Acknowledgments

We would like to express our appreciation to Professor S.V. Ulyanov for his continuous help, value corrections and comments to the organization of this paper.
We also wish to acknowledge YAMAHA Motor Co., Ltd. and YAMAHA Motor Europe N.V. and especially for T.Hagiwara, H.Watanabe and S.Fujii for their support and understanding during conducting research, making benchmarks simulations and developing new intelligent control system design technology.

Introduction

Control methods in complex dynamic systems applications have encountered many difficulties. Complex dynamic systems are characterized by uncertain model, a high degree of nonlinearity, instability, distributed sensors and actuators, high level of noise, abrupt changes in dynamics and so on. As a result, the reliability of control systems is decreased. The degree to which a control system deals successfully with above difficulties depends on the level of *intelligence* of control system.

A core component of intelligent control system is a fuzzy control system with a given knowledge base.

For many practical problems important information comes from human experts. Usually, information is not precise and is represented by vague terms like small, large, not very large, and so on. There are many reasons why expert information is usually expressed in a vague form, such as: for convenience, or lack of more precise knowledge, or ease of communication. May be also such sensitive way is much closer to human feelings than exact numerical information. Fuzzy systems arose from the desire to describe complex systems behavior and decision making process with linguistic descriptions mentioned above.

Fuzzy systems are based on fuzzy logic and fuzzy sets theory introduced by L. Zadeh (1973) [1]. Later, it was proven that fuzzy systems can be considered as universal approximator of systems with undefined dynamics and structure [2, 3, 4].Therefore they became so attractive in control engineering.

Fuzzy controllers (FC) allow for a simpler, more human approach to control design and provide reasonable, effective alternative to classical controllers (for example, see the introduction in [5]). Fuzzy logic approach enables us to translate qualitative knowledge about the problem into a reasoning system capable of performing approximate pattern matching and interpolation. But, in fuzzy logic based technology the generation of membership functions (MF) and fuzzy rules (FR) is a task mainly done by a human expert. Human expert also solves the task of refining (or tuning) of knowledge base. It means that fuzzy logic approach itself does not have adaptation and learning capabilities for self-constructing and tuning of MF's, and FR's. These tasks can be realized by using *soft computing technology* [5].

Soft Computing (SC) applied to design of intelligent control systems represents a combination of the following approaches: Fuzzy Systems Theory for a fuzzy control, Genetic Algorithms (GA) for global optimization of control laws, and Fuzzy Neural Networks (FNN) for physical realization of optimal control laws and for knowledge base (KB) design of FC using the extraction of necessary information by learning and adaptation methods.

Main problem in intelligent control systems design is obtaining optimal and robust knowledge base which guarantees high level of control quality in presence all mentioned above difficulties in complex dynamic systems control.

Research and development of new control technologies based on new types of computations has promoted automatic control to be at more and more higher level of intelligence.

1. Brief History of Intelligent Control Systems Design Technology

The current situation of intelligent control systems design technology can be introduced by a historical flow chart in Fig.1-1. This picture shows main steps of our technology for design intelligent control systems based on new types of soft and quantum computing [6,7].

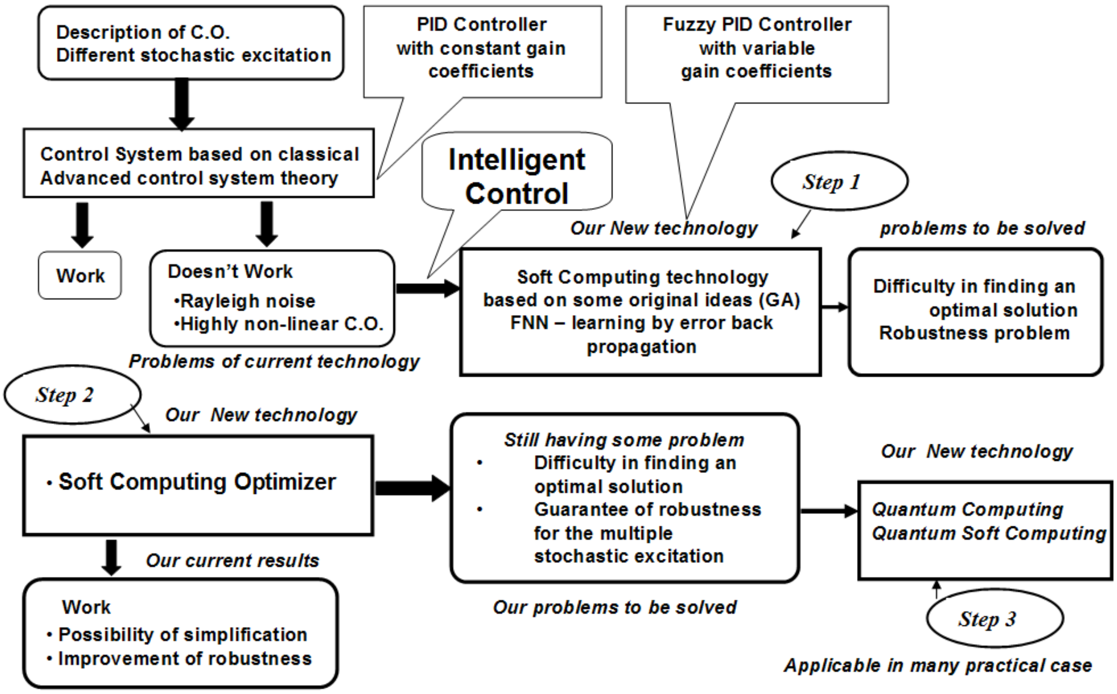


Figure 1-1. Historical flow-chart of SC based technology for design of intelligent control systems

1.1 Physical limitations and information bounds of classical control systems

We have investigated possibilities and limitations of classical (advanced) control theory for the cases of globally unstable and essentially non-linear control objects (C.O.) in the presence of stochastic noises with different probability distribution densities.

By using a set of benchmarks representing stable and unstable dynamic CO and by using our stochastic simulation system we found limitations of classical control approach. The structure of classical control system is shown in Fig.1-2.

Classical control systems are based on PID regulator with constant gain coefficients and on principle of global negative feedback. Thus, only one control quality criterion based on a minimum of control error may be considered.
Simulation results have shown that classical control system doesn't work well, if we have globally unstable or essentially non-linear CO in the presence of Rayleigh noises (with non-symmetric probability distribution densities), and/or if we have random noises in

sensor's measurement system in control channel loops, and/or if we have ill-defined parameters of mathematical model, and so on.
The control criterion based only on a minimum of control error cannot guarantee a robust and stable control achievement in these cases. How can we introduce into control system more complicated control quality criteria such as a minimum of entropy production in a plant, or a minimum of energy loss in controller, etc.?

