EChem++–An Object-Oriented Problem Solving Environment for Electrochemistry. 3. Classification of Voltammetric Signals by the Fuzzy ARTMAP Neural Network with Respect to Reaction Mechanisms[†]

Elena P. Sapozhnikova [a,1,*], *Martin Bogdan* [a], *Bernd Speiser* [b,*], *Wolfgang Rosenstiel* [a]

[a] Technische Informatik, Universität Tübingen,

Sand 13, D-72076 Tübingen, Germany

[b] Institut für Organische Chemie, Universität Tübingen,

Auf der Morgenstelle 18, D-72076 Tübingen, Germany

[†] Part 2: see ref. 1.

[1] The results presented in this paper were obtained during the postdoctoral research of E. Sapozhnikova within the Graduiertenkolleg "Chemie in Interphasen".

[*] corresponding authors: +49 7071/2978998, sapozhni@informatik.uni-tuebingen.de (E. Sapozhnikova); +49 7071/2976205, bernd.speiser@uni-tuebingen.de (B. Speiser).

Abstract

A method using the artificial neural network Fuzzy ARTMAP (FAM) was developed to classify cyclic voltammograms according to underlying reaction mechanisms. Different preprocessing methods for reducing input dimensionality, including Principal Component Analysis (PCA), feature extraction, and Wavelet Transform (WT), were compared. Results obtained for simulated and experimental voltammograms show that FAM can be applied to the classification into E, E_{qr}, EC and $E_{qr}C$ mechanisms successfully. The efficiency of WT for data compression was also confirmed. Experiments demonstrate a significant correspondence between misclassifications and intersections of class distributions for different reaction mechanisms. It was found by analyzing the error distributions of FAM that the most classification errors arise in the overlapping areas of two reaction mechanisms. The relationship of the resulting class distribution to the mechanistic zones of classical zone diagrams is discussed.

Keywords: Artificial Neural Networks, Classification, Fuzzy ARTMAP, Wavelet Transform, Voltammetry, Reaction Mechanism

1. Introduction

Electroanalytical techniques, which are based on the measurement of the current flowing through an electrode or the electrode potential, have found wide application due to their high sensitivity, adequate selectivity as well as the low cost of instrumentation and maintenance [2]. They have become more and more popular as diagnostic tools for analytical purposes [3]. Furthermore, these techniques provide detailed information about

reactions including or coupled to electron transfer steps, thus enabling mechanistic analysis [4].

Recently, laboratory automation and software-aided experimentation has been established as a noticeable trend in modern electrochemistry [5]. Voltammetric methods are in the main focus of automation owing to their wide applicability and the high speed of experiments. There is a trend to develop computer-based automatic systems for voltammetry [6, 7]. Such systems are capable of producing a huge amount of data for subsequent interpretation by electrochemists. To help them extract as much information as possible from measured signals and to facilitate the understanding of experimental results, powerful data analysis tools are required. These tools should be able to recognize relationships in acquired data, which is a supplement for the development of models from fundamental theory. It can be especially important to derive models from real data when there is possibly a lack of theoretical knowledge or a discrepancy between theory and experiments.

Intelligent techniques such as artificial Neural Networks (NNs) [8] meet this requirement and have been successfully applied to the analysis of voltammetric measurements in order to achieve high efficiency. NNs possess several valued properties: nonlinearity of modeling, adaptivity, and generalization. The first property means that they can represent any nonlinear relationship between their inputs and outputs. The second and third properties mean that an NN can learn from examples and is able to generalize from old data to new ones. This makes them very powerful in classification and pattern recognition tasks. The last decades were characterized by a growth in the amount of NN research and publications which proved their applicability to the interpretation of noisy, incomplete, or

inconsistent input data [9]. In recent years, NNs have also been used in electroanalytical chemistry [3, 10].

A widespread use of NNs in electrochemistry is calibration [10-13]: identifying and quantifying electroactive species in mixtures when electrochemical responses are complicated by highly overlapping signals or reactions between components [11]. A simple example of such an application is the estimation of the concentration for one or more analytes of interest from a multiple number of measurements. There are many recent studies of electrochemical calibration with NNs. A much less investigated approach is to determine automatically a reaction mechanism at the electrode on the basis of data analysis. No NN applications to this problem were found in the literature. Some comparable studies were previously performed for expert systems [14-17], and the nearest neighbor classifier [18, 19].

Automation plays an important role in modern industry: the development of systems with minimum functions for an operator seems likely to become ubiquitous in the near future. In this context, the use of NNs can help the chemist automatically classify an electrode reaction into one of the limiting cases for which comprehensive theoretical models have been developed (qualitative data analysis). Such elucidation of an electrode reaction is a typical stage in the analysis of electrochemical data. The importance of the discrimination between the limiting mechanisms can be confirmed by the fact that kinetic zone diagrams have been proposed already several decades ago to visualize the transition from one limiting mechanism to another [20, 21]. Exact identification of the reaction mechanism with respect to the limiting cases is an important goal of electrochemical experimentation and has great practical significance. The knowledge of the reaction

mechanism allows one to find relevant kinetic and thermodynamic reaction parameters in a subsequent step (quantitative data analysis).

The aim of the present research was the automatic classification of voltammetric signals for a limited set of reaction mechanisms with a chemical step coupled to an electron transfer. To solve this task, an intelligent classifying system based on an NN algorithm called Fuzzy ARTMAP (FAM) was designed and implemented in the framework of the EChem++ project [1, 22]. This project provides open source software for support of electrochemical experiments, including instrumentation control, data acquisition, numerical simulation, and data analysis.

The paper is organized as follows: Section 2 discusses the designed classification system as part of the EChem++ software package [23]. Brief descriptions of the algorithms used will also be given in this section. Section 3 is devoted to the experimental design and settings. Finally, Section 4 presents results of classification experiments with simulated and measured voltammograms.

2. Method: design of a classification system

2.1 An NN in the EChem++ framework

The objectives of EChem++ include real time control of an electrochemical experiment, simulation and data analysis. Software implementations for each of these tasks will be provided in separate modules [22]. For example, the module *Experiment* defines all settings which are needed to perform an experiment. Additionally, it stores experimental data. Analogously, the module *Model* consists of procedures necessary for simulation of an electrochemical reaction on the basis of model equations and procedures for storage of simulated data. It also contains some model-specific information which has no counterpart

in *Experiment*, such as, for example, the simulation algorithm. Finally, the module *Analysis* compares experimental and simulated data and estimates system parameters (e.g. rate constants of chemical reactions). In addition to such user output, it also produces feedbacks to the modules *Experiment* and *Model* defined either automatically or by the user. As a part of the data analysis tools, the classification of experimental data with an NN belongs to the module *Analysis*.

After acquiring the experimental data, usually, researchers suggest a reaction mechanism of the system under study. Then a computer simulation is performed for the model proposed, followed by a comparison of simulated and experimental results with the aim of verifying the reaction mechanism.

An NN is integrated into the procedure as shown in Figure 1. Such a use of the NN in the data analysis module of the EChem++ package corresponds to the last stage of a general problem-solving procedure of an electrochemical problem solving environment [7]. Measured data are preprocessed by a feature extraction procedure in order to reduce input dimensionality (i.e. the number of input components M) before supplying them to the NN. Preprocessing is a method to preserve only the most important information in the data, which improves the classification quality. A specific technique used in this work is discussed in Subsection 2.2. After preprocessing, feature vectors are formed and the previously trained NN can classify them into several possible classes according to the underlying reaction mechanisms. Classification results help the chemist (expert) draw conclusions about the investigated electrochemical process and optimize the experimental conditions. She/he can control the experiment settings or change simulation parameters properly if an unexpected discrepancy between simulated and measured signals indicates incorrect experimental conditions, simulation parameters or insufficient modeling of the

real mechanism and system. Of course, the simulation deals with an idealistic situation which neglects experimental errors such as noise, reference electrode shift or inadequate representation of complicating physical processes. Thus, the NN should be able to generalize in order to overcome such effects. Classification with the NN can be considered as a prerequisite to quantitative analysis.

