E-NOSE VAPOR IDENTIFICATION BASED ON DEMPSTER-SHAFER FUSION OF MULTIPLE CLASSIFIERS

Winston Li[1], Henry Leung[1], Chiman Kwan[2], and Bruce R. Linnell[3]

[1] Department of Electrical and Computer Engineering, University of Calgary, Calgary, Alberta, Canada T2N 1N4, Emails: {winston_li66@yahoo.com, leungh@ucalgary.ca}

[2] Intelligent Automation, Inc., 7519 Standish Place, Suite 200, Rockville, MD 20855 Email: {ckwan@i-a-i.com}

[3] Applied Chemistry Laboratory, NASA Kennedy Space Center, Florida USA 32899, Email: {bruce.linnell-1@ksc.nasa.gov}

Abstract

Electronic nose (e-nose) vapor identification is an efficient approach to monitor air contaminants in space stations and shuttles in order to ensure the health and safety of astronauts. Data preprocessing (measurement denoising and feature extraction) and pattern classification are important components of an e-nose system. In this paper, a wavelet-based denoising method is applied to filter the noisy sensor measurements. Transient-state features are then extracted from the denoised sensor measurements, and are used to train multiple classifiers such as multi-layer perceptions (MLP), support vector machines (SVM), k nearest neighbor (KNN), and Parzen classifier. The Dempster-Shafer (DS) technique is used at the end to fuse the results of the multiple classifiers to get the final classification. Experimental analysis based on real vapor data shows that the wavelet denoising method can remove both random noise and outliers successfully, and the classification rate can be improved by using classifier fusion.

Keywords: electronic nose, wavelet denoising, Dempster-Shafer, neural network, support vector machine, k nearest neighbor, Parzen classifier.

I. INTRODUCTION

An electronic nose (e-nose) is an instrument that combines gas sensor arrays and pattern recognition techniques for recognizing simple and complex odors [1]. Applications of an e-nose system include environmental monitoring for space programs, quality assurance of food, water, and drugs, safety and security, and military applications. Using e-nose to identify vapors is an effective approach to monitor air contaminants in the space shuttle and international space stations in order to ensure the health and safety of astronauts [2-4].

Data preprocessing (measurement denoising and feature extraction) and pattern classification are two important components of an e-nose system [5-7]. The raw measurements of gas sensor arrays are usually contaminated by random noise and outliers. Adaptive filters have been used to filter out the random noise, but they are sensitive to outliers. Median filters have been considered for removing outliers of the raw e-nose data, but the denoised sensor response curve is not smooth. Wavelet transforms can decompose a signal into several scales that represent different frequency bands, and at each scale, the signal structure can be determined approximately. Such a property is often used for denoising [8, 9]. There are mainly three steps for wavelet denoising: (1) decompose the noisy data into multiple scales by wavelet transform, (2) employ the hard or soft threshold at each scale to filter the noises, (3) get the estimated signal by the inverse wavelet transform. The advantages of wavelet denoising are: (1) both random noises and outliers can be filtered successfully, (2) the denoised sensor response curve is smooth, which is helpful to extract derivative features.

Feature extraction is another important process for classification [10-12]. Various vapor features have been considered in the literature, and they can be divided into mainly two categories: steady-state and transient-state features. Many commercial instruments use the steady-state features for vapor classification, and disregard the transient property of the sensor response. However, it is time-consuming for some gas sensors to stabilize. The transient–state features have also been considered recently. These features include: (1) sensor values at a specific sampling time such as 90 seconds, (2) means on time intervals, (3) windowed time slicing: the transient response is multiplied by a bell-shaped windowing functions, and integrated with time, (4) first and second derivatives of the denoised sensor response curves. The feature vector can be further compressed by principal component analysis (PCA) [13].

When features are extracted, the vapors can be classified using various approaches such as statistical methods (linear classifier, k-nearest neighbors (KNN) [13], Parzen classifier [19]), and neural networks (MLP [15, 16], SVM [17, 18]). KNN and Parzen classifier are two suboptimal methods of approximating the theoretically optimal Bayesian classifier. The advantage of these two classifiers is their simplicity with only little performance degradation compared with the Bayesian classifier. The MLP and SVM classifiers can obtain higher correct classification rates than the non-neural network methods such as KNN and Parzen classifier if the neural network parameters are suitably chosen [3, 16]. To increase the correct classification rates of single classifiers, the results of multiple classifiers are often fused by some methods such as fuzzy logic [20, 21], Bayesian weighted average [22], and Dempster-Shafer (DS) techniques [23, 24].

In this paper, we propose using multiple classifiers to classify the vapor features and then apply the DS technique to fuse the multiple (dis)similar classifiers to improve the classification performance of vapors collected by the electronic noses.

This paper is organized as follows. Section II introduces the block diagram of the DS fusion based e-nose vapor identification system. Data preprocessing is explained in more detail in Section III including sensor drift calibration, wavelet denoising, feature extraction, and the DS fusion based classification technique is presented in Section IV. Section V reports the experimental results. Finally, the conclusions are given in Section VI.

II. DS FUSION BASED E-NOSE VAPOR IDENTIFICATION SYSTEM

The block diagram of the DS fusion based e-nose vapor identification system is shown in Figure 1. The vapors are first collected by a chemical sensor array and then input into the data preprocessing component. Data preprocessing includes sensor drift calibration, wavelet denoising, and feature extraction. Because the sensor responses will drift with time, the sensor must be calibrated before features are extracted. Baseline manipulation is used here to reduce the effects of the sensor drift which refers to transformations based on the initial values of the transient sensor responses [7]. Then wavelet denoising is applied to filter the random noise and outliers in the raw measurements. Some transient features are finally extracted from the denoised sensor responses.

Pattern classification is another important component of the e-nose system. There are mainly two blocks in the classification component: individual classifiers and DS fusion. Each feature is used to train individual classifier which can be MLP, SVM, KNN, or Parzen classifier. The fuzzy classification results are then input into the DS fusion block to estimate the final classification results.