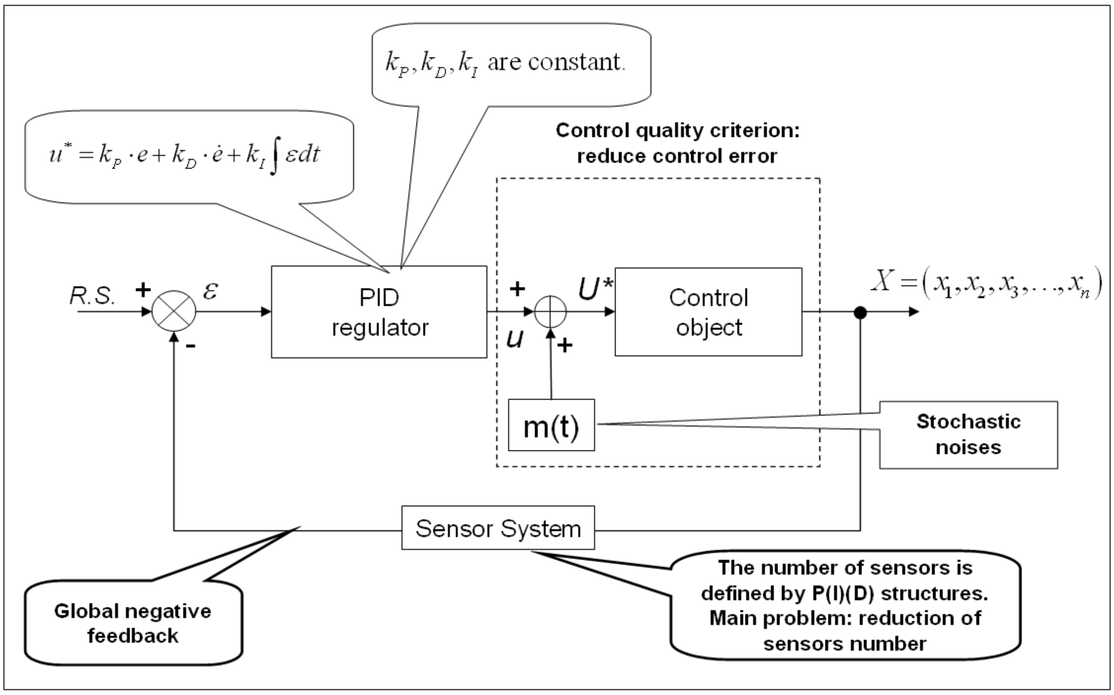


Figure 1-2. General structure of classical control system

Limitations of classical PID control inspired researchers to develop new types of control systems based on ideas of human intelligent control strategies.
A core component of intelligent control system (ICS) is a fuzzy control system with a knowledge base (KB) consisting of a set of fuzzy rules describing human control experience, i.e. it depends on human expert experience. Traditionally, so called expert systems have been used to design FC KB. Expert system looks like a friendly user interface which helps a human expert to represent his knowledge in standard fuzzy rules format. But, in the cases mentioned above (i.e. globally unstable or essentially non-linear CO in the presence of Rayleigh noises, or if we have random noises in sensor's measurement system in control channel loops), even for human expert it is difficult to find optimal control laws. This is a week point of traditional FC approaches based on expert systems.

1.2. General Structure of Intelligent Control System based on Soft Computing

To avoid the week point of traditional FC approaches, we developed step 1 technology for design intelligent control systems (ICS) based on SC approach.

Basic element in our technology is a Fuzzy PID Controller (FC-PID) with variable coefficients (Fig.1-3). But instead of knowledge base designed by human expert we use tools for obtaining and optimizing KB FC based on mathematical model of control object or by using some experimental data.

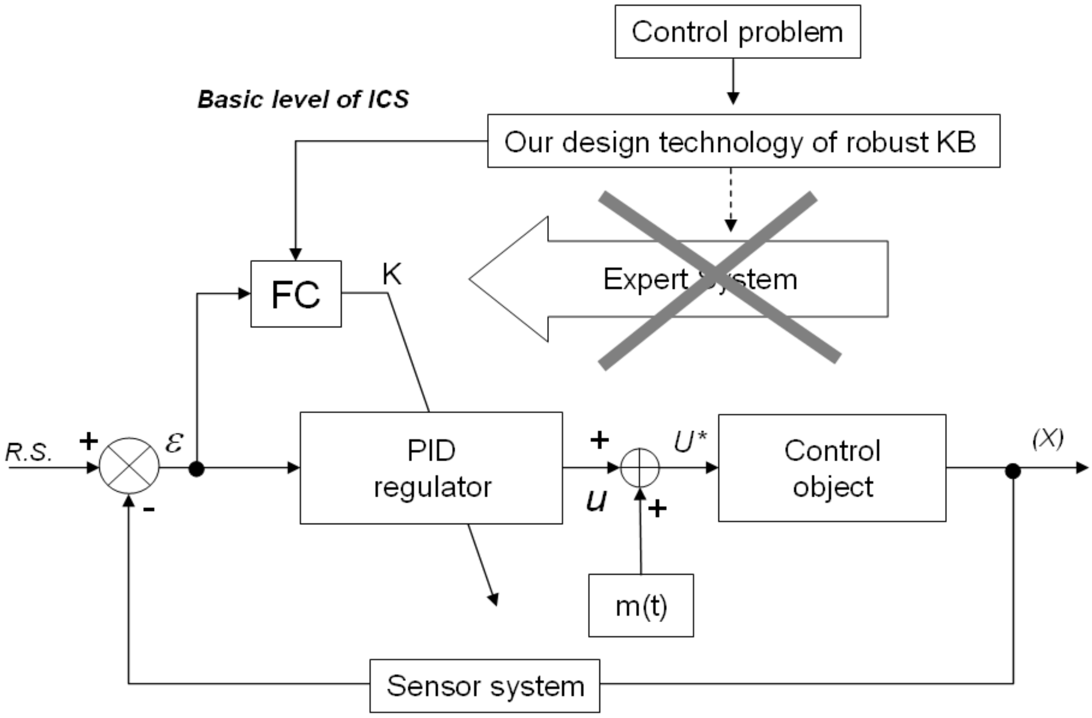


Figure 1-3. General structure of Fuzzy-PID control system

We have used a new model of intelligent feedback for simulation of FC KB and developed our original Genetic Algorithm (GA) with fitness function of control quality based on principle of minimum entropy production rate (MEP) (see, for details [8,9]).

In Fig. 1-4, a general structure of intelligent control system in step 1 technology of ISC design is shown. We use here the following designations:

GA - Genetic Algorithm; *f* - fitness function of GA; S - entropy production of a system; S_c – entropy production of controller; S_p - entropy production of controlled Plant;
K - global optimum solution of GA representing PID coefficients (PID gains) $K=(k_p,k_d,k_i)$;
KB – a knowledge base; FC - Fuzzy Controller; FNN - Fuzzy Neural Network;
ε - control error; u*- optimal control signal; m(t) - Disturbance; (*x*) – a plant state.

The main part of this structure is a *simulation system of control quality* (SSCQ) based on GA with chosen fitness function describing control quality criterion. Output of SSCQ is a vector $K=\{k_p(t),k_d(t),k_i(t)\}$ representing optimal (from the given GA fitness function) control laws. We call this vector K as a teaching signal (TS). Using fuzzy neural networks (FNN) tuning by error back propagation algorithm, we extract a knowledge base from the given TS.