Figure 1. An NN in electrochemical data processing of the EChem++ framework.

An NN is a computational model for information processing that represents a directed graph composed of units (known as neurons) connected by weighted links (weights) [8]. Typically, an NN classifier functions in one of two modes: training or classification. Training is an iterative process of input presentation and subsequent adaptation of network parameters, and possibly, a network structure. In the supervised learning mode which will be used here, a network obtains a class label together with an input vector. Adaptation should adjust the network parameters (typically the weights of connections between neurons) so that if the same input vector would be presented to the trained network, its output would be equal to the correct class label. Therefore the aim of training is to minimize misclassification. The classification stage, in turn, is a decision making process: a network predicts a class label for a previously unseen input vector.

In NN research, data are usually divided into a training and a test set. The training set is used for learning, while the test set is used to check the NN performance. From the viewpoint of statistical learning theory [24], the larger a training set, the more stable modeling with an NN is achieved and more reliable classification results are expected. Unfortunately, sufficiently large training sets are very rare when dealing with chemical

experimental data. In this domain, researchers often use only tens or even fewer training samples instead of hundreds or thousands required [11, 25].

It is worth noting that such small training sets are insufficient for providing satisfactory classification results, especially with high-dimensional data, for two reasons. First, in the case when the dimensionality of data exceeds the size of a training set, the classification task becomes very difficult because the number of observations should generally be greater than the number of explanatory variables [26]. And second, when degrees of freedom for an NN model (approximately the number of weights, or simply speaking tuned parameters) approach the number of training samples, the NN is prone to fitting irrelevant aspects of the training data (overfitting [27]). Since the theory of NN computation has been extensively discussed elsewhere [8, 9], only a brief description of data processing in the special case of FAM will be given in Subsection 2.3.

2.2. Wavelet Analysis

As one of the most powerful preprocessing methods wavelet analysis was used among other techniques for data compression in the present study. Recently, the use of wavelet analysis for signal processing has received intensive attention of researchers, and applications in electroanalytical chemistry have also been proposed [2, 28-31]. They range from de-noising of electrochemical signals to improving curve fitting procedures. In electrochemical data analysis, the Wavelet Transform (WT) has been established as a promising method for data reduction as well [25].

In contrast to the well-known Fourier transform which maps a function from the time domain into a pure frequency domain, WT maps a function from the time domain into a compound time-frequency space. Fourier analysis consists of breaking up a signal into sine

waves of various frequencies. Similarly, wavelet analysis decomposes a signal into basis functions or wavelets $\psi_{j,k}$ defined in the two-dimensional space of scale (a variable related to the frequency, see below) and time. Wavelets are finite, asymmetric, and non-periodic. The fact that wavelets are bounded in time provides wavelet decomposition with advantageous abilities, such as compressing information as well as approximating non-smooth and non-stationary signals. The discrete WT represents a function $f(t)$ as a linear combination of $\psi_{j,k}$:

$$f(t)=\sum_{j,k} c_{j,k}\psi_{j,k}(t) \tag{1}$$

The basis functions used in WT are shifted and scaled versions of a mother wavelet Ψ:

$$\psi_{j,k}(t)=\frac{1}{\sqrt{a}}\Psi\left(\frac{t-ka}{a}\right) \tag{2}$$

where j and k are integers, and $a = 2^j$ is called a scale. The scaling index j changes the behavior of $\psi_{j,k}$ in the frequency space, while the translation index k shifts the wavelet along the time axis. Eq. (2) represents dyadic sampling of the time-scale space. Thus, the mother wavelet is stretched/compressed in discrete steps to create different scales based on powers of two.

WT allows decomposition of a signal into multiresolution components. The lower the scale (smaller j), the more "compressed" is the wavelet. The fine and coarse resolution components are obtained at low and high scales, respectively. They also capture the high-frequency and the low-frequency parts of a signal. There are several families of mother wavelets with different shapes and properties, among them the Daubechies family is one of the most popular [32].

The discrete WT is most frequently computed by recursive multiplication of the signal with a particular filter matrix. At each scale level l a set of 2^{l-1} wavelet coefficients is calculated where $l < L$ and $N = 2^L$ is the length of the input signal s. Detailed coefficients (details) are computed by applying a high-pass filter to the signal and downsampling the result by a factor of 2. Approximation coefficients used as input at the next scale are computed by low-pass filtering with downsampling. The scheme of this process is shown in Figure 2. Starting from s, the first step produces two sets of coefficients: approximation coefficients cA_1, and detail coefficients cD_1. The next step splits cA_1 in two parts using the same scheme, replacing s by cA_1 and producing cA_2 and cD_2, and so on (Figure 3). The discrete WT consists of $\log_2 N$ stages at most. The wavelet decomposition of the signal s analyzed at level l has the following structure: cA_l, cD_l, cD_{l-1}, …, cD_1. The selection of a suitable level for the decomposition hierarchy depends on the signal type.

Figure 2. A discrete WT algorithm.

Figure 3. An example of a wavelet structure.

2.3. FAM Classifier

The classification system of EChem++ is based on the FAM algorithm [33] which belongs to a family of NNs, called Adaptive Resonance Theory (ART) [34]. Initially, the building blocks used in all ART networks were designed in order to simulate human information processing, in particular, visual processing, as plausibly as possible. After general ART principles were formulated by Grossberg [34], a large family of self-organizing NNs for unsupervised and supervised learning was developed. ART-based NNs

are well suited for practical classification applications in different fields [35-38] including some interesting applications in chemistry [12, 13, 39-42]. Their advantages in comparison to other neural algorithms are the ability of stable fast learning, adaptively growing structure, simple computations and tuning. They were found to be more accurate in solving classification tasks than some feedforward networks [38, 42]. ART structure is based on a two-layered competitive learning network [43], but the training paradigm is different. The principle of match-based learning in ART networks as opposite to the more usual error-based learning with global optimization ensures temporal stability and allows a network to be trained online. Match-based learning is tightly coupled with the term "resonance" in the name of the paradigm. It reflects the fact that in an ART network information reverberates back and forth between layers. If a close correspondence between neuron activities at both layers occurs, then this resonant state initiates the network's learning process. Unless the network has achieved resonance, no learning takes place.

FAM is a supervised classifier combining ART principles with fuzzy logic [44]. In this study, a simplified version of FAM as represented in Figure 4 was used. It consists of a module ART_a and an associative memory Map Field (see Figure 4). ART_a includes three layers: a preprocessing layer F_0, a comparison layer F_1, and a competitive layer F_2. It operates as follows. As an input vector $\boldsymbol{I}^a$ is presented, it is preprocessed at F_0 and transmitted to the next layer. Then the F_1 layer transfers it to the competitive layer F_2 to find a winner. Each neuron of F_2 stores in its weight connections a prototype, i.e. a set of relevant features, describing a cluster of inputs in the feature space (inputs belonging to the same cluster share common characteristics). The best matching prototype, that is the one with the largest activation, is said to be a winner.

Figure 4. A simplified FAM network; explanation of symbols, see text.

The winner sends its response back to the F_1 layer where the "vigilance test" occurs: the input vector has to be compared with the winner's prototype against a user-specified threshold (the so-called vigilance parameter ρ) which defines the minimum required similarity between an input vector and the prototype of the cluster it can be associated with. If the calculated similarity is smaller than ρ, the current winner is reset and the search process starts again without it. Once a good matching prototype is found, no reset signal is sent and the network attains resonance. When no existing prototype provides the satisfactory similarity, then the network will develop a resonant state adding a new neuron to the competitive layer, whose prototype is first installed as the current input vector. A result of the resonant state is learning: updating the weight vector of the winning neuron in order to code the current input.

