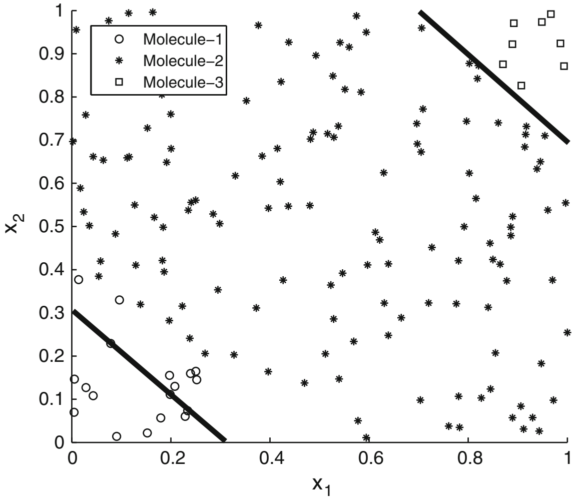


**Fig. 6.20** Response of the nonlinear AHN-classifier $\hat{f}$ of Example 6.9

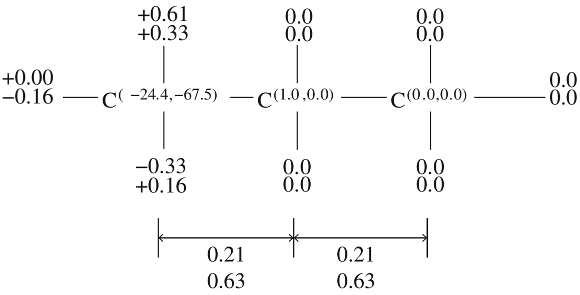


**Fig. 6.21** Artificial hydrocarbon network structure of the nonlinear classifier $\hat{f}$ of the XOR-classifier

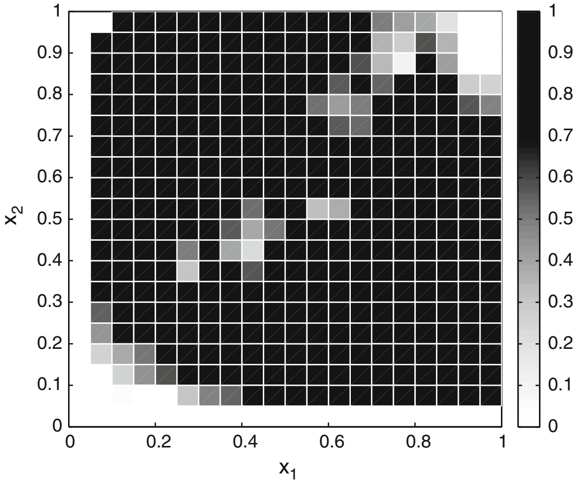


**Fig. 6.22** Response of the nonlinear self-organizing map of Example 6.9

Notice in Fig. 6.22 that some white regions are located inside black regions due to the topology of the SOM. In contrast, Fig. 6.20 reveals that the AHN-model can classify slightly better the XOR problem than the SOM net, i.e. misclassification rates of 4.5 and 9.5 %, respectively. To this end, it is important to highlight that an artificial hydrocarbon network classifier is a supervised algorithm while a self-organizing map classifier is an unsupervised algorithm.

Now, consider more attribute variables in a dataset like the Fisher's Iris dataset [1, 2]. It is a 150-samples dataset comes from three different species of Iris flowers to which four variables were measured: *sepal length* ($x_1$), *sepal width* ($x_2$), *petal length* ($x_3$), and *petal width* ($x_4$). All the 150-samples has numeric labels for the three different species of flowers: *Iris setosa* (1), *Iris versicolor* (2), and *Iris virginica* (3).

*Example 6.10* Build a nonlinear classifier $\hat{f}$ of the Iris dataset using artificial hydrocarbon networks.

**Solution 6.10** A nonlinear artificial hydrocarbon network classifier $\hat{f}$ was built using 75 % of the dataset, chosen random samples of the form $(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4, \mathbf{y}_1)$; where, $\mathbf{y}_1$ is the label of Iris flowers. Then, running Algorithm 5.3 a model $\hat{f}$ was obtained using the training data with a maximum number of molecules $n_{max} = 3$, a maximum number of compounds $c_{max} = 1$, a tolerance $\varepsilon = 1 \times 10^{-4}$, and a step size $\eta = 0.095$. Figures 6.23 and 6.24 show the response of the nonlinear AHN-classifier $\hat{f}$. In particular, the classifier has a misclassification rate of 10.7 % using the 150-samples as testing samples.

The nonlinear AHN-classifier is depicted in Fig. 6.25. As noted, three *CH*-primitive molecules clusters the dataset. In particular, each *CH*-primitive molecule

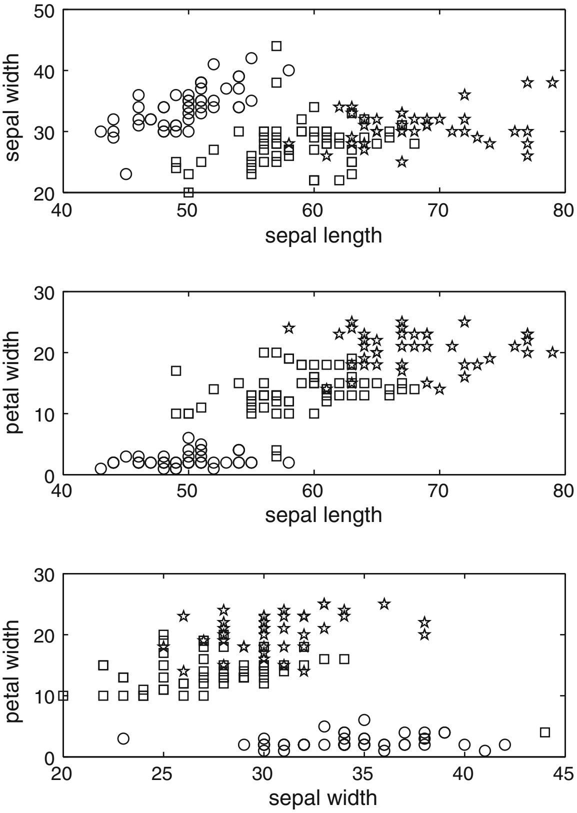


**Fig. 6.23** *(First part)* Response of the nonlinear AHN-classifier $\hat{f}$ over the Iris dataset: molecule-1 (○) represents the Iris setosa, molecule-2 (□) represents the Iris versicolor and molecule-3 (⋆) represents the Iris virginica

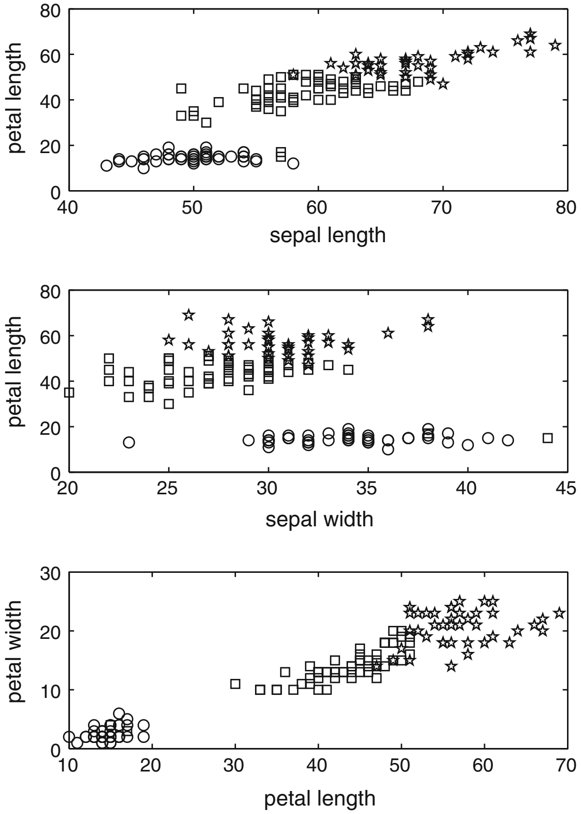


**Fig. 6.24** *(Second part)* Response of the nonlinear AHN-classifier $\hat{f}$ over the Iris dataset: molecule-1 (○) represents the Iris setosa, molecule-2 (□) represents the Iris versicolor and molecule-3 (⋆) represents the Iris virginica

coresponds to a different class of Iris flowers. Actually, the bounds of molecules $L$ are used as centers of molecules and partitions are calculated as (6.19). Notice that these centers are separated by intermolecular bonds $\mathbf{r}$.

## 6.3 Guidelines for Real-World Applications

In the previous section, some notes on modeling problems using artificial hydrocarbon networks were presented. As noted, built models are subjected to the structure of artificial hydrocarbon networks, and these structures with their parameters can be useful to train other artificial hydrocarbon networks or to analyze systems. Since molecules can be treated as packages of information, inheritance of data would be

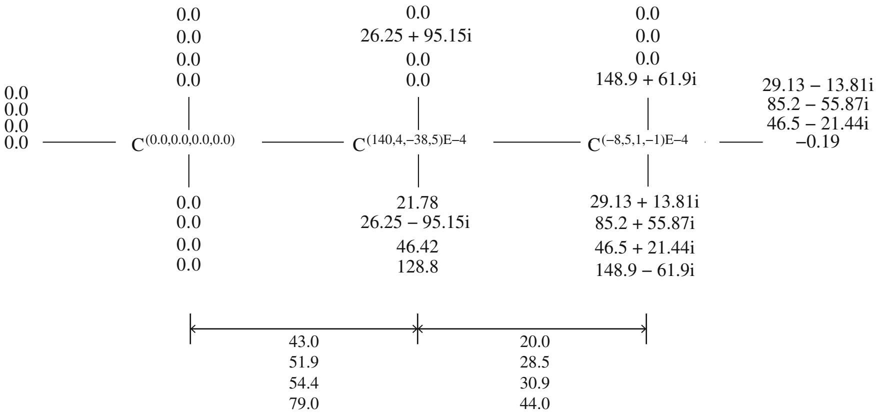


**Fig. 6.25** Artificial hydrocarbon network structure of the nonlinear classifier $\hat{f}$ of the Fisher's Iris dataset

useful to other AHNs or they can be stored in a catalog for future training models. In addition, hydrogen values, stoichiometric coefficients, bounds of molecules and intermolecular distances would be treated as metadata to analyze modeled systems or to perform new actions in an overall intelligent system. Then, three post-processes of artificial hydrocarbon networks are identified and discussed, as follows.

### 6.3.1 Inheritance of Information

Molecules in artificial hydrocarbon networks package information that can be inherited to other AHN-structures to preserve that information. For instance, consider the two artificial hydrocarbon compounds depicted in Fig. 6.26. As it can be seen, the first molecule in each structure is completely different while the other two molecules in both structures are similar. Then, it supposes that the two systems from which the AHN-structures were built, are similar. In fact, the first AHN-structure in Fig. 6.26 models the system $f$ depicted in (6.12) and the second AHN-structure models the system $f$ in (6.26).

$$f = \begin{cases} x + 0.2 & -1 \leq x < 0.1 \\ \sin(\pi x) & 0.1 \leq x < 0.9 \\ \cos(\pi x) & 0.9 \leq x \leq 1 \end{cases} \tag{6.26}$$

Moreover, combinations of molecules can be used to create new artificial hydrocarbon networks that exhibit behaviors as combinations of behaviors of these molecules. For example, Fig. 6.27 shows two different artificial hydrocarbon compounds made of three molecules. As a result, another artificial hydrocarbon compound

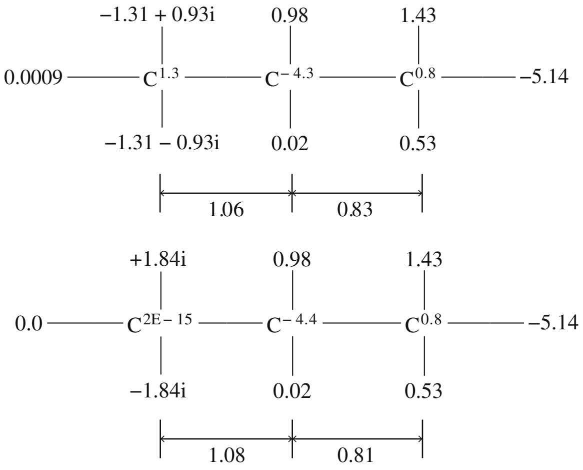


**Fig. 6.26** Two AHN-structures that reveal similar behaviors because they model two similar systems

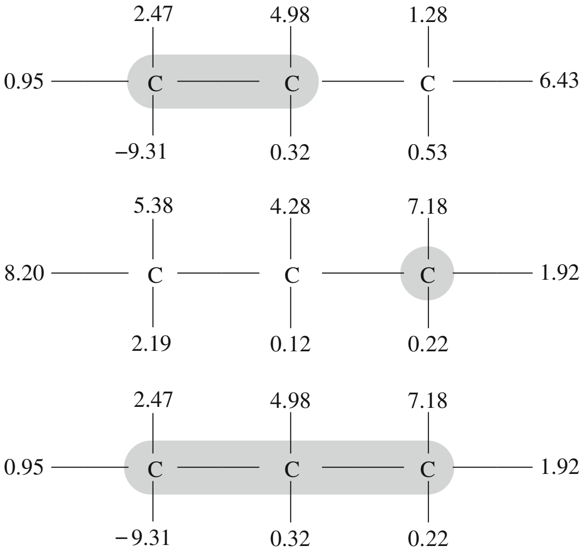


**Fig. 6.27** Three artificial hydrocarbon compounds. The first two compounds are made of molecules that are combined to make the resultant artificial compound shown at last

(see the last compound in Fig. 6.27) can be made with a combination of molecules in the other compounds.

### 6.3.2 Catalog Based on Artificial Compounds

If molecules inherit information, then they can be stored and create new compounds using these molecules. As an example of its application, consider the training process of an artificial hydrocarbon network. In some point, one can assume that an unknown system $\Sigma$ presents a particular behavior that can be associated to a molecule $M_p$ stored previously. Then, one can feed the training process with $M_p$. If the behavior of $M_p$ is similar to the unknown behavior of the system, then $M_p$ will be assimilated rapidly. Otherwise, $M_p$ will be vanished from the training process. For example, (6.26) was modeled using the second molecule shown in Fig. 6.26 as $M_p$, and the process finishes in 5 iterations while using Fig. 6.28 as $M_p$ the process finishes in 380 iterations vanishing $M_p$.

But, the most prominent application of storing molecules is that they can be analyzed in order to form a catalog of molecules with description. Then, the catalog of compounds can be implemented to inherit information, to create new compounds or to analyze if a system assimilates some previously recorded behaviors. In addition, the catalog can be implemented for large sets of observations in systems in order to reduce the time of training artificial hydrocarbon networks.

### 6.3.3 Using Metadata

As said before, different parameters in artificial hydrocarbon networks can be used for analyzing systems. Through this chapter hydrogen values have been used for packaging information and bounds of molecules have been used for explaining the limits of molecular behaviors. These and other characteristics were described in Sect. 4.4.

For instance, let assume that an intelligent system has the objective to identify if two functions $f_1$ and $f_2$ are similar or not (they can represent the characteristic vectors of two images, the target image and the testing image). One can prove that they are similar if $f_1$ is used as a training function and $f_2$ as the testing function of the model of $f_1$. Maybe in some regions they would be similar and in other input intervals they would not. Then, two artificial hydrocarbon networks can be used for modeling $f_1$ and $f_2$, let say, $\hat{f_1}$ and $\hat{f_2}$ as shown in Fig. 6.29.

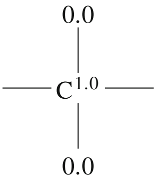


**Fig. 6.28** Molecule used in the training process to show the impact of stored molecules

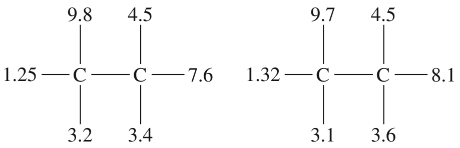


**Fig. 6.29** Two artificial hydrocarbon networks modeling two different functions. Molecular information is compared to find similarities on them, because implicitly, it means that the two functions are similar

Since two molecules are similar due to their parameters, then similarities in parameters mean similarities in modeled systems. Using the latter criterion, parameters can be compared in Fig. 6.29 and if these parameters are statistically equivalent, then both systems (or parts of them) are similar. Otherwise, systems are not similar. Finally, from Fig. 6.29, one can conclude that $f_1$ and $f_2$ are similar functions (e.g. matching of two images) because all hydrogen parameters and intermolecular distances reveal that there is a maximum absolute error of 0.5 between them. To this end, parameters in artificial hydrocarbon networks can be treated as metadata giving the opportunity to analyze systems or to perform new actions in an overall intelligent system.

## References

1. Bache K, Lichman M (2013) UCI machine learning repository. http://archive.ics.uci.edu/ml
2. Wong WC, Cho SY, Quek C (2009) R-POPTVR: a novel reinforcement-based POPTVR fuzzy neural network for pattern classification. IEEE Trans Neural Netw 20(11):1740–1755

# Chapter 7
# Applications of Artificial Hydrocarbon Networks

Artificial hydrocarbon networks (AHNs) present several characteristics that are useful for learning, classifying, predicting, analyzing, filtering and controlling tasks, as described in previous chapters. Precisely, *CH*-molecules in their structures allow to capture and to cluster information about systems that can be exploited to solve engineering problems. In that sense, artificial hydrocarbon networks can be applied successfully in many real-world engineering applications.

Thus, this chapter presents three different applications in which artificial hydrocarbon networks have been implemented, such as: design of adaptive filters for noisy audio signals, design of position controllers of direct current motors using the so-called AHN-fuzzy inference systems and design of a facial recognition system. Examples of program codes of these applications can be found in Appendix C.

## 7.1 Filtering Process in Audio Signals

Audio signal applications present problems like the addition of noise signals that interfere with the original ones, arriving in poor performance of audio. For signal analysis, this addition of noise causes imprecise data of information that blinds important features in time and frequency domain, while for human ears, it causes non-intelligibility of audio. Nowadays, the huge industry of mobile applications allow to search an unknown segment of music by recording it and sending over the web to query the basic information about the track. In fact, the main process of these applications is to filter the recorded signal in order to clean the original segment of music because it is corrupted with environmental and hardware noise.

In that sense, artificial hydrocarbon networks were proved in different scenarios in order to achieve noise reduction in audio signal to perform the music search problem. Three metrics were used: a direct comparison with a classical finite impulse response (FIR) filter, the short-time objective intelligibility (STOI) value, and the signal-to-noise ratio (SNR).

H. Ponce-Espinosa et al., *Artificial Organic Networks*,
Studies in Computational Intelligence 521, DOI: 10.1007/978-3-319-02472-1_7,

### *7.1.1 Background and Problem Statement*

In general, audio filters are analog or digital. In the following section, consider digital audio filters, e.g. filters implemented in software. In digital signal processing, audio filters can be classified into finite impulse response (FIR) filters and infinite impulse response (IIR) filters. The first one uses a filter kernel that reaches a finite zero frequency response while the second one uses a filter kernel that responds in frequency with infinite exponential decaying sinusoidal functions. In practice, convolution implements FIR filters and recursion implements IIR filters. In advance, FIR filters use convolution of the input signal while IIR filters uses convolution of the input but also of the output signals [13].

Several audio filtering applications consider two interests: improving either time domain response or frequency domain response. Smoothing, noise reduction and direct current (DC) removal are typical problems in time domain while separating frequencies is typical in frequency domain. However, it is very difficult to improve time and frequency responses with the same filter. Thus other techniques have been proposed. For example, the moving average and windowed-sinc filters are widely used in time domain. In contrast, the single 2-pole filter and Chebyshev's filter are used in frequency domain. The above examples fall into linear filters. These kinds of filters have a linear response from the input signal. However, there are other prominent filters that respond in a nonlinear way from the input signal. Classical nonlinear filters are based on the time-varying Wiener's filter, probabilistic filters and homomorphic filters [13].

Consider a special kind of noise so-called white or uniform noise. In particular, white noise reduction cannot be handled easily in audio signals because it contains similar low magnitude components in all the spectrum of frequencies that is typically confusing or overlapping with mainly voice signals and musical signals. In that sense, linear filters cannot clean the signal efficiently and nonlinear filters for these purposes have been proposed. In fact, recent tendencies on filtering signals fall into adaptive nonlinear filters. Those are primary nonlinear filters that react to segments of audio signals. Depending on the time and/or frequency responses in these intervals, the filter provides different actions. In particular, most of adaptive filters are covered by artificial intelligence techniques like artificial neural networks or fuzzy systems [1].

Furthermore, artificial hydrocarbon networks have a filtering property that in conjunction with molecular units can be adapted to audio signals in order to filter noisy signals. For instance, consider a segment of an audio signal corrupted with white noise. Then, an artificial hydrocarbon network can act as filter of this audio signal using the above properties, as stated in Problem 7.1.

**Problem 7.1** Let $s$ be a corrupted audio signal of the form as (7.1); where, $f$ is the original audio signal, $\rho$ is a white noise signal and $t$ is the time. Then, a filter $F$ is required in order to recover $f$ as much as possible, as expressed in (7.2).

$$s(t) = f(t) + \rho(t) \tag{7.1}$$

$$f(t) \approx F(s(t)) \tag{7.2}$$

On the other hand, since artificial hydrocarbon networks are filtering structures and they can adapt to different scenarios using *CH*-molecules, the problem to find a filter $F$ is reduced to find an artificial hydrocarbon network structure *AHN* given $s$, such that, (7.3) holds:

$$f(t) \approx AHN(s(t)) \tag{7.3}$$

Following, a methodology of using artificial hydrocarbon networks for noise reduction is presented as well as some experimental results over three different environments: filtering the segment of an audio track corrupted with white noise in software, filtering the segment of an audio signal recorded with a microphone and corrupted in software, and filtering the segment of an audio track recorded with a mobile phone and corrupted in a non-ideal environment.

### 7.1.2 Methodology

First, an audio signal $s$ is divided into batches, so-called windows. Each window that starts in time $t_0$ has a finite size of time $t_W$ measured in seconds, i.e. milliseconds. Then, each window is initially divided into 2 partitions with time length $t_P^{(i)}$ of $i$ partition, as shown in Fig. 7.1.

In order to determine the best set of time lengths $\{t_P^{(i)}\}^*$ for all partitions, an artificial hydrocarbon networks *AHN* implements its intermolecular distances $r^{jk}$, between two adjacent molecules $M_j$ and $M_k$, in time domain such that each time length $t_P^{(i)}$ is equal to the interval between two intermolecular distances as (7.4); where, $r^{01} = t_0$ and $r^{n,n+1} = t_0 + t_W$.