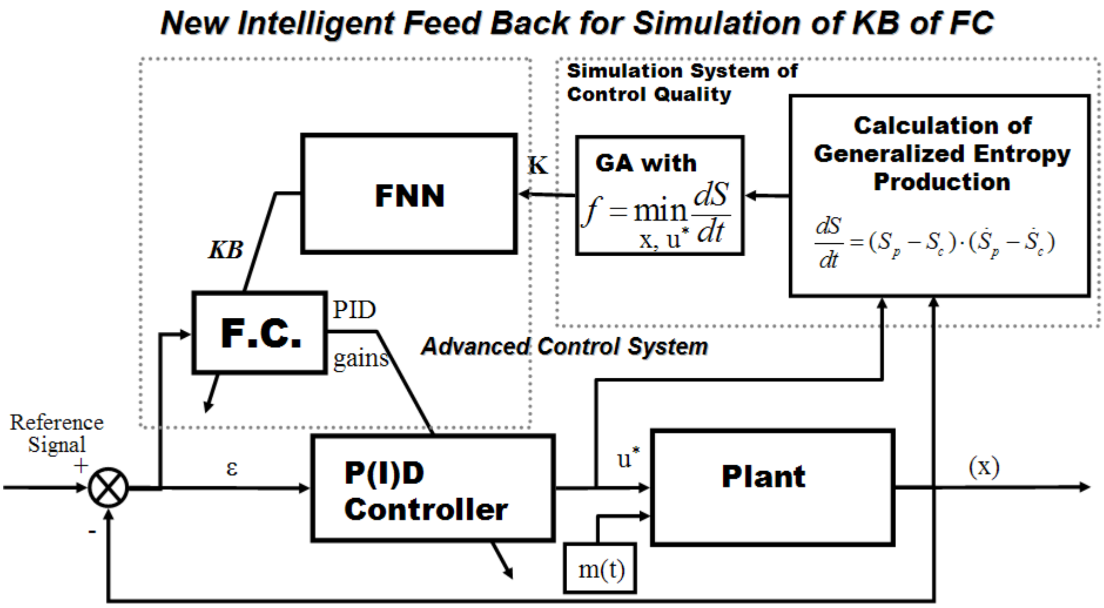


Figure 1-4. General structure of intelligent control system

Main problem of step 1 technology is robustness of designed FC.
By using the set of non-linear CO benchmarks, our stochastic simulation system and step 1 technology tools, we investigated possibilities and limitations of the step 1 technology based on FNN learning by error back propagation algorithm. We found that in the case of unstable or essentially non-linear CO and in the presence of Rayleigh noises, we cannot obtain robust KB FC. Main disadvantage of traditional FNN-based approaches is that the FNN structure must be given *a priori* (i.e., the number and type of MF must be introduced by a user), but sometimes it is difficult to *define optimal FNN structure manually*. To avoid all above mentioned disadvantages we developed step 2 technology based on *Soft Computing Optimizer* (SCO).
Still at step 2, we have some problems which we improve further, e.g. improve robustness in presence of multiple stochastic excitations and/or in presence of inaccuracy in the CO model, and/or in the presence of random noises in sensor's measurement system in control channel loops. For this aim we develop step 3 technology based on quantum computing (QC) and quantum soft computing (QSC).

Before to describe our ICS design tools, represent and analyze simulation results, consider briefly theoretical backgrounds of soft computing technology applied to intelligent control.

2. Brief Theoretical Backgrounds of Intelligent Control System Design based on a new Soft Computing technology and stochastic simulation

Soft Computing applied to design of intelligent control systems represents a combination of the following approaches: Fuzzy Systems Theory for a fuzzy control, Genetic Algorithms for global optimization of control laws, and Fuzzy Neural Networks for physical realization of optimal control laws and for knowledge base design of FC using the extraction of necessary information by learning and adaptation method based on error back propagation algorithm.

Traditional SC approach to control systems design has the following peculiarities:

- in accordance to the fuzzy system theory, a control object is considered as a “black-box”;
- “input-output” linguistic relations connected with this “black-box” are studied and optimized by using GA and FNN-based learning approaches.
- the coefficient gain schedule (control law) of PID-controller (representing a change of PID-gains in time) is described in the form of a KB of FC.

Consider briefly main concepts and ideas of mentioned topics. As we said above, new types of control systems are based on ideas of human intelligent control strategies. In order to formalize human control strategies, we need to use special mathematics introduced by L.Zadeh [1]. In 1965, *L. Zadeh* published the first paper on a novel way of characterizing no probabilistic uncertainties, which he called “fuzzy sets”.

Today fuzzy sets theory evolved into different disciplines. For our aims we will consider fuzzy systems theory.

2.1 Human intelligent control strategy description and Fuzzy sets

Consider the following example of human control strategies. Consider driving situation when a car runs on a road and a driver looks another car on the road in front of him (Fig.2-1). Having information from sensors about a distance between cars on the road and velocity of the car, the driver realizes one of control solutions which can be formulated by ordinary words, as, for example, follows: “if a distance between cars is *long* and a speed of the car is *slow*, hold the gas pedal steady (maintain the speed)”.

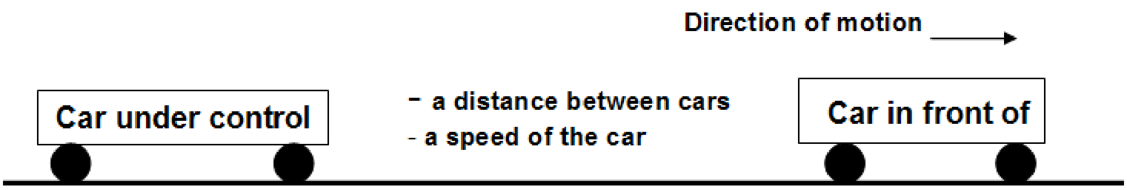


Figure 2-1. Driving a car control situation

So, human expert control strategies are described in linguistic form. Driving a car control strategy may be, for example, represented as following rules.

Rule 1: IF a *distance* between cars is **short** AND a *speed* is **fast**, **THEN** acceleration is negative big (step on the brake)
Rule 2: IF a *distance* between cars is **long** AND a *speed* is **slow**, **THEN** acceleration is positive big (increase the speed) and so on.

There are many reasons why human expert information is usually expressed in linguistic form:

- for convenience and ease of communication,
- in the case of lack of more precise information,
- for generalization and knowledge representation,
- may be also such sensitive way is much closer to human feelings than exact numerical information.

Fuzzy systems arose from the desire to describe complex systems behavior and decision making process with linguistic descriptions [10,11,12].
Fuzzy systems theory is based on two basic formalisms: *fuzzy sets theory* and *fuzzy logic*. Fuzzy sets are used for representation of linguistic values such as small, fast, short, etc., and fuzzy logic is used for flexible decision making based on rules like rule 1 and rule 2 above. Discuss main concepts and peculiarities of these two disciplines.

2.1.1 Fuzzy set definition and its main peculiarities

Definition: (a **fuzzy set**)
If X is the *universe of discourse*, then *fuzzy set A* in X is defined as a set of ordered pairs:

$$A = \{(x, \mu_A(x)) | x \in X\},$$

where $\mu_A(x)$ is called the *membership function* (MF) of x in A. The MF maps each elements of X to a continuous membership value (or membership grade, or a fuzzy measure) between 0 and 1**.** *Fuzzy measure can be considered as a degree of evidence that point x belongs to A.*

Definition: (a **support of a fuzzy set**)
The support of fuzzy set A is the crisp set of all points $x \in X$ such that $\mu_A(x) > 0$.

This definition of fuzzy set is an extension of the definition of a classical set in which the characteristic function is permitted to have continuous value between 0 and 1. Usually the universal set X may contain either discrete or continuous values.
A few classes of parameterized functions are commonly used to define membership functions.

Typical representations of fuzzy sets membership functions

Triangular MF (Fig.2-2, A) is specified by three parameters {*a,b,c*} which determine the *x* coordinates of three corners as follows

$$triangular(x,a,b,c) = \max\left[\min(\frac{x-a}{b-a},\frac{c-x}{c-b}),0\right].$$

Trapezoidal MF (Fg.2-2, B) is specified by four parameters {*a,b,c,d* } which determine the *x* coordinates of four corners as follows

$$trapezoid(x,a,b,c,d) = \max\left[\min(\frac{x-a}{b-a},1,\frac{d-x}{d-c}),0\right].$$