Thus, ART has a dynamically self-organizing structure within which the number of prototypes grows to adapt to the environment. Due to this property, an ART network is capable of incorporating new data preserving good performance on previously learned data and therefore successfully solves the so-called stability-plasticity dilemma [43]. Output neurons in ART represent clusters of predefined similarity among input patterns. Tuning of the vigilance parameter plays an important role in ART classification because the size of a network as well as its generalization ability depends on the value of ρ: with high values, the network tends to create a large number of small clusters to represent the underlying class distributions while with low ρ the clusters are large.

After the presentation of an input vector $\boldsymbol{I}^a$ with the components $I_i^a \in [0, 1]$ to the preprocessing layer F_0, it is complement coded into a $2M$-dimensional vector $\boldsymbol{A} = (I_1^a, \ldots,$

I_M^a, 1 - I_1^a, …, 1 – I_M^a). This means normalization to a constant vector length in terms of the "city-block" or "Manhattan" norm L_1 denoted as |…| [45]. For an M-dimensional vector $\boldsymbol{X}$, L_1 is calculated as

$$|\boldsymbol{X}| \equiv \sum_{i=1}^{M} X_i \ . \tag{3}$$

Thus, for all complement coded input vectors the norm is the same:

$$|\boldsymbol{A}| \equiv \sum_{i=1}^{M} I_i^a + (M - \sum_{i=1}^{M} I_i^a) = M \ . \tag{4}$$

The next step is the choice of a winner. At the layer F_2, the activation function T_j defined by the Weber Law [43] is computed for each neuron:

$$T_j = \frac{|\boldsymbol{A} \wedge \boldsymbol{W}_j|}{\alpha + |\boldsymbol{W}_j|} \tag{5}$$

where „∧" denotes the fuzzy AND operator, $(\boldsymbol{X} \wedge \boldsymbol{Y})_i \equiv \min(x_i, y_i)$, and $\alpha > 0$ is called the choice parameter. Then the best matching prototype J is found as belonging to the node with the maximal value of T_j.

$$T_J = \max\{T_j : j = 1, ..., N\}. \tag{6}$$

If more than one T_j is maximal, the node with the smallest index is chosen.

The choice of J must be confirmed by checking the vigilance test at the layer F_1:

$$C_J = \frac{|\boldsymbol{A} \wedge \boldsymbol{W}_J|}{|\boldsymbol{A}|} \geq \rho \ . \tag{7}$$

If inequality (7) fails, the system inhibits the winning node J and enables another neuron to be selected. This search process continues until the input is either assigned to an existing node which satisfies (7) or codes the prototype of a new neuron.

The ART_a module is part of the larger FAM network which is initialized with an arbitrary number N of neurons in F_2. Their weight vectors are set to unity $W_{j1}(0) = \ldots =$

= $W_{j2M}(0) = 1$ for all nodes $j = 1,\ldots, N$. The Map Field weights $\boldsymbol{w}_{jk}^{ab}$ are set to unity too. The vigilance parameter $\rho \in [0, 1]$ is initialized with a user-defined value.

During supervised training, input vectors $\boldsymbol{I}^a$ are presented to the network together with their class labels $\boldsymbol{I}^b$ which in simplified FAM can be coded directly in the binary K-dimensional vector $\boldsymbol{Y}^b$ of positional notation (i.e. '1 0 0' for the first of three classes and '0 0 1' for the third one). If the ART_a prototype chosen during the search stage for an input vector matches the corresponding class label, then an association between the prototype and the proper class is formed at the Map Field. Otherwise the Match Tracking (MT) process [33] initiates a choice of a new ART_a prototype by increasing the vigilance parameter to the value slightly greater than C_J of inequality (7).

A successful end of search leads to learning changes in the weight vector of the winning node:

$$\boldsymbol{W}_J^{\text{new}} = \beta(\boldsymbol{A} \wedge \boldsymbol{W}_J^{\text{old}}) + (1 - \beta)\boldsymbol{W}_J^{\text{old}} \tag{8}$$

where $\beta \in [0, 1]$ is the learning rate defining the FAM training dynamics. A β value close to 0 provides small gradual changes in the weights and therefore repetitive presentations of a training set are needed. The fast learning mode is achieved by setting $\beta = 1$. In this mode, a network is capable of learning during a single presentation of the training set and has a correspondingly short training time. Unfortunately, fast learning exhibits high sensitivity to the presentation order of input patterns. This means that the same data presented in a different order can cause a somewhat different result. Therefore, several training runs are usually needed to get a better estimate of the average classification performance.

Graphically expressed, weight vectors of the competitive neurons in FAM with complement coding can be represented as M-dimensional hyperrectangles (hyperboxes) with edges parallel to the coordinate axes in the input space (see Figure 5 for the two-

dimensional case). All points inside a hyperrectangle defined by the intervals [W_{ij}, 1 – – $W_{i+M,j}$] belong to the $\boldsymbol{W}_j$ prototype. The hyperbox size | R_j | is calculated as the sum of the lengths of its sides:

$$|R_j| = \sum_{i=1}^{M}((1-W_{i+M,j})-W_{ij}) = M-|\boldsymbol{W}_j|. \quad (9)$$

Figure 5. Growth of the rectangle R_1 to the point $\boldsymbol{A}$ during fast learning.

In this geometrical interpretation, the fast learning process could be seen as follows: if the hyperrectangle of the winning node J does not already contain an input point $\boldsymbol{A}$, then it expands just enough to include $\boldsymbol{A}$. Otherwise no update occurs. Starting from a point size, the hyperrectangles grow during training each time when it is necessary to enclose a new data point, until they achieve a maximum size predefined by the user through the vigilance parameter:

$$|R_j \oplus \boldsymbol{A}| \le (1-\rho)M \quad (10)$$

where $R_j \oplus \boldsymbol{A}$ denotes the minimum hyperrectangle containing both R_j and $\boldsymbol{A}$. Although it was supposed [42] that the clusters will oscillate during fast learning, this is not true for FAM learning dynamics. Clusters (hyperboxes) will only continuously expand. Inequality (10) corresponds to the match criterion (7).

3. Computational

For generation of simulated voltammograms the simulation program EASI [46] written in Fortran 77 was used. The PCA and WT procedures were implemented in MATLAB 6.5

(with the Wavelet Toolbox). For classification experiments, a FAM implementation from the simulator ARTSIM [47] written in Visual C++ was first used. Then the whole classification procedure was rewritten in order to incorporate it into the EChem++ package. Along with a new graphical user interface based on the QT library [48], it became a part of the *Analysis* module within the EChem++ program available from [23].

4. Results and discussion

4.1 Data sets

In order to examine the classification of cyclic voltammograms with FAM, several experiments using various data sets were designed. The first data set was composed of 2754 simulated voltammograms generated under the following conditions: The potential was changed between 0 and 0.5 V in 1 mV steps. The formal potential was set as 0.25 V. For each combination of the reaction parameters, nine cyclic voltammograms with different values of the scan rate (v = 0.02, 0.05, 0.1, 0.2, 0.5, 1, 2, 5, 10 V s^{-1}) were simulated.

The cyclic voltammograms should be classified into four classes depending on the underlying reaction mechanism. The mechanisms chosen have close relations to each other and in extreme cases they result in identical voltammograms. The mechanisms cover a single electron transfer step possibly coupled with a homogeneous irreversible chemical follow-up reaction. The first two classes were represented by a reversible (E) and quasi-reversible (E_{qr}) electron transfer mechanisms without a subsequent chemical reaction. With very large values of the heterogeneous constant k_s, the E_{qr} mechanism degenerates to E. A total of 17 values of k_s were used to produce E_{qr} curves (Table 1). The second two classes were represented by the same mechanisms but with a subsequent chemical reaction (EC

and $E_{qr}C$, respectively). Here, the rate constant k_1 of the chemical reaction plays an additional role. The 16 values of k_1 shown in Table 1 were used. The examples for the $E_{qr}C$ mechanism were generated with all possible 272 combinations of the two kinetic constants. A variation of this data set was obtained by merging nine voltammograms corresponding to different scan rates but with identical reaction parameters in one vector that provided 306 high-dimensional vectors.

