Atmospheric Turbulent Dispersion Modeling Methods using Machine learning Tools

Pierre Lauret, Frederic Heymes, Laurent Aprin, Anne Johannet, Gilles Dusserre, Emmanuel Lapebie, Antoine Osmont

Atmospheric Turbulent Dispersion Modeling Methods using Machine learning Tools

Pierre Lauret[a], Frédéric Heymes[a], Laurent Aprin[a], Anne Johannet[a], Gilles Dusserre[a], Emmanuel Lapébie[b], Antoine Osmont[b]

[a]Laboratoire de Génie de l'Environnement Industriel (LGEI), Ecole des Mines d'Alès, Alès, France
[b]CEA, DAM, GRAMAT, F-46500 Gramat, France
pierre.lauret@mines-ales.fr

Assessment of likely consequences of a potential accident is a major concern for loss prevention and safety promotion in process industry. Loss of confinement on a storage tank, vessel or piping on industrial sites could imply atmospheric dispersion of toxic or flammable gases. Gas dispersion forecasting is a difficult task since turbulence modeling at large scale involves expensive calculations. Therefore simpler models are used but remain inaccurate especially when turbulence is heterogeneous. The present work aims to study if Artificial Neural Networks coupled with Cellular Automata could be relevant to overcome these gaps. Two methods are reviewed and compared. An example database was designed from RANS k-ε CFD model. Both methods were then applied. Their efficiencies are compared and discussed in terms of quality, real-time applicability and real-life plausibility.

1. Introduction

Major industrial accidents are generally characterized by important mass release and huge consequences. Bhopal disaster sadly illustrates the impact of dispersion of a toxic gas such as methyl isocyanate (MIC) inside and beyond an industrial site. Sharan et al. (1997) showed the importance of both atmospheric conditions and topography. Indeed, local characteristics such as presence of lakes near Bhopal had a major contribution in transporting the MIC into the city area. Flammable gases emissions are also of interest in the studied topic because of the UVCE possibility. The Viareggio accident is a tragic example of LPG dispersion. Brambilla et al. (2009) reported the requirement for taking account of congested environments in such cases. Indeed, effects of the terrain and obstacles play a major role in gas dispersion, due to the eddies, wakes, stagnation and recirculation points they can introduce (Pontiggia et al., 2012).
In this context, it is important for decision maker to anticipate crisis situations by modeling several realistic scenarios. Need for operational tools gives important guidelines to create such a model: fast, accurate, reliable, easy-to-use and accounting well for obstacles in the simulation. Models presented in this paper have been designed to answer these criteria.

2. Atmospheric dispersion modeling

Models predicting gas dispersion are abundant. Computation of velocity field and gas dispersion of the commodity released is generally done separately. Atmospheric dispersion models use velocity field as an input. The main strategies are described hereafter.

2.1 Flow velocity field determination

Computational Fluid Dynamics (CFD) first aim is to describe a specific configuration of the flow. It solves the Navier-Stokes equations using different kinds of refinement depending of the goal to achieve. Reynolds decomposition is usually used. It divides the velocity in mean and fluctuation components. Turbulence models are used in order to close the equations system. Computation time, numerical convergence and required expertise are the drawbacks of such fineness. Experiments are often needed in

order to calibrate coefficients. CFD models require solving continuity, momentum and energy equations which is time consuming.
Semi-empirical models use Monin-Obukhov theory to determine vertical velocity profile. These profiles are dependent of the atmospheric stability state. Additionally, Hosker (1984) gives relations derived from experiments and theory to model flow behind obstacles. These methods are used to avoid long computing time and can be useful in first approach.
While semi-empirical models can provide quick results, CFD ones preserve consistent results with little error margin. In any cases, it is crucial to confront models and experimentations (Meroney, 2004).