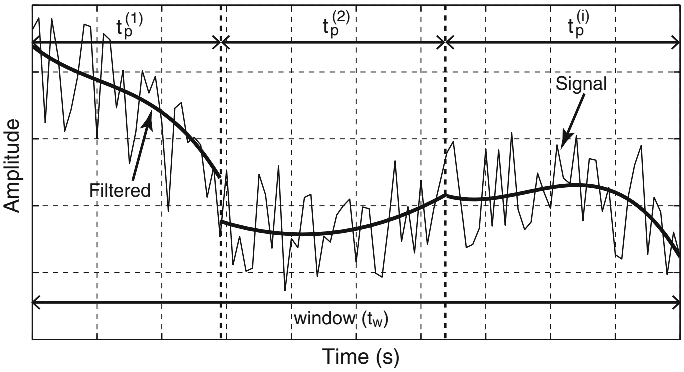


**Fig. 7.1** Segment of audio signal for training an artificial hydrocarbon network structure

**Algorithm 7.1** ADAPTIVE-AHN-FILTER($s, t_W, n_{max}, \varepsilon, \eta$): Adaptive filtering process to reduce noise in audio signals using artificial hydrocarbon networks.

```
Input: the corrupted audio signal $s = (t, s(t))$, window time $t_W$, the maximum number of molecules $n_{max} \geq 2$, the small tolerance value $\varepsilon > 0$ and the learning rate $\eta$.
Output: the filtered audio signal $\hat{f}$.

$c_{max} = 1$
$\hat{f} = \emptyset$
Split $s$ into $m$ equal windows $s_i$ using $t_W$.
for each $s_i$ do
    $[\Gamma, H, \Lambda, \Pi] =$ SIMPLE-AHN($s_i, n_{max}, c_{max}, \varepsilon, \eta$)
    $\hat{f}_i \leftarrow$ using $[\Gamma, H, \Lambda, \Pi]$ in the $i$-th input domain
    $\hat{f} = \hat{f} + \hat{f}_i$
end-for
return $\hat{f}$
```

$$t_P^{(i)} = r^{i,i+1} - r^{i-1,i} \ , \ \ \forall i = 1, \ldots, n \tag{7.4}$$

Thus, the structure of the *AHN* is adapting to each window with time $t_W$ using a compound, i.e. $c_{max} = 1$ in order to minimize computational timing, and a maximum of $n_{max}$ molecules, previously set. It is remarkable to say that an stoichiometric coefficient $\alpha_1$ has not to be computed since for one compound $\alpha_1 = 1$. For each window, an artificial hydrocarbon network *AHN* is performed, as summarized in Algorithm 7.1. At last, this process called *adaptive AHN-filter* will reduce noise in signal $s$. As noted, this is a simple and transparent way to perform a filter using artificial hydrocarbon networks.

On the other hand, three evaluations are implemented to measure the performance of the adaptive AHN-filter. The first one considers a comparison between the adaptive AHN-filter and the classical FIR-filter. For instance, the FIR-filter [1, 13] is expressed as a finite weighted sum as (7.5); where, $x[t]$ is a finite sequence of data representing the input signal in $t$, $\{b_k\}$ is a set of filter coefficients found by designing methods like windowing, $y[t]$ is the finite sequence representing the output signal in $t$, and $N$ represents the order of the FIR-filter.

$$y[t] = \sum_{k=0}^{N} b_k \cdot x[t-k] \tag{7.5}$$

The second evaluation considers the short-time objective intelligibility (STOI) value which it represents a monotonic relation with the average intelligibility of filtering signals. In fact, the STOI metric offers an objective, non-qualitative way to measure the intelligibility of filtered signals with respect to original signals, in the range of 0.0–1.0. Thus, a higher value of STOI stands for a better performance about filters [15].

The last evaluation considers the well-known signal-to-noise ratio (SNR) which it measures the ratio between the average power of the interested signal $P(s)$ and the

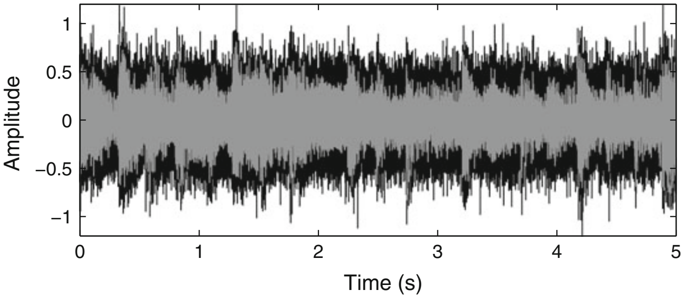


**Fig. 7.2** Audio signal of experiment 1, (*black*) original audio segment, (*gray*) noisy signal

average power of the noise signal $P(\rho)$ as written in (7.6). If the SNR-value is greater than 1, then the signal contains much more meaningful information than noise [14].

$$SNR = \frac{P(s)}{P(\rho)} \tag{7.6}$$

### *7.1.3 Results and Discussion*

In order to prove the performance of the adaptive AHN-filter, three experiments were designed. The first experiment uses an audio signal digitally corrupted with white noise; the second experiment was run on a digital signal processing (DSP) hardware; and the third experiment considers a naturally corrupted audio signal coming from a recording using a mobile phone.

#### 7.1.3.1 Filtering Audio Signal Digitally Corrupted with White Noise

The experiment considers a five-second mono-channel audio signal segment sampled at 44.10 kHz of a popular rock band. In addition, it was normalized and corrupted with white noise of 15 % SNR. Figure 7.2 shows the original and the corrupted signals. In particular, the adaptive AHN-filter was built using $n_{max} = 10$ and $t_W = 10$ ms. Figure 7.3 (top) presents the response of the adaptive AHN-filter in time domain. In addition, a classical 30-th order FIR-filter was implemented and its response is presented in Fig. 7.3 (bottom).

To better analyze the performance of the adaptive AHN-filter, Fig. 7.4 depicts a comparison in frequency domain of the original signal (Fig. 7.4 (top)), the response of adaptive AHN-filter (Fig. 7.4 (middle)) and the response of the FIR-filter (Fig. 7.4 (bottom)). Notice that the cut-off frequency is approximately at 5000 Hz in the AHN-filter response and 7000 Hz in the FIR-filter response, qualitatively demonstrating that adaptive AHN-filters can achieve lowpass responses. It is important to remark

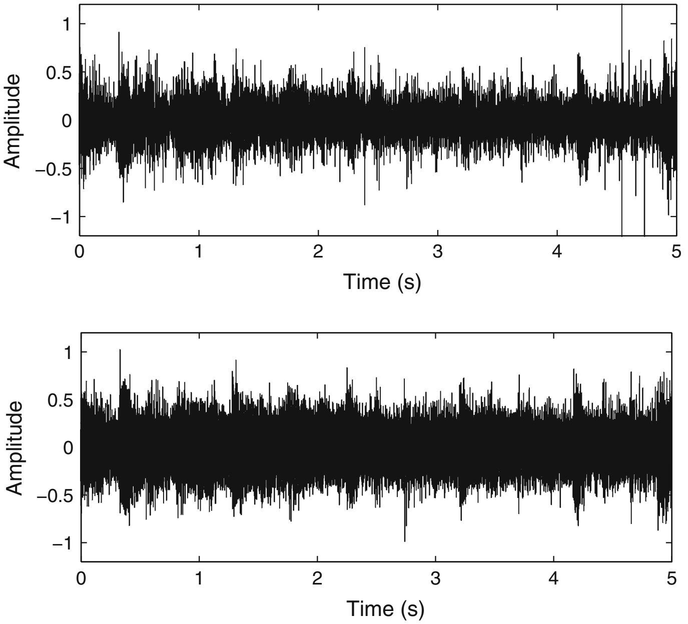


**Fig. 7.3** Audio segment filtered of experiment 1. (*top*) Response of the adaptive AHN-filter, (*bottom*) Response of the FIR-filter

that the cut-off frequency of the FIR-filter was set at 5000 Hz because important data of original audio signal is removed at the same cut-off frequency in the adaptive AHN-filter. In addition, the STOI-value was measured in both responses obtaining 0.6628 for the adaptive AHN-filter and 0.8083 for the FIR-filter. Thus, a FIR-filter is 18 % more intelligible than the adaptive AHN-filter. Finally, the SNR-value of the filtered signal when applying the adaptive AHN-filter is 2.3318 and 0.8609 when applying the FIR-filter, proving that adaptive AHN-filters can clean signals in a better way.

From the above experiment, a filter based on artificial hydrocarbon networks can reduce considerably the noise in audio signals. However, the STOI-value suggests that the filtered signal using the adaptive AHN-filter is not much intelligible, but in terms of noise reduction, the SNR-value reveals that the AHN-filter has better performance than the classical FIR-filter. In fact, Fig. 7.4 shows that a non-adaptive filter, i.e. FIR-filter, has poor performance of filtering white noise, while the adaptive AHN-filter can reach a better performance.

#### 7.1.3.2 Filtering Audio Signal on DSP Hardware

This experiment considers the implementation of the adaptive AHN-filter on DSP NI-Speedy 33 hardware; an educational board for signal processing that can be

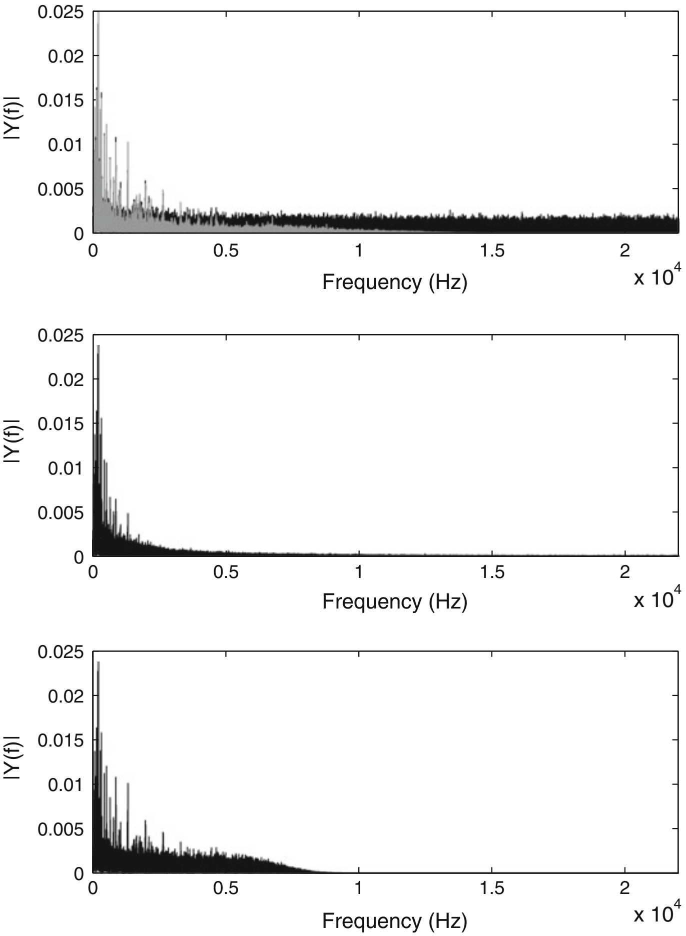


**Fig. 7.4** Analysis in frequency domain of experiment 1. (*top*) *black*: noisy signal, *gray*: original signal, (*middle*) Response of the adaptive AHN-filter, (*bottom*) Response of the FIR-filter

easily programmed using the LabVIEW platform. In that case, the audio signal was obtained via a microphone connected to the analog input of the hardware. Then, the audio signal was corrupted programmatically with white noise of 20 % SNR. The response of the adaptive AHN-filter is then obtained and finally the filtered signal is reproduced on headphones connected to the analog output port of the hardware. Figure 7.5 shows the block diagram of the program implementing this experiment in the DSP NI-Speedy 33 using LabVIEW.

Figure 7.6 (top) shows a 3-second audio signal collected from the hardware sampled at 44.10 kHz. Figure 7.6 (middle) shows the response of the adaptive AHN-filter and Fig. 7.6 (bottom) shows the response of the FIR-filter, in time domain. The AHN-filter was built using $n_{max} = 10$ and $t_W = 10$ ms. Notice that offline performance

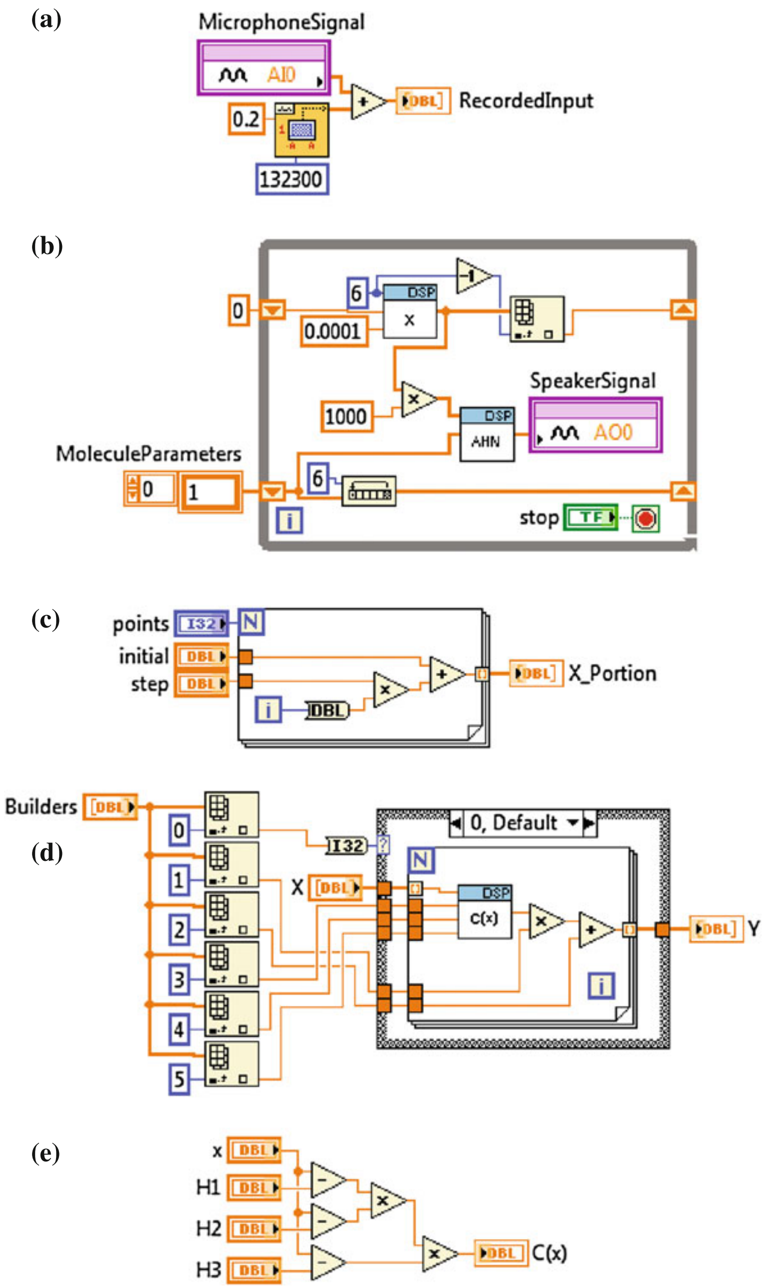


**Fig. 7.5** Block diagram of the program implemented on DSP NI-Speedy 33 using LabVIEW. **a** Block diagram for recording the input signal coming from a microphone. **b** Block diagram for reproducing the response of the AHN-filter. **c** Block diagram of the **X** SubVI in **b**. **d** Block diagram of the **AHN** SubVI in **b**. **e** Block diagram of the simplified *CH* molecule **C(x)** SubVI in **d**

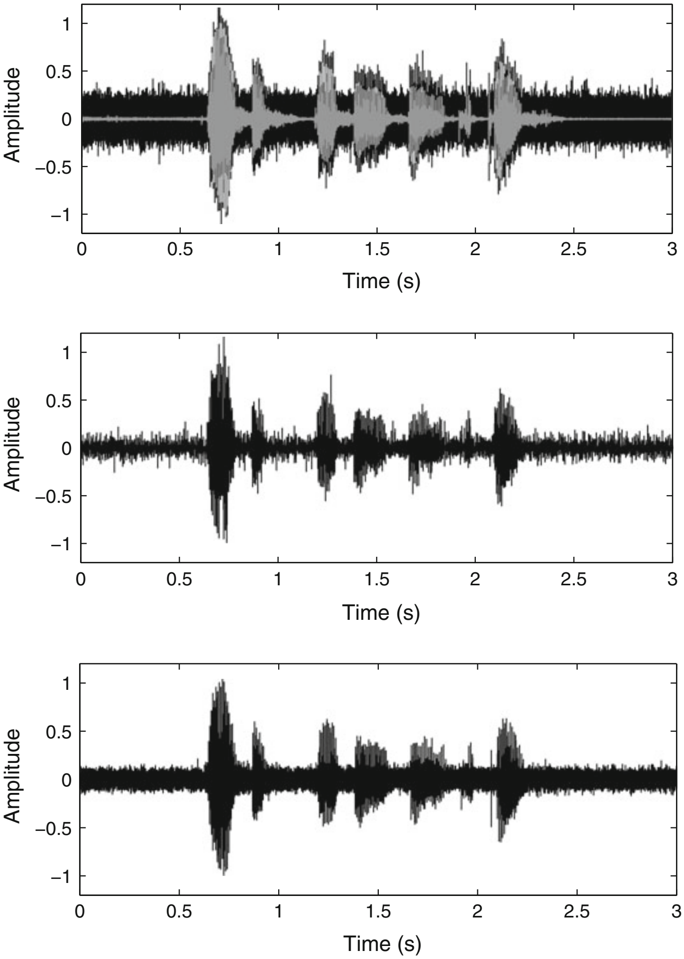


**Fig. 7.6** Audio segment of experiment 2. (*top*) *black*: noisy signal, *gray*: original signal, (*middle*) Response of the adaptive AHN-filter, (*bottom*) Response of the FIR-filter

is needed for calculating all parameters in the adaptive AHN-structure. If online filtering is required, real-time techniques may be used; but presetting parameters is mandatory.

In can be seen in Fig. 7.6 (bottom) that classical FIR-filter achieves lowpass filtering but does not attenuates noisy signal while voiceless, in comparison with the response of adaptive AHN-filter (Fig. 7.6 (middle)).

In addition, Fig. 7.7 shows the comparison of the adaptive AHN-filter and the FIR-filter in frequency domain. Figure 7.7 (middle) reveals that the adaptive AHN-filter can eliminate most of the white noise in the corrupted signal, but it does not cancel all white noise after 5000 Hz. Actually, the FIR-filter can deal with high frequencies,

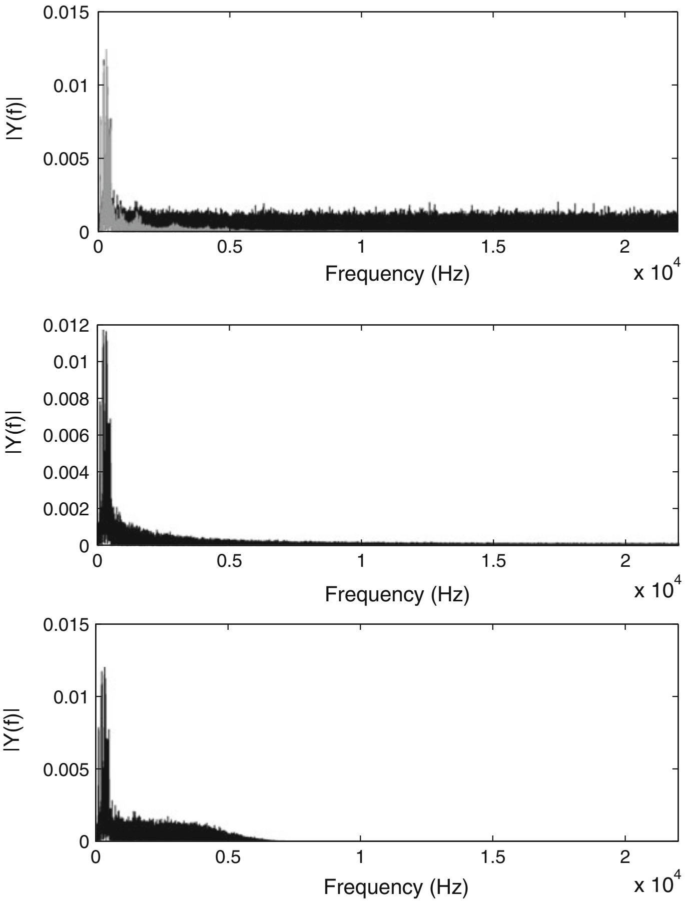


**Fig. 7.7** Analysis in frequency domain of experiment 2. (*top*) *black*: noisy signal, *gray*: original signal, (*middle*) Response of the adaptive AHN-filter, (*bottom*) Response of the FIR-filter

e.g. after 5000 Hz; but it cannot filter white noise in low frequencies, as revealed in Fig. 7.7 (bottom) in the interval from 3000 to 5000 Hz. A qualitative comparison is also complemented with the quantitative STOI metric; in which, the response of the adaptive AHN-filter has 0.6404 of STOI value and 0.6648 STOI value obtained from the filtered audio signal using the FIR-filter. Thus, FIR-filter is relatively 3.7 % more efficient than the adaptive AHN-filter, in terms of intelligibility. Finally, applying the SNR metric, the response of the adaptive AHN-filter obtained 8.3917 and the FIR-filter 1.4556, showing that the adaptive AHN-filter reduces white noise better than FIR-filter.

#### 7.1.3.3 Filtering Audio Signal Naturally Corrupted with White Noise

The last experiment considers an audio segment corrupted with noise of the environment. In that case, a mobile phone was used for recording the audio signal from an audio CD track. In fact, this is a real-world application in mobile devices to recognize audio segments. Actually, in order to compute the results of the artificial hydrocarbon network based filter and the FIR filter; the same segment of audio signal was extracted from the audio CD. At last, five-second mono-channel audio segment sampled at 44.10 kHz was used in the experiment as shown in Fig. 7.8 (top). Analyzing Fig. 7.8 (top), the recorded audio signal is corrupted with 15.1 % SNR. Figure 7.8 (middle) shows the response of the adaptive AHN-filter with $n_{max} = 10$ and $t_W = 10$ ms, and Fig. 7.8 (bottom) shows the response of a 30-th order FIR-filter, both in time domain.

In addition, Fig. 7.9 shows a comparison between the responses of both filters in frequency domain. The cut-off frequency of the adaptive AHN-filter is approximately 4500 and 5000 Hz in the FIR-filter. In fact, Fig. 7.9 (bottom) reveals that some of white noise could not be removed by FIR-filter in low frequencies, but it is very efficient in high frequencies. In contrast, the AHN-filter in Fig. 7.9 (middle) has a better performance in low frequencies eliminating white noise. In terms of intelligibility, the STOI-value of the adaptive AHN-filter is 0.3514 and 0.4918 for the FIR-filter. At last, the SNR-value for the AHN-filter and FIR-filter are 6.1314 and 1.9239, respectively. At last, from the above experiments, adaptive AHN-filters can achieve the music search problem because they can attenuate noise significantly; however, a tradeoff between the STOI and the SNR has to be considered.

## 7.2 Position Control of DC Motor Using AHN-Fuzzy Inference Systems

Controlling direct current (DC) motors is an important task in industrial applications because they are used in computer numeric control (CNC) machines, robotic systems, domotics, and so forth. Nowadays, fuzzy control systems have been widely used to perform these tasks. In particular, type-1 and type-2 fuzzy systems have been implemented; however, the first type of systems has not dealt appropriately in presence of noise while the second type of systems have done precisely. But the latter is much more complicated to compute because a large number of operational computations are required. Thus, in practice, type-1 fuzzy systems are implemented where a system is not subjected to large noise or when hardware does not allow too much computations; in contrast, type-2 fuzzy systems are implemented where a system is subjected to large noise and where the software and hardware is appropriate to perform all calculations [6].

In that sense, a new fuzzy inference system based on artificial hydrocarbon networks is presented in order to improve the design of both types of fuzzy systems.