Table 1. Values of k_1 and k_s used for simulation of voltammograms.

Besides the described data set there were three experimental data sets which consisted of 72 measured cyclic voltammograms. In the first case the electroactive species was an iridium P-C-P-pincer complex **I** [49]; in the second and third data sets voltammograms of two ruthenium complexes **II** and **III** [50] were used (Formulae **I** - **III**). The signals were measured with 9 scan rates of 0.05, 0.1, 0.2, 0.5, 1.003, 2.007, 5.120, 10.240 V s^{-1} for four different initial concentrations of the starting compounds and each scan rate was applied twice. The potential was varied between 0 and 0.6 V (iridium complex) or 0.5 V (ruthenium complexes) with an increment of 1 mV. From a direct comparison of experimental data with various simulated models it is assumed that the measured voltammograms correspond to the reaction mechanisms with the parameters presented in Table 2. In the classification experiments these data were added to the simulated voltammograms which resulted in three data sets of 2826 voltammograms.

Table 2. Parameters of experimental voltammograms.

4.2 Data preprocessing

A common preprocessing method in all experiments was scaling the data first by a factor of $(Ac^0\sqrt{v})^{-1}$ (A is the electrode area, c^0 is the initial concentration of the starting compound). This decreases or even eliminates the influence of A, v and c^0 on the data.

In the first experiment, there was no additional preprocessing. In the following four experiments, different preprocessing procedures, i.e. downsampling, Principal Component Analysis (PCA), feature extraction, and WT were applied to the data after scaling.

In the second experiment, downsampling was used to reduce the signals from 1000 to 200, 100, 40 or 20 points.

For the third experiment, four and five principal components covering 92.4 % and 95.7% of the variance in the data set, respectively, were calculated. PCA is a commonly used method to assess the redundancy of a data set [27, 51].

In the fourth experiment, a simple feature extraction procedure was performed: only the peak values were taken from each voltammogram for analysis because of their importance for human experts. The peak potential E_p^{for} and the corresponding current i_p^{for} of the forward peak were the first two features; the peak potential difference (E_p^{for} - E_p^{back}) and the peak current ratio (i_p^{for}/i_p^{back}) were the second two features. The peak potential difference is often used for discriminating between reversible and quasi-reversible processes, and the peak current ratio can serve as an indicator of a follow-up chemical reaction [52]. Note that the definition of the peak current ratio differs from that used earlier in [53] due to computational simplicity. In addition to these four features, the half peak potential $E_{p/2}$ (the potential value corresponding to a current equal to 50% of the peak current ($i_p/2$)) was also used for comparison. Figure 6 shows the used features by the example of a voltammogram belonging to the E mechanism. This preprocessing method as

well as PCA provided the highest reduction of data dimensionality (from 1000 to 4 or 5 components or features).

Figure 6. Features extracted from a voltammogram.

In the fifth experiment, the discrete WT based on the mother wavelets db4 and db5 (with four and five vanishing moments, respectively) from the Daubechies family was used for compression of the voltammograms. Other wavelet families such as coiflet-2 [32] were also tested; they provide very similar classification results because their shapes are not very different from db4 and db5. As a discrete WT requires the length of a signal or the number of sample points to be dyadic, i.e. 2^n, the initial voltammograms were extrapolated by so-called smooth padding. This method assumes that a signal is constant outside the time interval of a measurement, i.e. a signal extension on the left side is the repetition of the first value and on the right side of the last value. Thus, the extended length of signals was $1024 = 2^{10}$. The choice of the decomposition level was made taking into account the number of resulting approximation coefficients. Since simulated cyclic voltammograms are very smooth, the wavelet coefficients of lower scales (higher frequencies) are close to zero and the approximation part can be seen as a very compressed signal representation. Approximation coefficients at levels from 3 to 6 were used.

As an alternative to the analysis of approximation parts, selection of wavelet coefficients according to their variability was performed additionally in the fifth experiment. A simple method was used to choose the most important wavelet coefficients that exhibit the greatest potential to discriminate the classes: only 10 or 20 coefficients with the largest variances and large average values were selected for classification.

4.3 Training and classification parameters

In classification applications it is appropriate to use classification accuracy as a performance measure of a classifier. In general, it is unrealistic to expect error-free classification of a data set, except for really simple problems [45]. Real world data taken from distributions of several classes with significant inter-class overlap usually cannot conform to perfect class separability without overfitting. Achieving 100% classification accuracy would result in overfitting on a particular data set, which automatically leads to worse generalization (i.e., classification ability for a new unknown sample). A general way to measure the accuracy of a classifier is to count the percentage of misclassifications on a separate test set which was not used during training (generalization performance), i.e. its error rate. To assess the classification performance independently of presentation order, 30 random splits of data into training and test parts were undertaken, providing an average of error rates calculated for the respective test parts.

In experiments with simulated voltammograms, the training set consisted of 2000 samples and the test set of 754 samples for the case of single voltammograms and of 222 and 84 samples for the case when nine voltammograms corresponding to different scan rates were merged in one vector. Classification of experimental voltammograms was made using a leave-one-out technique: the classification performance was tested repeatedly on each exemplar of the data set after training on the remaining samples. This technique was also used for the error analysis of the classifier.

The FAM classifier was used in the fast learning mode ($\beta = 1$). The vigilance parameter ρ was set equal to 0.8.

4.4 Classification of simulated data

4.4.1 Comparison of classification with different preprocessing techniques

Table 3 summarizes the error rates averaged over 30 classification runs on the first data set. The results were obtained by applying different preprocessing methods. In all cases, the FAM network was able to recognize the reaction mechanisms of the 754 simulated voltammograms in the test set with high accuracy (85.8-95.7%). These results also demonstrate an essential advantage of FAM – its ability to cope well with high-dimensional data. Classification was successful for vectors consisting of 1000 components. Comparing the input dimensionalities after preprocessing, one can conclude that the best classification performance was obtained with a moderate reduction of data dimensionality. It was also found that simple downsampling does not affect classification results significantly. An error rate of about 5 % was achieved by downsampling of the data to as few as 20 points. This good result can be explained in part by the smooth and noise-free nature of simulated curves. At this extent of downsampling, however, the classification error starts to increase. Thus, a further decrease of the number of data points is not advisable. The situation could be relatively more difficult when classifying experimental signals with noise. In this case additional preprocessing would be necessary.

Table 3. Classification results for simulated cyclic voltammograms with different preprocessing techniques (In experiment 4, four components are without and five with half peak potential. Selection in experiment 5 means choosing the wavelet coefficients with the largest variances.)

It is worth pointing out that PCA (experiment 3) provided slightly worse classification results than extraction of peak values from voltammograms (experiment 4). This confirms the importance of human expertise in electrochemical data analysis. However, the use of

the few characteristic features usually determined by the chemists seems to be insufficient for achieving the best classification performance. The same result was obtained in [18]. Nevertheless, it must be mentioned that the dimensionality of the preprocessed data in experiments 3 and 4 was much less than in experiment 2.

Preprocessing by WT (experiment 5) showed classification accuracy very similar to downsampling of the signals. However, it is expected that experimental voltammograms with a low signal to noise ratio and poor quality would be better classified after WT preprocessing. The efficiency of using only approximation coefficients was confirmed by the experiments. It can be seen that an alternative method of choosing the most important wavelet coefficients by checking their variability in the sixth experiment turned out to be insufficient for feature selection.

For a variation of the first data set consisting of voltammograms merged from nine scan rates, two preprocessing techniques, downsampling to 40 points and WT db45, were compared. Table 4 contains error rates averaged over 100 classification runs. One can see that classification accuracy was again very similar for both preprocessing methods, but it is lower as compared to the classification of single voltammograms presented above. This is contrary to usual practice in the analysis of voltammograms where series of current/potential curves acquired at various scan rates are studied together. The reason for this decrease in classification performance is that the size of the training set was significantly smaller (9 times) when the merged voltammograms were used, especially relative to the increased data dimensionality. For example, only one exemplar of the E mechanism was available which could be used either for training or for classification.