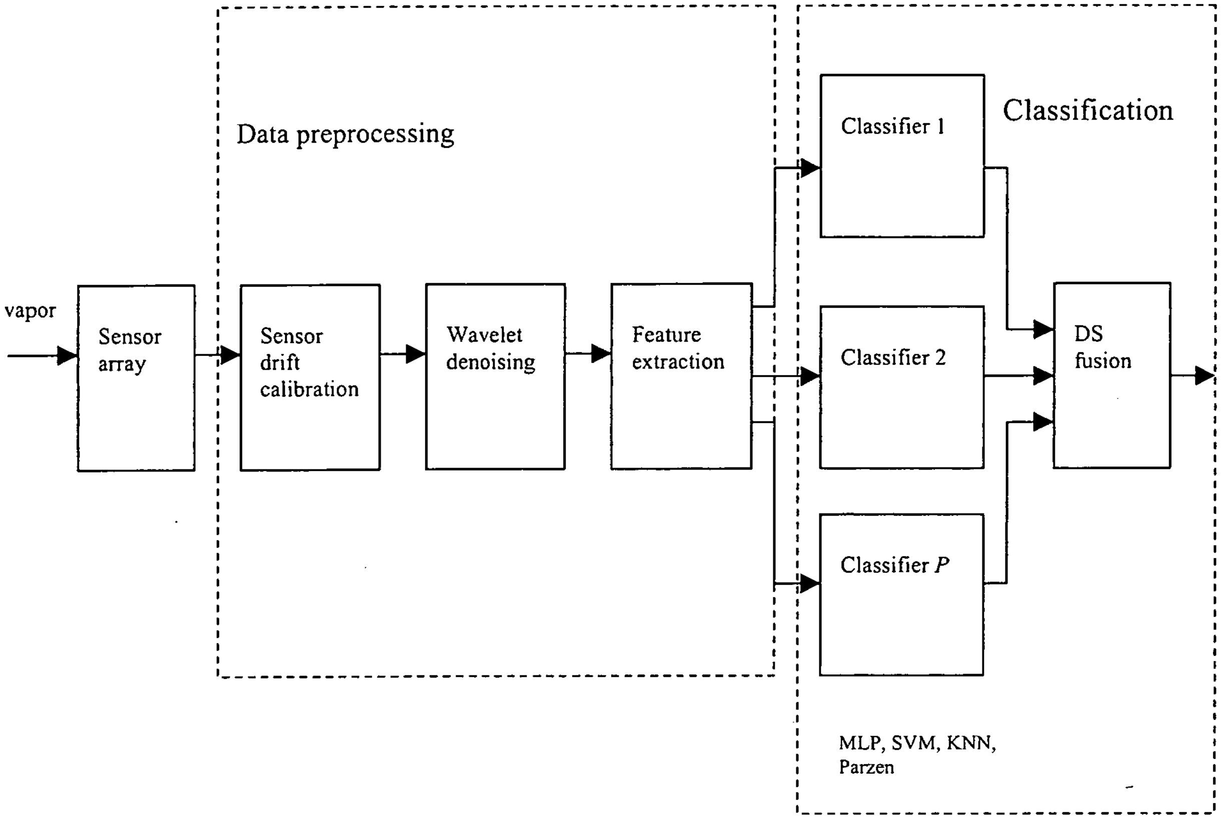


Figure 1. Block diagram of the DS fusion based e-nose vapor identification system.

III. PREPROCESSING OF RAW DATA SETS

A. Sensor drift calibration

The baseline of the sensor is defined as the initial value of the sensor transient response. Baseline manipulation is a simple and effective method to reduce the effect of the sensor drift [7]. Three basic transformations are: (1) differential: Subtract the baseline of each sensor; (2) Relative: Divide by the baseline of each sensor; (3) Fractional: A combination of the previous two methods. The fractional method is used in our experiments. The baseline is estimated by averaging some initial values of the sensor transient responses. The sample number is chosen to be 5 to 10. Suppose that the estimated baseline is f_0, and the raw sensor response at time t_k is denoted by $f(t_k)$, then the sensor response $f^B(t_k)$ after baseline manipulation is defined as

$$f^B(t_k) = \frac{f(t_k) - f_0}{f_0}. \tag{1}$$

B. Wavelet denoising

Because the measurements of each e-nose sensor are one-dimensional (1D) discrete signals, 1D discrete wavelet transform (DWT) is used for denoising the raw e-nose data. The 1D DWT is calculated using Mallat's algorithm. Suppose that filters *h* and *g* are quadrature mirror filters known as the scaling and wavelet filters, respectively. The scaling filter *h* is a lowpass filter, while the wavelet filter *g* is highpass which are determined by the wavelet basis functions. The wavelet transform coefficients, f_k and q_k at different scales, are calculated using the following convolution-like expressions:

$$f_k^{j-1} = \sum_n h_{n-2k} f_n^{\,j},$$

$$q_k^{j-1} = \sum_n g_{n-2k} f_n^{\,j} \tag{2}$$

where *j* denotes the resolution, $j = 1, 2, \cdots, J$, and *k* is the index for the samples. The operation defined in (2) is a linear digital filtering operation using filters *h* and *g*, followed by down-sampling. The top-level coefficients f_k^J represent the original signal f_k. The 1D inverse DWT (IDWT) is given by

$$f_k^{j+1} = \sum_n (h_{k-2n} f_n^{\,j} + g_{k-2n} q_n^{\,j}). \tag{3}$$

The operation defined in (3) is an up-sampling operation followed by linear digital filtering. At scale *j*, the scaling and wavelet coefficients are combined to form the level $j+1$ scaling coefficients.

Suppose that the gas sensor response f_k (i.e., the sensor response after transformation by equation (1)) is corrupted by white noise w_k, then the noisy sensor response can be expressed by

$$z_k = f_k + w_k, \tag{4}$$

where $k = 1, 2, \cdots, N$. *N* is the number of the e-nose samples.

Wavelet denoising usually involves three steps. The first step of denoising signal z_k is to take the wavelet transform by equation (2). The DWT of z_k can be considered as a linear superposition of the DWT of f_k with a small number of significant coefficients, and the DWT of w_k with a large number of small coefficients. In the second step, an approximated threshold is formed to exclude the small coefficients generated by w_k. The threshold is defined as [9]

$$\lambda = \frac{median\left|< z - \psi_{1,m} >\right|}{0.6745}\sqrt{2\ln\left(\frac{N}{2^m}\right)} \tag{5}$$

where ψ is the wavelet basis function, and m is the resolution level.

The soft thresholding is carried out by

$$\overline{q}_k^{\ j} = \begin{cases} sign(q_k^{\ j})(\left|q_k^{\ j}\right| - \lambda), if \left|q_k^{\ j}\right| > \lambda \\ 0, \qquad\qquad if \left|q_k^{\ j}\right| \leq \lambda \end{cases} . \tag{6}$$

In the third step, wavelet reconstruction given in (3) is used to yield the recovered signal $\hat{f}_k$ based on $f_k^{\ j}$ and $\overline{q}_k^{\ j}$. The reason why use soft threshold is that it can result in smoother sensor response curves.

C. Feature extraction

Feature extraction is another important data preprocessing step. There are two kinds of features: transient response and steady states which can be extracted from the denoised sensor response curve. A typical denoised sensor response curve is shown in Figure 2.

The whole sensor response curve can be exploited for e-nose feature extraction, from the absorption beginning to the desorption end [11]. In Figure 2, the first sample to about the 50th sample are the sensor absorption response, while the 50th sample to the 400th sample are the sensor desorption response. Traditionally the steady states are often used for classification because of its robustness. However, most sensors require a long time to stabilize. In order to obtain more features and keep shorter sampling time than the steady-state features, the transient features become more popular recently. The sensor values at specific times are used which are much earlier than the time needed for the sensors to arrive at the steady states. For example, the sensor measurements at time 90 seconds can be used as features for classification. Means on time intervals, and first and second derivatives of the denoised sensor response curves can also be obtained from the denoised sensor response curve. The feature vectors can be compressed by PCA. These features are often normalized to [-1, 1] before being input into the classifiers.

