# Nonlinear Prediction by Reinforcement Learning

Takashi Kuremoto, Masanao Obayashi, and Kunikazu Kobayashi

Dept. of Computer Science and Systems Eng., Eng. Fac., Yamaguchi Univ.
Tokiwadai 2-16-1, Ube, Yamaguchi 755-8611, Japan
{wu, m.obayas, koba}@yamaguchi-u.ac.jp

**Abstract.** Artificial neural networks have presented their powerful ability and efficiency in nonlinear control, chaotic time series prediction, and many other fields. Reinforcement learning, which is the last learning algorithm by awarding the learner for correct actions, and punishing wrong actions, however, is few reported to nonlinear prediction.
In this paper, we construct a multi-layer neural network and using reinforcement learning, in particular, a learning algorithm called Stochastic Gradient Ascent (SGA) to predict nonlinear time series. The proposed system includes 4 layers: input layer, hidden layer, stochastic parameter layer and output layer. Using stochastic policy, the system optimizes its weights of connections and output value to obtain its prediction ability of nonlinear dynamics. In simulation, we used the Lorenz system, and compared short-term prediction accuracy of our proposed method with classical learning method.

## 1 Introduction

Artificial neural network models, as a kind of soft-computing methods, have been considered as effective nonlinear predictors [1–4] in last decades. Casdagli employed the radial basis function network (RBFN) in chaotic time series prediction in early time [1]. Leung and Wang analyzed the structure of hidden-layer in RBFN, and proposed a technique called the cross-validated subspace method to estimate the optimum number of hidden units, and applied the method to prediction of noisy chaotic time series [3]. Oliveira ,Vannucci and Silva suggested a two-layered feed-forward neural network, where the hyperbolic tangent activation function was chosen for all hidden units, the linear function for the final output unit, and obtained good results for the Lorenz system, Henon and Logistic maps [2]. Such of neural network models are not only developed on fundamental studies of chaos, but also applied in many nonlinear predictions, e.g., oceanic radar signals [3], financial time series [4], etc. Kodogiannis and Lolis compared the performance of some neural networks, i.e., Multi-layer perceptron (MLP), RBFN, Autoregressive recurrent neural network (ARNN), etc., and fuzzy systems, used for prediction of currency exchange rates [4].

Meanwhile, reinforcement learning, a kind of goal-directed learning, is of great use for a learner (agent) adapting unknown environments [5, 6]. When

the environment belongs to Markov decision process (MDP), or Partially observable Markov decision process (POMDP), an learner acts some trial-and-error searches according to certain policies, and receives reward or punishment. Through the interactions between the environment and learner, both exploration and exploitation are carried out, the learner adapts to environment gradually. Though reinforcement learning has been showing more contributions on artificial intelligence, optimal control theory and other fields, however, this algorithm of machine learning is hardly applied in nonlinear prediction [7].

We have proposed a self-organized fuzzy neural network, which using a reinforcement learning algorithm called Stochastic Gradient Ascent (SGA) [6] to predict chaotic time series [7], and obtained a high precision result in simulation. However, the prediction system used multiple softcomputing techniques including fuzzy system, self-organization function, stochastic policy and so on, so it was complained too complex to use. In this paper, we intend to use a simple multilayer neural network but apply SGA on it, to predict nonlinear time series. This system includes 4 layers: input layer, hidden layer, stochastic parameter layer and output layer. Using stochastic policy, the system optimizes its weights of connections and output value to obtain its prediction ability of nonlinear dynamics. In simulation, we used the Lorenz system [8], and compared short-term prediction accuracy of our proposed method with classical learning method, i.e. error back propagation (BP) [9].

## 2 Prediction Systems

### 2.1 Conventional prediction system

Traditionally, multiple-layer feedforward neural networks serve as a good preditor of nonlinear time series [1, 2, 4]. Fig. 1 gives an example diagram of the networks. Units in each layer are linear functions, or monotonous functions i.e. sigmoid function, generally. Output of units are transfer by weighted connection to the units in next layer, and by adjusting the weights, network output approach to a teacher signal, e.g. training data of time series here. The optimal structure for chaotic time series prediction of this kind of networks are researched detailly in Ref. [2]. In convenient, multiple nodes in hidden layer accept input with weights $w_{kn}$, and their output is given by:

$$H_k(t) = \frac{1}{1 + e^{-\beta_H \sum x_n(t) w_{kn}}} \tag{1}$$

where $\beta_H$ is a constant.