Table 4. Classification results for merged voltammograms.

4.4.2 Analysis of error distribution

In order to investigate the classification performance of FAM in more detail, an analysis of error distributions was carried out on the simulated data. The aim was to find those voltammograms which are most difficult to classify. Figures 7 and 8 show the error distribution in the classes for experiments with downsampling to 200 points and preprocessing by WT db46, respectively. Two axes represent the values of constants k_1 and k_s, while the third axis quantifies the percentage of misclassifications for 9 voltammograms during 100 runs of leave-one-out classification. In this coordinate system, the EC and E_{qr} classes marked by pink and orange colors extend along the first two axes; the E class (red) is in the front corner; and the $E_{qr}C$ class (white) occupies the rest of the k_1k_s-plane. Although this visualization helps understand the spatial relationship between the mechanisms in the k_1k_s-space, it should be taken into account that the k-axes have nonuniform scales. The classification error was 3.5% and 3.8% for downsampling and WT, respectively.

Figure 7. Error distribution of voltammogram classification into classes E/E_{qr} and $EC/E_{qr}C$ after downsampling to 200 points.

It can be seen that most errors are due to misclassifications of $E_{qr}C$ examples as EC or E_{qr} ones and vice versa. This can be explained by the fact that the overlap between these classes is high, i.e. the voltammograms of the corresponding two classes in some regions of the parameter space become very similar. With increasing k_1, the number of misclassifications of $E_{qr}C$ as EC grows. As k_s decreases, E_{qr} samples become more likely to

be classified as $E_{qr}C$ ones. The results are consistent with expectations: Since class boundaries between reaction mechanisms are rather fuzzy, it can be difficult to classify voltammograms which lie close to them. It is also interesting to mention that the $E_{qr}C$ class is often misclassified as EC, and the E_{qr} class with small k_s as $E_{qr}C$. Possibly, for a fast chemical follow-up reaction (large k_1) small changes in the reversibility of the electron transfer (as controlled by k_s) do not change the shape of the voltammogram to an appreciable extent. On the other hand, for very small k_s the slow electron transfer remains determining the shape even when some material starts to interact in the follow-up reaction. This relation is independent of preprocessing. Visual comparison of both plots leads to the conclusion that WT preprocessing (Figure 8) has only a minor influence on the error distribution.

Figure 8. Error distribution of voltammogram classification into classes E/E_{qr} and $EC/E_{qr}C$ after WT preprocessing with db46.

As compared to the results of studies [14-17] where an expert system was used for an automatic elucidation of reaction mechanisms, classification with the NN presented here is advantageous because of its high speed and flexibility. The number of supported mechanisms can be increased simply by training on a proper data set and therefore it is not necessary to construct some theoretical criteria for discriminating between mechanisms. A disadvantage is the somewhat "black-box" data processing of an NN, i.e. it is very difficult to explain why the network has classified a voltammogram into a certain class. As a possible solution a combined neuro-fuzzy system [54] can be proposed. Such algorithms allow the user to interpret the decision-making process.

4.4.3 Classification and zone diagramms

In some classical voltammetric papers, e.g. [55], (see also the collection of graphical representations in [21]) zone diagrams have been constructed to visualize the relationship between similar reaction mechanisms as a function of the fundamental kinetic parameters (here k_1 and k_s) or their functional transformations. The borders between the zones in such diagrams were defined on the basis of some criterion, e.g. the peak potential within an error boundary of 2 mV [55]. The diagrams allow one to estimate a fraction of the parameter space within which the system's behavior does not depart from that expected for certain limiting cases by more than the experimental error. Consequently, they can be used to systematically identify limiting behaviors and the transition between such cases. Furthermore, the zone boundaries should be those regions in the parameter space where voltammograms from the neighboring zones inherently show high similarity. This approach corresponds to clustering in data analysis and can be performed by NNs during unsupervised learning on simulated data. In such a case the NN learns to group similar voltammograms together building clusters in the parameter space.

In contrast to the zone diagram approach, the present attempt of classification as a result of the supervised learning process assumes that only the most extreme parameter combinations belong to the limiting cases. Thus, for example, the border line between the EC and $E_{qr}C$ mechanisms is shifted very close to the left hand side of the k_1k_s-plane shown in Figures 7 and 8. Since, however, $E_{qr}C$ voltammograms with large values of k_s are almost indistinguishable from EC voltammograms, one would expect a high misclassification probability near the zone boundaries. Indeed this is the case in our classification experiments. A similar effect is, of course, observed for the $E_{qr}/E_{qr}C$ border line at small k_1.

4.5 Classification of experimental data

The data sets including experimental voltammograms were preprocessed by downsampling to 200 points as well as by WT db46. With the leave-one-out technique, all experimental samples were classified correctly as belonging to the corresponding E_{qr} or $E_{qr}C$ classes independent of preprocessing. The overall error rate was 3.4 % with downsampling and 3.9 % with WT which is comparable to 3.3 % obtained in [18] on a simulated data set consisted of 885 curves with leave-one-out classification by the nearest neighbor classifier.

5. Conclusion and future work

In this paper, a neural network classification system for cyclic voltammograms is introduced. It has been proposed in the framework of the EChem++ project and is incorporated into the software as a part of the data analysis module. It is based on the Fuzzy ARTMAP classifier and utilizes various data preprocessing techniques. Several classification experiments were carried out to examine the performance of the classifier and to compare the efficiency of preprocessing methods. They show that the developed classification system successfully discriminates between both simulated or experimental voltammograms corresponding to different electrode reaction mechanisms. Since the experimental data set used contained voltammograms of the E_{qr} and $E_{qr}C$ mechanisms only, additional experiments with extended data sets are planned. Another important aspect is to apply an unsupervised learned NN for clustering of simulated voltammograms in order to create zone diagrams automatically.

6. Acknowledgements

We thank the Deutsche Forschungsgemeinschaft, Bonn-Bad Godesberg, Germany, for financial support of this work within the Graduiertenkolleg 441 “Chemie in Interphasen”. We are grateful to Kai Ludwig and Filip Novak, Institut für Organische Chemie, Universität Tübingen, for preparing simulated and experimental data, respectively, as well as for fruitful discussions. We thank Nicolas Plumeré, Institut für Organische Chemie, Universität Tübingen, for drawing chemical structures **II** and **III**.

References

[1] K. Ludwig, B. Speiser, J. Chem. Inf. Comput. Sci. 44 (2004) 2051-2060.
[2] X. Lu, J. Mo, C. Yang, J. Kang, J. Gao, Anal. Lett. 33 (2000) 1167-1186.
[3] E. Richards, C. Bessant, S. Saini, Electroanalysis 14 (2002) 1533-1542.
[4] B. Speiser, in: A.J. Bard and M. Stratmann (Eds.), Encyclopedia of electrochemistry, Wiley-VCH, Weinheim, 2004, Vol. 8, pp. 1-23.
[5] C. Bessant, S. Saini, Electroanalysis 9 (1997) 926-931.
[6] A. Economou, S.D. Bolis, C.E. Efstathiou, G.J. Volikakis, Anal. Chim. Acta 467 (2002) 179-188.
[7] L. Bieniasz, in: B. Conway, White R. (Eds.), Modern aspects of electrochemistry, Kluwer Academic/Plenum Publishers, New York, 2002, Vol. 35, pp. 135-193.
[8] Z.B. Alfassi, Z. Boger, Y. Ronen. Statistical treatment of analytical data. Blackwell Science, Oxford, 2005.
[9] R.J. Mammone, Y.Y. Zeevi (Eds.), Neural networks: theory and applications, Academic Press: San Diego, 1991.
[10] A. Sundararaj, R. Ravi, T. Parthiban, G. Radhakrishnan, Bull. Electrochem. 15 (1999) 552-555.
[11] E. Cukrowska, L. Trnkova, R. Kizek, J. Havel, J. Electroanal. Chem. 503 (2001) 117-124.
[12] D. Wienke, G. Kateman, Chemom. Intell. Lab. Syst. 23 (1994) 309-329.
[13] D. Domine, J. Devillers, D. Wienke, L. Buydens, J. Chem. Inf. Comput. Sci. 37 (1997) 10-17.
[14] M. Palys, M.Bos, W.E. van der Linden, Anal. Chim. Acta 231 (1990) 59-67.
[15] –“–, Anal. Chim. Acta 248 (1991) 429-439.
[16] –“–, Anal. Chim. Acta 283 (1993) 811-829.