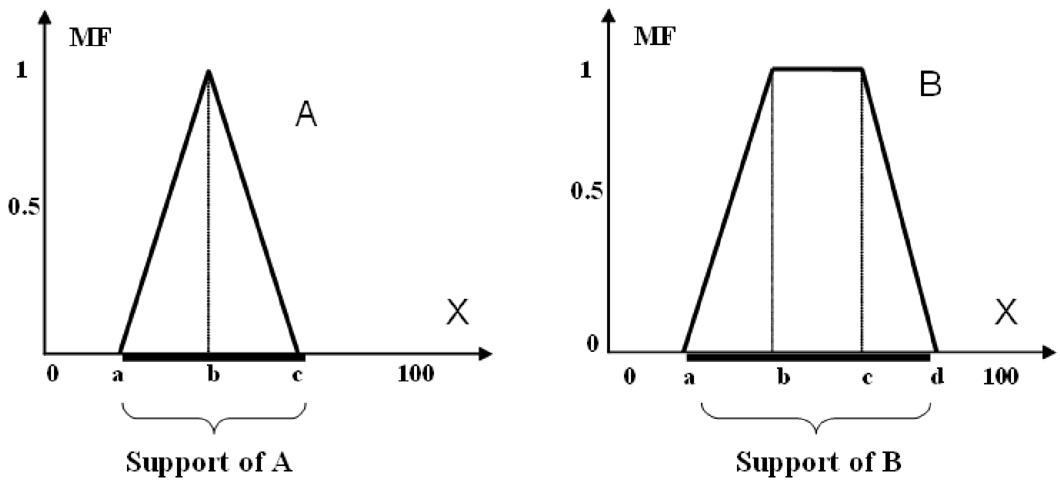


Figure 2-2. Triangular and trapezoidal membership functions representation

Gaussian MF's: Gaussian MF (Fig.2-3) is specified by two parameters as follows:

$$gaussian(x,\sigma,c) = e^{-[(x-c)/\sigma]^2}.$$

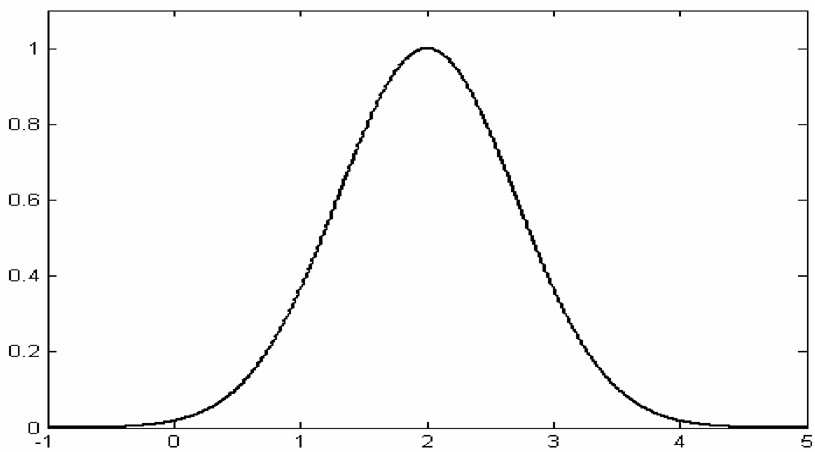


Figure 2-3. Gaussian membership functions representation

Driving a car: example of linguistic values description in the form of fuzzy sets

Return to driving a car example and represent linguistic values such as slow, fast (a speed) and short, long (a distance) by fuzzy sets. Let us consider the following linguistic variables: x is a distance between cars, y is a speed of a car, and z is an acceleration of a car (it allows to adjust the car's speed by gas pedaling or braking operations).

The membership functions need to be defined appropriately for the situations under consideration. For example, the speed of 70 km/h would be "fast" on a street road but it would be "slow" on a highway. The respective universal sets can be defined as $X = \{x|0 \le x \le 50\}[m]$, $Y = \{y|0 \le y \le 100 \quad [km/h]\}$ and $Z = \{z|-20 \le x \le 20\}\left[km/h^2\right]$.

Fuzzy sets "slow", "fast" (a speed) and "short", "long" (a distance) can be represented by membership functions shown in Fig.2-4.

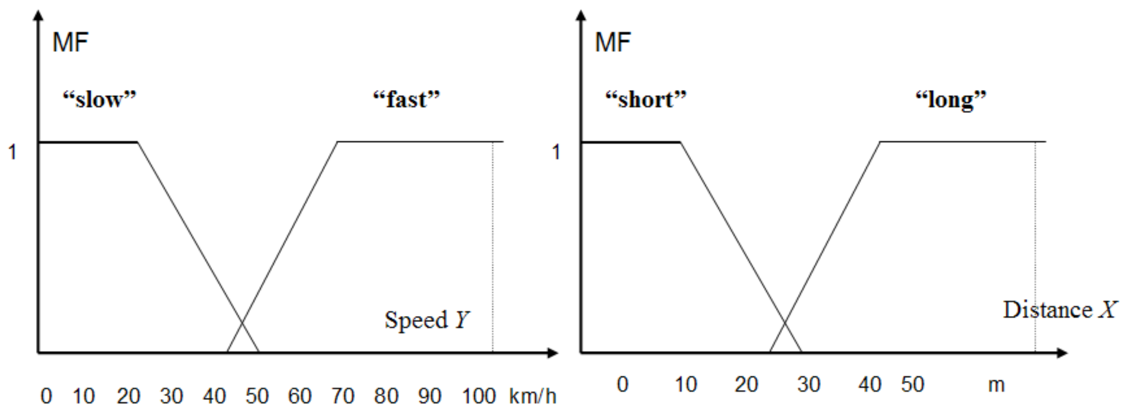


Figure 2-4. Fuzzy sets representation for driving a car control problem

Difference between classical and fuzzy representations

In Figure 2-5 a difference between classical and fuzzy representations of sets is shown by using the following example.

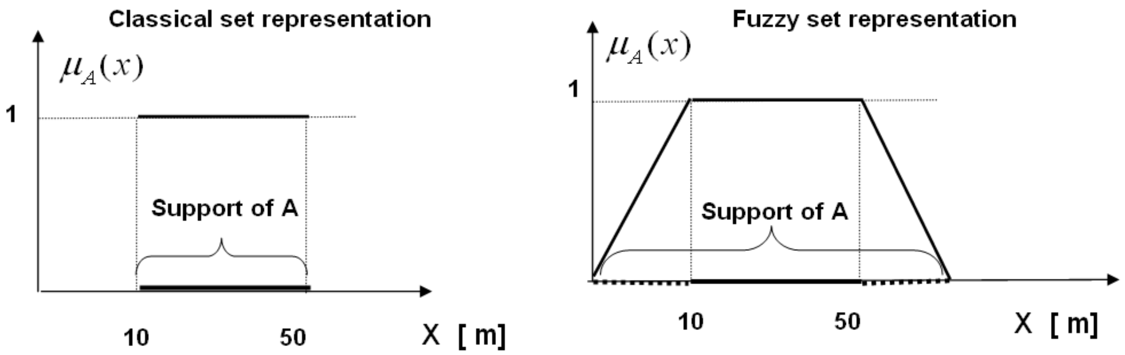


Figure 2-5.Classical and fuzzy representation of set A

Example. Consider classical and fuzzy representations of a set A defined for linguistic value "close" (for distance). Define classical representation of set A, for example, as (Fig.2-5, left):

$$\mu_A(x) = \begin{cases} 1, & 10 \le x \le 50 \\ 0, & otherwise \end{cases}$$

and fuzzy representation of set A as follows:

$$A:\{(x, \mu_A(x) > 0)\}; x \in A, \quad \mu_A(x) \in [0,1],$$

where $\mu_A(x)$ is shown in Fig.2-5 (right).

Discuss now the following problem: "Does the point $x = 9.9$ m or the point $x = 50.01$ m belong to the set A?

From classical representation point of view, the answer is "no". From human sensitive point of view, the answer is rather "yes" than "no". From fuzzy representation point of view the answer is also "yes". Thus, this simple example demonstrates that a fuzzy set representation is more close to human feelings and more flexible representation than classical one. By using fuzzy sets we can describe fuzzy boundaries.