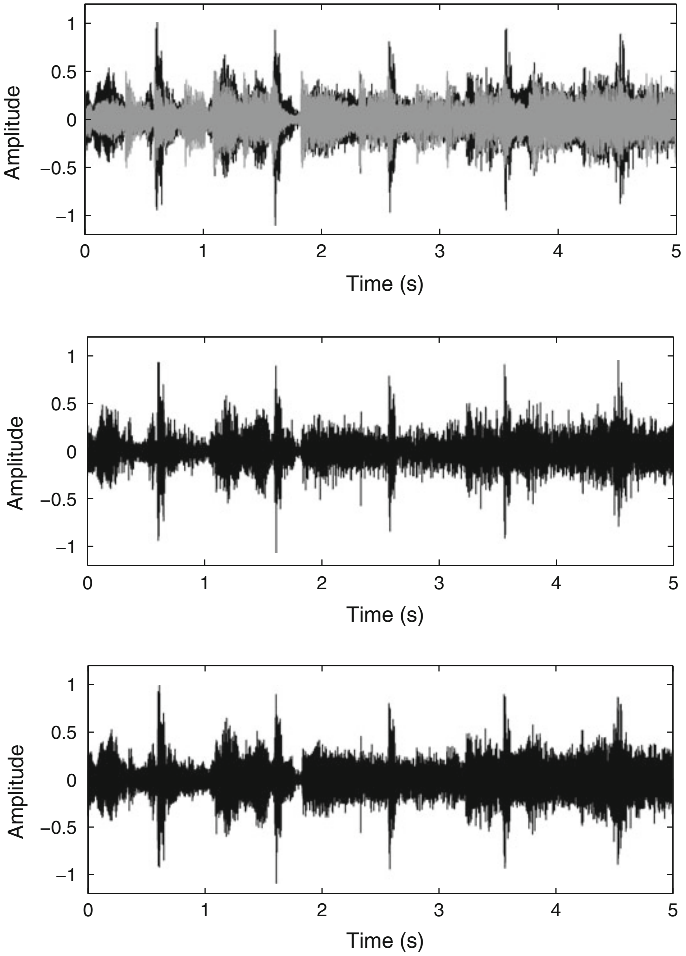


**Fig. 7.8** Audio segment of experiment 3. (*top*) *black*: mobile phone audio signal, *gray*: CD track audio signal, (*middle*) Response of the adaptive AHN-filter, (*bottom*) Response of the FIR-filter

Particularly, the proposed inference system based on AHNs was implemented on the design of a position controller of DC motors in a real-world application, demonstrating accuracy in both types with minimal modifications.

### *7.2.1 Background and Problem Statement*

In control theory, a control system can be considered as an autonomous system that regulates the behavior of other systems. For instance, regulating the temperature

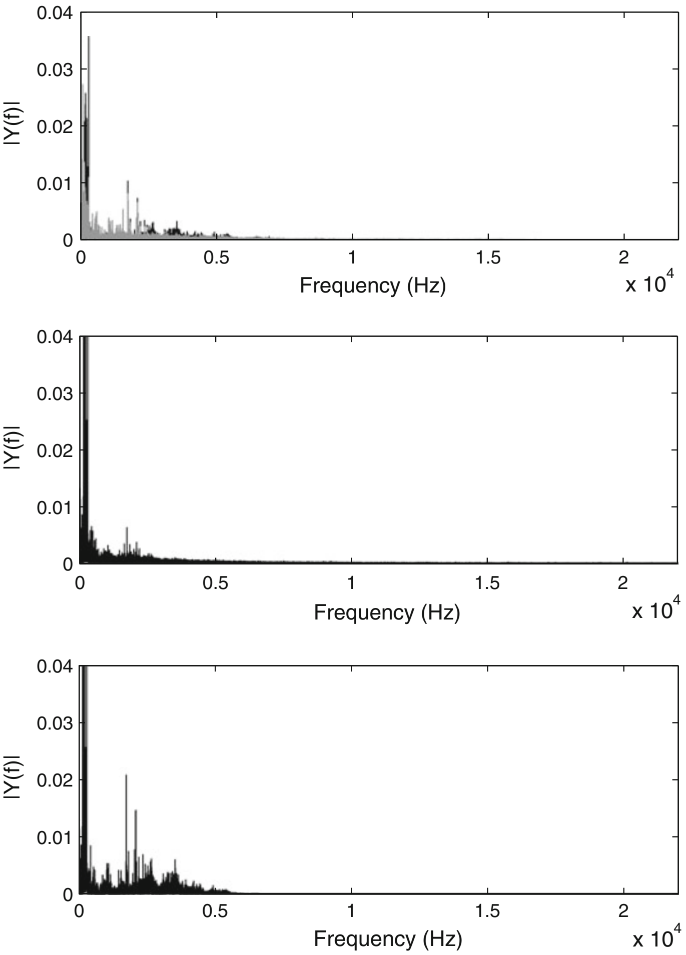


**Fig. 7.9** Analysis in frequency domain of experiment 3. (*top*) *black*: mobile phone audio signal, *gray*: CD track audio signal, (*middle*) Response of the adaptive AHN-filter, (*bottom*) Response of the FIR-filter

inside a room or controlling the velocity of a robotic car are examples of the response of control systems.

For instance, consider the following system, as shown in Fig. 7.10. A trainer hardware module is prepared for sending a reference signal $r(t)$ from a knob and a feedback signal $y(t)$ (i.e. the current position of a DC motor) to a host in which a control law is running. The correction signal $u(t)$ computed is sent back to the trainer module in order to feed a DC motor. In particular to this case study, a NI CompactRIO reconfigurable and embedded system based on field programmable gate arrays (FPGA) is used as the host. In addition, LabVIEW is used for programming

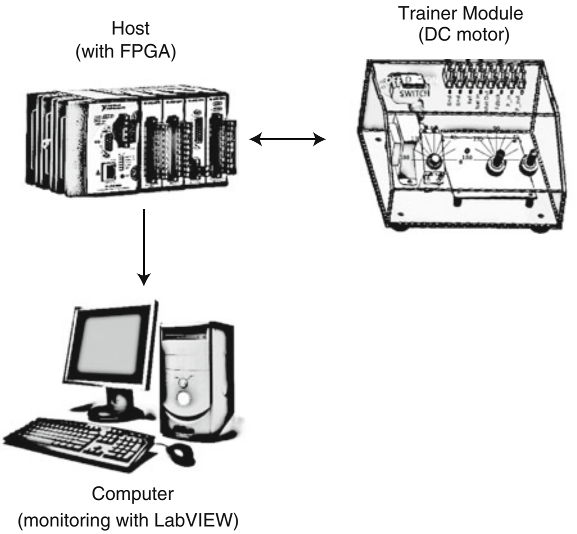


**Fig. 7.10** Overall system of the case study. The trainer module interacts with the NI CompactRIO (host) as a control law and it is monitored with LabVIEW

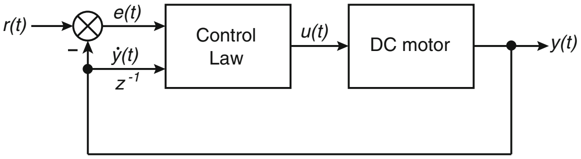


**Fig. 7.11** Block diagram of the PD position control of a DC motor

the control law on the NI CompactRIO and for monitoring the performance of the fuzzy-molecular control system.

On one hand, both the reference signal $r(t)$ that comes from a knob and the position signal $y(t)$ are in the voltage range [0.0, 5.0] V; where, 0.0 V represents an angle of 0° and 5.0 V represents an angle of 180°. On the other hand, the correction signal $u(t)$ is the input voltage of the DC motor in the range [0.0, 5.0] V; where, 0.0 V represents the maximum angular velocity of the motor to rotate counterclockwise, 5.0 V represents the maximum angular velocity of the motor to rotate clockwise and 2.5 V means no rotation (halt). It is remarkable to say that the position of the DC motor increases in counterclockwise direction and decreases in clockwise direction.

In addition, consider the control system of the case study depicted in Fig. 7.11 in which a position control of the DC motor is implemented. It consists of a control law that regulates the behavior of the overall system in order to reach a specific goal (i.e. control the position of a DC motor) using two inputs—the error signal $e(t)$ and the change of the position signal $\dot{y}(t)$—and one output—the input voltage of the DC motor $u(t)$; and the DC motor, referred as the plant.

The objective of a control system like Fig. 7.11 is to track a reference signal $r(t)$ with the position of the DC motor $y(t)$. In that sense, classical linear and nonlinear control system have been widely studied to this purpose. The most prominent control law is called proportional-integral-derivative (PID) controller [11] as expressed in (7.7); where, $P$ is the proportional constant, $I$ is the integral constant, $D$ is the derivative constant, $e(t)$ stands for the error between a reference signal $r(t)$ and the current value of the controlled variable $y(t)$ as in (7.8), $\dot{e}(t)$ stands for the derivative of error, and $u(t)$ is the correction signal.

$$u(t) = Pe(t) + I \int e(t)dt + D\dot{e}(t) \tag{7.7}$$

$$e(t) = r(t) - y(t) \tag{7.8}$$

If $I = 0$, then the control law is considered as a PD-controller and also if $D = 0$ then the resultant control law is called P-controller [11]. As noted, the PID-controller is based on the error signal $e(t)$; however, other control laws can act over different variables.

In most of the cases, a model of the plant is required, e.g. the model of the DC motor. Formally, the DC motor can be represented in the state space as (7.9) [5]; where, $I_a$ and $U_a$ are the flow current and the applied voltage in the rotor, $R_a$ and $L_a$ are the resistance and the inductance values of the rotor, $I_f$ and $L_f$ are the flow current and the inductance value of the stator, $\omega$ is the angular velocity of the rotor, $k$ is the construction constant of the motor related to the flux generation, $J$ is the moment of inertia of the rotor, $\beta$ is the viscous friction coefficient, $T_l$ is the load torque; and, $\dot{a}$ stands for the first derivative of $a$. Figure 7.12 shows the scheme of a DC motor.

$$\begin{bmatrix} \dot{I}_a \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} -\frac{R_a}{L_a} & -\frac{L_f I_f}{L_a} \\ \frac{k}{J} & -\frac{\beta}{J} \end{bmatrix} \begin{bmatrix} I_a \\ \omega \end{bmatrix} + \begin{bmatrix} \frac{1}{L_a} & 0 \\ 0 & -\frac{1}{J} \end{bmatrix} \begin{bmatrix} U_a \\ T_l \end{bmatrix} \tag{7.9}$$

However, there are other controllers like the so-called fuzzy controllers that can be implemented without knowing the model of the plant, e.g. (7.9). In addition, those are very important in real-world applications because they can deal with uncertain and imprecise information, mapping nonlinearities from the input domain (attribute variables) to a fuzzy space and transforming the latter into the output domain (target variables). Focusing on fuzzy control systems, three models can be found in literature: Takagi-Sugeno inference systems [16], Mamdani's fuzzy control systems [4, 7] and Tsukamoto's inference models [17].

In a nutshell, Takagi-Sugeno inference systems apply polynomial functions to construct the consequent values using pairs of input-output data of a given system to model [16]. Mamdani's fuzzy control systems refer to control laws that apply fuzzy inference models with fuzzy sets in the defuzzification phase [7]. In contrast, Tsukamoto's inference models implement monotonical membership functions [17].

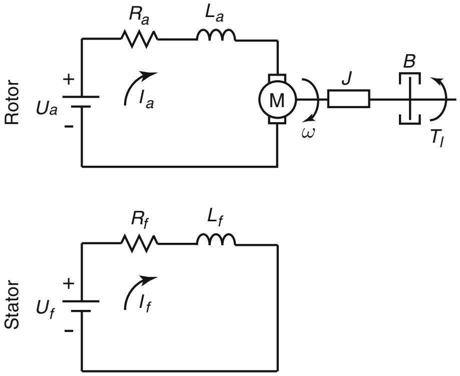


**Fig. 7.12** Scheme of a DC motor

In any case, the basic fuzzy inference model, known as type-1 fuzzy system, consists of three steps, as shown in Fig. 7.13. The first step is the fuzzification process in which the input domain is mapped into a fuzzy space using fuzzy sets that assign a membership value of any element in the input domain to be part of the fuzzy set. Once all membership values are computed, the inference step is performed using propositional rules and computed consequent membership values, and the third step is the defuzzification process in which membership values obtained from the inference process are transformed into an element of the output domain.

As said before, fuzzy inference models use fuzzy partitions, so-called fuzzy sets and they are of two types: the type-1 fuzzy sets (Fig. 7.14 (top)) which assign a membership value of any element to be part of the fuzzy set; and the type-2 fuzzy sets (Fig. 7.14 (bottom)) which assigns a two membership values, i.e. primary and secondary memberships, in order to model uncertainties on boundaries of type-1 fuzzy sets. In particular, the region inside these two membership functions is called the footprint of uncertainty [6, 9].

The generalized fuzzy system is the so-called type-2, as shown in Fig. 7.15. In fact, type-1 is a particular case of type-2 fuzzy systems because the footprint of uncertainty in type-2 systems embeds type-1 fuzzy systems, e.g. a type-2 fuzzy system can be reduced to a type-1 fuzzy system if the secondary membership function is equivalent to the primary membership function.

Then, type-2 fuzzy systems consist of three steps. The first two steps are fuzzification and inference, while the third step is known as output processing. The latter

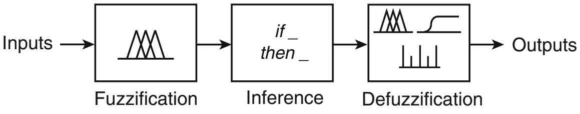


**Fig. 7.13** Block diagram of type-1 fuzzy inference systems

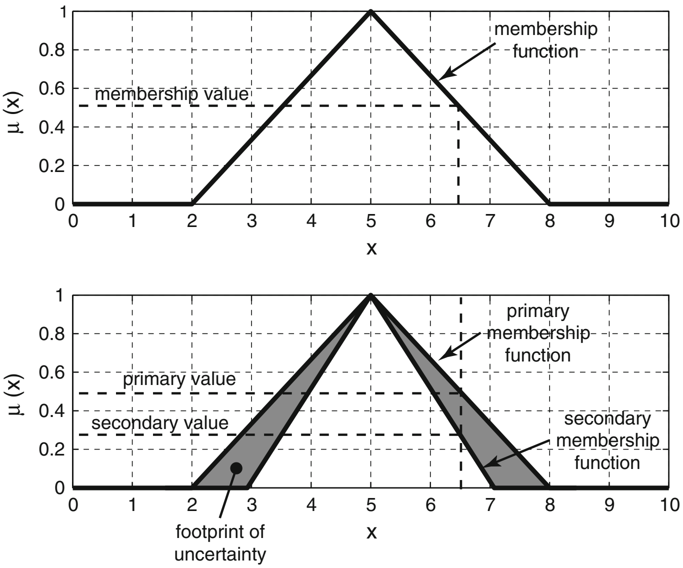


**Fig. 7.14** Types of fuzzy sets. (*top*) Type-1 fuzzy set representation, and (*bottom*) type-2 fuzzy set representation

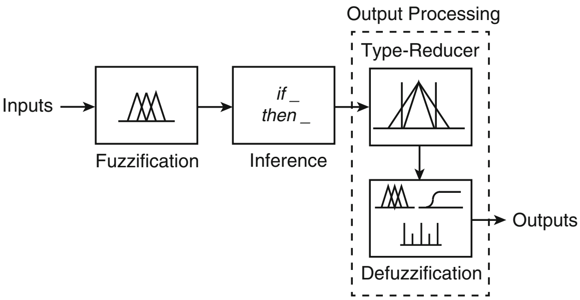


**Fig. 7.15** Block diagram of type-2 fuzzy inference systems

is subdivided in type-reducer and defuzzification steps. In that sense, type-reducer step computes the centroids of all embedded type-1 fuzzy systems in a footprint of uncertainty. As a result, these centroids are used for calculating the output value in the defuzzification step.

In particular, consider Mamdani's fuzzy control systems [4]. Roughly speaking, type-1 Mamdani's fuzzy control systems are important because fuzzy sets can deal with uncertain data, but when the input domain is corrupted with noise the response of type-1 is significantly poor. In addition, type-2 Mamdani's fuzzy control systems enhance the latter problem; but the implementation of type-2 systems are highly complex in comparison with type-1 in terms of the number of operational computations and the ideal unlimited fuzzy sets to evaluate under the footprint of uncertainty region [6, 10]. Then, applications based on Mamdani's fuzzy control systems have to make a tradeoff between using type-1 or type-2 fuzzy systems.

In that sense, a fuzzy inference system based on artificial hydrocarbon networks has been proposed in order to exploit both types of fuzzy systems when a tradeoff between them is required. Formally, it is stated in Problem 7.2.

**Problem 7.2** Let $F$ be the fuzzy control law in a position controller of a DC motor like Fig. 7.11, such that, (7.10) holds; where, $e(t)$ and $\dot{y}(t)$ are the inputs, i.e. the error signal and the derivative of position, respectively; and $u(t)$ is the output, i.e. the input voltage of the DC motor.

$$u(t) = F(e, \dot{y}, t) \tag{7.10}$$

Moreover, suppose that $F$ has to be implemented in both type-1 and type-2 fuzzy systems with minimal modifications. Then, design a fuzzy control law $F$ such that a position controller of a DC motor can be performed with accuracy in both fuzzy types with minimal modifications.

## *7.2.2 Methodology*

The application consists in two parts. At first, a fuzzy inference system based on artificial hydrocarbon networks named fuzzy-molecular inference (FMI) model is proposed. Then, the design of the fuzzy controller $F$ is described using the FMI-model.

### 7.2.2.1 Fuzzy-Molecular Inference Model

Consider a type-1 fuzzy inference system like Fig. 7.16. The fuzzy-molecular inference (FMI) model can be viewed as a block with inputs that are mapped to a fuzzy space in which processes are run and then resulting values are returned to as outputs. In order to do so, the fuzzy-molecular inference model considers three steps: *fuzzification*, *fuzzy inference engine* and *defuzzification*. In addition, the exists a *knowledge base* in which information is stored about a given specific problem.

Fuzzification is the step of the FMI-model in which an input $x$, known as a linguistic variable, is mapped to a fuzzy value in the range [0, 1]. Then, let $A$ be a fuzzy set and $\mu_A(x)$ the membership function of $A$ for all $x \in X$; where $X$ is the input domain space. Moreover, $\mu_A(x)$ is a real value in the interval [0, 1] representing the degree of belonging $x$ to the fuzzy set $A$ as shown in (7.11).

$$\mu_A : x \mapsto [0, 1] \tag{7.11}$$

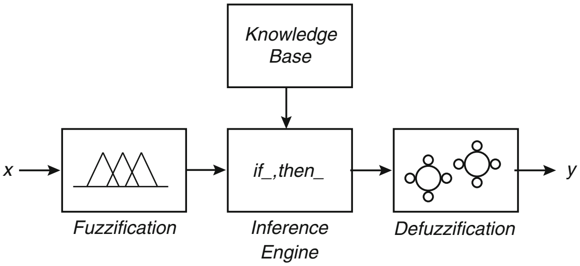


**Fig. 7.16** Block diagram of the fuzzy-molecular inference model

In fact, this linguistic variable is partitioned into $k$ different fuzzy sets $A_i$, for all $i = 1, \ldots, k$. As an example, consider a linguistic variable $x$ partitioned into fuzzy sets like: "small", "medium" and "large". Then, its membership value is calculated over the set of membership functions $\mu_{A_i}(x)$ for all $i = 1, \ldots, k$. The shape of these membership functions depends on the purpose of the problem domain. Typical membership functions are the triangular function (7.12), the S-function (7.13) and the Z-function (7.14); where, $a$, $b$ and $c$ are real value parameters for each function as shown in Fig. 7.17.

$$\mu_A(x) = \begin{cases} \frac{x-a}{b-a} & a \leq x \leq b \\ \frac{c-x}{c-b} & b \leq x \leq c \\ 0 & otherwise \end{cases} \tag{7.12}$$

$$\mu_A(x) = \begin{cases} 0 & x < a \\ \frac{x-a}{b-a} & a \leq x \leq b \\ 1 & x > b \end{cases} \tag{7.13}$$

$$\mu_A(x) = \begin{cases} 1 & x < a \\ \frac{b-x}{b-a} & a \leq x \leq b \\ 0 & x > b \end{cases} \tag{7.14}$$

The next step in the FIM-model is the fuzzy inference engine. This is a mechanism that computes the consequent value $y_l$ of a set of fuzzy rules that accept membership values of linguistic variables. Let, $R_l$ be the $l$-th fuzzy rule of the form as (7.15); where, $\{x_1, \ldots, x_k\}$ is the set of linguistic variables in the antecedent, $\{A_1, \ldots, A_k\}$ is the set of fuzzy partitions of the input space $X$, $y_l$ is the variable of the consequent in the $l$-th rule, $M_j$ is the $j$-th $CH$-primitive molecule of an artificial hydrocarbon network excited with the resulting fuzzy implication value $\mu_\Delta(x1, \ldots, x_k)$ (see below), and $\Delta$ is any $T$-norm function.

$$R_l : \; if \; \Delta \left(x_1 \; is \; A_i, \ldots, x_k \; is \; A_k\right), \; then \; y_l \; is \; M_j \tag{7.15}$$

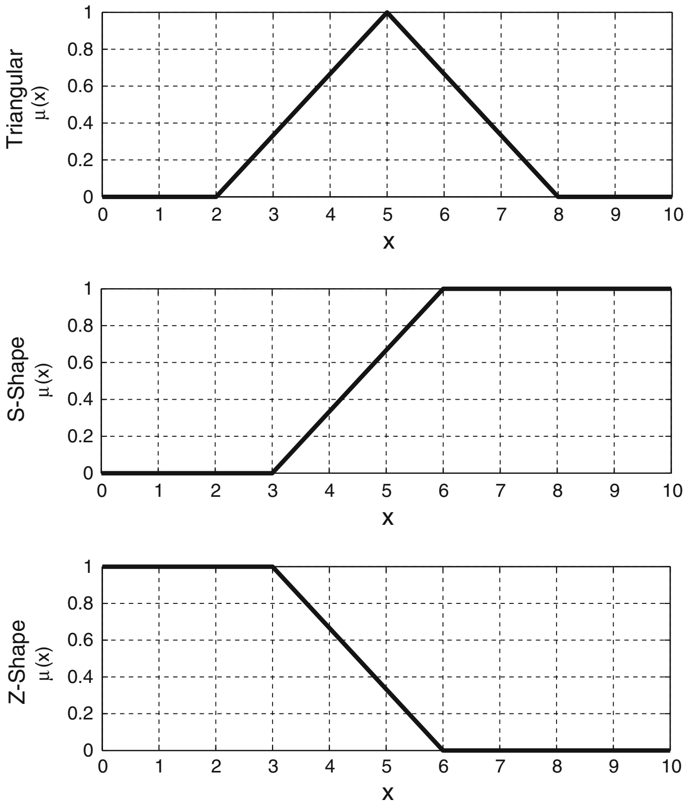


**Fig. 7.17** Common membership functions in fuzzy inference systems

If assuming that $\mu_{\Delta}(x_1, \ldots, x_k)$ is the result of any $T$-norm function, then (7.15) can be rewritten as (7.16); where, $\varphi_j$ is the molecular behavior of $M_j$.

$$R_l : \textit{ if } \Delta\left(x_1 \textit{ is } A_i, \ldots, x_k \textit{ is } A_k\right), \textit{ then } y_l = \varphi_j\left(\mu_{\Delta}(x_1, \ldots, x_k)\right) \tag{7.16}$$

For simplicity to this case study, let $\mu_{\Delta}(x_1, \ldots, x_k)$ be the min function (7.17):

$$\mu_{\Delta}(x_1, \ldots, x_k) = \min\{\mu_{A_1}(x_1), \ldots, \mu_{A_k}(x_k)\} \tag{7.17}$$

The last step in the FMI-model is the defuzzification. It calculates the crisp output value $y$ as (7.18) using all fuzzy rules of the form as (7.16); where, $y_l$ is the $l$-th consequent value and $\mu_{\Delta l}(x_1, \ldots, x_k)$ is the fuzzy evaluation of the antecedents, for all fuzzy rules.

$$y = \frac{\sum \mu_{\Delta l}\left(x_1, \ldots, x_k\right) \cdot y_l}{\sum \mu_{\Delta l}\left(x_1, \ldots, x_k\right)} \tag{7.18}$$

Since, the fuzzy-molecular inference model has a generic fuzzy inference engine, proper knowledge of a specific problem domain can be enclosed into a knowledge base. It is a matrix that summarizes all fuzzy rules of the form as (7.16) in the following way:

- For all input variables $x_1, \dots, x_k$, represent all possible combinations of them using the label of each set in the fuzzy partition of inputs, such that all antecedents in the fuzzy rules will be covered.
- For each combination (summary of antecedents), assign the corresponding label of molecule $M_j$ that will act when the fuzzy rule is fired.