[17] –"–, Anal. Chim. Acta 284 (1993) 107-118.
[18] S.D. Schachterle, S.P. Perone, Anal. Chem. 53 (1981) 1672-1678.
[19] W.A. Byers, B.S. Freiser, S.P. Perone. Anal. Chem. 55 (1983) 620-625.
[20] C. Amatore, J.M. Saveant, Electroanal. Chem. 102 (1979) 21-40.
[21] A.J. Bard, L.R. Faulkner, Electrochemical Methods: Fundamentals and applications, pp. 486, 489, 500, Wiley, New York, 2001.
[22] K. Ludwig, L. Rajendran, B. Speiser, J. Electroanal. Chem. 568 (2004) 203-214.
[23] http://echempp.sourceforge.net.
[24] T. Hastie, R. Tibshirani, J.H. Friedman. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer, New York, 2001.
[25] J.M. Palacios-Santander, A. Jimenez-Jimenez, L.M. Cubillana-Aguilera, I. Naranjo-Rodriguez, J.L. Hidalgo-Hidalgo-de-Cisneros, Microchim. Acta 142 (2003) 27-36.
[26] A. Alexandridis, P. Patrinos, H. Sarimveis, G. Tsekouras, Chemom. Intell. Lab. Syst. 75 (2005) 149-162.
[27] M. Berthold and D.J. Hand (Eds.), Intelligent data analysis: an introduction, Springer, Berlin, 2003.
[28] H. Fang, H.-Y. Chen, Anal. Chim. Acta 346 (1997) 319-325.
[29] S. Wu, L. Nie, J. Wang, X. Lin, L. Zheng, L. Rui, J. Electroanal. Chem. 508 (2001) 11-27.
[30] X. Zhang, J. Zheng, H. Gao, Analyst 125 (2000) 915-919.
[31] X. Zou, J. Mao, Anal. Chim. Acta 340 (1997) 115-121.
[32] F.-T. Chau, Y.-Z. Liang, J. Gao, X.-G. Shao. Chemometrics: from basics to wavelet transform. Wiley, Hoboken, NJ, 2004.
[33] G.A. Carpenter, S. Grossberg, N. Markuzon, J.H. Reynolds, D.B. Rosen, IEEE Transactions on Neural Networks 3 (1992) 698-713.
[34] S. Grossberg, Studies of mind and brain: neural principles of learning, perception, development, cognition, and motor control, Reidel Press, Boston, 1982.
[35] E. Granger, M.A. Rubin, S. Grossberg, P. Lavoie, Neural Networks 14 (2001) 325-344.
[36] J. Downs, R.F. Harrison, S.S. Cross, Neural Computing and Applications 7 (1998) 147-165.
[37] R.K. Aggarwal, Q.Y. Xuan, A.T. Johns, F. Li, A. Bennet, IEEE trans. on Neural Networks, 10 (1999) 1214-1221.
[38] L. Ludwig, E. Sapozhnikova, V. Lunin, W. Rosenstiel, Neural Computing and Applications, 9 (2000) 202-210.
[39] D. Wienke, W. van den Broek, W. Melssen, L. Buydens, R. Feldhoff, T. Kantimm, T. Huth-Fehre, L. Quick, F. Winter, K. Cammann, Anal. Chim. Acta 317 (1995) 1-16.
[40] D. Wienke, L. Buydens, Trends Anal. Chem. 14 (1995) 398-406.
[41] D. Wienke, L. Buydens, Chemom. Intell. Lab. Syst. 32 (1996) 151-164.
[42] D. Wienke, W. van den Broek, L. Buydens, T. Huth-Fehre, R. Feldhoff, T. Kantimm, K. Cammann, Chemom. Intell. Lab. Syst. 32 (1996) 165-176.
[43] G.A.Carpenter and S. Grossberg (Eds.), Pattern recognition by self-organizing neural networks, MIT Press, Cambridge, 1991.

[44] B. Kosko. Neural networks and fuzzy systems. Prentice Hall, Englewood Cliffs, 1994.
[45] R.O. Duda, P.E. Hart, D.G. Stork, Pattern classification, Wiley, New York, 2001.
[46] B. Speiser, Comput. Chem. 14 (1990) 127 -140.
[47] E. Sapojnikova, ART-based fuzzy classifiers: ART fuzzy networks for automatic classification, Cuvillier Verlag, Goettingen, 2004.
[48] J. Blanchette, M. Summerfield. C++ GUI Programming with Qt 3. Prentice Hall, Upper Saddle River, 2004.
[49] F. Novak, B. Speiser, H.A.Y. Mohammad, H.A. Mayer, Electrochim. Acta 49 (2004) 3841-3853.
[50] E. Lindner, S. Al-Gharabli, H.A. Mayer, Inorg. Chim. Acta 334 (2002) 113-121.
[51] S. Verboven, M. Hubert, Chemom. Intell. Lab. Syst. 75 (2005) 127-136.
[52] R.S. Nicholson, I. Shain, Anal. Chem. 36 (1964) 706-723.
[53] R.S. Nicholson, Anal. Chem. 38 (1966) 1406.
[54] S. Mitra and Y. Hayashi, IEEE trans. on Neural Networks 11 (2000) 748-768.
[55] L. Nadjo, J.M. Saveant, J. Electroanal. Chem. 48 (1973) 113-145.

Figure captions

Figure 1. An NN in the framework of the electrochemical data processing of the EChem++ project.

Figure 2. A discrete WT algorithm.

Figure 3. An example of a wavelet structure.

Figure 4. A simplified FAM network; explanation of symbols, see text.

Figure 5. Growth of the rectangle R_I to the point $\boldsymbol{A}$ during fast learning.

Figure 6. Features extracted from a voltammogram.

Figure 7. Error distribution of voltammogram classification into classes E/E_{qr} and $EC/E_{qr}C$ after downsampling to 200 points.

Figure 8. Error distribution of voltammogram classification into classes E/E_{qr} and $EC/E_{qr}C$ after WT preprocessing with db46.

Tables

Table 1. Values of k_1 and k_s used for simulation of voltammograms.

class	me-chan.	parameters		number of parameter combinations	number of cyclic voltamm.
		k_1/s^{-1}	k_s/cm s^{-1}		
1	E	0	10000	1	9
2	E_{qr}	0	10^{-5},10^{-4}, 10^{-3}, 0.005, 0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09, 0.1, 0.5, 1, 10	17	153
3	EC	10^{-7},10^{-6}, 10^{-5},10^{-4},10^{-3}, 10^{-2}, 10^{-1},1, 10,10^{2},10^{3}, 10^{4}, 10^{5},10^{6},10^{7}, 10^{8}	10000	16	144
4	$E_{qr}C$	as EC	as E_{qr}	272	2448
total				306	2754

Table 2. Parameters of experimental voltammograms.

complex	mechanism	parameters	
		k_1/s^{-1}	k_s/cm s^{-1}
I	$E_{qr}C$	0.3	0.05
II	E_{qr}	0	0.05
III	E_{qr}	0	0.06

Table 3. Classification results for simulated cyclic voltammograms with different preprocessing techniques (In experiment 4, four components are without and five with half peak potential. Selection in experiment 5 means choosing the wavelet coefficients with the largest variances.)