Basic fuzzy sets operations

Definition: **Fuzzy Containment (or Fuzzy Subset)**

Fuzzy set A contained in fuzzy set B (or, equivalently, A is a subset of B) if and only if

$$\mu_A(x) \le \mu_B(x) \text{ for all } x.$$

In symbol form: $A \subseteq B \Leftrightarrow \mu_A(x) \le \mu_B(x)$.

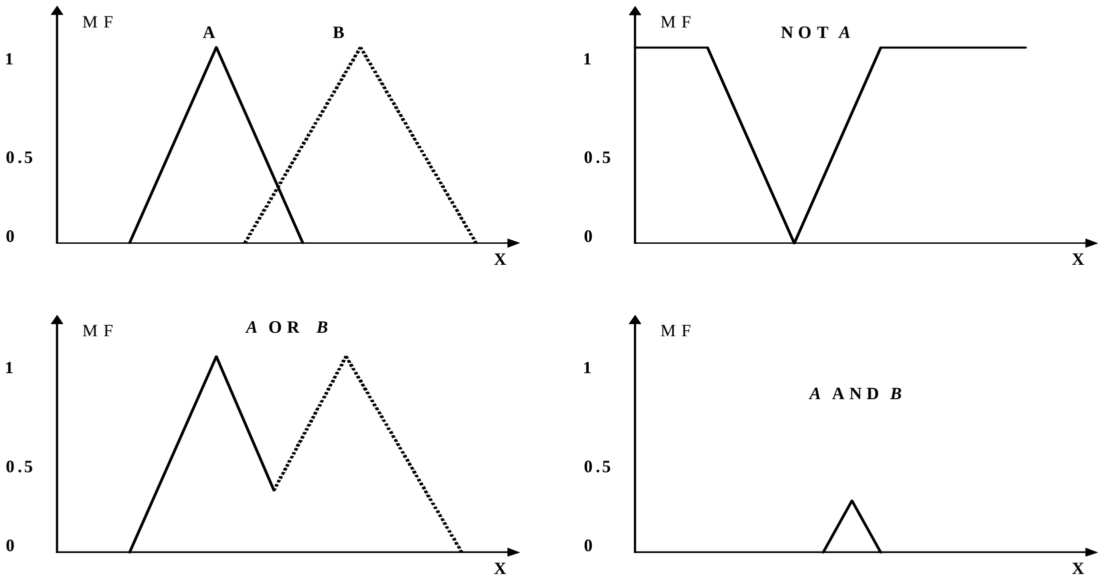


Figure 2-6. Examples of fuzzy operations on fuzzy sets

Definition: **Equality of Fuzzy Sets**
The equality of fuzzy sets *A* and *B* is written as $A = B$ and defined as

$$A = B \Leftrightarrow \mu_A(x) = \mu_B(x) \text{ for every } x \in X .$$

Definition: **Fuzzy Union (or Fuzzy Disjunction)**
The union of two fuzzy set *A* and *B* is fuzzy set *C*, written as $C = A \bigcup B$ or *A* OR *B* or A $\vee$ B, whose MF is defined as

$$\mu_C(x) = \max\{\mu_A(x), \mu_B(x)\} = \mu_A(x) \vee \mu_B(x) .$$

Definition: **Fuzzy Intersection (or Fuzzy Conjunction)**
The intersection of two fuzzy sets *A* and *B* is fuzzy set *C*, written as $C = A \bigcap B$, or *C* = *A* AND *B*, or *C* = A $\wedge$ B, whose MF is defined as

$$\mu_C(x) = \min\{\mu_A(x), \mu_B(x)\} = \mu_A(x) \wedge \mu_B(x) .$$

Figure 2-6 illustrates an example of fuzzy operations introduced by above definitions.

Main peculiarities of Fuzzy Sets

The law: $A \bigcup \overline{A} = X$ (called as the law of *excluded middle*) and the law: $A \bigcap \overline{A} = \phi$, where ϕ means an empty set (called as the law of *contradiction*) are valid for crisp sets, but in general are not valid for fuzzy sets.
The law of excluded middle for fuzzy sets is $A \bigcup \overline{A} \neq X$ .The law of contradiction for fuzzy sets is $A \bigcap \overline{A} \neq \phi$.
Let us make important conclusions from these peculiarities.

In classical case one point x_1 has only one from two possibilities: $x_1 \in A$ or $x_1 \notin A$. In fuzzy case one point x_1 may belong to the set A and to the set $\overline{A}$ with different membership values *a* and *b* as shown in Fig.2-7. It means that in fuzzy reasoning process we can simultaneously consider two possibilities, which makes decision making process more flexible than classical decision. Figure 2-7 illustrates this property.

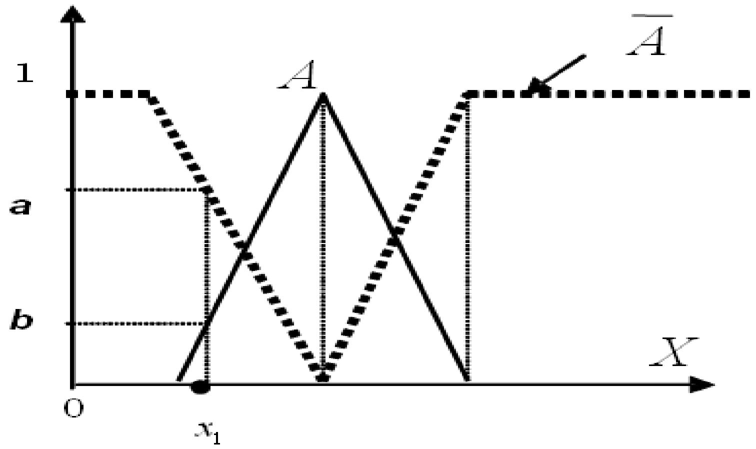


Figure 2-7. Main peculiarities of fuzzy sets

2.2 Fuzzy Logic and its main peculiarities

The *fuzzy logic* can be considered as the extension of an infinite-many-valued logic. Its ultimate goal is to provide foundations for *approximate reasoning with imprecise propositions* using fuzzy set theory as the principle tool. In order to deal with imprecise propositions, fuzzy logic allows the use of *fuzzy predicates***.**

Each simple fuzzy predicate such as " x is P " is represented by a fuzzy set. Assume, for example, that *x* stands for the age of a person and a property *P* has the meaning of *young*.

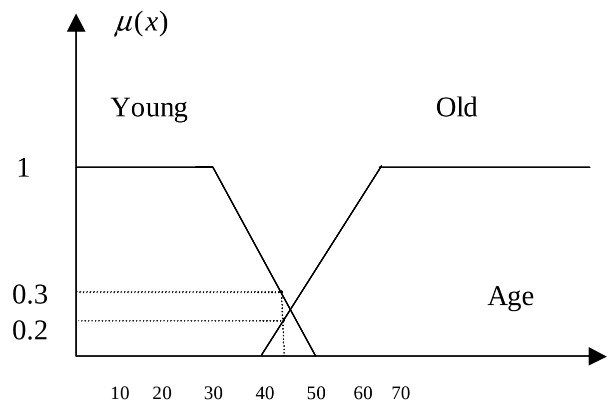


Figure.2–8. Membership values of fuzzy predicates

Then, assuming that the universal set is the set of integers from 0 to 70 representing different ages, the predicate "x is P" may be represented by a fuzzy set whose membership function is shown, for example, in Fig.2 –8.

Consider now the truth value of a proposition obtained by a particular substitution for x into predicate, such as "Sergei is young". The truth value of this proposition is defined by the membership grade of Sergei's age (for example, his age is 45 years old) in the fuzzy set chosen to characterize the concept of a young person (Fig.2–8).

Fuzzy logic as any formal logical system consists of three main components: a set of fuzzy predicates; a set of fuzzy logic operations; and a set of fuzzy inference rules.