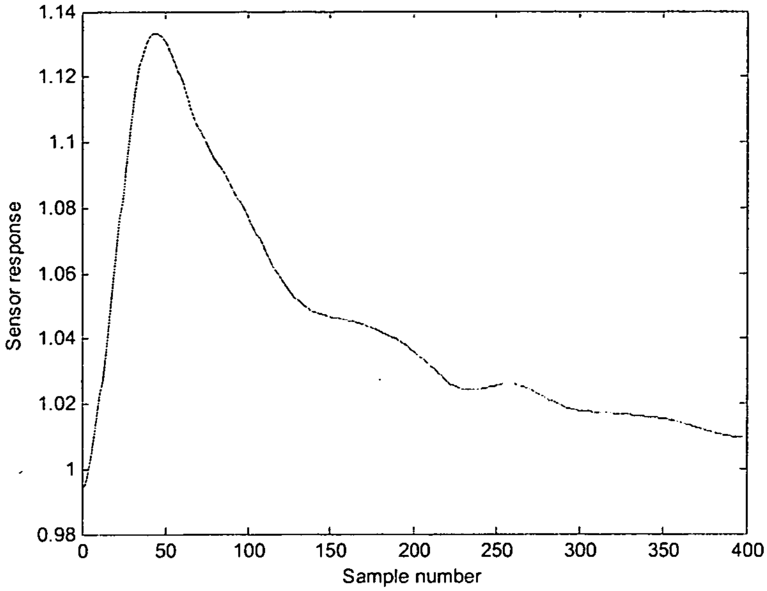


Figure 2. Sensor response after wavelet denoising.

IV. DS FUSION OF MULTIPLE CLASSIFIERS

A. Introduction of individual classifiers

A.1 KNN

KNN is a powerful technique of approximating the optimal Bayesian classifier [19]. Besides its simplicity advantage, it can generate good highly nonlinear classification results. The KNN classifier searches for the K nearest neighbors of the input sample among a set of training samples. Let C denote the cluster number, and $K_1,\cdots,K_C$ denote the numbers of the nearest neighbors for the C clusters, respectively, then the fuzzy classification result is given by

$$d_i = K_i / K, \quad i=1,\cdots,C. \tag{7}$$

A.2 Parzen classifier

Parzen classifier is a popular kernel-type method of approximating the optimal Bayesian classifier. It uses the estimation of a density function to design a classifier. Gaussian kernel functions are used in our experiments. The Gaussian kernel function requires the estimation of a covariance matrix through the training data sets. The detailed Parzen method can be found in [19, 25].

A.3 MLP

MLP is a popular neural network classifier in the e-nose community. The detailed structure and training algorithms of MLP can be found in [15]. A three-layer perceptron structure is chosen. The number of the first-layer neurons is L which is equal to the dimension of the sensor measurements while the number of output neurons is C which is the same as the number of vapor classes. The number of the hidden neurons is determined by trial and is in the range of 10 to 30. The MLP is trained by the back-propagation learning algorithm. The outputs of the third-layer neurons represent the classification results of the datum input into the first-layer. If a training datum belongs to the cth class, the output of the cth neuron in the third-layer is 1, and the outputs of all other neurons in the third-layer are zero. For each testing datum, the outputs of the third- layer neurons should be normalized in order to guarantee that the sum of the classification results is 1.

A.4 SVM

SVM is known as a very good tool for pattern classification [3, 17, 18]. The classification problem is changed to a quadratic programming with linear constrains when training the SVM. The advantages of SVM include: (1) Unlike MLP which is sometimes trapped in the local minima, SVM can obtain the global optimization solution because it is a quadratic learning algorithm; (2) There is no over-training problem; (3) The classification performance of SVM is often better than other classifiers.

For the SVM classifier, the number of the input neurons is equal to L and the number of the hidden neurons is set equal to the number of the number of the training samples. There is only one neuron in the third-layer. To apply SVM for multiple classes, one frequently used method is the one-against-others decomposition. It works by constructing a SVM for each class at first to separate that class from all other classes and then uses an expert to arbitrate between each SVM output in order to produce the final decisions. The SVM program used in our study is called LIBSVM which is produced by Chih-Jen Lin's group at the National Taiwan University.

The data sets are divided into two parts: training data sets and testing data sets. We use different features of the training data sets to train the classifiers. When the training of multiple classifiers is finished, the test data is used to evaluate the classifier performance. The classification results of these individual classifiers are then fused by using the DS technique.

B. DS fusion of multiple classifiers

DS theory of evidence is a tool for representing and combining evidence, which is considered to be a generalization of the Bayesian theory [23, 24]. Combination of evidence in the DS theory is implemented using Dempster's rule of combination, assuming that the evidences are independent. Rather than representing the probability of a hypothesis *H* by a single value *P*(*H*), DS uses two measures, i.e., *Plausibility*(*H*) and *Belief*(*H*), to describe the imprecision and uncertainty. These two measures satisfy the following condition

$$Belief(H) \le P(H) \le Plausibility(H).$$

Fusion of multiple classifiers are based on the DS combination rule defined by

$$m_{1\oplus 2}(H) = \frac{\sum_{B\cap C=H} m_1(B)m_2(C)}{1-\sum_{B\cap C=\Phi} m_1(B)m_2(C)}. \tag{8}$$

The output of the DS fusion is the following interval of belief:

$$\begin{aligned} &Belief(H) = m_{1\oplus 2}(H), \\ &Plausibility(H) = 1 - Belief(\bar{H}), \\ &Belief\ interval = [Belief(H),\ Plausibity(H)], \end{aligned} \tag{9}$$

where $\bar{H}$ is the complementary hypothesis of *H*: $H \cup \bar{H} = whole\ set\ of\ hypotheses$, and $H \cap \bar{H} = 0$.

The upper probability function *Plausibility*(*H*) can also be computed by (8) and (9). The popular decision-making criteria for DS fusion include: (1) maximum of *Belief*(*H*), (2) maximum of *Plausibility*(*H*), (3) other rules such as maximum(*Plausibility*(*H*)+*Belief*(*H*)). Since maximum of *Belief*(*H*) is the most widely used criterion, it is employed here. In addition, the performances of these criteria are quite similar for this vapor classification problem, but the computation complexity of using *Belief*(*H*) is the lowest. If the belief interval information is required as well, *Plausibility*(*H*) can also be obtained by (8) and (9).