Similarly, output of unit in output layer of system can be described as:

$$\hat{Y}(t+1) = \frac{1}{1 + e^{-\beta_Y \sum H_k(t) w_{yk}}} \tag{2}$$

where $\beta_Y$ is a constant.

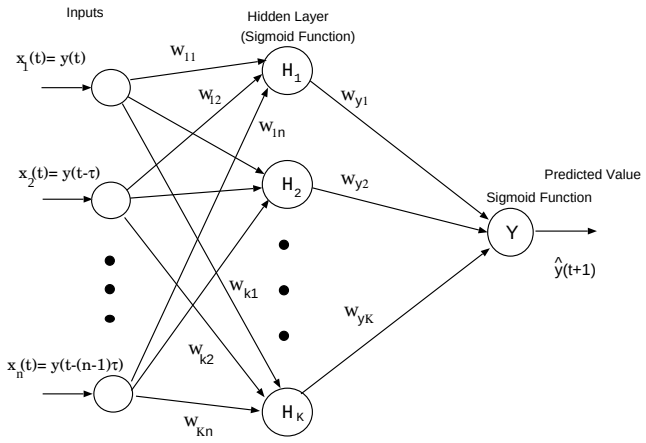


**Fig. 1.** Architecture of a prediction system using error back propagation learning algorithm (conventional system)

### 2.2 Proposal prediction system

To deal with nonlinear dynamcs, we could not neglect stochastic methods, which are more effective on resolving problems in real world. We propose a multiple-layer neural network here, as a nonlinnear predictior, using reinforcement learning algorithm which has a stochastic policy (Fig. 2).

This hierarchical network is composed by 4 layers:

1) Input layer which receiving information of environment, i.e., reconstructed data of time series;

2) Hidden layer which is constructed by multiple nodes of sigmoid function;

3) Stochastic layer (Distribution of prediction layer in Figure 2) which are parameters of probability function, and the nodes fire according to sigmoid function too;

4) Output layer which is a probability function, we use Gaussian function here. Stochastic gradient ascent (SGA) [6], which respects to continuous action, is naturally served into the learning of our predictor. The prediction system and its learning method will be described in detail in this section.

**Reconstructed Inputs** According to the Takens embedding theorem [10], the inputs of prediction system on time $t$, can be constructed as a $n$ dimensions vector space $X(t)$, which includes $n$ observed points with same intervals on time series $y(t)$.

$$X(t) = (x_1(t), x_2(t), \cdots, x_n(t)) \tag{3}$$

$$= (y(t), y(t-\tau), \cdots, y(t-(n-1)\tau) \tag{4}$$

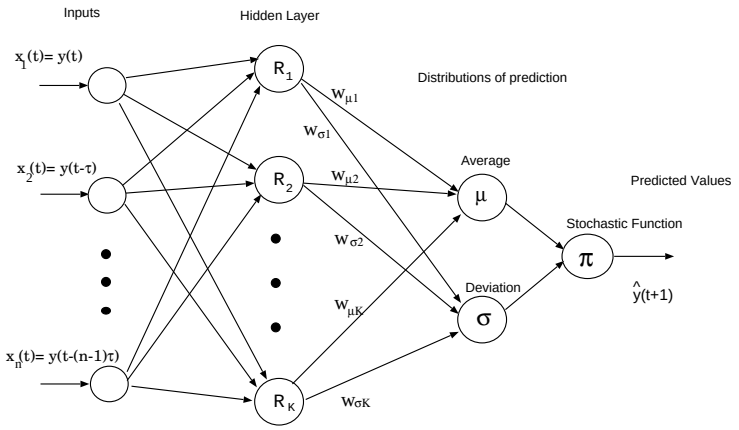


**Fig. 2.** Architecture of a prediction system using reinforcement learning algorithm (proposal system)

where $\tau$ is time delay (interval of sampling), $n$ is the embedding dimension.

If we set up a suitable time delay and embedding dimension, then a track which shows the dynamics of time series will be observed in the reconstructed state space $X(t)$ when time step $t$ increases.