2.2.1 Typical Fuzzy Logic Operations

Fuzzy logic operations can be introduced, by different ways. Consider typical fuzzy logic operations used in control application.

Remark. Because fuzzy predicates truth values are described by fuzzy sets, for fuzzy logic operations we will use fuzzy sets operations defined above.

Definition: (**Fuzzy conjunction** (or **fuzzy AND**))

The truth value of fuzzy conjunction $A \wedge B$ is given by

$$\mu_{A \wedge B}(x) = \min(\mu_A(x), \mu_B(x)),$$

where $\mu_A(x), \mu_B(x)$ are truth values of fuzzy predicates A, B respectively.

Definition: (**Fuzzy disjunction** (or **fuzzy OR**))

The truth value of fuzzy disjunction $A \vee B$ is given by

$$\mu_{A \vee B}(x) = \max(\mu_A(x), \mu_B(x)),$$

where $\mu_A(x), \mu_B(x)$ are truth values of A, B respectively.

Definition: (**Fuzzy negation**)

The truth value of fuzzy negation $\overline{A}$ (or $\neg A$) is given by

$$\mu_{\overline{A}}(x) = 1 - \mu_A(x),$$

where $\mu_A(x)$ is truth value of A.

Fuzzy implication or a **fuzzy rule**

A fuzzy rule assumes the following expression:

$$\text{IF } x \text{ is } A \text{ THEN } y \text{ is } \mathrm{B},$$

where *A* and *B* are linguistic variables defined by fuzzy sets on universes of discourse *X* and *Y*, respectively. (We will write this rule as $R = A \to B$).

The part "IF" (x is A) is called the *antecedent*, *or premise*. The part "THEN" (y is B) is called the *consequence or conclusion*.

Definition: (**Fuzzy implication)**

$$\mu_{A\to B}(x,y) = \mu_A(x)\wedge\mu_B(y)$$, where $\wedge$ is a fuzzy and operation.

Interpretation of fuzzy implication given in the definition above is called *Mamdani implication.*

Remark. Note that other definitions for fuzzy logic operations have been proposed in the literature. In general case, these operations are called as *"T-norm"* and *"T-conorm"* respectively (see, for example [14a]).
A popular alternative for fuzzy AND and OR are:

$$\mu_{A\wedge B}(x)=\mu_A(x)\mu_B(x);\ \mu_{A\vee B}(x)=\mu_A(x)+\mu_B(x)-\mu_A(x)\mu_B(x).$$

The alternative definitions for fuzzy implication are, for example, following operators:

Lukasiewicz's implication: $\mu_R(x,y)=1\wedge(1-\mu_A(x)+\mu_B(y))$;

Algebraic product (or Larsen implication): $\mu_R(x,y)=\mu_A(x)\cdot\mu_B(y)$;

Zadeh implication: $\mu_R(x,y)=(\mu_A(x)\wedge\mu_B(y))\vee(1-\mu_A(x))$.

2.2.2 Fuzzy Inference Process

Before discussion about what is a fuzzy inference (or fuzzy reasoning process), let us talk about *reasoning process in general case* illustrated by Fig.2-9.

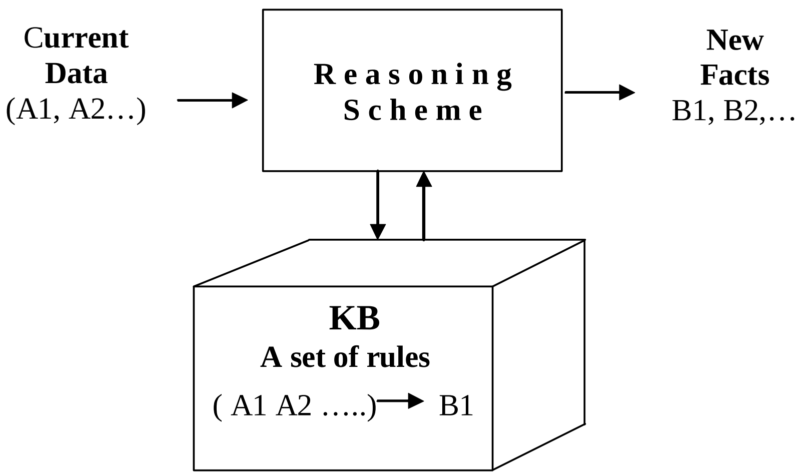


Figure.2–9. General scheme of reasoning process

In general case, the reasoning, or inference, process consists of two stages:

(1) *matching* input data (A1,A2,…) with a left part of a rule containing in a Knowledge Base (KB); and
(2) *Inferring* output data (facts) (B1, B2,…) by using a law of inference.

Classical reasoning scheme

Consider laws of inference in classical propositional logic.

The most popular laws of inference in propositional logic are called *modus ponens* and *modus tollens*.

Modus ponens inference rule: $\frac{A \to B, A}{B}$, which means that if we have a rule $A \to B$ and true input Y and $Y = A$, then we can infer that B is true. B is considered as an output of the inference process.

In this scheme A is called an *antecedent* or *premise*, and B is called a *conclusion* or a *consequent*.

Modus tollens inference rule: $\frac{A \to B, \neg B}{\neg A}$, which means that if we have true input $\neg B$ (not B) and a rule $A \to B$, then $\neg A$ is true, that is, A is false.

Other inference laws are also valid in propositional logic, for example following:

Law of syllogism: $\frac{A \to B, B \to C}{A \to C}$; *Law of contra positive*: $\frac{A \to B}{\neg B \to \neg A}$;

Law of double negation: $\frac{\neg(\neg A)}{A}$.

Write now traditional *modus ponens* inference rule by the following way:

$$\frac{\begin{array}{l}\text{premise 1 (an input fact): } x \textit{ is } A\\ \text{premise 2 (a rule): IF } x \textit{ is } A \text{ THEN } y \textit{ is } B\end{array}}{\text{a consequent (or conclusion): } y \textit{ is } B}$$

Main conclusion about <u>classical reasoning scheme</u>: we make *exact matching* of input data with left parts of rules; we have only *two-valued truth values* of input/output data (true or false).

Fuzzy reasoning scheme

Fuzzy reasoning process is shown in Fig. 2-10. We match input fuzzy data with left parts of fuzzy rules and deduce fuzzy output by using fuzzy inference law.

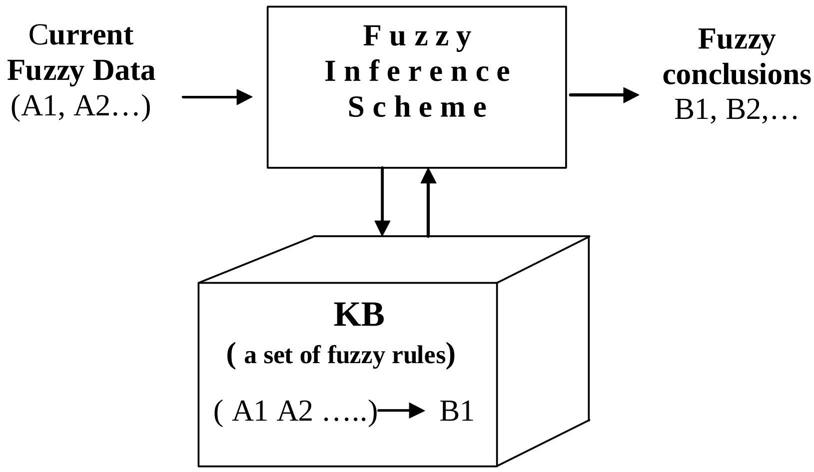


Figure.2–10: General scheme of fuzzy inference process

Consider laws of inference in fuzzy logic. This is a *generalized modus ponens* inference rule:

$$\frac{A \to B, A'}{B'},$$

which means that if we have a fuzzy rule $A \to B$ and a fuzzy input A' ($A' \neq A$), then we can do the output fuzzy conclusion B' ($B' \neq B$).
Rewrite this rule as follows:

$$\begin{array}{l} \text{premise 1 (a fuzzy input): } x \textit{ is } A' \\ \underline{\text{premise 2 (a fuzzy rule): IF } x \textit{ is } A \text{ THEN } y \textit{ is } B} \\ \quad \text{a fuzzy consequent (or conclusion): } y \textit{ is } B' \end{array}$$

Main conclusion about fuzzy reasoning scheme: we make *approximate matching* of input data with left parts of rules; we have *continuous-valued truth values* of input/output data.