The detailed DS fusion method is described as follows. Let *P* denote the number of classifiers. The classification of the *i*th test datum by the *j*th classifier is denoted by $d_{i,j} = [d_{i,j,1}, d_{i,j,2}, \cdots, d_{i,j,C}]^T$, where $d_{i,j,c} \in [0,1]$, $\sum_{c=1}^{C} d_{i,j,c} = 1$. For the *i*th test datum, the combined probability of the *c*th class for the *P* classifiers is given by

$$d_{i,c}^{combined} = \sum_{k=1}^{P} \sum_{j=k+1}^{P} d_{i,k,c} d_{i,j,c}, \quad c = 1, \cdots, C, \quad i = 1, \cdots, N. \tag{10}$$

The classification of the *i*th test datum can be expressed as

$$D_i^{combined} = \left[d_{i,1}^{combined}, d_{i,2}^{combined}, \cdots, d_{i,C}^{combined}\right] \Big/ \sum_{c=1}^{C} d_{i,c}^{combined} \tag{11}$$

where $D_i^{combined}$ is the belief probability *Belief*(*H*) in (9).

The correct classification rates are calculated by using the hard classification results. The *i*th test datum is classified to the *c*th class if its combined probability is the largest, i.e.,

$$c = \arg\max_{m} d_{i,m}^{combined}, \quad m = 1, \cdots, C. \tag{12}$$

V. EXPERIMENTAL RESULTS

A. Raw data sets

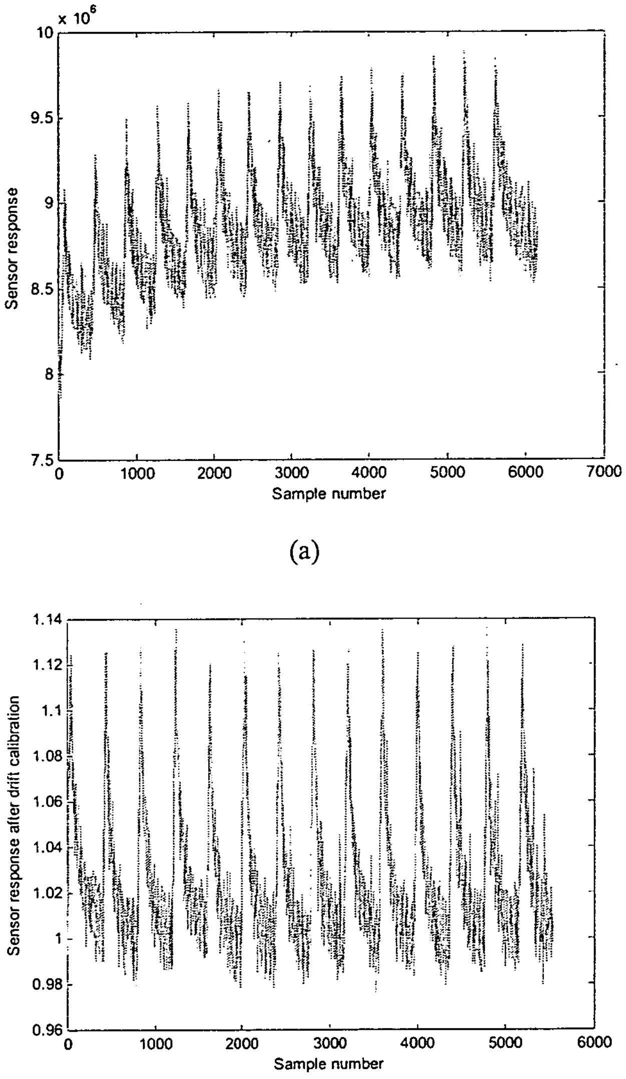
NASA provided us four raw experimental data sets named Air2o, Air3f, Kam15o, and Kam15f, respectively. Metal oxide semiconductor (MOS) sensors were used in the experiments. Air2o and Air3f were collected by i-Pen2 and i-Pen3 sensor arrays manufactured by Airsense Inc. Kam15o and Kam15f were collected by Kamina sensor arrays manufactured by Karlsruhe Research Centers. Each raw data set was collected under three humidity conditions ("low", "middle", "high") and three different concentrations ("low", "middle", "high"). Detailed experimental parameters are shown in Table 1.

Table 1. Parameters of the four raw data sets

Raw data sets	Air2o	Air3f	Kam15o	Kam15f
Sensor type	i-Pen2	i-Pen3	Kamina	Kamina
Sensor number	10	10	38	38
Sampling time (second)	1	1	3	3
Total sample number	1077	261	647	397
Cluster number	8	3	2	4
Relative humidity	low, middle, high	low, middle, high	low, middle, high	low, middle, high
Vapor concentration	low, middle, high	low, middle, high	low, middle, high	low, middle, high

B. Sensor drift calibration

Up to now, it is impossible to fabricate chemical sensors without drift. Without drift calibration, the features of different clusters will overlap which will result in low correct classification rates. Baseline manipulation is used to reduce sensor drift. One experimental result for drift calibration is presented here. The original response of the 4th sensor for the data set named kt4lohi.dat in Kam15f is shown in Figure 3(a). kt4lohi.dat was collected by the Kamina sensor array under low humidity and high concentration. There are 15 sniffs included. The initial value of the first sniff is about 7.8×10^6. The baseline is drifted to about 8.7×10^6 for the 15th sniff. The sensor response after baseline manipulation is shown in Figure 3(b). The sensor drift is effectively reduced.



(a)

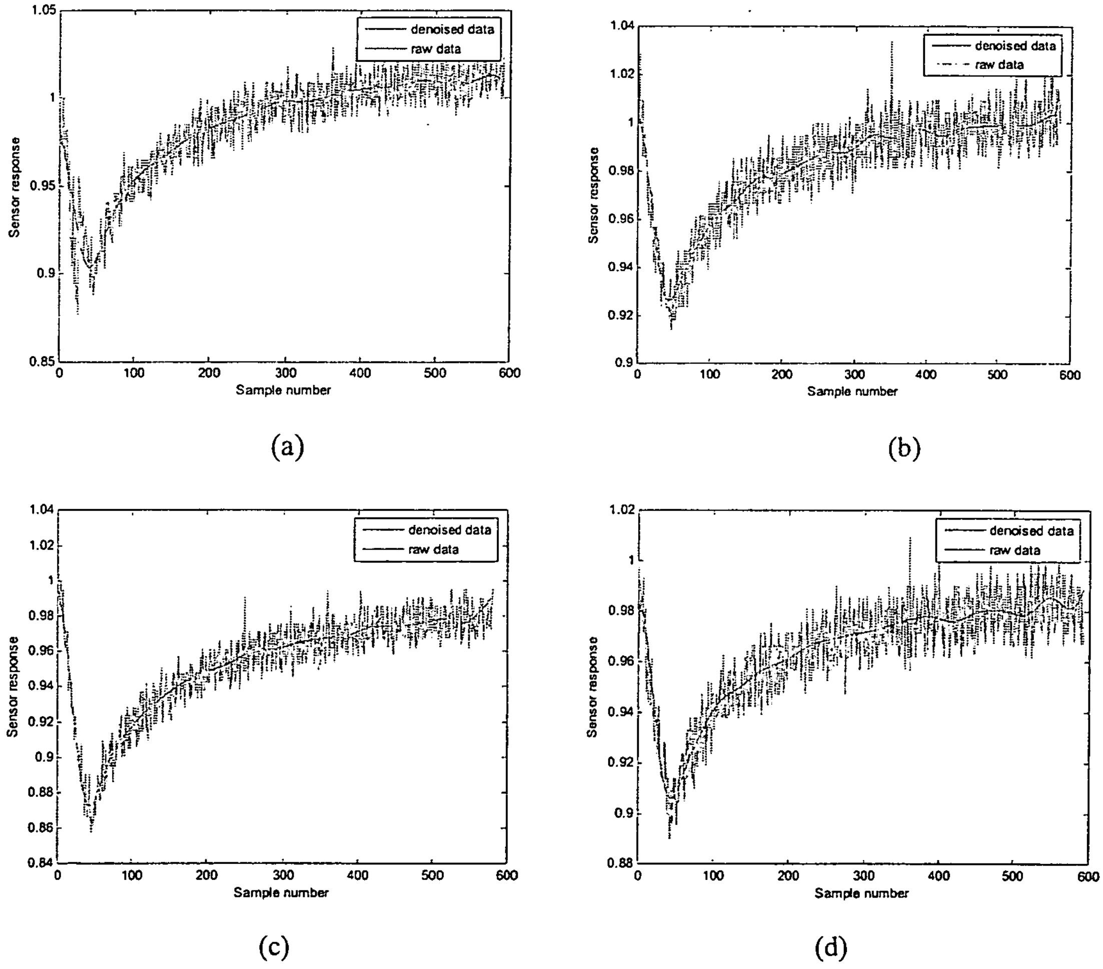
(b)