At last, the fuzzy-molecular inference model combines interesting properties from both fuzzy logic and artificial hydrocarbon networks. Advantages of the FMI-model are the following:

- Fuzzy partitions in the output domain might be seen as linguistic units, e.g. "small", "large".
- Fuzzy partitions have a degree of understanding (parameters are metadata).
- Molecular units deal with noise and uncertainties.

It is remarkable to say that molecules are excited by consequent values, thus molecules does not model a given system, but transfer information from a fuzzy subspace to a crisp set. Moreover, molecular units have the property of filtering noise and uncertainties, especially important in real-world control applications.

#### 7.2.2.2 Design of Control Law Based on the FMI-Model

Consider the control law depicted in Fig. 7.11. It is based on the fuzzy-molecular inference model to implement a position controller of a DC motor.

For instance, the fuzzification step considers the two input variables (error signal $e(t)$ and first derivative of position signal $\dot{y}(t)$) in the partitioned space with three type-2 fuzzy sets: "negative" $(N)$, "zero" $(Z)$ and "positive" $(P)$. Figure 7.18 shows the fuzzy sets for input $e(t)$ and Fig. 7.19 shows the fuzzy sets for input $\dot{y}(t)$ . All parameters were tuned manually. Notice that type-2 fuzzy sets are determined using the primary membership function $\mu_A^U(x)$ and the secondary membership function $\mu_A^L(x)$, the latter as an additional value of uncertainty. As a matter of fact, the region inside both membership functions is called the footprint of uncertainty (7.19) of a fuzzy set $A$.

$$FOU(A) = \bigcup_{x \in X} \left( \mu_A^L(x), \mu_A^U(x) \right) \tag{7.19}$$

It is remarkable to say that when the secondary membership function is equal to the primary membership function, then the type-2 fuzzy inference system is reduced to a type-1 fuzzy inference system.

Then, the fuzzy inference engine calculates the consequent values of both primary and secondary membership values $\mu_A^U(x)$, $\mu_A^L(x)$ using the knowledge base shown

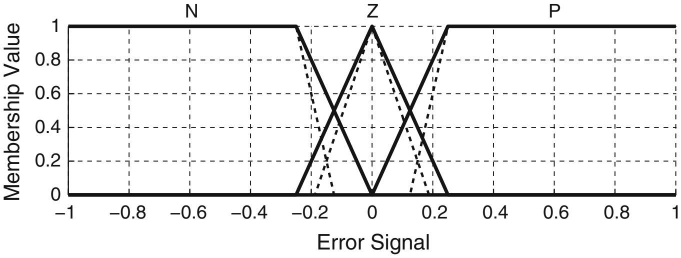


**Fig. 7.18** Fuzzy sets of the input error signal. (*solid line*) Primary membership function, (*dashed line*) secondary membership function

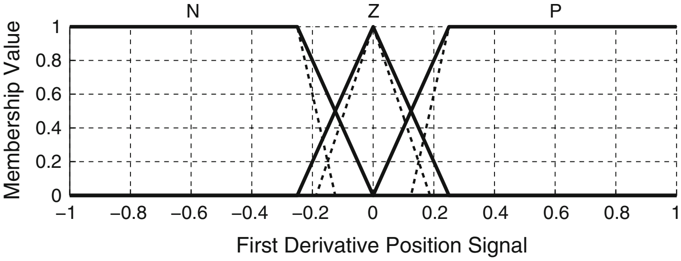


**Fig. 7.19** Fuzzy sets of the input first derivative position signal. (*solid line*) Primary membership function, (*dashed line*) secondary membership function

in Table 7.1. Consequent values $y_L$ and $y_U$ are similarly obtained using (7.18) for both primary and secondary membership values, respectively.

As noted in Table 7.1, the output signal was partitioned into three *CH*-molecules $M_j$ for all $j = 1, \ldots, 3$ that represent the action to be held. In particular, the output signal was partitioned into the following molecules: "clockwise" (*CW*), "halt" ($H$) and "counterclockwise" (*CCW*). Figure 7.20 shows the artificial hydrocarbon compound used for this position controller. Parameters were found using Algorithm 4.5.

At last, the Nie-Tan method was used for computing the final value of the output variable $u(t)$ for the type-2 fuzzy controller because of its simplicity of computation. Consider other methods like Karnik-Mendel, Greenfield-Chiclana, or Wu-Mendel for type-reduction. In fact, the Nie-Tan method generates a type reduction of the form as (7.20); where, $y$ is the crisp output value $u(t)$.

**Table 7.1** Knowledge base of FMI-model based fuzzy controller

| $e(t) \mid \dot{y}(t)$ | ***N*** | ***Z*** | ***P*** |
|---|---|---|---|
| ***N*** | $M_{CW}$ | $M_{CW}$ | $M_{CW}$ |
| ***Z*** | $M_{CW}$ | $M_H$ | $M_{CCW}$ |
| ***P*** | $M_{CCW}$ | $M_{CCW}$ | $M_{CCW}$ |

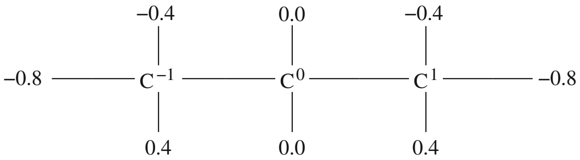


**Fig. 7.20** Artificial hydrocarbon network used in the FMI-model based position controller

$$y = \frac{y_L + y_U}{2} \tag{7.20}$$

### 7.2.3 *Results and Discussion*

In order to demonstrate that the fuzzy-molecular inference model for fuzzy control systems can be used as an alternative of type-2 fuzzy control systems, an experiment over system of Fig. 7.10. The experiment considers a comparison between the FIM-model based fuzzy control system designed previously and a Mamdani's fuzzy controller system designed with the same parameters for type-1 and type-2 fuzzy systems. Actually, the output variable was partitioned for the Mamdani's fuzzy controller system into three type-2 fuzzy sets: "clockwise" (*CW*), "halt" ($H$) and "counterclockwise" (*CCW*). Figure 7.21 shows this partition for the output variable $u(t)$.

#### 7.2.3.1 Type-1 Fuzzy System Comparison

For this experiment, the fuzzy-molecular position controller for a DC motor was reduced to a type-1 fuzzy system by only considering the primary membership functions in the fuzzification step, as well as in the Mamdani's fuzzy controller.

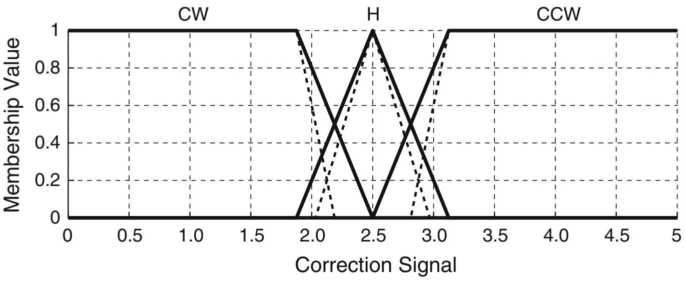


**Fig. 7.21** Fuzzy sets of the defuzzification step in the Mamdani's fuzzy control system. (*solid line*) Primary membership function, (*dashed line*) secondary membership function

The system was subjected to a step function without noise as shown in Fig. 7.22. Results of the FMI controller determine that it had a step response of 0 % of maximum overshoot, a rise time of 1.0 s and a maximum error of 2.5° in steady state. On the other hand, the system was subjected to a step function with 35 % of noise as shown in Fig. 7.23. Results of the FMI controller reports a 0 % of maximum overshooting, a rise time of 1.1 s and a maximum error of 5.8° in steady state measured from position 180°. For contrasting, Table 7.2 summarizes the overall results of the FMI and Mamdani fuzzy controllers.

Notice in Fig. 7.22 that the response of the FMI controller is 50 % faster than the response of the Mamdani controller and has a less value of maximum error in steady state than then Mamdani controller. In comparison, Fig. 7.23 shows that both fuzzy controllers are well stable as measured (5.8° and 5.5° of maximum error in steady state). However, FMI controller is still faster (1.1 s of rise time) than the response of the Mamdani controller (2.5 s of rise time). As noted, FMI controller has a better response for dynamic uncertainties than the Mamdani controller.

Also, the system was subjected to a ramp function without noise as shown in Fig. 7.24. Results determine that the FIM controller has a maximum error of 3.6° in steady state while the Mamdani controller has 6.7°. On the other hand, the system was subjected to a ramp function with 35 % of noise as shown in Fig. 7.25. The FMI controller reports 11.0° of maximum error in steady state and the Mamdani controller reports 12.3°. Also, Table 7.2 summarizes the overall results of this experiment with respect to the response of FMI and Mamdani fuzzy controllers.

It is evident from Table 7.2 that both fuzzy controllers decrease their performance in presence of noise. However, the FIM controller can track the reference signal better than the Mamdani controller, as shown in the steady state error. In addition, note that the FMI controller is slightly faster than the Mamdani controller.

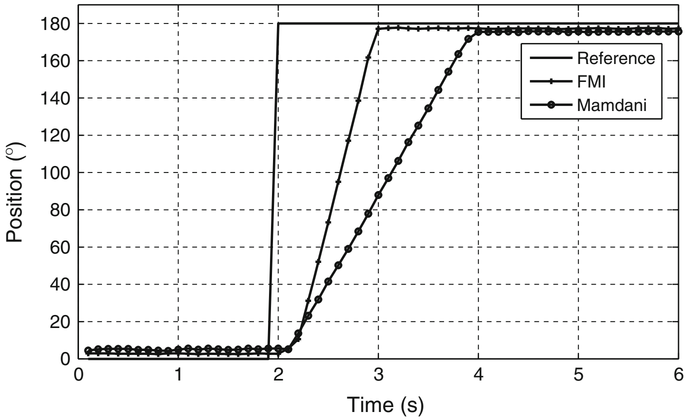


**Fig. 7.22** Comparison of the step response without noise of the FMI and Mamdani type-1 controllers

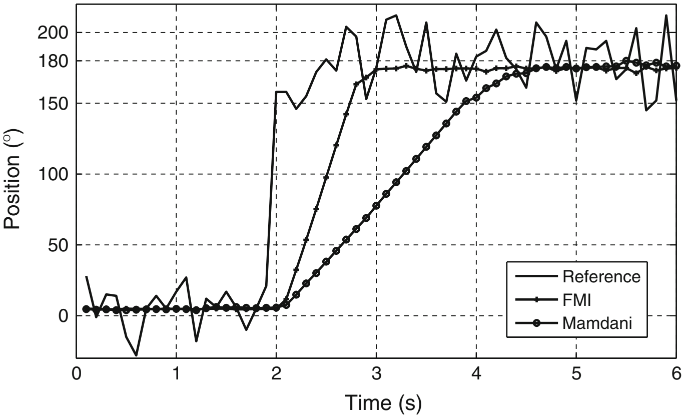


**Fig. 7.23** Comparison of the step response with 35 % noise of the FMI and Mamdani type-1 controllers

**Table 7.2** Experimental results of type-1 fuzzy controllers

| Fuzzy controller | Noise (%) | Rise time (s) | Steady-state error (°) |
|---|---|---|---|
| Step response | | | |
| *FIM* | 0 | 1.0 | 2.5 |
| | 35 | 2.0 | 5.8 |
| *Mamdani* | 0 | 1.1 | 4.7 |
| | 35 | 2.5 | 5.5 |
| Ramp response | | | |
| *FIM* | 0 | – | 3.6 |
| | 35 | – | 11.0 |
| *Mamdani* | 0 | – | 6.7 |
| | 35 | – | 12.3 |

#### 7.2.3.2 Type-2 Fuzzy System Comparison

The type-2 fuzzy-molecular position controller for a DC motor described previously was implemented as well as the type-2 Mamdani controller. Again, the system was subjected to a step function with 35 % noise and without it as shown in Figs. 7.26 and 7.27, respectively. The same process was done with a ramp function and the responses of both controllers are shown in Figs. 7.28 and 7.29, respectively. The overall results are summarized in Table 7.3.

As noted from Tables 7.2 and 7.3, the step responses of both FIM and Mamdani type-2 fuzzy controllers remain similar to type-1 controllers, as expected. Thus, type-1 and type-2 FIM fuzzy controllers are slightly equivalent with or without perturbations.

From Figs. 7.28 and 7.29, it can be seen that the response of type-2 fuzzy controllers are slightly better than type-1 controllers, as expected. From the point of view

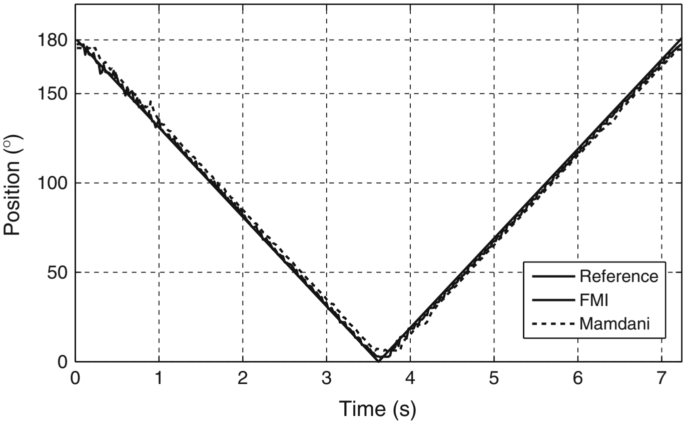


**Fig. 7.24** Comparison of the ramp response without noise of the FMI and Mamdani type-1 controllers

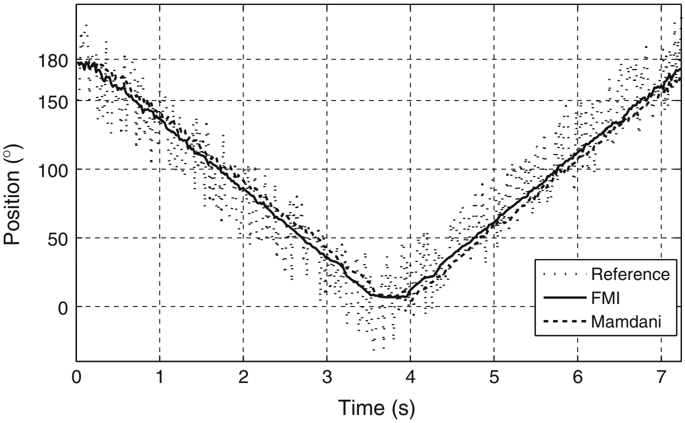


**Fig. 7.25** Comparison of the ramp response with 35 % noise of the FMI and Mamdani type-1 controllers

of ramp response, the FIM controller presents similar performance than the Mamdani controller without noise (3.8° and 3.7° maximum steady state errors, respectively). Again, both controllers present the same tendency when they are exposed to noise, and in comparison with type-1 controllers, type-2 fuzzy controllers act slightly better as found in Tables 7.2 and 7.3 (FIM: 17.2 % better, Mamdani: 1.7 % better).

On one hand, from the above results, fuzzy-molecular inference models can achieve fuzzy control applications. Moreover, these FIM-model based controllers can be used as an alternative for type-2 fuzzy control systems. This statement comes from the evaluation and comparison of step and ramp responses between the FIM-controller and the Mamdani fuzzy controller, both models subjected to static and

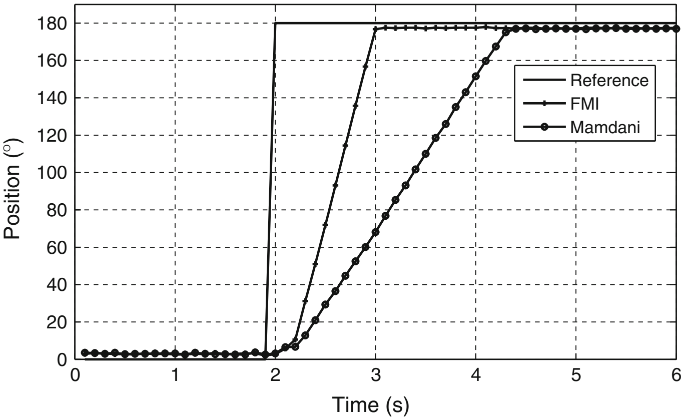


**Fig. 7.26** Comparison of the step response without noise of the FMI and Mamdani type-2 controllers

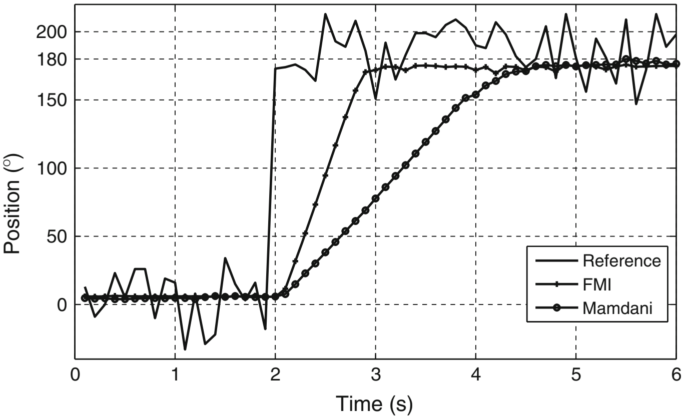


**Fig. 7.27** Comparison of the step response with 35 % noise of the FMI and Mamdani type-2 controllers

dynamic uncertainties. In this case study, a Mamdani's fuzzy control system was used because it is the fuzzy inference system most implemented in industry.

On the other hand, it is important to distinguish the fuzzy-molecular inference model from other fuzzy inference models like Takagi-Sugeno inference systems or Mamdani's fuzzy control systems. For instance, the defuzzification process in each fuzzy inference model is different. As FMI-model uses artificial hydrocarbon networks, each molecule represents a linguistic partition of the output variable. In above results simple *CH*-molecules were implemented, but complex molecules can either be used. Thus, defuzzification can have complex nonlinear mappings in the FMI-model. In contrast, Takagi-Sugeno's model uses polynomial functions and Mamdani's model represents linguistic partitions with membership functions

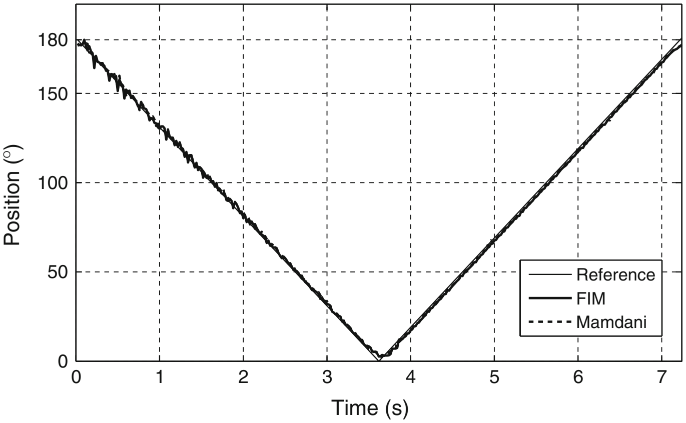


**Fig. 7.28** Comparison of the ramp response without noise of the FMI and Mamdani type-2 controllers

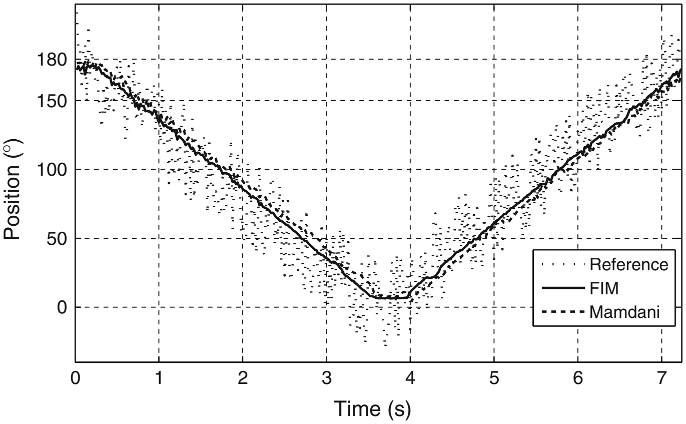


**Fig. 7.29** Comparison of the ramp response with 35 % noise of the FMI and Mamdani type-2 controllers

associated to fuzzy sets. Parameters inside artificial hydrocarbon networks are hydrogen and carbon values, polynomial coefficients for Takagi-Sugeno's model, and parameters of membership functions in Mamdani's model. In addition, molecules in FMI-model make a mapping from membership or truth-values to output values, also dealing with uncertainties. This is remarkable because Takagi-Sugeno's model maps from input values to output values and using fuzzy inference values linearly acts on the final output value. At last, Mamdani's model makes a mapping from membership values to output values. In fact, the fuzzy-molecular inference model combines linguistic partitions of output variables with molecular structures.

**Table 7.3** Experimental results of type-2 fuzzy controllers

| Fuzzy controller | Noise (%) | Rise time (s) | Steady-state error (°) |
|---|---|---|---|
| Step response | | | |
| *FIM* | 0 | 1.0 | 2.5 |
| | 35 | 1.0 | 5.0 |
| *Mamdani* | 0 | 2.4 | 4.7 |
| | 35 | 2.6 | 5.5 |
| Ramp response | | | |
| *FIM* | 0 | – | 3.8 |
| | 35 | – | 9.1 |
| *Mamdani* | 0 | – | 3.7 |
| | 35 | – | 12.1 |

## 7.3 Facial Recognition Based on Signal Identification Using AHNs

Nowadays, facial recognition systems are widely used for identifying, verifying or tracking people from images or video frames, security, government activities, entertainment and so forth. In security, facial recognition can be considered as a biometric system that analyzes, extracts and classifies features, landmarks or any other information within images. In that context, the following case study applies artificial hydrocarbon networks with the discrete cosine transform to easily implement a facial recognition system.

### *7.3.1 Background and Problem Statement*

Facial recognition is a challenging problem in computer vision because it is computationally complexity due to position, illumination, environmental and instrumental noise, pose, etc. that affect the performance of algorithms [8]. In that sense, many robust facial recognition algorithms have been developed [3, 8]. However, undesirable effects are drawn when they are applied in real-world applications. Several algorithms are widely used, for example [8]: background removal, illumination normalization, artificial neural networks, discrete cosine transform, and hybrid algorithms. In that context, the purpose of this case study is to propose a novel facial recognition system based on metadata of artificial hydrocarbon networks in order to minimize timing of feature matching.

For instance, consider the facial recognition system of Fig. 7.30. As noted, it consists of the following steps: *pre-processing*, *feature extraction* and *classification*.

The first block applies a transformation of input images in order to obtain suitable images for a specific problem. Several pre-processing techniques typically used in

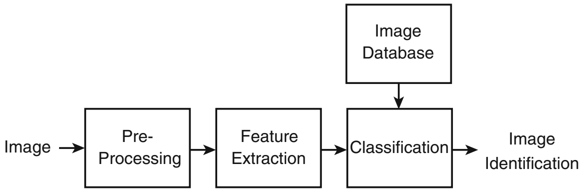


**Fig. 7.30** Block diagram of a facial recognition system

facial recognition systems are: scaling, edge detection, histogram equalization and color to grayscale [8].