We will consider only one law of a fuzzy inference as *generalized modus ponens*. Discuss now: how can we calculate the membership value of a conclusion?
Other words, if we know the membership values of $\mu_{A'}(x)$ and $\mu_A(x), \mu_B(y)$ of left-right parts of the rule, how can we find the membership value of conclusion $\mu_{B'}(y)$?

A fuzzy rule can be defined as *binary fuzzy relation* R on the product space $X \times Y$ ($(x,y) \in X \times Y$) with some membership value $\mu_R(x,y)$.
Consider the following task:

Let A' is a fuzzy set on X and R is a fuzzy relation on $X \times Y$: $R = A \to B$.
Consider the following fuzzy inference problem:

Premise 1 (a fact) : x is A' ,

premise 2 (rule R): IF x is A *THEN* y is B

---

consequent (conclusion): y *is* B';

Find the resulting fuzzy set B' *on Y.*

Fuzzy Reasoning Based on a Max-Min Composition

Consider typical types of calculation of fuzzy conclusion. This calculation is based on a so called *max-min composition* which is defined as follows [13, 14]:

$$\mu_{B'}(y) = \max_x \min\left[\mu_{A'}(x), \mu_R(x,y)\right] = \vee_x \left[\mu_{A'}(x) \wedge \mu_R(x,y)\right], \qquad (2\text{-}1)$$

where $\vee_x$, $\wedge$ are fuzzy disjunction and fuzzy conjunction operations, respectively.

Denote this composition by symbol "$\circ$". Then we may write

$$B' = A' \circ R.$$

If we choose a fuzzy AND operation as the product operation (of membership functions) and a fuzzy OR operation as the *max* operation, then we have *max-product composition* and

$$\mu_{B'}(y) = \vee_x \left[\mu_{A'}(x) \cdot \mu_R(x,y)\right] \qquad (2\text{-}2)$$

Formulae (2-1) and (2-2) are mostly usable formulae in fuzzy inference.

Fuzzy reasoning with one (single) fuzzy rule with one (single) antecedent

For this case we simplify (2-1) by using interpretation of fuzzy implication as

$$\mu_{A\to B}(x,y) = \mu_A(x) \wedge \mu_B(y).$$

Then

$$\mu_{B'}(y) = \vee_x \left[\mu_{A'}(x) \wedge \mu_R(x,y)\right] = \vee_x \left[\mu_{A'}(x) \wedge \{\mu_A(x) \wedge \mu_B(y)\}\right] =$$
$$= \max_x \left[\mu_{A'}(x) \wedge \mu_A(x)\right] \wedge \mu_B(y) = \omega \wedge \mu_B(y). \qquad (2\text{-}3)$$

ω is called a *firing strength of a rule* .

Fuzzy reasoning with one (single) fuzzy rule with two antecedents

A fuzzy rule with two antecedents is usually written as:

"**IF** x is A *AND* y is B **THEN** z *is* C ."

This fuzzy rule represents a ternary fuzzy relation *R* which can be defined by the following membership function:

$$\mu_R(x,y,z) = \mu_{(A\times B)\times C}(x,y,z) = \mu_A(x) \wedge \mu_B(y) \wedge \mu_C(z) \qquad (2\text{-}4)$$

Resulting fuzzy set C' can be represented as:

$$C' = (A' \times B') \circ R \qquad (2\text{-}5)$$

By using (2-4) and extension of (2-1) for the case (2-5), we can calculate $\mu_{C'}(z)$ as:

$$\mu_{C'}(z) = \vee_{x,y} [(\mu_{A'}(x) \wedge \mu_{B'}(y)) \wedge \mu_R(x,y,z)] =$$
$$= \vee_{x,y} [(\mu_{A'}(x) \wedge \mu_{B'}(y)) \wedge (\mu_A(x) \wedge \mu_B(y) \wedge \mu_C(z))] =$$
$$= \vee_{x,y} [(\mu_{A'}(x) \wedge \mu_{B'}(y) \wedge \mu_A(x) \wedge \mu_B(y))] \wedge \mu_C(z) = \quad (2.6)$$
$$= \underbrace{\left\{ \vee_x [\mu_{A'}(x) \wedge \mu_A(x)] \right\}}_{\omega_1} \wedge \underbrace{\left\{ \vee_y [\mu_{B'}(y) \wedge \mu_B(y)] \right\}}_{\omega 2} \wedge \mu_C(z) = \omega_1 \wedge \omega_2 \wedge \mu_C(z)$$

The mechanism of calculating of formula (2-6) is shown graphically in Fig.2-11.

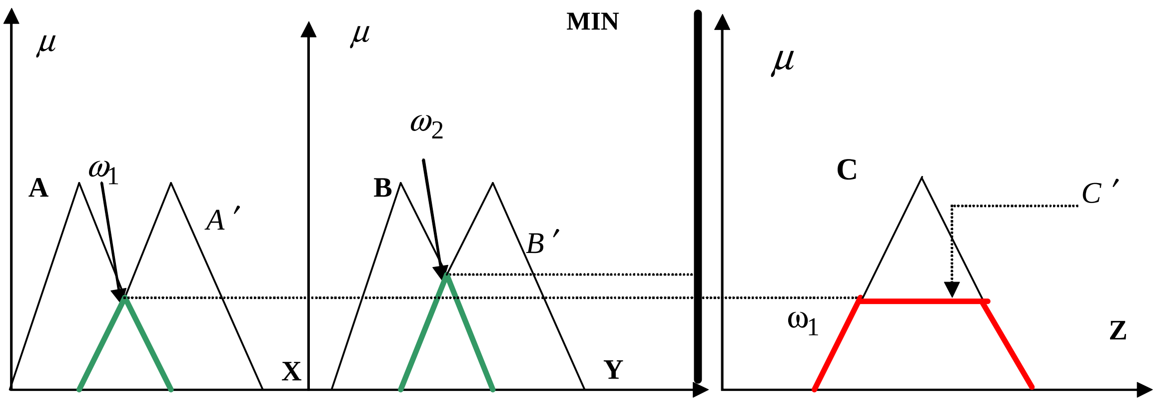


Figure.2–11. Simple graphical interpretation of max-min fuzzy inference scheme

We find ω_1 and ω_2 (firing strengths) as follows: ω_1 is the maximum of intersection of fuzzy sets A and A', ω_2 is the maximum of intersection of fuzzy sets B and B'. Then we give the value of ω as a minimum from the values ω_1 and ω_2. The resulting fuzzy set C' is constructed by cutting the membership function of C by ω.

Remark. Consider another type of fuzzy rule like:

“**IF** x is A *OR* y is B **THEN** z *is* C.”

Then a firing strength ω is given as the maximum from ω_1 and ω_2.