**Hidden Layer** Multiple nodes accept input with weights $w_{ij}$, and their output is given by:

$$R_j(t) = \frac{1}{1 + e^{-\beta_R \sum x_i(t) w_{ij}}} \tag{5}$$

where $\beta_R$ is a constant.

**Stochastic Layer** To each hidden node $R_j(t)$ in hidden layer, parameters of distribution function are connected in weight $w_{j\mu}$ and weight $w_{j\sigma}$ when we consider the output is according to Gaussian distribution. Nodes in stochastic layer give their output $\mu$, $\sigma$ as:

$$\mu(R_j(t), w_{j\mu}) = \frac{1}{1 + e^{-\beta_\mu \sum R_j(t) w_{j\mu}}} \tag{6}$$

$$\sigma(R_j(t), w_{j\sigma}) = \frac{1}{1 + e^{-\beta_\sigma \sum R_j(t) w_{j\sigma}}} \tag{7}$$

where $\beta_\mu$, $\beta_\sigma$ is constant, respectively.

**Output Layer** The node in output layer means a stochastic policy in reinforcement learning. Here we use a 1-dimension Gaussian function $\pi(\hat{y}(t+1), W, X(t))$ simply, to predict time series data:

$$\pi(\hat{y}(t+1), W, X(t)) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(\hat{y}(t+1)-\mu)^2}{2\sigma^2}} \tag{8}$$

where $\hat{y}(t+1)$ is the value of one-step ahead prediction, produce by regular random numbers. $W$ means weights $w_{ij}$, $w_{j\mu}$ and $w_{j\sigma}$. This function causes learner's action so it is called stochastic policy in reinforcement learning.

**Reinforcement learning: SGA algorithm** Kimura and Kobayashi suggested a reinforcement learning algorithm called stochastic gradient ascent(SGA), to respect to continuous action[6]. Using this stochastic approximation method, we train the proposed multiple-layer neural network to be nonlinear predictor. The SGA algorithm is given under.

1. Accept an observation $X(t)$ from environment.
2. Predict a future data $\hat{y}(t+1)$ under a probability $\pi(\hat{y}(t+1), W, X(t))$.
3. Collate training samples of times series, take the error as reward $r_i$.
4. Calculate the degree of adaption $e_i(t)$, and its history for all elements $\omega_i$ of internal variable $W$. where $\gamma$ is a discount($0 \leq \gamma < 1$).

$$e_i(t) = \frac{\partial}{\partial \omega_i} ln\big(\pi(\hat{y}(t+1), W, X(t))\big) \tag{9}$$

$$D_i(t) = e_i(t) + \gamma D_i(t-1) \tag{10}$$

5. Calculate $\Delta\omega_i(t)$ by under equation.

$$\Delta\omega_i(t) = (r_i - b) D_i(t) \tag{11}$$

where $b$ is a constant.

6. Improvement of policy: renew $W$ by under equation.

$$\Delta W(t) = (\Delta\omega_1(t), \Delta\omega_2(t), \cdots, \Delta\omega_i(t), \cdots) \tag{12}$$

$$W \leftarrow W + \alpha(1-\gamma)\Delta W(t) \tag{13}$$

where $\alpha$ is a learning constant, non-negative.

7. Advance time step $t$ to $t+1$, return to (1).

## 3 Simulation

Using time series data of the Lorenz system [8], we examine efficiency of proposed prediction system and compare with error back propagation (BP) method, a classical learning of hierarchical neural network. Both prediction system act in same procedure: observe the Lorenz time series till 1,500 steps, use the beginning 1,000 steps to be training samples, then perform learning loops till prediction errors going to a convergence. After the architecture of system becomes stable, it is employed to predict data from 1,001 step to 1,500 step.

### 3.1 The Lorenz System

The Lorenz system, which is leaded from convection analysis, is composed with ordinary differential equations of 3 variables$o(t)$, $p(t)$, $q(t)$. Here, we use their discrete difference equations (Equ. 14 – 16), and predicts the variable $o(t)$(Equ. 16).

$$o(t+1) = o(t) + \Delta t \cdot \sigma \cdot (p(t) - o(t)) \tag{14}$$

$$p(t+1) = p(t) - \Delta t(o(t) \cdot q(t) - r \cdot o(t) + p(t)) \tag{15}$$

$$q(t+1) = q(t) + \Delta t(o(t) \cdot p(t) - b \cdot q(t)) \tag{16}$$

here,we set $\Delta t = 0.005, \sigma = 16.0, \gamma = 45.92, b = 4.0$.