Then, feature extraction refers to the process of representing an image in a punch of relevant parameters. One of the most prominent feature extraction techniques in imaging is the so-called discrete cosine transform (DCT) [3, 12]. It is a separable linear transformation that compact the energy of an image in frequency. For a two-dimensional image, the DCT is expressed as (7.21); where $I(x, y)$ is the intensity of pixel at $(x, y)$ coordinates, $N$ and $M$ are two integers values related to the size of the image $M \times N$, $u$ and $v$ vary from $0, \ldots, N-1$ and $0, \ldots, M-1$, respectively, and $\alpha(u)$, $\alpha(v)$ hold (7.22).

$$F(u, v) = \alpha(u)\alpha(v) \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} I(n, m) \cos\left(\frac{\pi u(2n+1)}{2N}\right) \cos\left(\frac{\pi v(2m+1)}{2M}\right) \tag{7.21}$$

$$\alpha(u) = \begin{cases} \sqrt{\frac{1}{N}} & u = 0 \\ \sqrt{\frac{2}{N}} & u \neq 0 \end{cases}$$
$$\alpha(v) = \begin{cases} \sqrt{\frac{1}{M}} & v = 0 \\ \sqrt{\frac{2}{M}} & v \neq 0 \end{cases} \tag{7.22}$$

Moreover, low frequencies of DCT contain the average intensity of an image that are very useful in facial recognition systems. High frequencies of DCT contain other components of an image like noise and illumination variations [8]. Since, the first values of the DCT are required, a zigzag ordering or raster scan order is computed to reorder the values of DCT in a vector, placing the most important values at first. The above technique avoids some problems of pose and position.

Finally, a classification process is done using the resulting vector from feature extraction analysis and obtaining a matching image, previously recorded on a database. The most common technique is the Euclidean classifier (7.23) [8]; where, $D$ is the distance between two vector of images, $p_i$ is the $i$-th element of the target vector (i.e. from the repository) and $q_i$ is the $i$-th element of the testing vector image. To this end, the minimum distance $D$ associated to a testing image means the matching of it with a target image.

$$D = \sqrt{\sum_{i=1}^{N} (p_i - q_i)^2} \tag{7.23}$$

As noted, (7.23) is computed $k$ times for the $k$ target images in the repository. Moreover, if the database of target images is too large, then the computation of the set of (7.23) will take a huge amount of time. Then, other techniques have to be proposed.

**Problem 7.3** Consider the above facial recognition system. The objective is to design an easier facial recognition system under variant pose using artificial hydrocarbon networks.

### *7.3.2 Methodology*

The proposed facial recognition system is shown in Fig. 7.31. As reviewed earlier, the facial recognition system considers three main components: pre-processing, feature extraction and classification.

For this application, consider an input red-green-blue (RGB) image $I(x, y)$ with $M \times N$ pixels. Initially, the pre-processing step computes a RGB-color to grayscale transformation of image $I(x, y)$ to a grayscale image $G(x, y)$.

Once the latter is obtained, a feature extraction step over $G$ is computed. The two-dimensional discrete cosine transform is applied to $G(x, y)$ obtaining a transformed image $F(u, v)$ in the frequency domain. Later, a zigzag ordering is applied to $F$ giving a vector $\mathbf{v}$ with low frequency elements at first. In particular to this application, the first 50 elements in $\mathbf{v}$ are considered. Furthermore, $\mathbf{v}$ is highly dynamic but for a set of face images with different poses, the set of $\mathbf{v}$ vectors tends to be equal. Then, taking

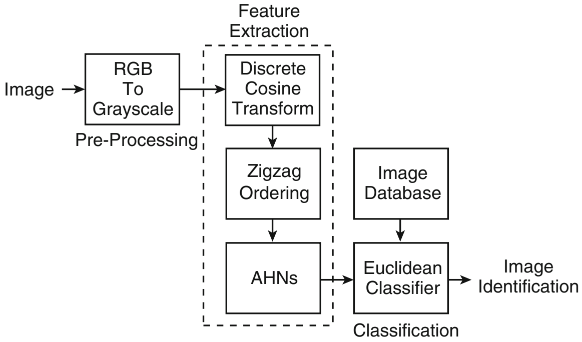


**Fig. 7.31** Facial recognition system based on artificial hydrocarbon networks

advantage of the latter, $\mathbf{v}$ is approximated with an artificial hydrocarbon network *AHN* with $c_{max} = 1$ and a fixed number of *CH*-primitive molecules $n = 6$. Recall that an AHN-model filters a system but also finds interesting points.

For example, consider a facial image like in Fig. 7.32. Doing the above procedure, the image is transformed to a grayscale image and the DCT is computed. Later on, the zigzag ordering rearranges the DCT frequency image into a vector of features truncated to its first 50 elements. Finally, an approximation of that vector is obtained with an artificial hydrocarbon network. Notice that the transitions between *CH*-primitive molecules capture the tendency of elements in the vector. Then, intermolecular distances (or alternatively bounds of molecules) can be used for capturing information in the vector of DCT elements of an image.

Finally, a feature vector $f_T$ with intermolecular distances (or alternatively bounds of molecules) of a target image $I_T$ is stored in the database. Then, $k$ different feature vectors $f_T$ will be stored in the image gallery. In order to compute the feature vector $f_T$ for a set of similar target images, the set of DCT ordering vectors $\{v\}$ of each sample image are used for training the *AHN*-structure associated to $f_T$.

On the other hand, the facial recognition system calculates the feature vector $f_S$ of a testing image $I_S$. Thus, the classification step uses the Euclidean classifier (7.23) with $p_i$ as the $i$-th element of a target vector $f_T$ and $q_i$ as the $i$-th element of a testing vector $f_S$. The minimumdistance $D_T$ associated to a target vector $f_T$ will determine

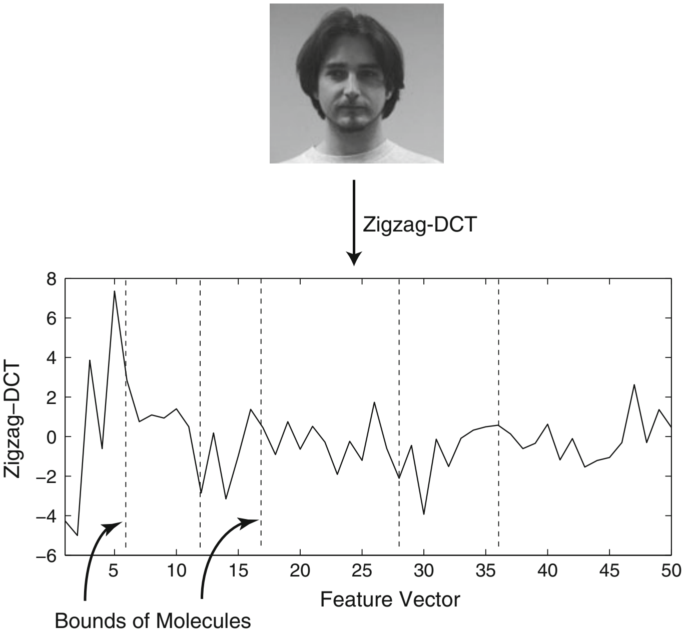


**Fig. 7.32** An example of using AHN-metadata for capturing features of facial images

the matching of $I_S$ as (7.24). To this end, $C$ is the matching index of two images $I_S$ and $I_T$.

$$C = \arg\min_{T} \{D_T\} \tag{7.24}$$

### 7.3.3 Results and Discussion

Fifty images from the public head pose image database of Gournier, Hall and Crowley [2] were used. Selected images come from 10 different people with 5 different poses each. Figure 7.33 shows an example of pose variations ranging from +30° to −30° in pan angles with 0° tilt angle, in the set of selected images. In that sense, the training set consists of 60 % of the latter set, i.e. 3 random poses per person. Figure 7.34 shows one pose of the 10 people in the training sample.

The above facial recognition system based on artificial hydrocarbon networks were implemented. Ten different AHN-structures were obtained, one for each target person. These structures were trained with a fixed number of molecules $n = 6$, one compound $c = 1$ and a step size of $\eta = 0.05$. At last, one feature vector $f_T$ with 7 elements (bounds of molecules) was found for each person. Table 7.4 shows the resultant set of feature vectors for each person.

Then, the 50 selected images were used as the testing set. The facial recognition system based on artificial hydrocarbon networks has a misclassification rate of 10 %. In fact, the system can order the facial identification from the most probable person to the less probable one. Actually, the system can identify 64 % of the images as

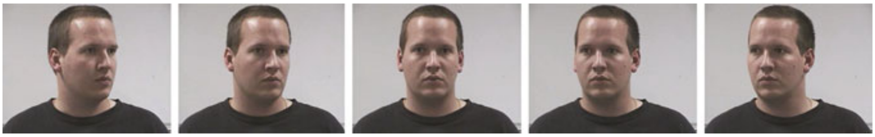


**Fig. 7.33** Collection of images from *person* 1 in the set of images in the case study. All faces have 0° tilt-angle, and the pan-angle are +30°, +15°, 0°, −15° and −30° from *left* to *right*

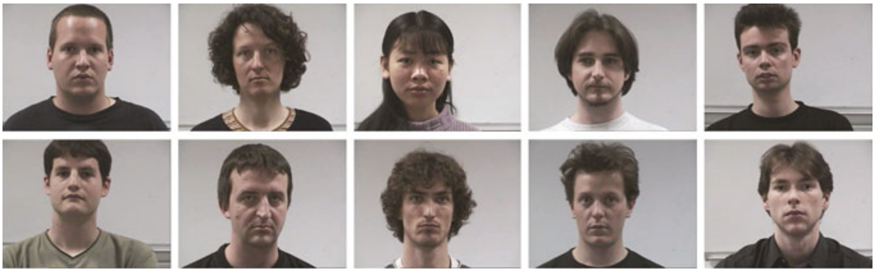


**Fig. 7.34** Image samples, (*top*) *person* 1 to *person* 5, (*bottom*) *person* 6 to *person* 10

its first choice, 18 % of the images as its second choice, and 8 % of the images as its third choice. Figure 7.35 shows the recognition rate from each person versus the choice position.

As shown in Fig. 7.35, the system can easily recognize images from people 1, 3, 6, 7, 9 and 10, while people 2, 5 and 8 are the most difficult ones to recognize. It is evident from Fig. 7.34 that person 2 and person 8 are visually similar, explaining the difficulty of recognition.

Concluding, the above experimental results demonstrate that a facial recognition system based on artificial hydrocarbon networks can be easily implement for variant poses. Interestingly, this approach uses metadata of AHN-structures, in contrast to other applications.

**Table 7.4** Set of feature vectors for each person in the set of images in the case study

| *Person* | $L_0$ | $L_1$ | $L_2$ | $L_3$ | $L_4$ | $L_5$ | $L_6$ |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 0 | 0 | 6 | 10 | 12 | 50 |
| 2 | 1 | 5 | 9 | 13 | 25 | 38 | 50 |
| 3 | 1 | 5 | 10 | 21 | 27 | 38 | 50 |
| 4 | 1 | 5 | 11 | 17 | 27 | 37 | 50 |
| 5 | 1 | 4 | 8 | 12 | 19 | 33 | 50 |
| 6 | 1 | 5 | 10 | 15 | 24 | 33 | 50 |
| 7 | 1 | 5 | 10 | 16 | 30 | 39 | 50 |
| 8 | 1 | 4 | 9 | 17 | 26 | 38 | 50 |
| 9 | 1 | 5 | 10 | 18 | 29 | 39 | 50 |
| 10 | 1 | 1 | 4 | 5 | 10 | 20 | 50 |

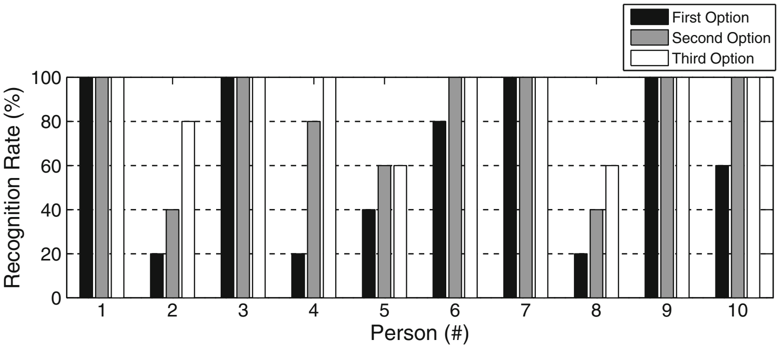


**Fig. 7.35** Variation of recognition rate with the option number. It represents the cumulative probability of image $I_S$ to be correctly classified

## References

1. Giura MD, Serina N, Rizzotto G (1997) Adaptive fuzzy filtering for audio applications using a neuro-fuzzy modelization. In: Proceedings of the international joint conference on neural networks, vol 4. MD, USA, pp 2162–2166
2. Gournier N, Hall D, Crowley JL (2004) Estimating face orientation from robust detection of salient facial features. In: Proceedings of pointing, ICPR, international workshop on visual observation of deictic gestures, Cambridge, UK, pp 17–25
3. Hafed ZM, Levine MD (2001) Face recognition using the discrete cosine transform. Int J Comput Vision 43(3):167–188
4. Iancu I (2012) A Mamdani type fuzzy logic controller. In: Dadios EP (ed) Fuzzy logic: controls, concepts, theories and applications. InTech Croatia, Rijeka, pp 325–350
5. Jablonski R, Turkowski M, Szewczyk R (eds) (2007) Recent advances in mechatronics. Springer, Berlin
6. Linda O, Manic M (2010) Comparative analysis of type-1 and type-2 fuzzy control in context of learning behaviors for mobile robotics. In: IECON—36th annual conference on IEEE industrial electronics society, pp 1092–1098
7. Mamdani HE (1974) Application of fuzzy algorithms for control of simple dynamic plant. In: IEEE proceedings of the institution of electrical engineers, vol 121, pp 1585–1588
8. Manikantan K, Govindarajan V, Kiran VVSS, Ramachandran S (2012) Face recognition using block-based DCT feature extraction. J Adv Comput Sci Technol 1(4):266–283
9. Mendel JM, John RIB (2002) Type-2 fuzzy sets made simple. IEEE Trans Fuzzy Syst 10(2):117–127
10. Musikasuwan S, Garibaldi JM (2006) On relationships between primary membership functions and output uncertainties in interval type-2 and non-stationary fuzzy sets. In: IEEE international conference on fuzzy systems, pp 1433–1440
11. Ogata K (2002) Modern control engineering. Prentice Hall, Upper Saddle River
12. Rao KR, Yip P (eds) (2010) The transform and data compression handbook. Taylor & Francis, London
13. Smith J (2010) Introduction to digital filters with audio applications. Stanford University Press, Stanford
14. Smith SW (1997) The scientist and engineers guide to digital signal processing. California Technical Publishing, San Diego
15. Taal C, Hendriks R, Heusdeus R, Jensen J (2011) An algorithm of intelligibility prediction of time-frequency weighted noisy speech. IEEE Trans Audio Speech Lang Process 19(7):2125–2136
16. Takagi T, Sugeno M (1985) Fuzzy identification of systems and its applications to modeling and control. IEEE Trans Syst Man Cybern 15(1):116–132
17. Tsukamoto, Y (1979) An approach to fuzzy reasoning method. In: Gupta M, Ragade R, Yager R (eds) Advances in fuzzy set theory and applications. Elsevier, Amsterdam, pp 137–149

# Appendix A
# Brief Review of Graph Theory

Artificial organic networks as well as artificial hydrocarbon networks involve key concepts of graph theory. In this context, relevant definitions of graphs are summarized following.

**Definition A.1** ***(general graph)*** *Let $G = (V, E, \phi)$ be an ordered triple, where $V \neq \varnothing$, $V \cap E = \varnothing$ and $\phi \rightarrow P(V)$ is a map from the set $E$ to the power set of $V$, such that, $|\phi(e)| \in \{1, 2\}$ for each $e \in E$. Then, $G$ is called a* general graph*, $V$ is a set of* vertices *of $G$, $E$ is a set of edges of $G$, and $\phi$ is an* edgemap *with $\phi(e)$ called* endvertices.

In Fig. A.1 are shown different general graphs. Notice that edgemaps assure that edges has associated up to two vertices, i.e. $|\phi(e)| = 1$ if the edge loops at the same vertex and $|\phi(e)| = 2$ if the edge connects two different vertices.

**Definition A.2** ***(simple graph)*** *Let $G = (V, E)$ be an ordered pair, where $V \neq \varnothing$ and $E$ is a set of 2-element subsets of $V$, such that $E \subseteq \{\{u, v\}|u, v \in V, u \neq v\}$. Then, $G$ is called a simple graph.*

As noted, a simple graph is a particular case of general graphs. In fact, simple graphs do not accept edges connecting one vertex to itself, as shown in Fig. A.2. Actually, artificial organic networks and artificial hydrocarbon networks use the notion of simple graphs to define molecular units.

**Definition A.3** ***(degree of vertex)*** *Let $G = (V, E)$ be a graph and $u \in V$ be a vertex of $G$. The* degree *of $u$, denoted by $d_G(u)$, is defined as:*

$$d_G(u) = |\{e \in E | e = \{u, u_i\}, \forall u_i \neq u, u_i \in V\}| \tag{A.1}$$

Where*,* $|S|$ stands for the cardinality of the set $S$.

Figure A.3 shows a simple graph with the degree of its vertices, denoting the number of edges (relationships) that a vertex has.

H. Ponce-Espinosa et al., *Artificial Organic Networks*,
Studies in Computational Intelligence 521, DOI: 10.1007/978-3-319-02472-1,

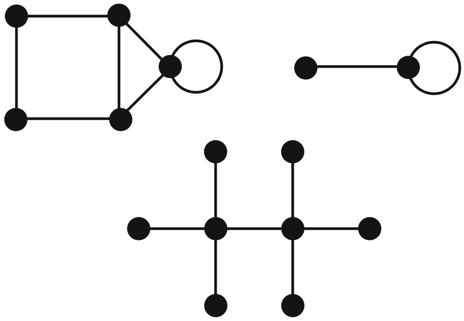


**Fig. A.1** Examples of general graphs

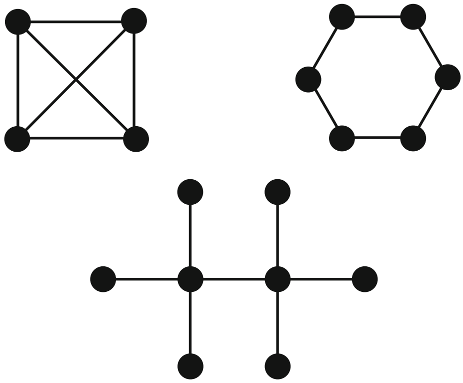


**Fig. A.2** Examples of simple graphs

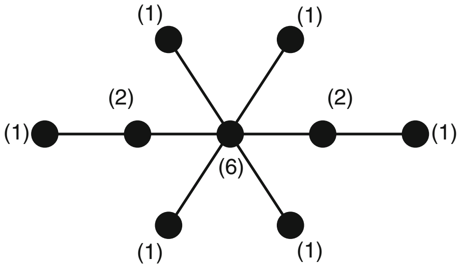


**Fig. A.3** Degree of vertices in a simple graph

**Definition A.4** ***(subgraph)*** *Let $G = (V, E)$ and $G' = (V', E')$ be two graphs. Then, $G'$ is a subgraph of $G$, denoted $G' \subseteq G$, if $V' \subseteq V$ and $E' \subseteq E$.*

From Definition A.4, it can be concluded that $G$ is *induced* by $G'$. This means that $G$ is an *extension* or it is *spanned* from $G'$. Thus, artificial organic networks and artificial hydrocarbon networks use the terms *induced* and *spanned* indistinctly. Figure A.4 shows some examples of subgraphs contained in a simple graph.

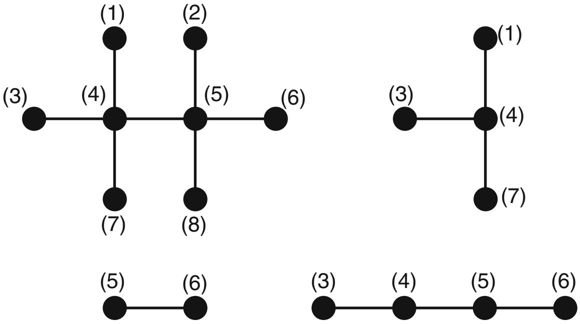


**Fig. A.4** Examples of subgraphs in a simple graph

# Appendix B
# Experiment of Signal-Molecule Correlation

The following experiment was run in order to find any correlation between the output response of a system and the type of *CH*-primitive molecules that form an artificial hydrocarbon compound, using the energy information of the system. The methodology and results of the experiment are present below.

## B.1 Overview of the Experiment

Let $\Sigma$ be a system with an input signal $x$ and an output signal $y$ such that the behavior of the system relates the input and the output as $\Sigma(x) = y$. Also, let suppose that there is no other information about the system. Then, assume that an artificial hydrocarbon network $AHN = \langle \Gamma_f, \Phi, \Lambda, x \rangle$ with behavior $S(x)$ has to approximate $\Sigma$. Hence, the build topological structure problem (BTSP) claims for finding molecules and compounds, $\Gamma_f$, of an artificial hydrocarbon network using an algorithm $f$, such that *AHN* can approximate $\Sigma$. In that sense, an experiment was designed and run in order to find any correlation between $\Sigma$ and $\Gamma_f$.

For instance, consider the energy $E$ of signal $y$ in the interval $x \in [a, b]$, calculated as (B.1); where, $\|g\|$ stands for the norm of $g$.

$$E(y) = \int_a^b \|y\|^2 dx \ , \ x \in [a, b] \tag{B.1}$$

Thus using (B.1), the energy $E_i$ of a partition $\Sigma_i$ is expressed as (B.2); where, $x_i$ is the input and $y_i$ is the output signals of $\Sigma_i$ in the interval $x_i \in [r_1, r_2]$.

$$E_i = E(y_i) = \int_{r_1}^{r_2} \|y_i\|^2 dx \ , \ x_i \in [r_1, r_2] \tag{B.2}$$

H. Ponce-Espinosa et al., *Artificial Organic Networks*,
Studies in Computational Intelligence 521, DOI: 10.1007/978-3-319-02472-1,

From signal theory, an energy signal is said to be any signal $s$ such that $0 < E(s) \leq \infty$. Thus, $y$ is an energy signal, and also $S(x)$. In that sense, the output response $y$ might be related to the behavior $S(x)$.

## B.2 Design of the Experiment

The experiment aims to determine the relationship between the energy signal of the system and the structure of an artificial hydrocarbon compound. Initially, a set of energy signals was defined and a prototype of an artificial hydrocarbon compound was designed.

### *B.2.1 Sample Definition*

The sample of the experiment consists in a subset $S_T$ of test signals $s_k \in S_T$ of the infinite set of all possible causal, aperiodic, energy signals in the closed interval $x \in [a, b]$ such that the energy of those is $0 < E(s_k) \leq \infty$. From signal theory, the subset of test signals was randomly constructed with a composite of fundamental signals (unit step, rectangular pulse, signed function, ramp, sine, sinc, exponential, and unit impulse signals). In practice, the input domain was arbitrary chosen to be $x \in [0, 5]$ and the experiment required a 99 % confidence. Thus, a total of 22,312 test signals were chosen randomly.