2.2 Turbulent dispersion

Turbulent diffusion is involved through every turbulent dispersion model with a specific form. Gaussian models result from a simplified analytical solution of the Advection Diffusion Equation (ADE) (1). Indeed, some assumptions are made on the wind field: it is considered as constant in space and time, turbulence is considered isotropic and stationary. Standard deviation coefficients are determined from weather stability, and calibrated from experimental data.

$$\frac{\partial C}{\partial t} + u_i \frac{\partial C}{\partial x_i} = D_t \frac{\partial^2 C}{\partial x_i^2} + S_C \quad (1)$$

With C the concentration of a pollutant,
U_i the wind velocity in direction xi,
S_c an eventual pollutant source term,
D_t the turbulent diffusion coefficient,
$u_i \frac{\partial C}{\partial x_i}$ is the advection term, $D_t \frac{\partial^2 C}{\partial x_i^2}$ is the diffusion term.
It is also possible to solve the ADE in space and time using discrete approach. This method is used by CFD software, calculating D_t from Navier-Stokes system closure equations. This is an iterative computing, using discretization in space and time.
The last method to mention here is Lagrangian particle tracking. A wide number of particles are introduced in the numerical domain. Using advection from the previous calculated velocity field, particles are followed through the iterations. To take turbulence into account, a random controlled distribution is added to the mean velocity value. At the end of the simulations, the particle density is evaluated and linked to a pollutant concentration. Lagrangian models need expert control to adjust the correct velocity fluctuation. Summary of method cited here is schematize on Figure 1:

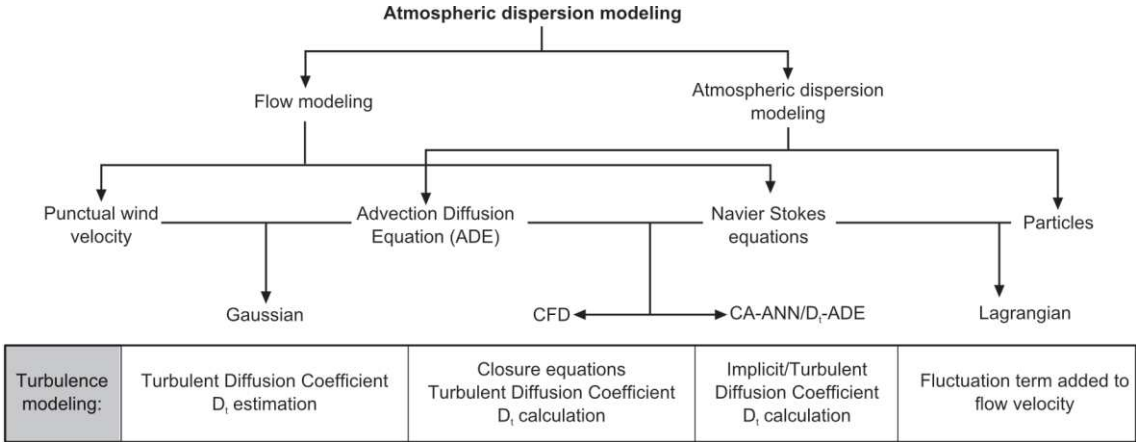

Figure 1: Flow and atmospheric dispersion modelling.

This study considers ADE resolution because no important expertise is needed once the setup is made. Two methods both using artificial neural networks are used here to take turbulence into account.

3. Numerical scheme for Advection Diffusion Equation (ADE) resolution

Solving ADE on a grid can be built using several discretization techniques such as finite differences, finite volumes, finite elements or spectral method. Finite differences are used in this work because of the setup simplicity.

Many different methods exist to discretize partial differential equations (Fletcher, 1991). It is commonly assumed to use respectively classic upwind scheme and centered scheme for advection and diffusion term:

$$\frac{C_{i,j}^{t+1}-C_{i,j}^{t}}{\Delta t}+U_{x}{}_{i,j}^{t}\frac{C_{i,j}^{t}-C_{i-1,j}^{t}}{\Delta x}+U_{y}{}_{i,j}^{t}\frac{C_{i,j}^{t}-C_{i,j-1}^{t}}{2.\Delta y}=D_{t}.\frac{C_{i+1,j}^{t}-2.C_{i,j}^{t}+C_{i-1,j}^{t}}{\Delta x^{2}}+D_{t}.\frac{C_{i,j+1}^{t}-2.C_{i,j}^{t}+C_{i,j-1}^{t}}{\Delta y^{2}} \quad (2)$$

With $C_{i,j}^{t}$ the concentration in the cell *i* (x direction), *j* (y direction) at time step *t*.

In this case, discretization of the time dependent equation is explicit. The concentration at time step t+1 depends only of the concentration at time step t.
Each numerical scheme has to be consistent and stable i.e. convergent. Stability condition of this scheme had been largely discussed (Fletcher, 1991):

$$\Delta t \leq \frac{\Delta x^{2}}{U_{x}.\Delta x+4.D_{t}+U_{y}.\Delta x} \quad (3)$$

This numerical scheme and associated limitations are used in the following.