Figure 3. Baseline manipulation for the 4th sensor of kt4lohi.dat in Kam15f:

(a) original; (b) after baseline manipulation.

C. Wavelet denoising

Denoising is required for the noisy Kamina data and the low concentration Airsense fuel data, but the organic Airsense data does not need any filtering. We apply the wavelet denoising to all of the noisy raw vapor data sets, but only show two examples here because of the space limit. Symmetrical wavelet and soft threshold are used, and there are totally 5–scale wavelet decomposition. Kam15o has an array of 38 gas sensors. For each data set in Kam15o, there are multiple sniffs, and each of these sniffs must be denoised. Figures 4(a) to 4(d) show the denoised results of the first to 4th sniffs of the 20th sensor for kt1hihi.dat in Kam15o. Figures 5(a) to 5(d) show the denoised results of the first to 4th sniffs of the 10th sensor for kt3hihi2.dat in Kam15f. It can be seen that both random noise and outliers are filtered successfully by the wavelet denoising. Wavelet denoising is compared with medium filter in Figure 6. Apparently, the sensor response curve denoised by the wavelet approach is much smoother than that of the median filter.



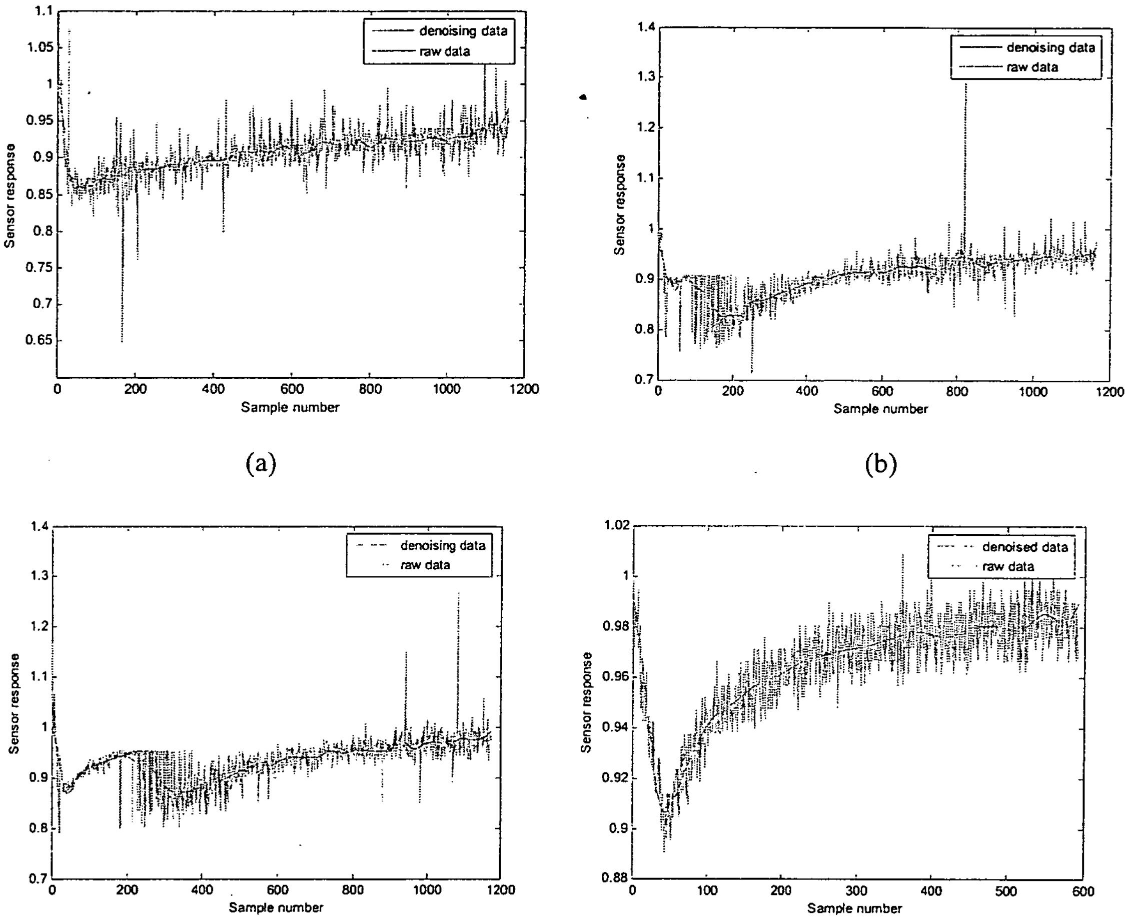
(a)

(b)

(c)

(d)

Figure 4. Wavelet denoising of the first to 4th sniffs for the 20th sensor of kt1hihi.dat in Kam15o: (a) first sniff; (b) second sniff; (c) third sniff; (d) fourth sniff.