### *B.2.2 Design of Artificial Hydrocarbon Compound Prototypes*

A set of artificial hydrocarbon compounds $S_C$ was designed using $n = 2$ $CH$-primitive molecules. In order to create the most stable compounds with two $CH$-primitive molecules, three compounds $C_i \in S_C$ for all $i = 1, 2, 3$ were proposed: a simple bond based molecule $C_1 = (CH_3 - CH_3)$, a double bond based molecule $C_2 = (CH_2 = CH_2)$ and a triple bond based molecule $C_3 = (CH \equiv CH)$. Actually, each molecule has behavior $\varphi_j$ for all $j = 1, 2$, and each compound $C_i$ has behavior $\psi_i$ expressed as a piecewise function (B.3); where, $r_1$ is the middle position in the interval, i.e. $r_1 = \frac{1}{2}(a + b)$.

$$\psi_i(x) = \begin{cases} \varphi_1(x) & a \leq x < r_1 \\ \varphi_2(x) & r_1 \leq x \leq b \end{cases} \tag{B.3}$$

**Algorithm B.1** Methodology of the signal-molecule correlation experiment.

```
for each test signal $s_k \in S_T$ do
    Measure $E(s_k)$ using (B.2)
    for each $i = 1 : 3$ do
        Approximate $s_k$ with behavior $\psi_i$ of $C_i$ using least squares estimates
        Measure $E(\psi_i)$ using (B.2)
    end-for
    Classify $s_k$ into group of $Class_i$
end-for
Get statistics of each class
```

### B.2.3 Methodology

The experiment consists of classifying each test signal $s_k \in S_T$ depending on how close the energy of the signal $E(s_k)$ is equal to the energy of the compound behavior $E(\psi_i)$ via one of the three compounds $C_i \in S_C$. Three groups for classification are possible:

- $Class_1$: The best approximation via simple bond based molecules.
- $Class_2$: The best approximation via double bond based molecules.
- $Class_3$: The best approximation via triple bond based molecules.

The methodology for the experiment adopted is depicted in Algorithm B.1.

In order to classify each test signal, the criterion is based on the minimum distance $e_i$ between the energy of test signal $E(s_k)$ and the energy of each compound behavior $E(\psi_i)$ of compounds $C_i \in S_C$ for all $i = 1, 2, 3$. This criterion is synthetized in (B.4) with distance (B.5); where, $|g|$ stands for the absolute value of $g$.

$$s_k \in Class_m \ , \ m = \arg\min_i(e_i) \tag{B.4}$$

$$e_i = |E(s_k) - E(\psi_i)| \ , \ i = 1, 2, 3 \tag{B.5}$$

## B.3 Results and Discussion

After the experiment was run, the distribution of each class was obtained, as shown in Fig. B.1. Qualitatively, it can be seen that $Class_1$ has the greatest number of samples, in comparison with $Class_3$ which it has the lowest number of test signals. Table B.1 summarizes statistics from Fig. B.1. In fact, from Table B.1, $Class_1$ has 96.4 % of test signals, $Class_2$ has 2.8 % of test signals, and $Class_3$ only has 0.08 % of test signals. In that sense, it shows a strong relationship to real organic tendencies in which the most number of forming bonds in nature are subjected to simple bonds, then double bonds appear less, and the number of triple bonds appears last.

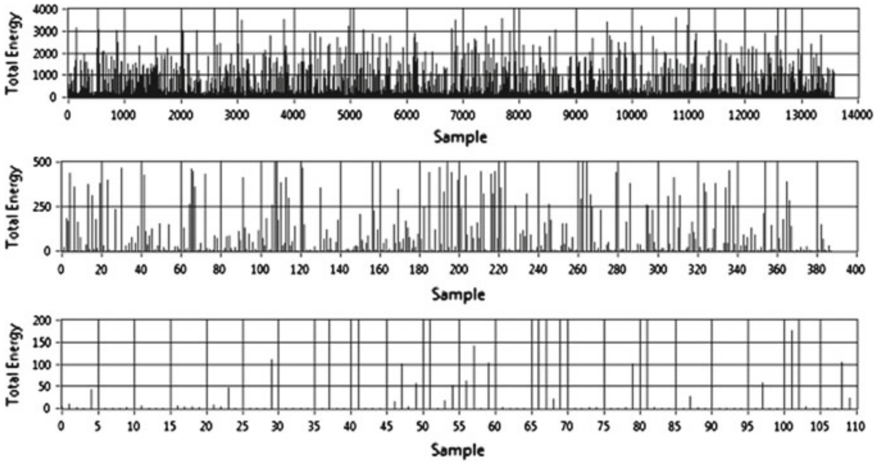


**Fig. B.1** Distribution of classes. **a** *Class*$_1$: the best approximation via *simple bond* molecules, **b** *Class*$_2$: the best approximation via *double bond* molecules, and **c** *Class*$_3$: the best approximation via *triple bond* molecules

**Table B.1** Statistics of classes

| Type of class | Mean energy | Standard deviation | Size |
|---|---|---|---|
| 1 | 152.74 | 341.64 | 13,577 |
| 2 | 132.21 | 488.04 | 388 |
| 3 | 48.18 | 108.27 | 110 |

Table B.1 also reports the mean energy. Roughly, it can be interpreted as the measurement of dynamics of signals: the greater the mean energy is, the more dynamic the signal is. Actually, dynamics of signals is due to oscillations and large values of amplitude signals, as shown in Figs. B.2, B.3 and B.4. Interestingly, all signals in each class are very similar.

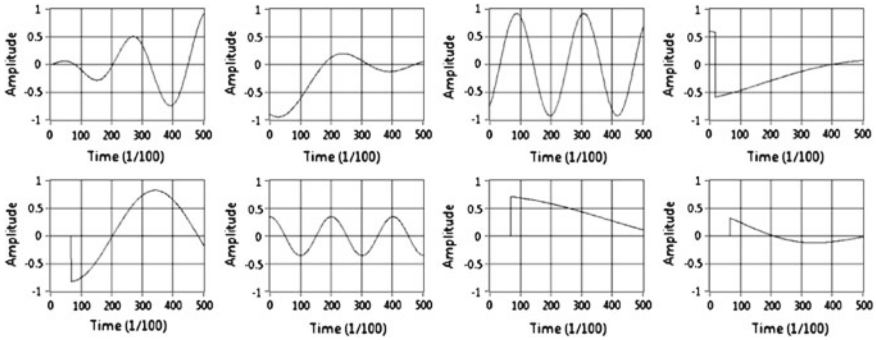


**Fig. B.2** *Class*$_1$: *highly dynamic*, unstable signals

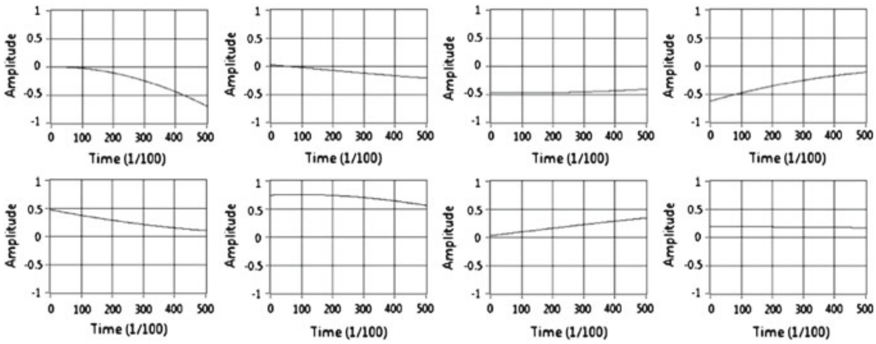


**Fig. B.3** *Class*$_2$: *medium dynamic*, quadratic-based signals

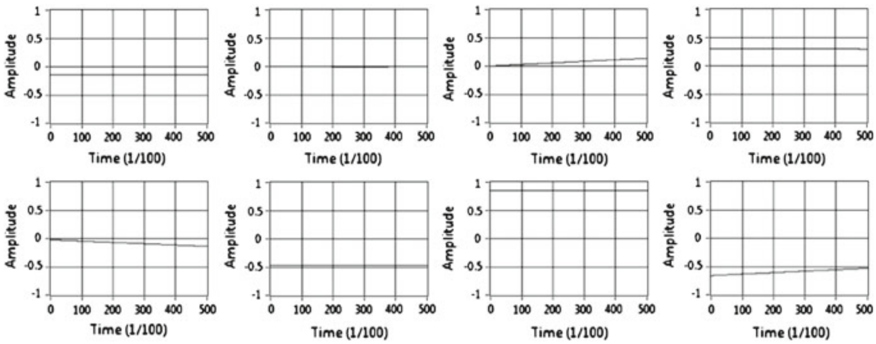


**Fig. B.4** *Class*$_3$: *low dynamic*, linear and/or constant quasi-stable and stable signals.

For instance, Fig. B.2 shows the tendency of profiles in *Class*$_1$; where signals are irregular and with some peaks, a few more are periodic—several presenting phase offset—, and they may have discontinuities. *Class*$_2$ presents smooth signals with concavities shaping quadratic functions as observed in Fig. B.3. Finally, *Class*$_3$ depicted in Fig. B.4 presents constant and quasi-constant signals and some others are linear. Thus, a relationship between energy in signals and compound behaviors (explained by the dynamics of signals) was revealed. In nature, the most flexible organic compounds present large staturated, linear chains and branched chains, while the most rigid and stable organic compounds are made of double or triple bonds (or by symmetric geometries, out of scope). This notion is shown in artificial hydrocarbon networks by the above discussion in which signals with simple bonds (*Class*$_1$) are not too stable, but dynamic; signals with double bonds (*Class*$_2$) are less dynamic; and, signals with triple bonds (*Class*$_3$) are the most stable ones. In fact, another interpretation of experimental results deals with the notion of flexibility. Structures of artificial hydrocarbon network with lower order of bonds are more flexible than structures of artificial hydrocarbon networks with higher order of bonds. Finally, Table B.1 is used for selecting the best pair of molecules when the energy of an output signal of any given system can be measured.

# Appendix C
# Practical Implementation of Artificial Hydrocarbon Networks

This appendix provides example codes of the programs employed in the real-world applications depicted in Chap. 7. For detailed information about background and description of these applications, refer to the latter chapter. In addition, a quick guide of implementing artificial hydrocarbon networks is presented at the end of the appendix. If required, Appendix D summarizes the documentation of the Artificial Organic Networks Toolkit using LabVIEW$^{TM}$.

## C.1 Program Codes for Audio Filtering

The adaptive audio filter based on artificial hydrocarbon networks of Algorithm 7.1 is implemented in LabVIEW$^{TM}$ as shown in Fig. C.1. First, the original signal is obtained from the **WAV** SubVI (see Fig. C.2). As shown in Fig. C.2, the audio signal is extracted from a *.wav file using the first channel (monochannel). The signal is transformed into a $(X, Y)$ system. Then, this signal is corrupted with white noise using the **Uniform White Noise** VI (Fig. C.1).

The corrupted signal, also expressed as a $(X, Y)$ system, enters to a **While Loop** in which the signal is splitted into windows (batches) using the **BATCH** SubVI depicted in Fig. C.3. This batch code receives an initial time value (**initial**) and the length in time (**WindowTime**). The outputs of the **BATCH** SubVI are two arrays containing a portion of the system $(X, Y)$ in the given interval $[WindowTime + initial, initial]$.

Once a segment of corrupted signal is obtained, it is introduced to the **ModifiedSimpleAHN** VI (see Fig. C.1) which it computes the modified artificial hydrocarbon networks algorithm. In fact, it receives its training parameters (**Training Properties**) with the step size (**StepSize**), the maximum number of molecules in the compound (**NumberOfMolecules**), the maximum number of compounds fixed to one in this application, and the tolerance value (**Tolerance**). As a result, this VI gives the response (**MixtureResponse**) of the trained artificial hydrocarbon network. At last, both the corrupted signal and the adaptive AHN-filter are shown in a graph (**XY Graph**).

H. Ponce-Espinosa et al., *Artificial Organic Networks*,
Studies in Computational Intelligence 521, DOI: 10.1007/978-3-319-02472-1,

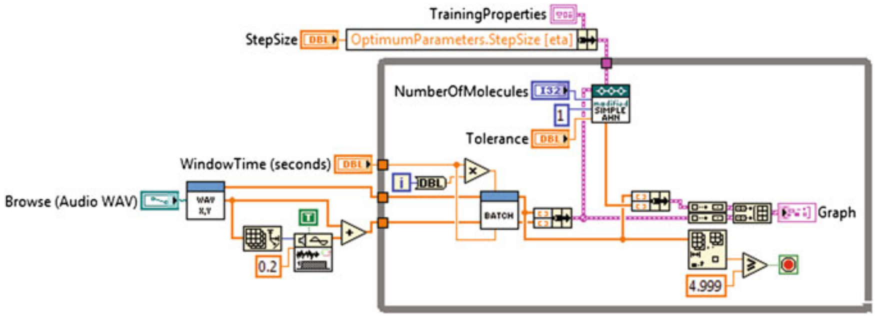


**Fig. C.1** Block diagram of the simple adaptive AHN-filter

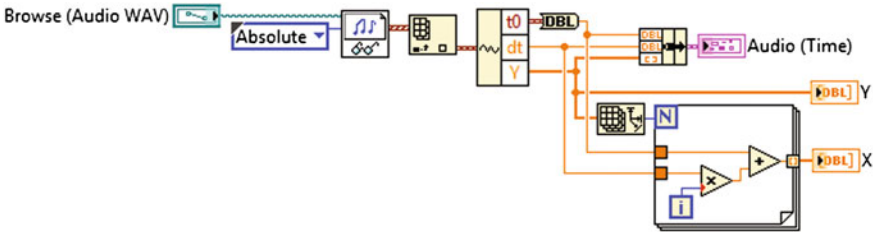


**Fig. C.2** Block diagram of the **WAV** SubVI in Fig. C.1

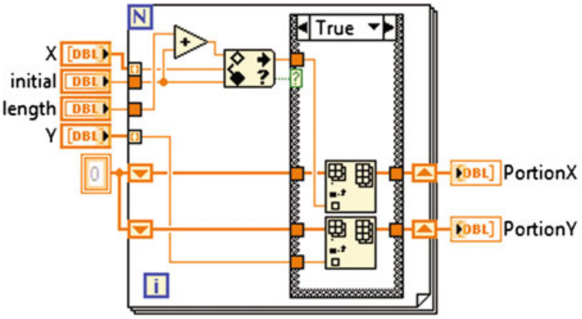
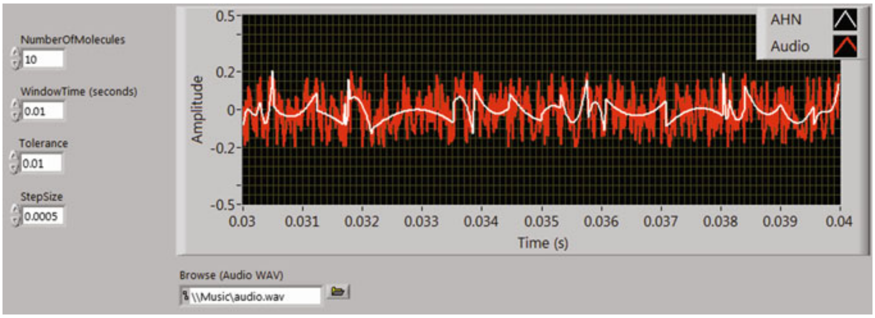


**Fig. C.3** Block diagram of the **BATCH** SubVI in Fig. C.1

For this example code, the audio signal had 5 s. Thus, the stop criterion of the **While Loop** is fixed to compare the time of the signal to be greater or equal to the value 4.999. The front panel of the code is shown in Fig. C.4.



NumberOfMolecules 10

WindowTime (seconds) 0.01

Tolerance 0.01

StepSize 0.0005

Browse (Audio WAV) \\Music\audio.wav

**Fig. C.4** Front panel of the simple adaptive AHN-filter

## C.2 Program Codes for Position Control of DC Motors

The position control of the direct current (DC) motor designed in Chap. 7 requires three steps: fuzzification, fuzzy inference engine and defuzzification. Since the first two steps can be any fuzzification and fuzzy inference engine procedures detailed in proper literature, this appendix only shows the defuzzification program code.

The defuzzification step of the fuzzy-molecular inference (FMI) controller written in LabVIEW$^{TM}$ is shown in the block diagram of Fig. C.5. In that case, the fuzzy membership values of antecedents in rules (**FuzzyEvaluationValues**) are collected in an array. This array enters to a **For Loop** in order to evaluate the consequent values. The evaluation is done via an artificial hydrocarbon compound implemented in the **Compound** VI which receives the **Molecular Parameters** (i.e. type of molecule, the number of molecules, order of bonding and hydrogen values), the **Compound Parameters** (i.e. the type of behavior, bounds and optional properties of covalent bonds) and the fuzzy membership values of the antecedents.

Notice that consequent values are multiplied by the fuzzy membership values of antecedents. Then, these results are summed as well as the fuzzy membership values of antecedents in order to compute the **Crisp Output Value**. It is remarkable to say that the VI shown in Fig. C.5 can be used directly for type-1 fuzzy systems, and reproducing it another time can be used for type-2 fuzzy systems.

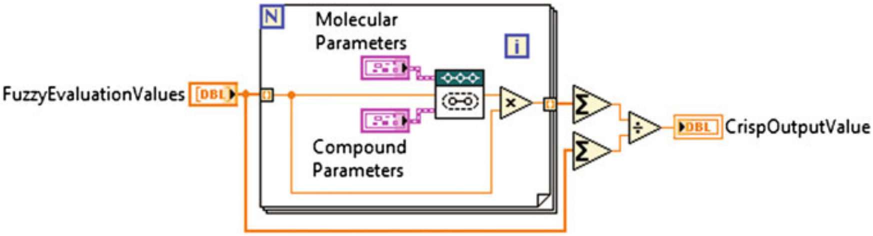


**Fig. C.5** Block diagram of the defuzzification step in the FMI controller

## C.3 Program Codes for Facial Recognition

The facial recognition system using artificial hydrocarbon networks was implemented in MATLAB®. The main script for training an artificial hydrocarbon network and for extracting the feature vector made of bounds of molecules, is shown following:

```
clear all;
close all;
warning off all;

% Lower and upper bounds of input domain
a = min(X);
b = max(X);

% Uniformly initialize intermolecular distances
over the input domain
r = a:(b-a)/(n-2):b;

% Starts the gradient descent method for length bonds
t = 1;

while (t < = 300)

    % Calculate bounds of molecules
    Intervals = zeros(1,n);
    Intervals(1) = r(1) + a;
    SizeR = size(r,2);
    for k = 2:SizeR
        temp = r(k) + Intervals(k-1);
        if (temp <= b)
          Intervals(k) = temp;
        else
          Intervals(k) = b;
        end
    end
    Intervals(n) = b;

    % Split the system
    counter = 1;
    SizeX = size(X,2);
    SizeY = size(Y,2);
    s = struct(’X’,[],’Y’,[]);
    for k = 1:n
       i = 1;
       while ((X(counter) < Intervals(k)) && (counter <
        SizeX))
           s(k).X(i) = X(counter);
           s(k).Y(i) = Y(counter);
           i = i + 1;
           counter = counter + 1;
        end
    end
    s(n).X(i) = X(SizeX);
    s(n).Y(i) = Y(SizeY);

    % Create compound using a saturated linear chain
    ord = 2 * ones(n);
    ord(1) = 3;
    ord(end) = 3;

   % Calculate hydrogen values
   AHNApproximation = [];
   EMolecule = zeros(1,n);
   HTemp = struct(’H’,[]);
   HydrogenValues = struct(’H’,[],’A’,[]);
   for k=1:n
       HTemp(k).H = polyfit(s(k).X,s(k).Y,ord(k));
       HydrogenValues(k).H = roots(HTemp(k).H);
       HydrogenValues(k).A = HTemp(k).H(1);
       f = polyval(HTemp(k).H,s(k).X);
       EMolecule(k) = sum((s(k).Y - f).^2);
       s(k).Y = f;
      AHNApproximation = [AHNApproximation f];
   end

   % Update all intermolecular distances
   for k = 1:SizeR
      r(k) = r(k) - StepSize * (Em(k) - Em(k+1));
   end
   t = t + 1;

end
Intervals = [a Intervals];
```

As shown above, the inputs of the script are: an array of the numbers of input vectors coming from the discrete cosine transform (DCT) rastered in zigzag denoted as `X`, an array of the feature vector coming from the DCT rastered in zigzag denoted as `Y`, the number of molecules in the artificial hydrocarbon compound denoted as `n`, and the step size denoted as `StepSize`. The resultant feature vector using artificial hydrocarbon networks is obtained from the `Intervals` variable.

## C.4 Quick Guide for Implementing Artificial Hydrocarbon Networks

This section presents a guick guide for using artificial hydrocarbon networks to model systems.

Observations to chemical organic compounds reveal enough information to derive the artificial organic networks technique. From studies of organic chemistry, organic compounds are the most stable ones in nature. In addition, molecules can be seen as units of packaging information; thus, complex molecules and its combinations can determine a nonlinear interaction of information. Moreover, molecules can be used for encapsulation and potential inheritance of information. Thus artificial organic networks take advantage of this knowledge, inspiring the artificial hydrocarbon networks algorithm that infer and classify information based on stability and chemical rules that allow formation of molecules.

In that sense, artificial organic networks define four components, i.e. atoms, molecules, compounds and mixtures; and two basic interactions among components, i.e. covalent bonds and chemical balance interaction. In order to obey chemical rules, the following definitions of artificial hydrocarbon networks hold:

- *Atoms*. They are the basic units with structure. No information is stored in them. In addition, when two atoms have the same number of degrees of freedom they are called *similar atoms*, and *different atoms* otherwise. The degree of freedom is the number of valence electrons that allow atoms to be linked with others. Only hydrogen atoms and carbon atoms can be found.
- *Molecules*. They are the interactions of two or more atoms made of covalent bonds. These components have structural and behavioral properties. Structurally, they conform the basis of an organized structure while behaviorally they can contain information. Thus, molecules are known as the basic units of information. If a molecule has filled out all of the valence electrons in atoms, it is stable; but if a molecule has at least one valence electron without filling, it is considered as unstable. Two behaviors of molecular units can be found in artificial hydrocarbon networks: the *sum form* (5.10)–(5.11) and the *product form* (5.18)–(5.19).
- *Compounds*. Structurally, they are two or more molecules interacting each other linked with covalent bonds. Their behaviors are mappings from the set of molecular behaviors to real values.
- *Mixtures*. They are the interaction of two or more molecules and/or compounds without physical bonds. Mixtures are linear combinations of molecules and/or compounds like (5.28) forming a basis of molecules with weights so-called stoichiometric coefficients.
- *Covalent bonds*. They are of two types. Polar covalent bonds refer to the interaction of two similar atoms while nonpolar covalent bonds refer to the interaction of two different atoms. Three models of covalent bond behaviors can be identified: the *weighted sum interaction* (4.5)–(4.7), the *exponential interaction* (4.8) and the *piecewise interaction* (4.16). For implementation purposes, the latter is the easiest covalent bond behavior.
- *Chemical balance interaction*. It refers to find the proper values of stoichiometric coefficients in mixtures in order to satisfy constrains in artificial hydrocarbon networks.

On the other hand, artificial hydrocarbon networks algorithm is divided in two steps: the training process and the inference process (refer to Sect. 3.4.3 for further details). Following, both processes are summarized in order to implement them easily.

### *C.4.1 Training Process*

This process will build and train an artificial hydrocarbon network $AHN$ for modeling a given system $\Sigma$. In order to do this, compute the following steps:

1. Collect $q$ samples of attribute $\mathbf{X}$ and target $\mathbf{Y}$ variables from the system and represent it as a pair of variables $\Sigma = (\mathbf{X}, \mathbf{Y})$.
2. Determine the maximum number of compounds $c_{max} > 0$ that the artificial hydrocarbon network $AHN$ can have.
3. Determine the maximum number of molecules $k_{max} \geq 2$ each compound can have.
4. Fix the learning rate value $0 < \eta < 1$ that is used for training the artificial hydrocarbon network $AHN$. Small values of $\eta$ work well.
5. Determine the tolerance value $\varepsilon > 0$ (error between the system and the model) expected in the artificial hydrocarbon network $AHN$.
6. Use Algorithm 5.3 for training multidimensional artificial hydrocarbon networks.