4. Using Artificial Neural Networks to solve ADE

Artificial Neural Networks (ANNs) consist of several mathematical functions, termed as neurons, which are linked in a network. ANN are powerful as non-linear statistical data modeling tools. They are generally used when the process to model is not fully known thanks to two essential properties: first the universal approximation (Hornik et al., 1989), and second the parsimony (Barron, 1993). Thanks to these properties ANNs are able to predict efficiently future behaviors on never encountered situations. ANN can be used to forecast physical phenomenon, presenting powerful models (Ak et al., 2013). The information about the non-linear phenomenon to simulate or forecast must be provided using a database. As previously presented, ANNs act generally like a black-box: the physics cannot be extracted from the results. A neuron is a nonlinear, parameterized, bounded function. Variables are assigned to the inputs of the neuron. Output of a neuron is the result of nonlinear combination of the inputs, weighted by the parameters and using an activation function. Sigmoid s-shaped functions are generally used. A neural network is the composition of several neurons. Parameters calibration is done through application of an algorithm using the training database and designed to decrease the model error, in this work the Levenberg-Marquardt method is adopted (Hagan et al., 1994). The function realized by the ANN is continuously tested on a disjoined set of examples, namely the stop set. This last set is employed to avoid overtraining using early stopping (Sjöberg et al., 1995). Lastly, performances of the model must be measured on another set, never used during training or stopping: test or validation set.

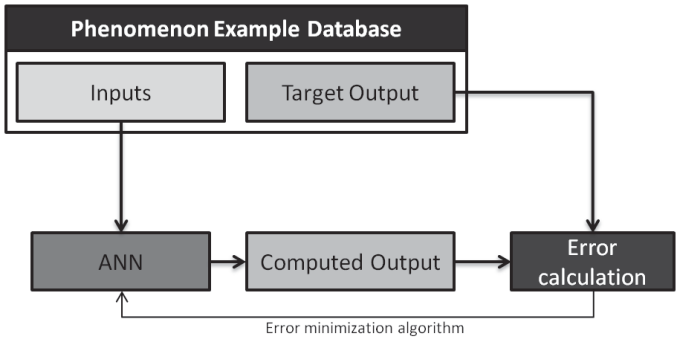

Figure 2 : ANN Training method

4.1 Artificial Neural Networks ruled (CA-ANN) Cellular Automata

This method uses both Cellular Automata (CA) and Artificial Neural Networks. Cellular automata are tools used for modeling physical phenomena in discrete space-time coordinates. The impact of local interactions on the evolution of the phenomenon is an important feature that promotes the use of CA. Itami (1994) formalized cellular automata, defining Q, as the global state of the system:

$$Q=\langle S,N,T\rangle \quad (4)$$

S represents the discrete states accepted for the cellular automaton. N represents the neighborhood of cells providing input values for the transition rules, depending on the aim of the study. The transition rule T

defines how a cell changes his state from the current time step to the next. A large number of quantitative mathematical techniques can be used such as Artificial Neural Networks, genetic algorithms, Markov Chain, Monte Carlo simulations, fuzzy logic. Cellular automata can be used in real-time to forecast phenomena (Russo et al., 2013). In the present work, ANN are used as transition rules. The inputs are directly selected from the ADE as described in following Figure 3:

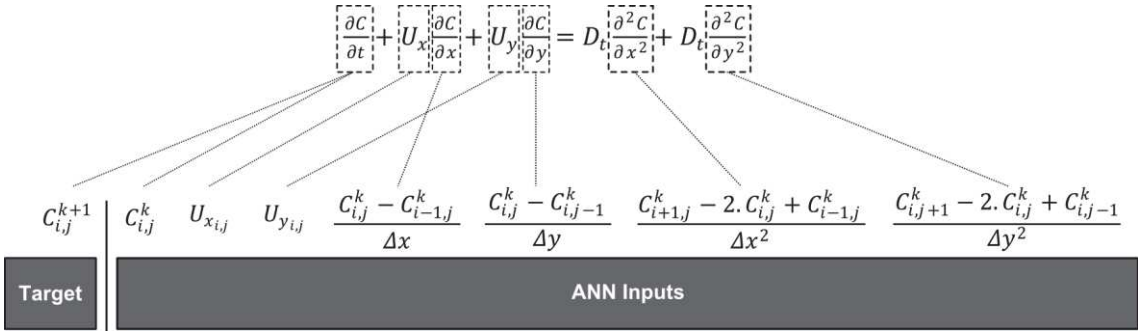

Figure 3: ANN inputs creation

An example database has to be created for the training phase of the ANN. It may be constituted of real life or small scale experiments or from CFD simulation representing reality. Database has to be representative of the atmospheric dispersion to be forecast. Data are recovered from a grid corresponding to the one used with the CA. Each input of the ANN is built from the neighborhood of the considered cell. According to 3.2, spatial and temporal steps are linked together. Data are then normalized and can be used to calibrate ANN parameters.

Once the training algorithm is achieved, the process is similar to CFD techniques. Boundary conditions of the domain to simulate are set. Cells are initialized with concentration and wind field values corresponding to test scenario. The iterative process is then applied: cells concentrations and velocities are recovered. They are combined to compute the ANN inputs, using scheme of Figure 3. Inputs data feed the ANN synchronously through the entire domain and an output concentration is provided for each cell *i,j*.Hence, these concentrations are used to compute the next time step. The Figure 4 sums up the iterative process.

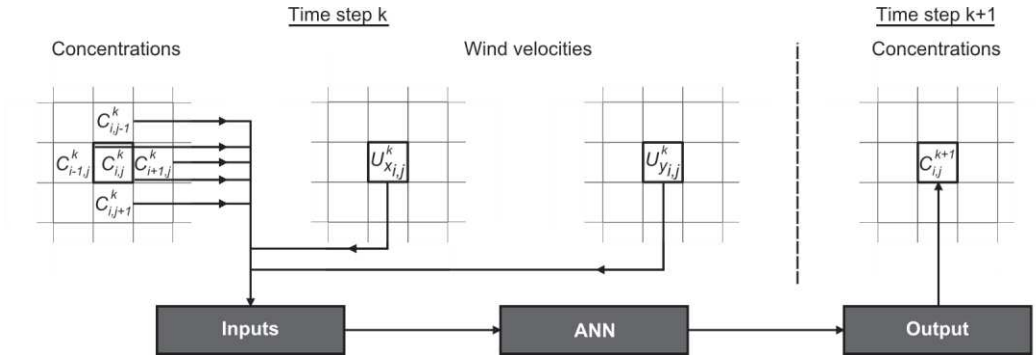

Figure 4: CA transition rule algorithm

Advantages and drawbacks of this method are discussed further.

4.2 ANN determination of the turbulent diffusion coefficient

In this work, the stationary velocity field is considered as known. ADE represents dispersion through time of an initial concentration distribution. Thereby, considering terms of equation 1, the only missing parameter to solve the ADE is the turbulent diffusion coefficient. Method presented here is to forecast this coefficient using ANN. As explained previously, turbulence over horizontal plane is not homogeneous. Especially, obstacles within the flow can critically affect the values of the turbulent diffusion coefficient. Moreover, meteorological stability conditions also affect this coefficient. Finally, altitude of the horizontal plane considered is a concern, related to the height of the atmospheric boundary layer. All these criteria have to be studied to determine the best ANN inputs. Because the ANN output is a single value, inputs have to include geometric data (Lauret et al., 2013). As in 4.1, an example database is built. Because initial hypothesis deals with passive gas, there is no need of information about atmospheric dispersion in the database. Instead, the different examples represent different flows with turbulence variations.

Once the training algorithm is done, the turbulent diffusion coefficient is computed on each cell of the grid. Then, a classical iterative process of the numerical scheme of the ADE described at 3 is applied as described in Figure 5.