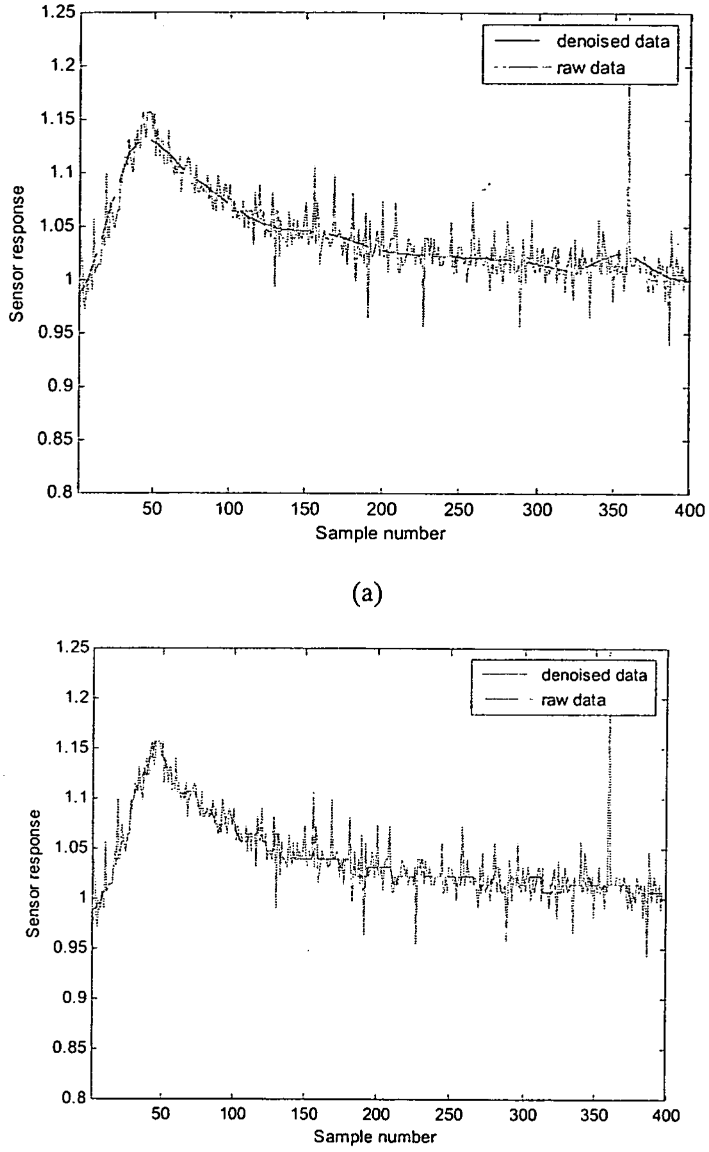
(a)

(b)

(c)

(d)

Figure 5. Wavelet denoising of the first to 4th sniffs for the 10th sensor of kt3hihi2.dat in Kam15f (a) first sniff; (b) second sniff; (c) third sniff; (d) fourth sniff.



(a)

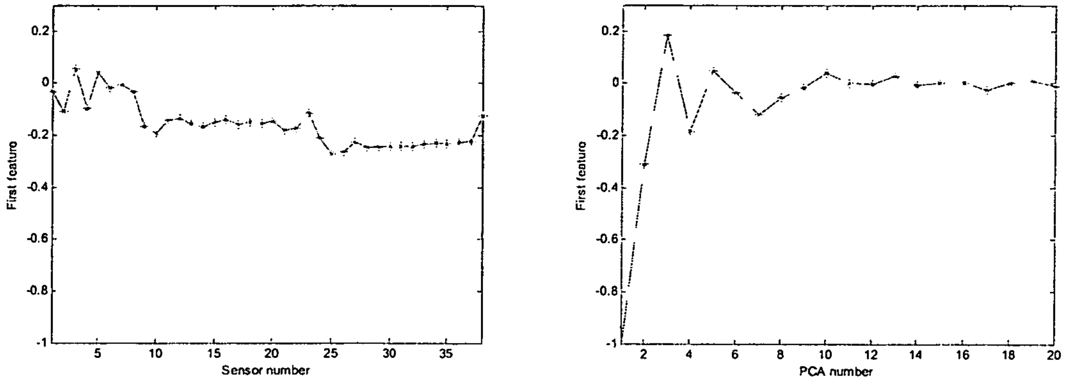
(b)

Figure 6. Comparison of wavelet denoising and median filter: (a) wavelet denoising; (b) median filter.

D. Feature extraction

In our experiments, the transient-state features are extracted from the denoised sensor response curves including: (1) sensor value at the 350th sample, (2) sensor value at the 100th sample, (3) means on the sample interval [101, 120], (4) means on the sample interval [121, 140], (5) means on the sample interval [141, 160], (6) means on the sample interval [161, 180]. All the above features extracted from the sensor desorption response curves are used for classification. The first feature for the 10th sniff of kt3hihi2.dat in Kam15f is shown in Figures 7(a) for illustration. Each of these features is used to train an individual classifier, and the classification results of multiple classifiers are then fused by the DS technique.

When the features in Figure 7(a) are used directly for vapor classification, the data dimension becomes high. There are also some feature redundancies. Principal components analysis (PCA) is an effective method to project the features into a lower-dimension space. Figure 7(b) shows the PCA projection of Figure 7(a) where each feature space with 38 dimensions is projected onto a 20-dimensional space.



(a) (b)

Figure 7. The first feature for the 10th sniff of kt3hihi2.dat in Kam15f (a) original; (b) after PCA.

E. Classification results

The transient-state features after PCA transformations are fed to the multiple classifiers. Half of the data set is used for training and the whole data set is used for testing. Six features are extracted as described in the above section. Each feature is used to train one of the six classifiers.

In the first classification experiment, we use the six features to train six individual SVM classifiers. There are two parameters which are needed to be chosen for SVM: r_1 and r_2. For different data sets, the optimal r_1 and r_2 may be different. $r_1 = 1$ and $r_2 = 100$ are chosen by trials which can result in good classification results for all data sets. Half of the data sets are selected randomly for each SVM classifier. Table 2 shows that the classification performance can be improved effectively by DS fusion.

Table 2. Classification results of fusing NN classifiers based on different features

Raw data	Individual classifiers						Fusion
	Feature 1 (SVM1)	Feature 2 (SVM1)	Feature 3 (SVM3)	Feature 4 (SVM4)	Feature 5 (SVM5)	Feature 6 (SVM6)	
Air2o	0.992	0.987	0.992	0.996	0.998	0.997	0.999
Air3f	0.971	0.965	0.944	0.954	0.970	0.972	0.995
Kam15o	0.972	0.983	0.970	0.985	0.972	0.975	0.990
Kam15f	0.971	0.836	0.900	0.960	0.971	0.975	0.986

In the second classification experiment, we use the six features to train six individual KNN classifiers. Table 3 shows that the classification performance can also be improved effectively by DS fusion. Compare Table 2 with Table 3, we can see that the performance of the individual SVM classifiers and fusion of SVMs are better than the individual KNN classifier and fusion of KNNs.

Table 3. Classification results of fusing multiple KNN classifiers based on different features

Raw data	Individual classifiers						Fusion
	Feature 1 (KNN1)	Feature 2 (KNN2)	Feature 3 (KNN3)	Feature 4 (KNN4)	Feature 5 (KNN5)	Feature 6 (KNN6)	
Air2o	0.957	0.953	0.973	0.982	0.987	0.989	0.995
Air3f	0.964	0.920	0.921	0.940	0.948	0.958	0.983
Kam15o	0.951	0.972	0.952	0.974	0.972	0.972	0.984
Kam15f	0.941	0.745	0.837	0.929	0.963	0.974	0.978

In the third classification experiment, we use the six features to train six different classifiers: MLP1, MLP2, KNN, Parzen classifier, SVM1, and SVM2. The fused classification results are better than the six individual classifiers. Compare Table 4 and Table 2, we can see that the performance of fusing six SVMs are almost the same as that of fusing different classifiers.