Experiment	Preprocessing	Dimensionality	Classification error (%)	Standard deviation
1	-	1000	4.3	1.2
2	Downsampling	200	4.3	0.9
	Downsampling	100	4.3	0.9
	Downsampling	40	4.5	1.0
	Downsampling	20	5.1	1.2
3	PCA	4	14.2	2.4
	PCA	5	12.3	2.2
4	Feature extraction	4	12.4	2.0
	Feature extraction	5	10.1	1.7
5	Wavelet db53	135	4.3	0.8
	Wavelet db45	38	4.5	1.0
	Wavelet db46	22	5.0	1.3
	Wavelet db53 with selection	20	10.3	3.3
	Wavelet db53 with selection	10	10.7	3.2

Table 4. Classification results for merged voltammograms.

Preprocessing	Dimensionality	Classification error (%)	Standard deviation
Downsampling	360	10.9	2.8
Wavelet db45	342	11.2	3.1

Figure 1.

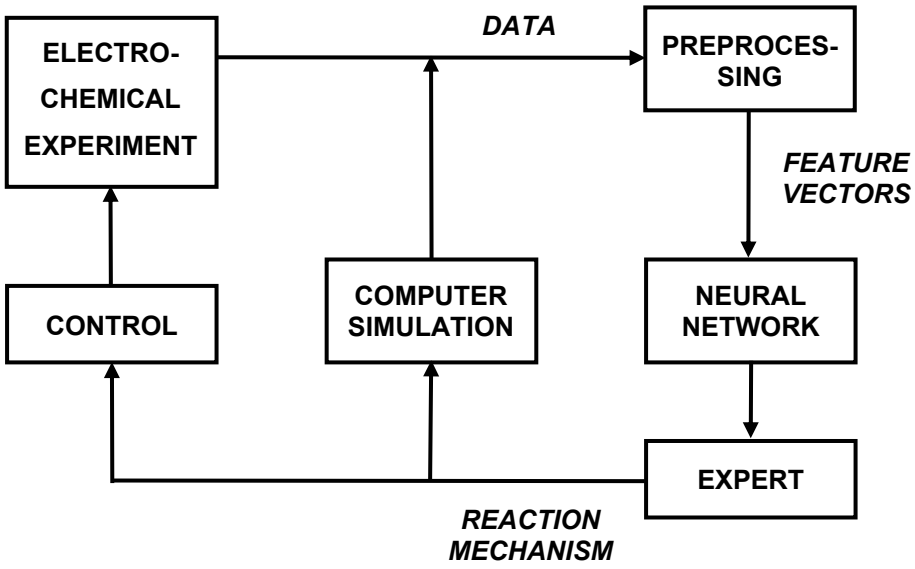

ELECTRO-
CHEMICAL
EXPERIMENT
DATA
PREPROCES-
SING
FEATURE
VECTORS
CONTROL
COMPUTER
SIMULATION
NEURAL
NETWORK
EXPERT
REACTION
MECHANISM

Figure 2.

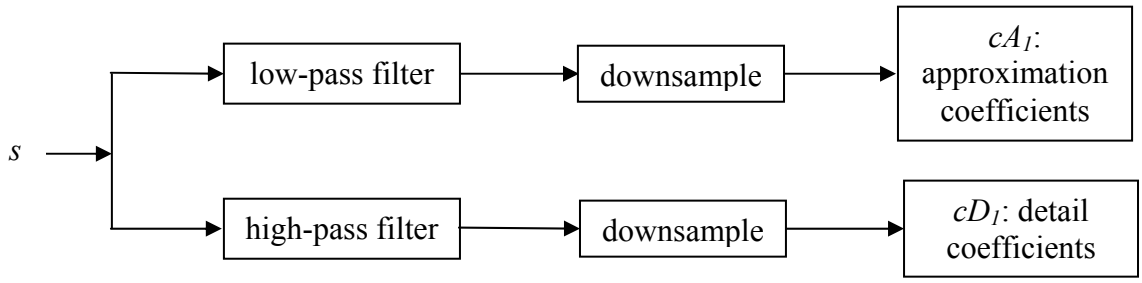

s
low-pass filter
downsample
cA1: approximation coefficients
high-pass filter
downsample
cD1: detail coefficients

Figure 3.

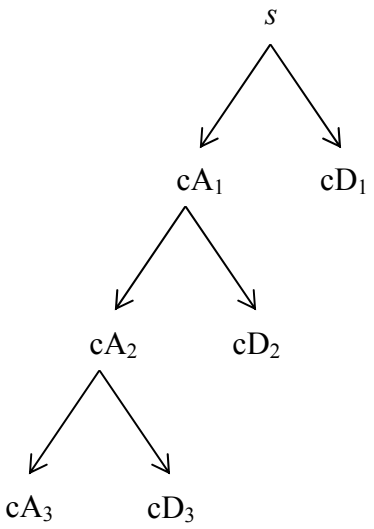


Figure 4.

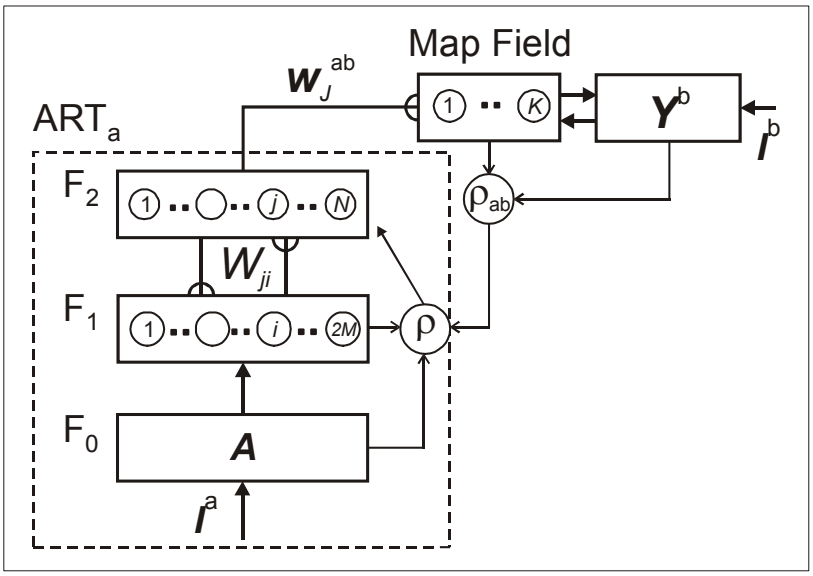


Figure 5.

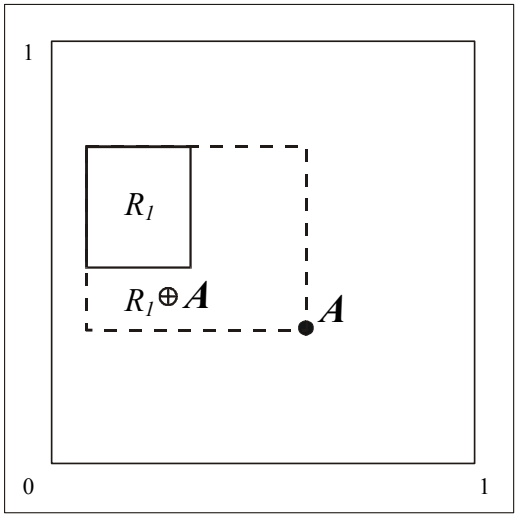


Figure 6.

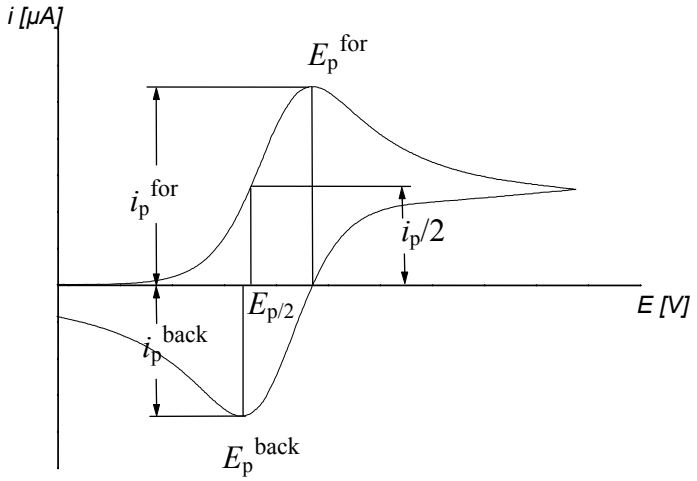


Figure 7.