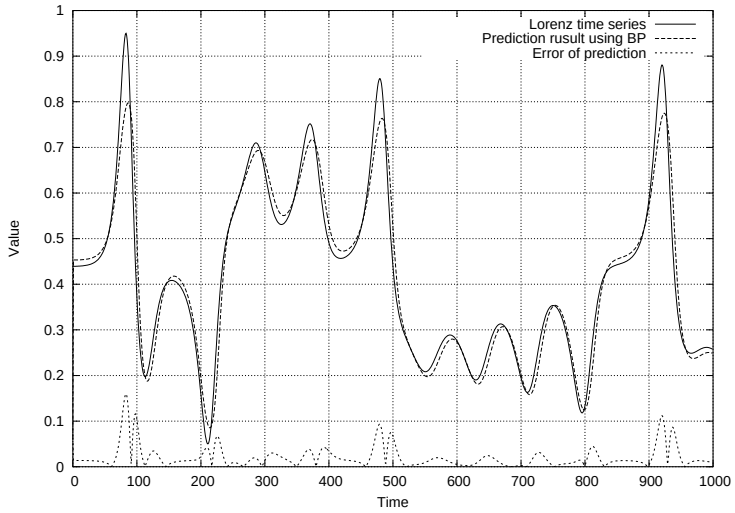


**Fig. 3.** Error back propagation (BP): learning result (2,000 iterations)

### 3.2 Parameters of Prediction System

Parameters in every part of prediction system are reported here.

1. Reconstruction of input space by embedding(Equ.(1),(2)): Embedding dimension $n : 3$ , Time delay $\tau : 1$ , (i.e.,in the case of input to be data of step 1,2,3, then the data of step 4 will be predicted).
2. Multiple neural network using BP:
   Number of hidden layer: 1, Number of hidden layer's nodes: 6, Constant $\beta_H$ of units in hidden layer: 1.0, Constant $\beta_Y$ of unit in output layer: 1.0, Learning rate: 0.01, Maximum value of error: 0.1.