Table 4. Classification results of fusing NN and non-NN classifiers based on different features

Raw data	Individual classifiers						Fusion
	Feature 1 (MLP1)	Feature 2 (MLP2)	Feature 3 (KNN)	Feature 4 (Parzen)	Feature 5 (SVM1)	Feature 6 (SVM2)	
Air2o	0.983	0.968	0.973	0.965	0.998	0.997	0.999
Air3f	0.982	0.941	0.921	0.854	0.970	0.972	0.995
Kam15o	0.970	0.973	0.952	0.960	0.972	0.975	0.988
Kam15f	0.962	0.817	0.837	0.851	0.971	0.975	0.989

VI. CONCLUSIONS

In this paper, we investigated the data preprocessing and classification techniques for e-nose vapor classifications. The raw sensor measurements are first divided by the baseline (average of the first few samples) of each sensor to reduce the effect of sensor drifts. Then wavelet denoising is proposed to remove random noise and outliers in the sensor measurements. Compared with the median filter method, although both methods can remove outliers successfully, the sensor response curve after wavelet denoising is much smoother. Various transient-state features are then extracted from the wavelet-denoised sensor response curves. Multiple classifiers (MLP, SVM, KNN, and Parzen classifier) are applied to these features for vapor classification and the classification results are combined using the Dempster-Shafer fusion method. The DS method is shown to have a higher correct classification rate compared to the individual classifiers.

ACKNOWLEDGEMENT

We would like to thank Rebecca C. Young for many valuable inputs and her helps in conducting the electronic nose experiment.

REFERENCES

[1] J. W. Gardner and P. N. Bartlett, *Electronic Noses: Principles and Applications*. Oxford: Oxford University Press, 1999.

[2] R. C. Young, W.J. Buttner, B. R. Linnell and R. Ramesham, "Electronic nose for space program applications," *Sensors and Actuator B: Chemical*, vol. 93, pp. 7-16, 2003.

[3] T. Qian, R. Xu, C. Kwan, B. Linnell, and R. Young, "Toxic vapor classification and concentration estimation for space shuttle and international space station," *Proceedings of International Symposium on Neural Networks Part I*, Dalian, China, pp. 543-551, Aug. 2004.

[4] M. A. Ryan, H. Zhou, M. G. Buehler, K. S. Manatt, V. S. Mowrey, S. P. Jackson, A. K. Kisor, A. V. Shevade, and M. L. Homer, "Monitoring space shuttle air quality using the jet propulsion laboratory electronic nose," *IEEE Sensors Journal*, vol. 4, no. 3, pp. 337-347, June 2004.

[5] E. L. Hines, E. Llobet, and J. W. Gardner, "Electronic noses: a review of signal processing techniques," *IEE Proc. –Circuits Devices Syst.*, vol. 146, no. 6, pp. 297-310, Dec. 1999.

[6] H. Nagle, R. Gutierrez-Osuna, and S. Schiffman, "The how and why of electronic noses", *IEEE Spectrum*, pp. 22-34, Sept. 1998.

[7] R. Gutierrez and H. T. Nagle, "A method for evaluating data-preprocessing techniques for odor classification with an array of gas sensors," *IEEE Trans. Systems, Man, and Cybernetics-Part B: Cybernetics*, vol. 29, no. 5, pp. 626-632, Oct. 1999.

[8] O. Farooq and S. Datta, "Wavelet-based denoising for robust feature extraction for speech recognition," *Electronics Letters*, vol. 39, no. 1, pp. 163-165, Jan. 2003.

[9] L. Shark and C. Yu, "Denoising by optimal fuzzy thresholding in wavelet domain," *Electronics Letters*, vol. 36, no. 6, pp. 581-582, March 2000.

[10] D. M. Wilson and S. P. DeWeerth, "Odor discrimination using steady-state and transient characteristics of tin-oxide sensors," *Sensors and Actuators B: Chemical*: vol. 28, pp. 123-128, 1995.

[11] S. Roussel, G. Forsberg, V. Steinmetz, P. Grenier, and V. Bellon-Maurel, "Optimization of electronic nose measurements. Part I: methodology of output feature selection," *Journal of Food Engineering*, vol. 37, pp. 207-222, 1998.

[12] L. Carmel, S. Levy, D. Lancet, and D. Harel, "A feature extraction method for chemical sensors in electronic nose," *Sensors and Actuators B: Chemical*, vol. 93, pp. 67-76, 2003.

[13] R. Gutierrrez-Osuna, "Pattern analysis for machine olfaction: a review," *IEEE Sensors Journal*, vol. 2, no. 3, pp. 189-202, June 2002.

[14] A. Perera, T. Sundic, A. Pardo, R. Gutierrez-Osuna, and S. Marco, "A portable electronic nose based on embedded PC technology and GNU/Linux: hardware, software and applications," *IEEE Sensors Journal*, vol. 2, no. 3, pp. 235-246, June 2002.

[15] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, pp. 359-366, 1989.

[16] M. Pardo and G. Sberveglieri, "Remarks on the use of multilayer perceptions for the analysis of chemical sensor array data, " *IEEE Sensors Journal*, vol. 4, no. 3, pp. 355-363, June 2004.

[17] K. Brudzewski, S. Osowski, and T. Markiewicz, "Classification of milk by means of an electronic nose and SVM neural networks," *Sensors and Actuators B: Chemical*, vol. 98, pp. 291-298, 2004.

[18] K. I. Kim, K. Jung, S. H. Park, and H. J. Kim, "Support vector machines for texture classification," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 11, pp. 1542-1550, Nov. 2002.

[19] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. John Wiley & Sons, 2000.

[20] A. Verikas, A. Lipnickas, K. Malmqvist, M. Bacauskiene, and A. Gelzinis, "Soft combination of neural classifiers: a comparative study," *Pattern Recognition Letters*, vol. 20, pp. 20, pp. 429-444, 1999.

[21] Y. Lu and F. Yamaoka, "Fuzzy integration of classification results," *Pattern Recognition*, vol. 30, no. 11, pp. 1877-1891, 1997.

[22] L. Bruzzone, R. Cossu, and G. Vernazza, "Detection of land-cover transitions by combining multidate classifiers," *Pattern Recognition Letters*, vol. 25, pp. 1491-1500, 2004.

[23] H. Leung and J. Wu, "Bayesian and Dempster-Shafer target identification for radar surveillance," *IEEE Trans. Aerospace and Electronics*, vol. 36, no. 2, pp. 432-447, April 2000.