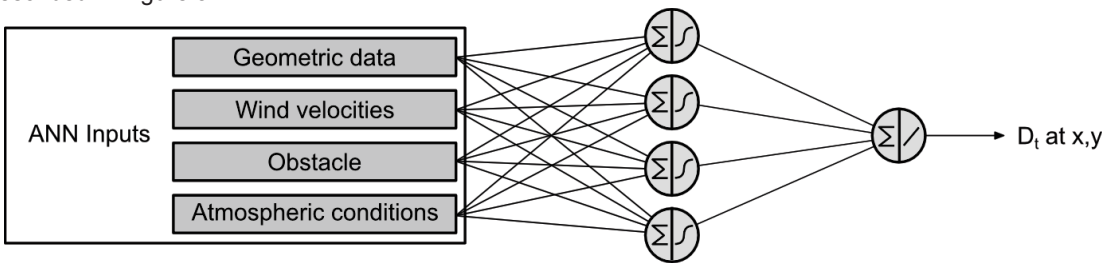

Figure 5: ANN inputs for D_t determination

5. Pros and cons of both methods

Both methods are fast because no direct/indirect solution of a linear system is required compared to CFD models. Moreover, modeling atmospheric dispersion over complex terrain can be considered: spatial heterogeneity of turbulence and obstacles can be implemented. Both methods use ANN. The CA-ANN uses it as a transition rule for the cellular automata whereas in the other hand, ANN are used to determine turbulent diffusion coefficient on a geometrical domain, once (Figure 6).

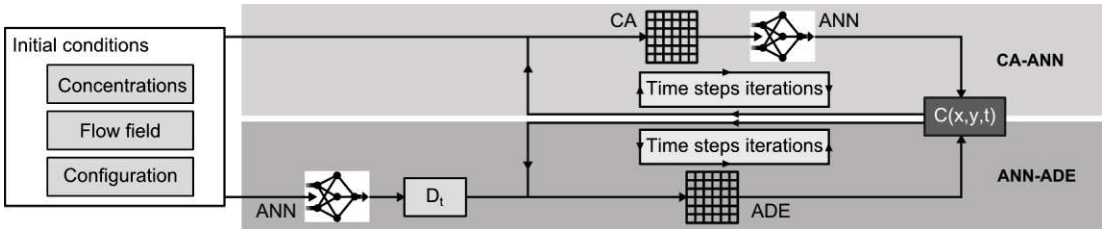

Figure 6: Sum up of both methods

CA-ANN is strictly a black box model: concentration is directly forecast. ANN-ADE can be considered as a grey box: D_t is computed from statistical training method and the phenomenon is modeled using physical ADE.

Even if the training phase is very accurate, it implies some error on the prediction. This error is made at each time step for the CA-ANN. Therefore, global error potentially increases. This configuration can move towards solution instability as can be seen on Figure 7:

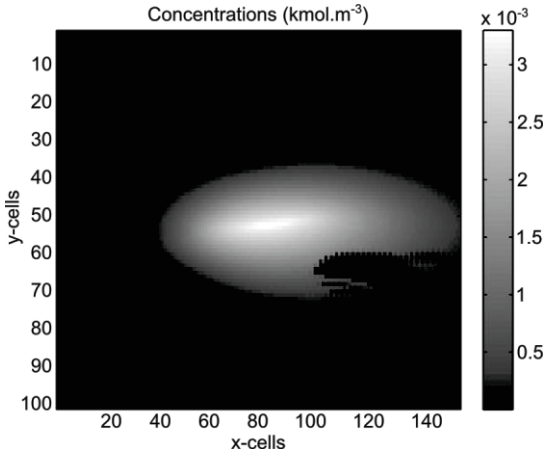

Figure 7: CA-ANN error propagation

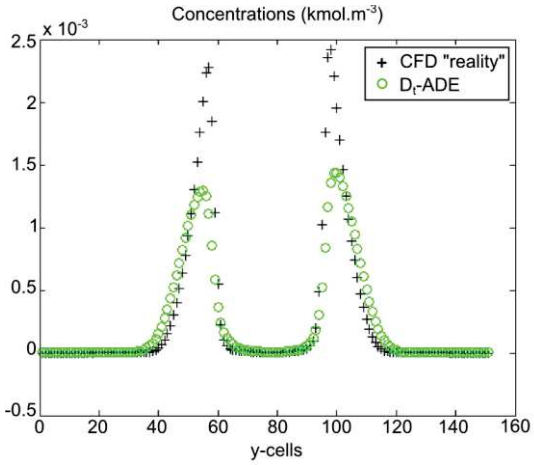

Figure 8: Diffusion error due to D_t calculation error

Dealing with CPU-time savings, using ANN rule is nearly instantaneous, because of the application of a simple explicit equation. Nevertheless, the rule is applied for each cell. As the number of cells increases,