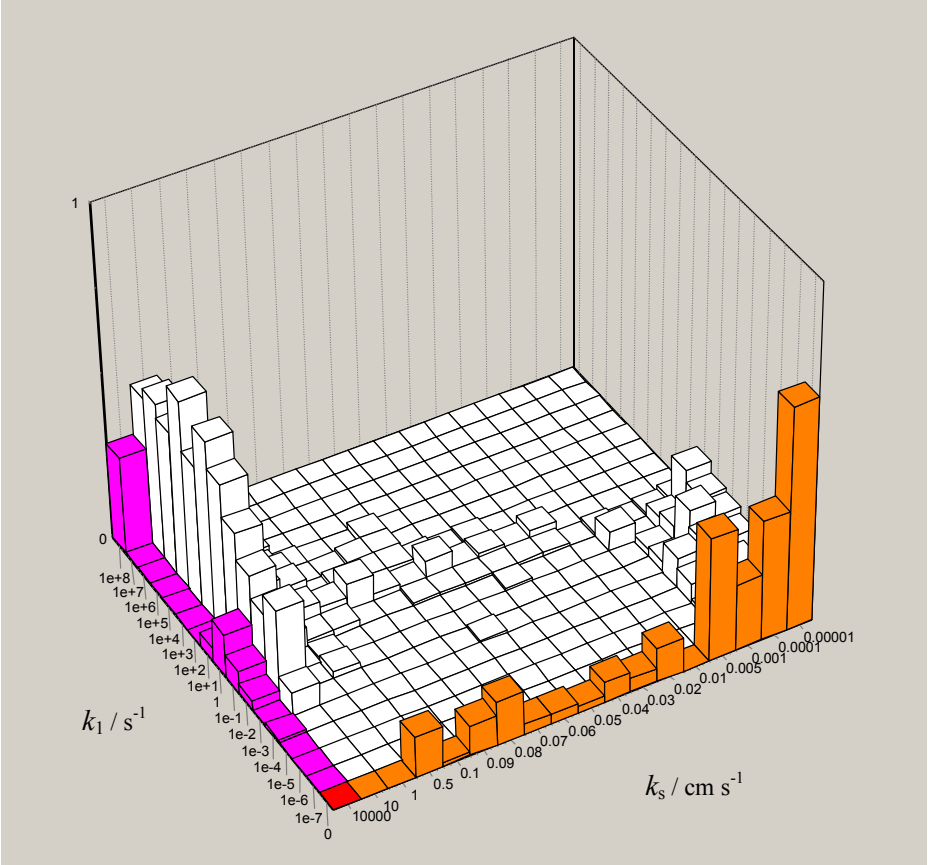
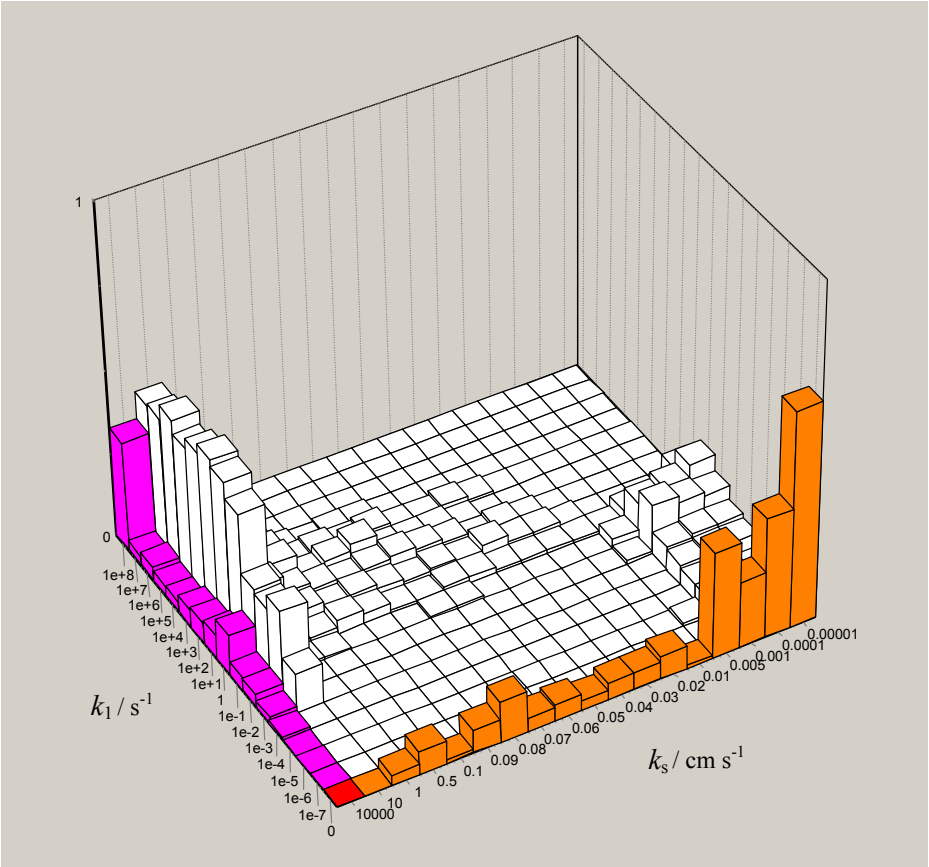
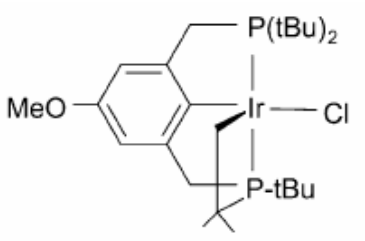
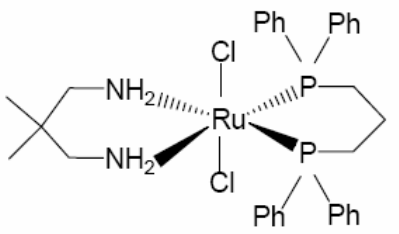

1
0
1e+8
1e+7
1e+6
1e+5
1e+4
1e+3
1e+2
1e+1
1
1e-1
1e-2
1e-3
1e-4
1e-5
1e-6
1e-7
0
k_1 / s^{-1}
10000
10
1
0.5
0.1
0.09
0.08
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0.005
0.001
0.0001
0.00001
k_s / cm s^{-1}

Figure 8.

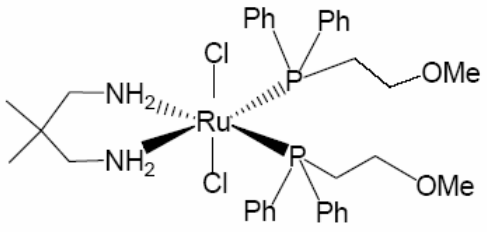

1
0
1e+8
1e+7
1e+6
1e+5
1e+4
1e+3
1e+2
1e+1
1
1e-1
1e-2
1e-3
1e-4
1e-5
1e-6
1e-7
0
k_l / s^{-1}
10000
10
1
0.5
0.1
0.09
0.08
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0.005
0.001
0.0001
0.00001
k_s / cm s^{-1}


P(tBu)2
MeO
Ir
Cl
P-tBu

I


Ph
Ph
Cl
NH2
P
Ru
NH2
P
Cl
Ph
Ph

II


Ph
Ph
Cl
P
OMe
NH2
Ru
NH2
P
Cl
OMe
Ph
Ph

III