At last, Algorithm 5.3 will return the artificial hydrocarbon network $AHN$ in terms of its structure $\Gamma$ (the total number of compounds used, the total number of molecules per compound used and the links between molecules), and all parameters required in that structure (the set of hydrogen values $H$, the stoichiometric coefficients $\Lambda$ and the intermolecular distances $\Pi$).

In order to implement Algorithm 5.3, consider the flow diagram shown in Fig. C.6. As noted, it consists of two main procedures: generate a set of compounds (build and train a mixture of compounds) and mix them up to find optimal stoichiometric coefficients.

For instance, consider Algorithm 5.3. In order to generate the set of compounds, it is neccesary to generate one compound at a time (see Fig. C.6).

To generate a compound, it is required to initialize the number of the current compound $i$ and to set the minimal number of molecules $k = 2$ in the compound. Once it is done, the CREATE-COMPOUND $(R, k)$ procedure (Algorithm 4.1) is called to build the structure of the compound; where, $R$ stands for the residual system (initially set to be the system $\Sigma$) and $k$ is the number of molecules in the compound. In particular, this procedure has to split the system $\Sigma = (X, Y)$ in equal partitions. Thus, it requires to initialize the set of bounds of molecules $L$ using (4.24) and (4.25). Three different splitting processes were presented through the book. For example, to split multivariate systems, it can be applied (6.16) to use a hyperplane criterion or (6.19) to use a centers-of-molecules criterion. If the system is univariate, it can be applied (4.26). After that, it is required to measure the energy of partitions using (5.33) and compare this energy with Table 4.2.

Then, that compound is optimized and trained using the OPTIMUM-COMPOUND $(R, C_i, k, \eta)$ procedure (Algorithm 4.4); where $C_i$ stands for the current compound in the set of all compounds. This procedure initializes the set of intermolecular distances $\Pi_i$ and recalculates the set of bounds of molecules $L$ using (4.25). Then, it has to compute the least squares estimates using (5.35) and finally it has to update intermolecular distances using (5.36) and (5.37).

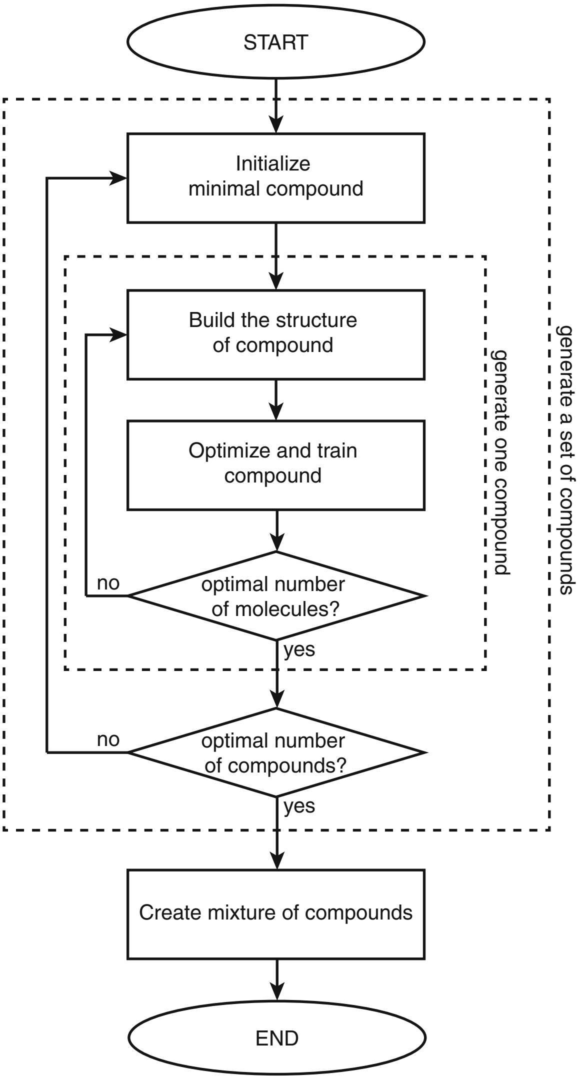


**Fig. C.6** Flow diagram of the artificial hydrocarbon networks algorithm

Finally, it is required to determine optimality in the number of molecules that conform the compound. In that sense, the ENTHALPY-RULE $(R, C_i, k)$ procedure (Algorithm 5.1) is called. It returns the updated number of molecules $k$ and the heuristic value $m$. This finishes the process of generating a compound.

To determine if the number of compounds is useful to reach a tolerance value $\epsilon$ between the system and the model, (4.15) is computed and $R$ is updated. This finishes the process of generating a set of compounds in the artificial hydrocarbon network.

The last step of Algorithm 5.3 mixes the set of compounds using the least squares estimates (4.19) in order to obtain a set of stoichiometric coefficients $\Lambda$ which it terminates the algorithm.

### C.4.2 Inference Process

This process can be used for making a query **x** to the artificial hydrocarbon network *AHN* and obtaining a value response **y** (prediction). In order to do this, compute the following steps:

1. Build the structure of the *AHN* using $\Gamma$, $H$, $\Lambda$ and $\Pi$.
2. Feed *AHN* with a query **x**.
3. Calculate the value response **y**.
4. Repeat steps 2 and 3 until all queries are respond.

# Appendix D
# Artificial Organic Networks Toolkit Using LabVIEW$^{\mathrm{TM}}$

This appendix presents a quick starting guide for designing applications with the Artificial Organic Networks Toolkit using LabVIEW$^{\mathrm{TM}}$. First, an installation guide is presented. Then, the functions palette is described. Finally, a complete list of function blocks and their descriptions is summarized.

## D.1 Installation Guide

Refer to the online reference of the book in order to download the release version of the Artificial Organic Networks Toolkit using LabVIEW$^{\mathrm{TM}}$. Then, follow these instructions in order to install it:

1. Close the LabVIEW$^{\mathrm{TM}}$ software if running.
2. Unzip the **AONToolkit_Package.zip** file.
3. Locate the **AON_Toolkit** folder and the **dir.mnu** file and put them inside the **user.lib** folder following the path: **Program Files** ≫ **National Instruments** ≫ **LabVIEW** ≫ **user.lib**. It is remarkable to say that the **dir.mnu** file already present in the system will be replaced.
4. Open the LabVIEW$^{\mathrm{TM}}$ software and verify that there is the Artificial Organic Networks Toolkit (AON) under the **User Libraries** in the functions palette.

## D.2 Functions Palette

When the Artificial Organic Networks Toolkit is installed, it appears under the User Libraries in the functions palette. Figure D.1 shows the whole toolkit which it is classified into four categories: *Components*, *Interactions*, *Chemical Rules* and *Training*.

H. Ponce-Espinosa et al., *Artificial Organic Networks*,
Studies in Computational Intelligence 521, DOI: 10.1007/978-3-319-02472-1,

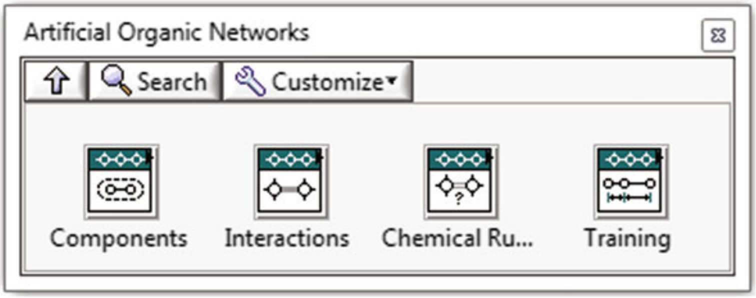


**Fig. D.1** Artificial Organic Networks Toolkit

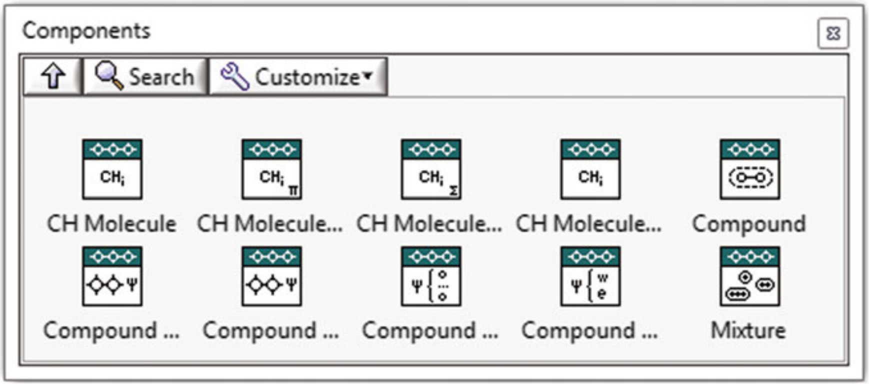


**Fig. D.2** *Components* palette of the Artificial Organic Networks Toolkit

The *Components* palette (Fig. D.2) contains function blocks to implement the basic components of the artificial hydrocarbon networks like molecules, compounds and mixtures.

The *Interactions* palette (Fig. D.3) contains function blocks to implement covalent bonds between components of artificial hydrocarbon networks. In fact, these virtual instruments (VI) functions implement three models of covalent bonding behaviors.

The *Chemical Rules* palette (Fig. D.4) contains function blocks to implement rules to create compounds, to split systems, to calculate intermolecular distances, and to measure energy in components and systems.

Finally, the *Training* palette (Fig. D.5) contains function blocks to implement training algorithms of articial hydrocarbon networks. In particular, both the simple-AHN and the modified-AHN algorithms are programmed.

Refer to the next section to obtain general information of each function block in the toolkit.

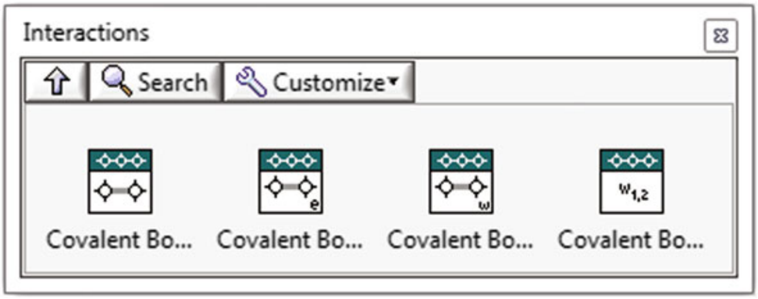


**Fig. D.3** *Interactions* palette of the Artificial Organic Networks Toolkit

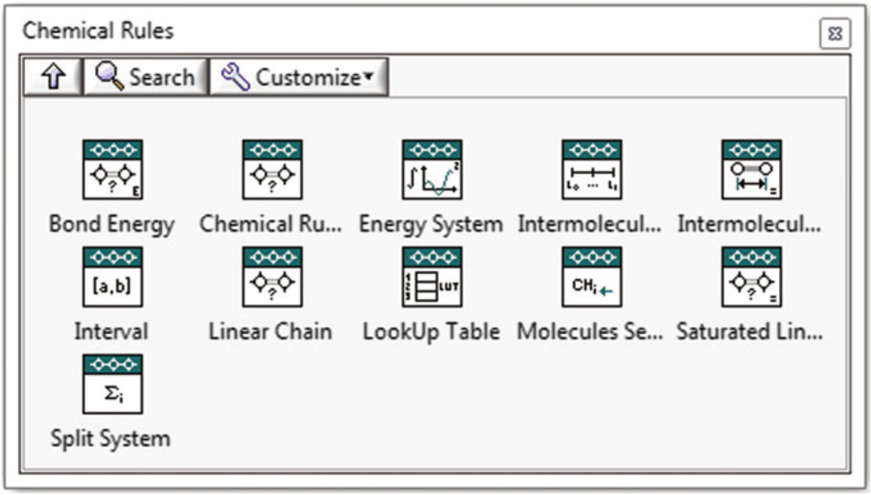


**Fig. D.4** *Chemical Rules* palette of the Artificial Organic Networks Toolkit

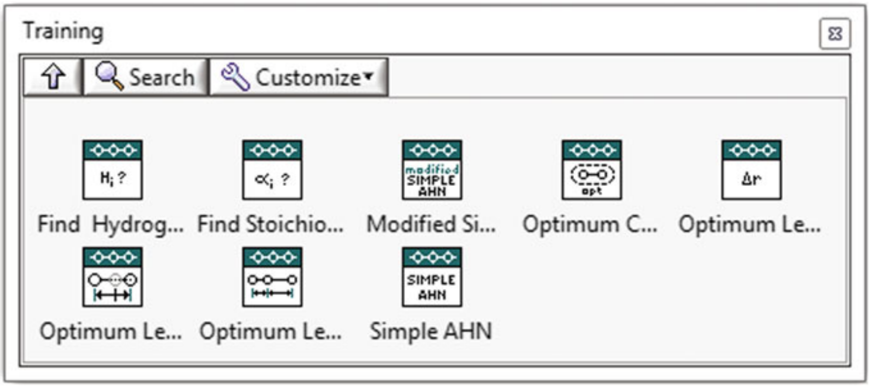


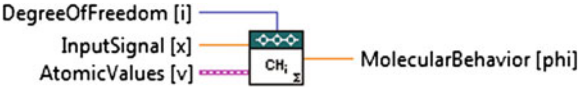
**Fig. D.5** *Training* palette of the Artificial Organic Networks Toolkit

## D.3 Function Blocks and Descriptions

This section summarizes the function blocks of the Artificial Organic Networks Toolkit using LabVIEW$^{TM}$.

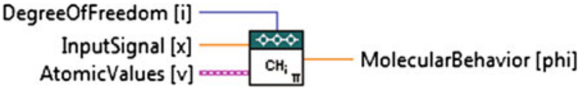
### *D.3.1 Functions of Components Palette*

**CH Molecule Sum:**



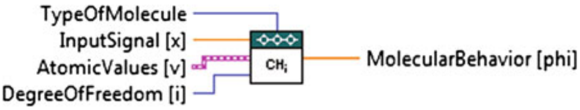
It computes the **Molecular Behavior** of a *CH*-primitive molecule, in its sum form, due to an **Input Signal**. In that case, **Atomic Values** represents a cluster of both **Hydrogen Values** of size equals to the order of bond (**Degree Of Freedom**) and the carbon value **(Offset**) of the molecule.

**CH Molecule Product:**



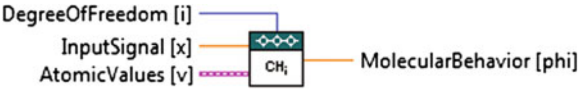
It computes the **Molecular Behavior** of a *CH*-primitive molecule, in its product form, due to an **Input Signal**. In that case, **Atomic Values** represents a cluster of both **Hydrogen Values** of size equals to the order of bond (**Degree Of Freedom**) and the carbon value **(Offset**) of the molecule.

**CH Molecule Selector:**



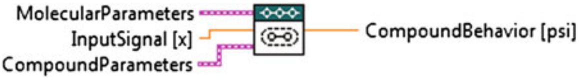
This VI computes the **Molecular Behavior** of a *CH*-primitive molecule due to an **Input Signal** using a specific form determined manually in the **Type Of Molecule** (i.e. *Product Form*, or *Sum Form*). Again, **Atomic Values** represents a cluster of both **Hydrogen Values** of size equals to the order of bond (**Degree Of Freedom**) and the carbon value **(Offset**) of the molecule.

**CH Molecule (Polymorphic):**



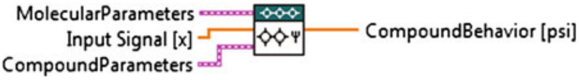
This is a polymorphic VI that computes the **Molecular Behavior** of a *CH*-primitive molecule due to an **Input Signal** depending on the **Atomic Values** input (*Product Form* if **Hydrogen Values** are in complex representation and *Sum Form* if they are in double representation). In that case, **Atomic Values** represents a cluster of both **Hydrogen Values** of size equals to the order of bond (**Degree Of Freedom**) and the carbon value **(Offset**) of the molecule.

**Compound:**



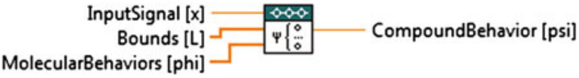
It computes the **Compound Behavior** due to an **Input Signal**. It requires the set of **Molecular Parameters** (**Type of Molecule**, **Number of Molecules**, **Degrees Of Freedom** and **Hydrogen Values**) and the **Compound Parameters** (**Type Of Compound Behavior**, **Bounds** and optional **Properties** of the *Weighted Sum* behavior).

**Compound N Points:**



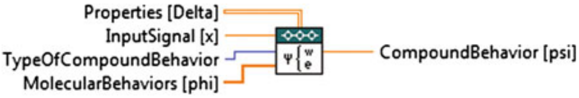
It computes an array of **Compound Behaviors** due to an array of values in the **Input Signal**. It requires the set of **Molecular Parameters** (**Type of Molecule**, **Number of Molecules**, **Degrees Of Freedom** and **Hydrogen Values**) and the **Compound Parameters** (**Type Of Compound Behavior**, **Bounds** and optional **Properties** of the *Weighted Sum* behavior).

**Compound Piecewise Behavior:**



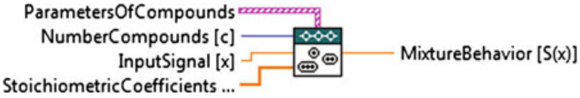
It computes the **Compound Behavior**, using the piecewise behavior, due to an **Input Signal**. It requires the **Bounds** parameters to define the piecewise molecular actions and an array of **Molecular Behaviors**.

**Compound Weighted Or Exponential Behavior:**



It computes the **Compound Behavior**, using the weighted or exponential behavior, due to an **Input Signal**. It requires to set manually the **Type Of Compound Behavior** (*Weighted Sum* or *Exponential*) and the **Properties** of the *Weighted Sum* if necessary. Also, it requires an array of **Molecular Behaviors**.

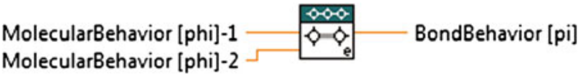
**Mixture:**



It computes the **Mixture Behavior** of an artificial hydrocarbon network due to an **Input Signal**. It requires the **Parameters Of Compounds** (**Molecular Parameters** and **Compound Parameters** as in the **CH Compound** VI), the **Number of Compounds** and an array of **Stoichiometric Coefficients**.

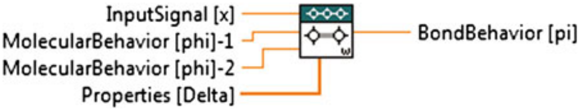
## *D.3.2 Functions of Interactions Palette*

**Covalent Bonds Exponential:**



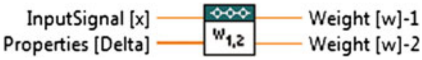
It computes the covalent **Bond Behavior** between two **Molecular Behaviors** using the exponential interaction.

**Covalent Bonds Weighted Sum:**



It computes the covalent **Bond Behavior** between two **Molecular Behaviors** using the weighted sum interaction. It requires the **Input Signal** and the **Properties** of the bond.

**Covalent Bonds Weighted Sum Weights:**



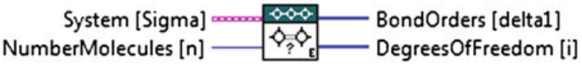
This VI calculates the **Weights** of two molecules. It requires an **Input Signal** and the **Properties** of bond interactions.

**Covalent Bonds (Polymorphic):**

This is a polymorphic VI that computes the covalent **Bond Behavior** between two **Molecular Behaviors**. The polymorphic VI can be selected from the *Select Type* item of the contextual menu of this VI. The *Weighted Sum* selection requires an **Input Signal** and the **Properties** of bond interactions.

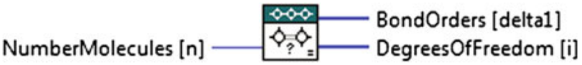
### *D.3.3 Functions of Chemical Rules Palette*

**Bond Energy:**



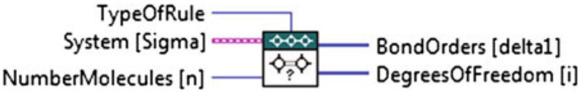
It creates a linear chain of *CH*-primitive molecules using the bond energy criterion. This VI requires the **System** to model and the **Number Of Molecules** in the compound, and returns the order of bonds (**Bond Orders**) between the **Number Of Molecules** and the **Degrees Of Freedom** of each molecule.

**Saturated Linear Chain:**



It creates a saturated linear chain of *CH*-primitive molecules. This VI requires the **Number Of Molecules** in the compound, and returns the order of bonds (**Bond Orders**) between the **Number Of Molecules** and the **Degrees Of Freedom** of each molecule.

**Chemical Rule Selector:**

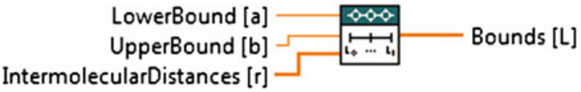


It creates a linear chain of *CH*-primitive molecules to form a compound using one of the **Type Of Rules** (*Simple Bond* or *Bond Energy*). It requires the **Number Of Molecules** in the compound and the **System** to model, if required. This VI returns the order of bonds (**Bond Orders**) between the **Number Of Molecules** and the **Degrees Of Freedom** for each molecule. The *Simple Bond* type returns a saturated linear chain, and the *Bond Energy* type returns a linear chain based on the bond energy criterion.

**Linear Chain (Polymorphic):**

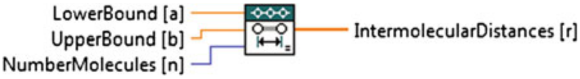
This is a polymorphic VI that creates a linear chain of CH-primitive molecules to form a compound: 1. Saturated Linear Chain VI. 2. Bond Energy Criterion VI.

**Intermolecular Distances Bounds:**



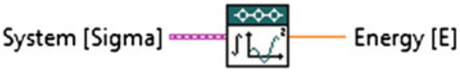
It determines the **Bounds** of action of *CH*-primitive molecules in a compound. It requires the interval of the input domain [**Lower Bound**, **Upper Bound**] and the **Intermolecular Distances** that represent the length of bonds between molecules.

**Intermolecular Distances Equal:**



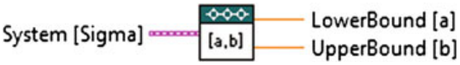
This VI determines a set of **Intermolecular Distances** between a **Number Of Molecules** with equal lengths distributed in the interval [**Lower Bound**, **Upper Bound**].

**Energy System:**



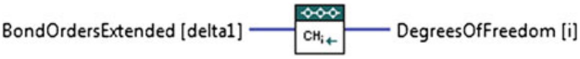
It calculates the **Energy** of a **System** using the energy signal theory.

**Interval:**



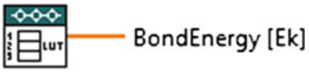
It determines the **Lower Bound** and the **Upper Bound** in the input domain of the **System**.

**Molecules Selector:**



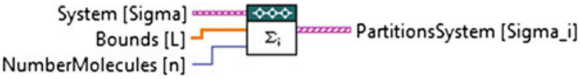
It determines the **Degrees Of Freedom** for each molecule in a compound using the order of bonds (**Bond Orders Extended**) between molecules.

**Look Up Table:**



This is the look-up-table that helps to find the best order of bond (**Bond Energy**) between two *CH*-primitive molecules.

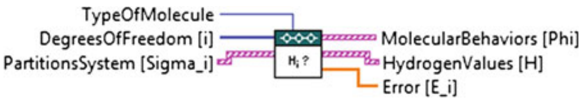
**Split System:**



This VI splits the **System**, using the set of **Bounds** for each partition, in a **Number Of Molecules** specified. It returns the set of partitions of the system (**Partitions System**).