the CPU-time increases too. For the same domain, CA-ANN appears to be approximately 1.5 times faster than CFD calculation to get final state. If each time step is required, then, the CA-ANN is 20 times faster.
In the case of D_t calculation, error is made once. It implies increase or decrease of diffusion process over advection. This kind of error is illustrated on Figure 8. Peaks of concentration are synchronized in space but not in value. In this case, the coefficient of turbulent diffusion is overestimated compared to CFD transport process, resulting in ANN-ADE peak decrease. In addition, mass balance is difficult to do for the ANN because the lack of information on where the maximum error is located. Nevertheless, a global mass balance can be done and used for basic error correction. As the objective to take into account for 3D atmospheric dispersion, improving the calculation of D_t seems easier. Modifications in the ANN inputs, especially for geometric data, are required. Then, application of finite differences techniques seen at §3 poses no major difficulties. In case of CA-ANN modification, database creation and ANN training phase mobilize considerable computer capacities.

6. Conclusions

Both methods presented here use machine training tools. CA-ANN method is a black box model, focused on forecasting directly concentration of atmospheric gas. Cellular automata are used to emulate ADE through specific domain. Transition rule is implemented by an Artificial Neural Networks. Inputs are created based on ADE parameters. General behavior is satisfactory. Main difficulty is to avoid error amplification due to small ANN training error inevitably increasing inside the domain. ANN-ADE method is a grey box model. It combines the use of ANN to determine heterogeneous Turbulent Diffusion Coefficient throughout the domain. ADE finite differences resolution is then applied. Major concerns are on the error made during the D_t ANN training phase. It can lead to change the ratio between advection and diffusion in the considered phenomenon. Nevertheless, physics of atmospheric dispersion is kept.
Future work will be focused on tridimensional atmospheric dispersion over multiple obstacles and error correction.

References

Ak, R., Li, Y., Vitelli, V., Zio, E., 2013. A Genetic Algorithm and Neural Network Technique for Predicting Wind Power under Uncertainty. Chem. Eng. Trans. 33, 925–930.

Barron, a. R., 1993. Universal approximation bounds for superpositions of a sigmoidal function. IEEE Trans. Inf. Theory 39, 930–945.

Brambilla, S., Totaro, R., Manca, D., 2009. Simulation of the LPG release, dispersion, and explosion in the Viareggio railway accident. Chem. Eng. Trans. 19, 195–200.

Fletcher, C.A.J., 1991. Computational Techniques for Fluid Dynamics, Volume 1, Fundamental and General Techniques, 2nd editio. ed. Springer Verlag, New York.

Hagan, M.T., Menhaj, M.B., 1994. Training Feedforward Networks with the Marquardt Algorithm. IEEE Transations Neural Networks 5, 989–993.

Hornik, K., Stinchcombe, M., White, H., 1989. Multilayer Feedforward Networks are Universal Approximators. Neural Networks 2, 359–366.

Hosker Jr, R.P., 1985. Flow and Diffusion Near Obstacles. In: Atmospheric Science and Power Production. pp. 241–326.

Itami, M., 1994. Simulating spatial dynamics : cellular automata theory. Landsc. Urban Plan. 30, 27–47.

Lauret, P., Heymes, F., Aprin, L., Johannet, A., Munier, L., Lapébie, E., 2013. Near Field Atmospheric Dispersion Modelling on an Industrial Site Using Neural Networks. Chem. Eng. Trans. 31, 151–156.

Meroney, R.N., 2004. Wind Tunnel and Numerical Simulation of Pollution Duspersion: A Hybrid Approach. Working paper, Croucher Advanced Study Institute on Wind Tunnel Modeling, Hong Kong University of Science and Technology, -610 December 2004, 60pp.

Pontiggia, M., Busini, V., Gattuso, M., Uguccioni, G., Rota, R., 2012. Consequences Assessment of an Accidental Toxic Gas Release Through a CFD Tool : Effect of the Terrain and Major Obstacles. Chem. Eng. Trans. 26, 537–542.

Russo, L., Vakalis, D., Siettos, C., 2013. Simulating the Wildfire in Rhodes in 2008 with a Cellular Automata Model. Chem. Eng. Trans. 35, 1399–1404.

Sharan, M., Gopalakrishnan, S.G., 1997. Bhopal gas accident: a numerical simulation of the gas dispersion event. Environ. Model. Softw. 12, 135–141.

Sjöberg, J., Zhang, Q., Ljung, L., Benveniste, A., Deylon, B., Glorennec, P.Y., 1995. Nonlinear Black-Box Modeling in System Identification: a Unified Overview. Automatica 31, 1691–1724.