[24] D. Fixsen and R. Mahler, "The modified Dempster-Shafer approach to classification," *IEEE Trans. Systems, Man, and Cybernetics*, vol. 27, no. 1, pp. 96-104, Jan. 1997.

[25] Y. Hamamoto, Y. Fujimoto, and S. Tomita, "On the estimation of a covariance matrix in designing Parzen classifiers," *Pattern Recognition*, vol. 29, no. 10, pp. 1751-1759, 1996.

- Expensive methods require large step sizes to keep computation time low, and at large step sizes, close encounters of bodies look like singularities to the integrator. However, since the bodies will move around each other, small step sizes will show that all functions are smooth, thus fulfilling the key requirement for predictor-corrector methods to be competitive.

1.3.1 nbody_sh1

We used the program nbody_sh1 by Hut and Makino (2002), which is documented line by line in great detail by Hut and Makino (2003b). Below, we will present the algorithm used to perform the next integration step at time t.

nbody_sh1 uses adaptive step size control. It is called with a *step size control parameter* d_{control} via the -d option. By default, $d_{\text{control}} = 0.03$, and one should use $d_{\text{control}} \ll 1$.

At time t, nbody_sh1 has an estimate of the time increment until "something drastic happens", such as a collision of two particles. This time increment is called θ_{coll} and is determined in the course of the step at time t, see below. First, the step size θ at time t is determined as

$$\theta := d_{\text{control}} \cdot \theta_{\text{coll}}.$$

Now the predictor step is performed to predict the positions and velocities at time $t+\theta$. This is nothing but a simple Taylor expansion:

$$\begin{aligned} x_i^{\text{pred}}(t+\theta) &:= x_i(t) + \dot{x}_i(t)\theta + \frac{1}{2}\ddot{x}_i(t)\theta^2 + \frac{1}{6}\dddot{x}_i(t)\theta^3 && \text{for } 1 \le i \le N \\ \dot{x}_i^{\text{pred}}(t+\theta) &:= \dot{x}_i(t) + \ddot{x}_i(t)\theta + \frac{1}{2}\dddot{x}_i(t)\theta^2 && \text{for } 1 \le i \le N. \end{aligned}$$

Next, the accelerations and jerks are updated based on the predicted positions and velocities:

$$\begin{aligned} r_{ji}^{\text{pred}} &:= x_j^{\text{pred}}(t+\theta) - x_i^{\text{pred}}(t+\theta) \\ v_{ji}^{\text{pred}} &:= \dot{x}_j^{\text{pred}}(t+\theta) - \dot{x}_i^{\text{pred}}(t+\theta) \\ \ddot{x}_i^{\text{pred}}(t+\theta) &:= \sum_{j \neq i} \frac{m_j r_{ji}^{\text{pred}}}{\|r_{ji}^{\text{pred}}\|^3} \quad \text{for } 1 \le i \le N \\ \dddot{x}_i^{\text{pred}}(t+\theta) &:= \sum_{j \neq i} \frac{m_j}{\|r_{ji}^{\text{pred}}\|^3} \cdot \left(v_{ji}^{\text{pred}} - \frac{3\langle r_{ji}^{\text{pred}}, v_{ji}^{\text{pred}} \rangle r_{ji}^{\text{pred}}}{\|r_{ji}^{\text{pred}}\|^2} \right) \quad \text{for } 1 \le i \le N. \end{aligned} \tag{3}$$

The acceleration value results directly from Newton's Law, while the jerk formula is a simple differentiation of the acceleration. Now the collision time variable is reset:

$$\theta_{\text{coll}} := \min \left\{ \frac{\|r_{ji}^{\text{pred}}\|}{\|v_{ji}^{\text{pred}}\|}, \sqrt{\frac{\|r_{ji}^{\text{pred}}\|^3}{m_i + m_j}} \,\middle|\, 1 \le i < j \le N \right\}.$$

Here, the first entry models a linear unaccelerated motion of bodies i and j, while the second term is an estimate for the collision time in a head-on free fall, cf. Hut and Makino (2003a), sections 7.4 and 7.5.

Finally, a corrector step is performed:

$$\begin{aligned}\dot{x}_i(t+\theta) &:= \dot{x}_i(t) + \frac{1}{2}(\ddot{x}_i(t) + \ddot{x}_i^{\text{pred}}(t+\theta))\theta + \frac{1}{12}(\dddot{x}_i(t) - \dddot{x}_i^{\text{pred}}(t+\theta))\theta^2 \\ x_i(t+\theta) &:= x_i(t) + \frac{1}{2}(\dot{x}_i(t) + \dot{x}_i(t+\theta))\theta + \frac{1}{12}(\ddot{x}_i(t) - \ddot{x}_i^{\text{pred}}(t+\theta))\theta^2.\end{aligned}$$

Note that $\dot{x}_i$ is calculated first, since this value is used in the calculation of x_i. This again is simply a Taylor expansion of the position to a higher order than the predictor, although this is not obvious from the formula. In fact, the position is developed to the fifth, the velocity to the fourth, the acceleration to the third and the jerk to the second order. Then, the derivatives $\frac{d^4x_i}{dt^4}$ and $\frac{d^5x_i}{dt^5}$ are expressed in terms of lower derivatives and re-substituted into the equations for the position and the velocity. Thus, a Taylor expansion to the fifth order can be written using only the first three derivatives, and the resulting predictor-corrector scheme is a fourth-order ODE solver.

The same idea could in principle be applied to construct a higher-order solver. However, while $\dddot{x}_i$ can be calculated by (3) in a single pass over all particles, $\frac{d^4x_i}{dt^4}$ would already require a double pass, severely increasing the algorithm's complexity (cf. equation 6.5 in Hut and Makino 2003a).

Details on the derivation can be found in Hut and Makino (2003a), sections 6.1 and 6.3.

1.3.2 Maple

As an alternative, we considered the built-in numerical integrators of the computer algebra system Maple. The different integrators are compared in the next section.

1.4 Comparison of Different Integrators

To compare different integrators, we considered the following masses and initial positions and velocities:

$$\begin{aligned}
m_1 &= 1, & x_1(0) &= \begin{pmatrix}0\\0\\0\end{pmatrix}, & \dot{x}_1(0) &= \begin{pmatrix}0\\0\\0\end{pmatrix} \\
m_2 &= 0.1, & x_2(0) &= \begin{pmatrix}0\\0.5\\0\end{pmatrix}, & \dot{x}_1(0) &= \begin{pmatrix}0\\0\\0.5\end{pmatrix} \\
m_3 &= 1, & x_3(0) &= \begin{pmatrix}10\\1\\0\end{pmatrix}, & \dot{x}_3(0) &= \begin{pmatrix}-1\\0\\0\end{pmatrix} \\
m_4 &= 0.13, & x_4(0) &= \begin{pmatrix}10\\1.5\\0\end{pmatrix}, & \dot{x}_4(0) &= \begin{pmatrix}-1\\0\\0.5\end{pmatrix}.
\end{aligned}$$