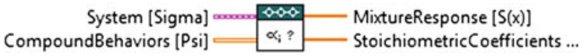
### D.3.4 Functions of Training Palette

**Find Hydrogen Values:**



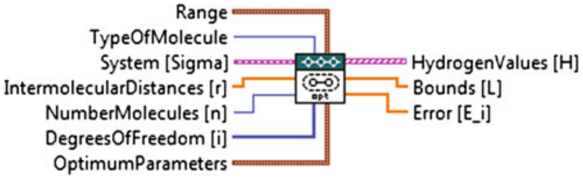
This VI determines the best **Hydrogen Values** of each *CH*-primitive molecule in a compound. It requires the **Degrees Of Freedom** of each molecule, the **Type Of Molecules** used and the partitions of the system (**Partitions System**) to model. It also returns the **Error** between the **Molecular Behaviors** and the partitions of the system (**Partitions System**), and the **Molecular Behaviors**.

**Find Stoichiometric Coefficients:**



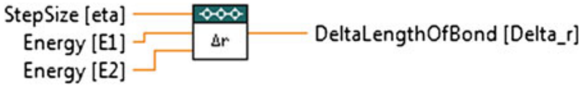
It calculates the **Stoichiometric Coefficients** and the **Mixture Response** of the artificial hydrocarbon network, using the **System** to model and the **Compound Behaviors**.

**Optimum Compound:**



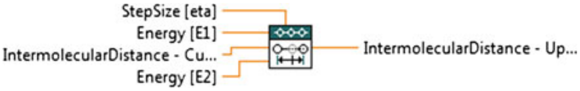
This VI builds an artificial hydrocarbon compound using the Optimum Compound Algorithm. It requires the **Range** of the input domain, the **Type Of Molecule**, the **System** to model, the **Intermolecular Distances**, the **Number Of Molecules**, the **Degrees Of Freedom** and the **Optimum Parameters** (**Step Size** and **Max Iterations**).

**Optimum Lengths Delta Rule:**



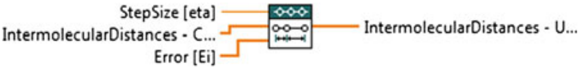
It calculates the change of intermolecular distances (**Delta Length Of Bond**) using the error functions **Energy 1** and **Energy 2** with a real number **Step Size** between 0 and 1.

**Optimum Lengths Updated Lengths:**


StepSize [eta]
Energy [E1]
IntermolecularDistance - Cu...
Energy [E2]
IntermolecularDistance - Up...

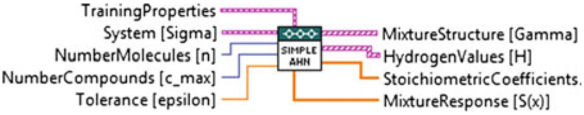
It calculates the **Updated Intermolecular Distance** between two molecules using the **Current Intermolecular Distance**, and the error functions **Energy 1** and **Energy 2** with a real number **Step Size**.

**Optimum Lengths Updated Lengths N Points:**


StepSize [eta]
IntermolecularDistances - C...
Error [Ei]
IntermolecularDistances - U...

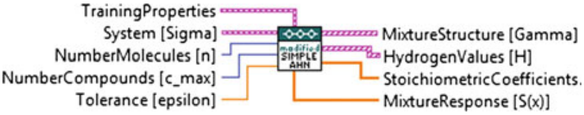
It calculates the **Updated Intermolecular Distances** of $N$-molecules using the **Current Intermolecular Distance**, the **Error** functions and the **Step Size**.

**Simple AHN:**



This VI trains an artificial hydrocarbon network using the Simple AHN Algorithm. It requires the **Training Properties**, the **System** to model, the **Number Of Molecules**, the **Number of Compounds** and the **Tolerance** value. It outputs the **Mixture Structure**, the **Hydrogen Values**, the **Stoichiometric Coefficients** and the **Mixture Response**.

**Modified Simple AHN:**



This VI trains an artificial hydrocarbon network using the modified version of the Simple AHN Algorithm. It requires the **Training Properties**, the **System** to model, the **Number Of Molecules**, the **Number of Compounds** and the **Tolerance** value. It outputs the **Mixture Structure**, the **Hydrogen Values**, the **Stoichiometric Coefficients** and the **Mixture Response**.

# Appendix E
# Examples of Artificial Hydrocarbon Networks in LabVIEW$^{\text{TM}}$

The Artificial Organic Networks Toolkit using LabVIEW$^{\text{TM}}$ is available for designing, training and implementing artificial hydrocarbon networks (AHNs). Following, a series of examples show the way how components of AHNs can be used. Then, training virtual instruments (VI) nodes are described.

## E.1 Using Components

*Example E.1* Consider a *CH*-primitive molecules of with three hydrogen atoms $h_1 = 0.2$, $h_2 = 0.6$, $h_3 = 0.4$ and a carbon value $v_C = 1$. Build a LabVIEW$^{\text{TM}}$ application for drawing the behavior of that molecule in the input domain $x \in [-1, 1]$. In particular:

1. Use the first model of *CH*-molecules (sum form) of Proposition 4.1.
2. Use the second model of *CH*-molecules (product form) of Proposition 4.2.

**Solution E.1** Create a new blank VI and select the **CHMoleculeSelector** function from the following path: **AHN** $\gg$ **Components**. This VI allows to select the type of *CH* model to use (**TypeOfMolecule**), the degree of freedom (**DegreeOfFreedom**), the hydrogen values and carbon value (**AtomicValues**) and the input (**InputSignal**). Then, it performs the behavior of molecule (**MolecularBehavior**). In this case, select *Sum Form* for **TypeOfMolecule**, 3 for **DegreeOfFreedom**, and the hydrogen values $h_1, h_2, h_3$ and carbon value $v_C$ for **AtomicValues**.

However, this VI only calculates the behavior of molecule from one data point of the input domain. Then, place it inside a **For Loop** and create an array of values from $x = -1$ to $x = 1$, considering a step value between elements, e.g. 0.01. Figure E.1 shows the global block diagram of this example. As noted, a graph (**XY Graph**) was connected to the output terminal of the **CHMoleculeSelector** VI. Figure E.2 shows the front panel of the this application.

H. Ponce-Espinosa et al., *Artificial Organic Networks*,
Studies in Computational Intelligence 521, DOI: 10.1007/978-3-319-02472-1,

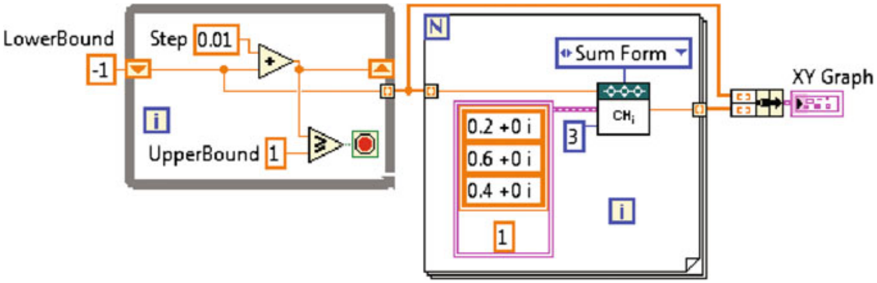


**Fig. E.1** Block diagram of a *CH*-primitive molecule (*sum form*) implementation

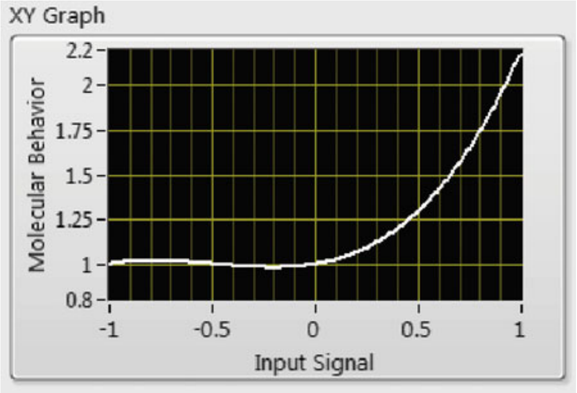


**Fig. E.2** Front panel of Example E.1 (*sum form*)

In addition, the CHMoleculeSelector VI can perform the product form by changing to *Product Form* in **TypeOfMolecule** selector. Figure E.3 shows the block diagram for this implementation and Fig. E.4 shows the front panel. Notice that the molecular behavior depends on the type of the *CH* model selected, as well as the order of hydrogen values (only in the sum form).

***Optional:*** Create a SubVI node for the input signal in order to use it in the following examples.

*Example E.2* Consider an artificial hydrocarbon compound made of two *CH*-primitive molecules $M_1, M_2$ with hydrogen values $h_{11} = 0.3, h_{12} = 0.6, h_{21} = 0.4$ and $h_{22} = 0.7$ and $v_{C1} = v_{C2} = 1$. Both molecules are modeled with the product form. In addition, consider a piecewise interaction between molecules like (E.1). Build a LabVIEW$^{TM}$ application to calculate the value of the compound behavior at $x = 0.45$.

$$\psi(x) = \begin{cases} M_1 & -1 \leq x < 0 \\ M_2 & 0 \leq x \leq 1 \end{cases} \quad \text{(E.1)}$$

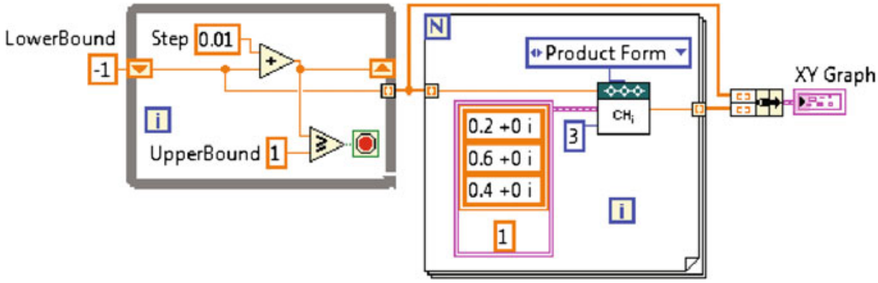


**Fig. E.3** Block diagram of a *CH*-primitive molecule (*product form*) implementation

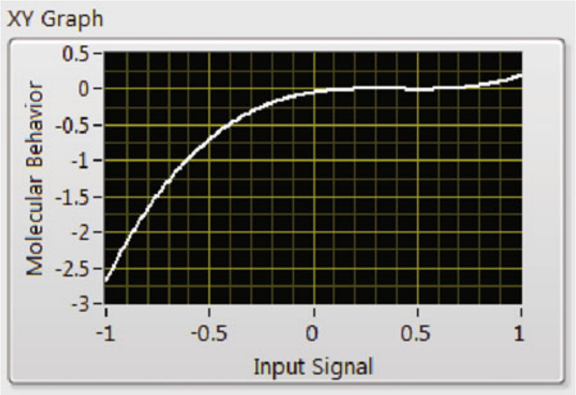


**Fig. E.4** Front panel of Example E.1 (*product form*)

**Solution E.2** Create a new blank VI and select the **Compound** function from the following path: **AHN** ≫ **Components**. This VI requires three inputs. The first input terminal (**MolecularParameters**) is a cluster of the molecular parameters, including: type of *CH* model (*Product Form*), the number of molecules in the compound (2), the degrees of freedom of all molecules in the compound expressed as a vector (2, 2), and an array of atomic parameters (hydrogen parameters and the carbon value). The second input refers to the input value (**InputSignal**) of the compound ($x = 0.45$) and the third input refers to paramaters of the compound (**CompoundParameters**), including: type of compound interaction (*Piecewise* in this case), the bounds of molecules in as a vector $(-1, 0, 1)$ and an optional vector for additional parameters to the *Weighted Sum* interaction. The *Exponential* interaction among molecules is also available. The result of the VI is the compound behavior (**CompoundBehavior**).

Figure E.5 shows the global block diagram of this example. Run the application and verify that the compound behavior at $x = 0.45$ is $\psi(x) = -0.0125$.

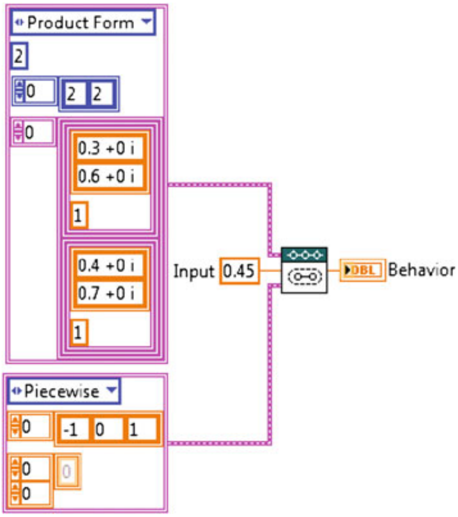


**Fig. E.5** Block diagram of the artificial hydrocarbon compound implementation

**Table E.1** Parameters of the mixture in Example E.3

| Compound 1 | | Compound 2 | |
|---|---|---|---|
| Description | Value | Description | Value |
| Type of interaction | Piecewise | Type of interaction | Piecewise |
| Bounds of molecules | $(-1, 0.5, 1)$ | Bounds of molecules | $(-1, 0, 1)$ |
| Number of molecules | 2 | Number of molecules | 2 |
| *Molecule 1* | | | |
| Type of *CH* model | Product form | Type of *CH* model | Product form |
| Hydrogen values | $(0.6, 0.7)$ | Hydrogen values | $(0.4, 0.9)$ |
| Carbon value | 1 | Carbon value | 1 |
| *Molecule 2* | | | |
| Type of *CH* model | Product form | Type of *CH* model | Product form |
| Hydrogen values | $(0.3, 0.8)$ | Hydrogen values | $(0.1, 0.6)$ |
| Carbon value | 1 | Carbon value | 1 |

*Example E.3* Consider a mixture of two compounds with parameters shown in Table E.1. Also, consider a set of stoichiometric coefficients $\Lambda = (0.9, 0.4)$. Build a LabVIEW$^{TM}$ application to calculate the value of the mixture at $x = 0.8$.

**Solution E.3** Create a new blank VI and select the **Mixture** function from the following path: **AHN** ≫ **Components**. This VI requires four inputs. The first input terminal (**ParametersOfCompounds**) is an array of two clusters representing the parameters of molecules of each compound (i.e. **MolecularParameters** and **CompoundParameters** like in Example E.1). The second input is the number of compounds (**NumberCompounds**) in the mixture. The third input refers to the input

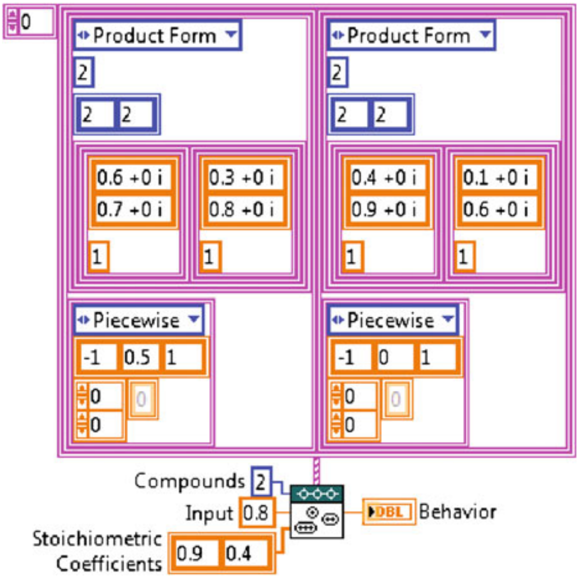


**Fig. E.6** Block diagram of the mixture implementation

value (**InputSignal**) of the compound ($x = 0.8$). Finally, the fourth input is a vector of sotichiometric coefficients (**StoichiometricCoefficients**).

Figure E.6 shows the global block diagram of this example. Run the application and verify that the behavior of the mixture at $x = 0.8$ is $S(x) = 0.056$.

*Example E.4* Consider the same mixture of Example E.3. Improve the LabVIEW$^{\text{TM}}$ application for drawing the behavior of the mixture in the input domain $x \in [-1, 1]$.

**Solution E.4** Open the application of Example E.3 and place it inside a **For Loop**. In order to reduce the size of the **ParametersOfCompound** array, right-click in the outer cluster inside the array and select the **View Cluster As Icon** option in the context menu. Then, create an array of values from $x = -1$ to $x = 1$ with a step of 0.01, or place the input signal VI created in Example E.1. Replace the constant input signal (**InputSignal**) with the latter array of inputs, and relink the output of the **Mixture** VI to a graph control (**XY Graph**). Figure E.7 shows the overall block diagram and Fig. E.8 shows the front panel with the behavior of mixture.

## E.2 Training Artificial Hydrocarbon Networks

*Example E.5* Consider the function $f$ as written in (E.2). Build a LabVIEW$^{\text{TM}}$ to train an artificial hydrocarbon network for this function in the input domain $x \in [0, 5]$. Use 5 *CH*-primitive molecules in each compound, a maximum number of compounds

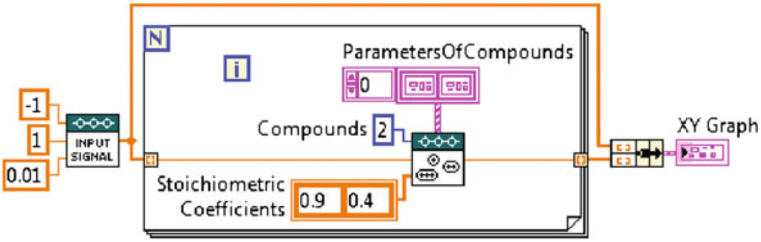


Fig. E.7 Block diagram of the mixture implementation

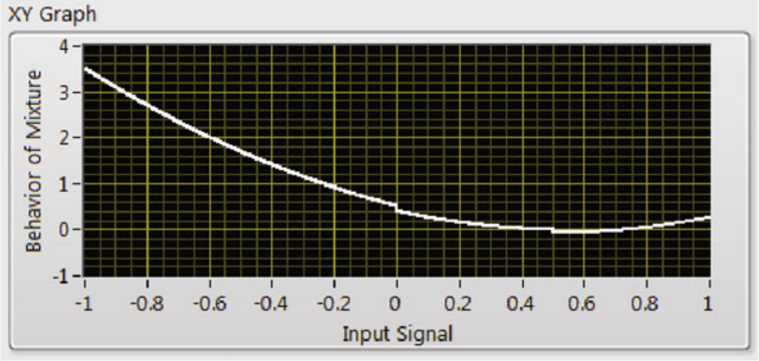


Fig. E.8 Front panel of Example E.4

$c_{max} = 2$, a step size of $\eta = 0.05$, a tolerance value of $\varepsilon = 0.001$ and a maximum number of iterations $t_{max} = 100$.

$$f(x) = \sin(\pi x) \tag{E.2}$$

**Solution E.5** Create a new blank VI and select the **ModifiedSimpleAHN** function from the following path: **AHN** ≫ **Training**. This VI implements Algorithm 4.5 for training an artificial hydrocarbon network. Five inputs are required. The first input corresponds to some parameters (**TrainingProperties**) that determine how linear chains of molecules are created (*Simple Bond Based* or *Bond Energy Based*), the type of *CH* molecules in compounds (*Sum Form* or *Product Form*), the type of interactions between molecules (*Piecewise*, *Weighted Sum* or *Exponential*), optional properties for the *Weighted Sum* interactions, and a cluster for the step size and the maximum number of iterations. In this case, *Simple Bond Based* algorithm, *Product Form* type, *Piecewise* type, and $\eta = 0.05$ and $t_{max} = 100$ values were used.

The second input requires a cluster (**System**) with two arrays containing the input signal and the output signal, respectively. In this example, an array of inputs was implemented (using the VI created in Example E.1) with lower bound 0 and upper bound 5 and a step of 0.01. Function (E.2) was implemented with the **Sine** node

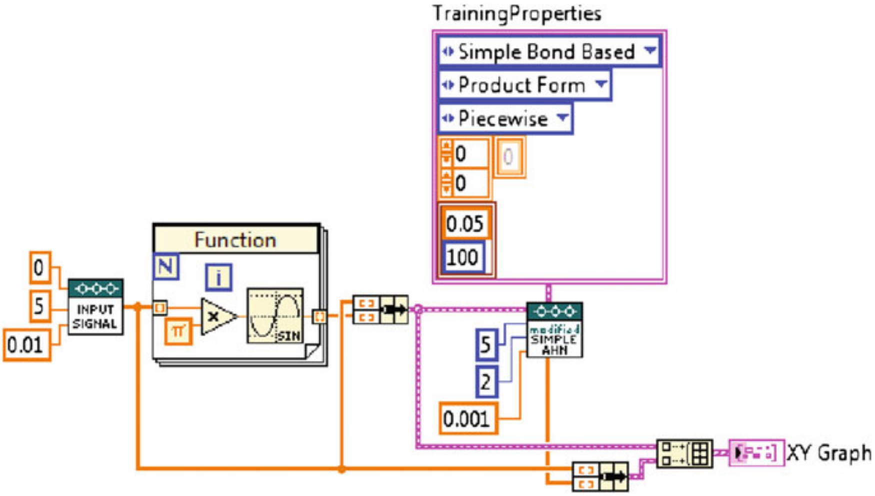


**Fig. E.9** Block diagram for training an artificial hydrocarbon network

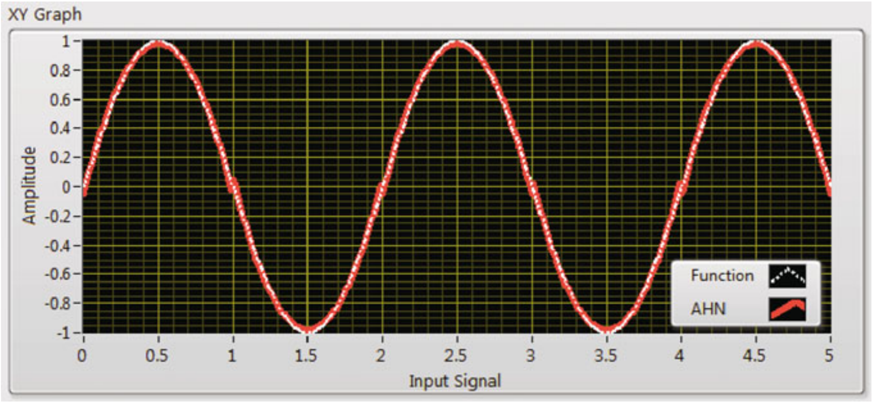


**Fig. E.10** Response of the trained artificial hydrocarbon network of Example E.5

inside a **For Loop**, as depicted in Fig. E.9. Both arrays are clustered with the **Bundle** node.

The third input refers to the number of molecules (**NumberMolecules**) in the AHN-structure, while the next input refers to the maximum number of compounds in the mixture (**NumberCompounds**), and the fifth input is the tolerance value (**Tolerance**). For this example, the corresponding values were 5, 2 and 0.001, respectively.

Finally, the response of the trained artificial hydrocarbon network was plotted in a graph (**XY Graph**). In order to graph the original function and the response (i.e. the output terminal **MixtureResponse**), the latter is clustered with the input signal using the **Bundle** node. To this end, the original function and the response clusters are used for creating a 2D-array with the **Build Array** node. Figure E.9 shows the overall block diagram of the application and Fig. E.10 shows the response of the trained AHN-structure in comparison with the original function.

If the AHN-structure and all trained parameters are required for a reasoning process (predicting, forecasting, etc.), the **ModifiedSimpleAHN** function has other three outputs: the structure of compounds in the mixture (**MixtureStructure**), the hydrogen values with carbon values (**HydrogenValues**) and the set of stoichiometric coefficients (**StoichiometricCoefficients**).