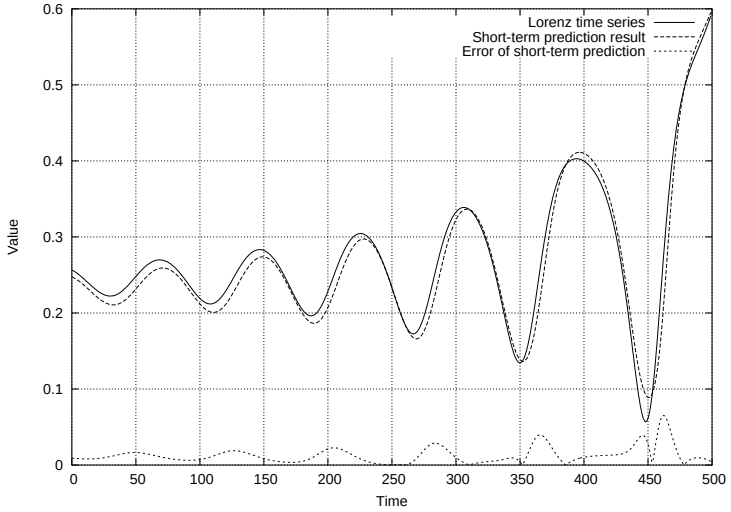


**Fig. 4.** Error back propagation (BP): Short-term (1-step ahead) prediction result

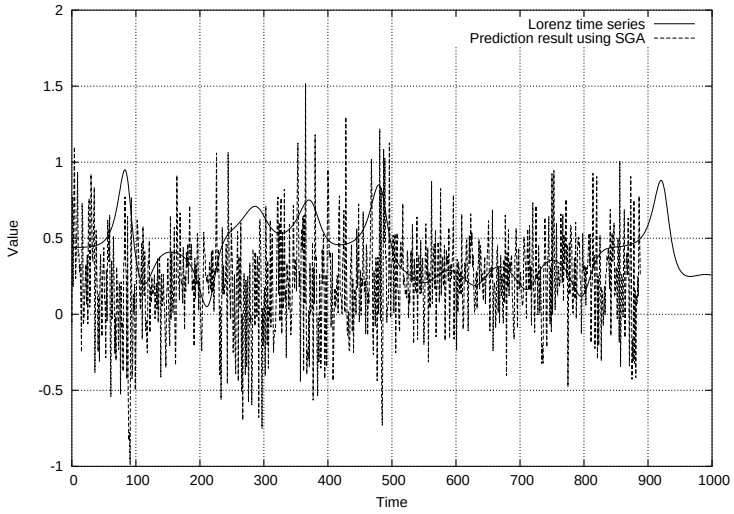


**Fig. 5.** Stochastic Gradient Ascent (SGA): Before learning (0 iteration)

3. Proposed neural network:
Number of nodes $R_j$: 60, Constant $\beta_R$ of units in hidden layer: 10.0, Constant $\beta_\mu$ of unit $\mu$ in stochastic layer: 8.0, Constant $\beta_\sigma$ of unit $\sigma$ in hidden layer: 18.0, Learning constant: For weight $w_{ij}$, $\alpha_{ij}$: 2.0E-6, for weight $w_{j\mu}$, $\alpha_{j\mu}$: 2.0E-5, for weight $w_{j\sigma}$, $\alpha_{j\sigma}$: 2.0E-6, Reinforcement learning of SGA: Reward from prediction error $r_t$ is

$$r_t = \begin{cases} 4.0E-4 & if|\hat{y}(t+1) - y(t+1)| \leq \varepsilon \\ -4.0E-4 & if|\hat{y}(t+1) - y(t+1)| > \varepsilon \end{cases}$$

. Limitation of errors$\varepsilon$: 0.1, Discount$\gamma$: 0.9.

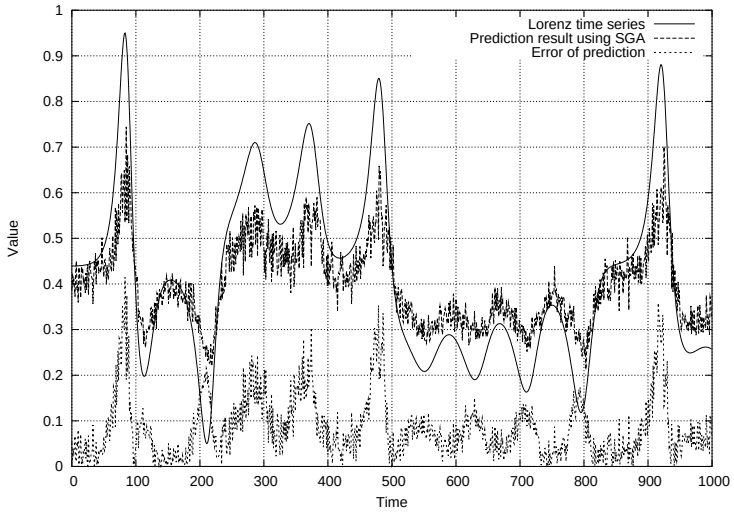


**Fig. 6.** Stochastic Gradient Ascent (SGA): Learning result (5,000 iterations)

### 3.3 Simulation Result

For conventional learning algorithm (BP), Fig. 3 shows its learning result after 2,000 times iteration, and one-head prediction result is shown in Fig. 4. The average value of prediction error in 500 steps short-term prediction is 0.0129 (values of time series data are regularized into (0, 1)).

For proposed system using reinforcement learning, Fig. 5 and Fig. 6 show its learning aspects, Fig. 7 shows its learning result after 30,000 times iteration, and one-head prediction result is shown in Fig. 8. The average value of prediction error in 500 steps short-term prediction is 0.0112 (values of time series data are regularized into (0, 1)), i.e., prediction precision is raised 13.2% than conventional algorithm.

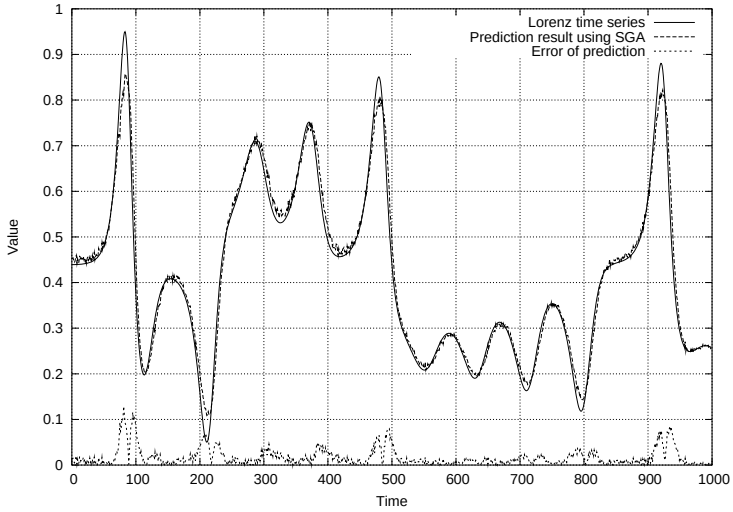


**Fig. 7.** SGA: Learning result (30,000 iterations)

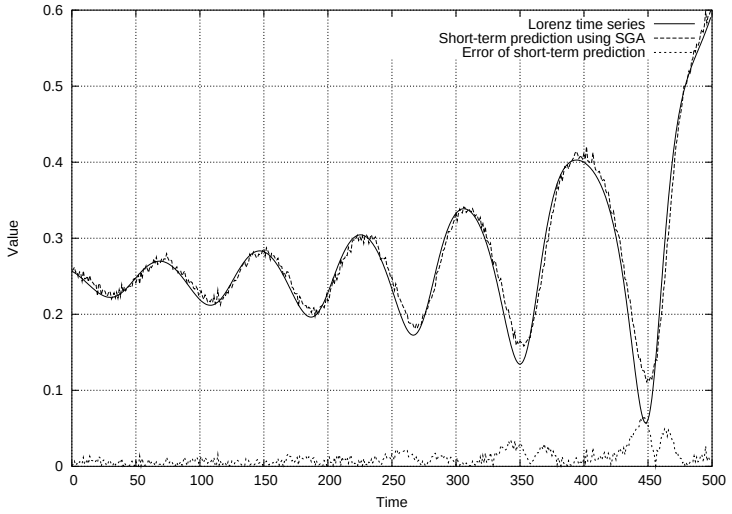


**Fig. 8.** SGA: Short-term (1-step ahead) prediction result

## 4 Conclusions

An algorithm of reinforcement learning with stochastic policy, SGA, was applied to a multi-layer neural network to predict nonlinear time series in this paper. Using Lorenz chaotic time series, prediction simulation demonstrated the proposed system provided successful learning results and prediction results comparing with conventional learning algorithm. For its stochastic output, the proposed system is expected to be applied on the noise contained complex dynamics, or particially observable Markov decision process in real world.

### Acknowledgments

A part of this work was supported by MEXT KAKENHI (No. 15700161).

## References

1. M. Casdagli: Nonlinear prediction of chaotic time series. Physica D: Nonlinear Phenomena **35** (1989)335–356
2. K. A. de Oliveira, A. Vannucci, E. C. da Silva: Using artificial neural networks to forecast chaotic time series. Physica A **284** (1996)393–404
3. H. Leung, T. Lo, S. Wang: Prediction of noisy chaotic time series using an optimal radial basis function. IEEE Trans. on Neural Networks **12** (2001)1163–1172
4. V. Kodogiannis, A. Lolis: Forecasting financial time series using neural network and fuzzy system-based techniques. Neural Computing & Applications **11** (2002)90–102
5. R.S.Sutton and A.G. Barto: Reinforcement learning: an introduction. The MIT Press (1998)
6. H. Kimura, S. Kobayashi: Reinforcement learning for continuous action using stochastic gradient ascent. Intelligent Autonomous Systems **5** (1998)288–295
7. T. Kuremoto, M. Obayashi, A. Yamamoto, K. Kobayashi: Predicting chaotic time series by reinforcement learning. Proc. of The 2nd Intern. Conf. on Computational Intelligence, Robotics and Autonomous Systems (CIRAS 2003)
8. E. N. Lorenz: Deterministic nonperiodic flow. J. atomos. Sci. **20** (1963)130–141
9. D. E. Rumelhart, G.E. Hinton, R. J. Williams: Learning representation by back-propagating errors. Nature **232**(9) (1986)533–536
10. F. Takens: Detecting strange attractor in turbulence. Lecture Notes in Mathematics (Springer-Verlag) **898** (1981)366–381