Fuzzy reasoning with multiple fuzzy rules with multiple antecedents

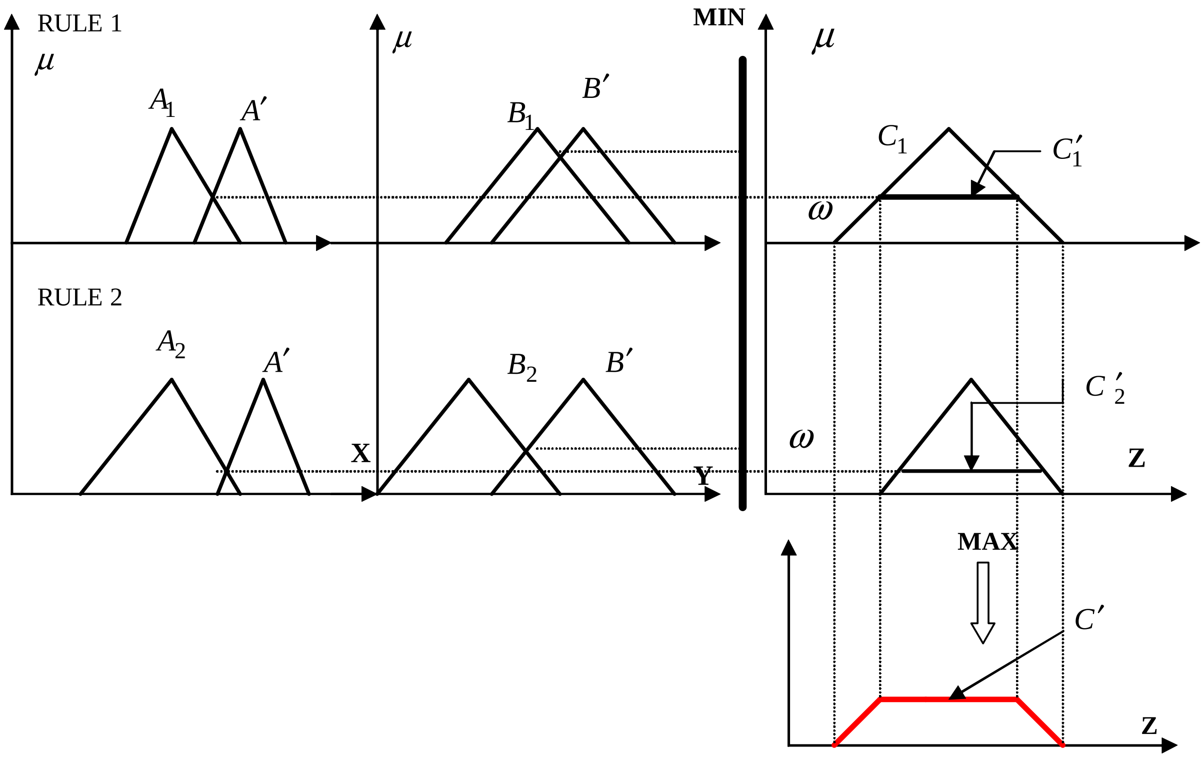


Figure 2-12. Fuzzy Reasoning for multiple rules with multiple antecedents

In this case a fuzzy inference scheme is represented as follows:

premise 1 (a fact) : x *is* A', and y *is* B',
premise 2 (a rule 1): IF x *is* A_1 AND y *is* B_1 THEN z *is* C_1
premise 3 (a rule 2): IF x is A_2 AND y is B_2 THEN z is C_2

---

a *consequent* (conclusion) : z *is* C'

We will design a fuzzy set C' as

$$C' = (A' \times B') \circ (R_1 \cup R_2). \tag{2-7}$$

Since the max-min composition operator is distributive over the '$\cup$' operator we can rewrite Eq.(2-7) as:

$$C' = (A' \times B') \circ (R_1 \cup R_2)) = \\ [(A' \times B') \circ R_1] \cup [(A' \times B') \circ R_2] = C_1' \cup C_2' \,, \tag{2-8}$$

where C_1' and C_2' are the inferred fuzzy sets for rule 1 and rule 2.
So, the final result is constructed as a sum (i.e. max) of two inferred fuzzy sets C_1' and C_2'.

In Fig.2-12 the mechanism of calculating of $\mu_{C'}(z)$ is shown graphically. This method of fuzzy inference is called *min-max method of a fuzzy inference*.

2.2.3. From Fuzzy Logic to Fuzzy Systems and Fuzzy Controllers

In the chapter above we have considered main ideas of fuzzy inference process with fuzzy input and fuzzy output. But, in real control applications the input data are crisp and the output value are expected to be crisp too. To realize this step we make the transition from a fuzzy logic to a fuzzy system (or fuzzy model).

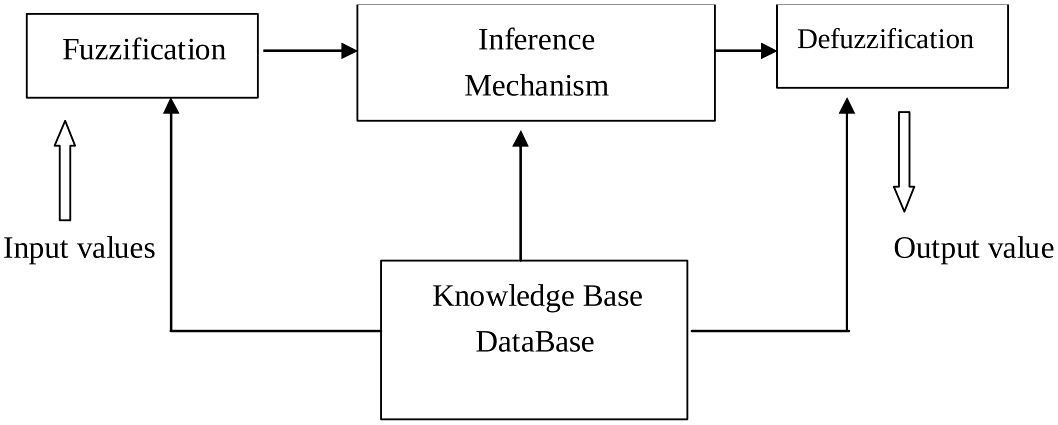


Figure 2-13. Basic structure of a Fuzzy System (or Fuzzy Controller)

The basic structure of the fuzzy system is shown in Fig.2-13. We will use this structure in control engineering area. Therefore, the structure of a Fuzzy Controller repeats the structure of a fuzzy system given in Fig.2-13.

So, Fuzzy Controller consists of the following components:

- Knowledge Base including Fuzzy Rules Base and Database,
- Reasoning (or Inference) Mechanism, and
- Fuzzification/Defuzzification modules.

Fuzzy Rules Base contains a set of fuzzy rules “if-then”. The Database defines membership functions used in the fuzzy rules.

The *fuzzification* component converts input variable’s values (crisp values) to fuzzy set values.

The *defuzzification* component converts fuzzy set values to output crisp value.

2.2.4. Typical fuzzifiers and defuzzifiers

The fuzzifier performs a mapping from a crisp point $a \in X$ (where X is a universal set) into a fuzzy set *A* in X. There are two possible choices of this mapping:

- *singleton fuzzifier*: in this case fuzzy set *A* is defined as: $\mu_A(x) = 1$ for $x = a$ and $\mu_A(x) = 0$ for all other $x \neq a$;

- *nonsingleton fuzzifier*: in this case $\mu_A(a)=1$ and $\mu_A(x)$ decreases from 1 as x moves away from a. For example, $\mu_A(x)=\exp\left[-(x-a)^2/\sigma^2\right]$, where σ^2 is a parameter characterizing the shape of $\mu_A(x)$.

In many applications including control area, the singleton fuzzifier has been used. A nonsingleton fuzzifier may be useful if the inputs are corrupted by a noise.

If we have a partition of a chosen universe of discourse *X* on some parts described by appropriate membership functions, then the fuzzification process may be considered as shown in Fig.2-14.

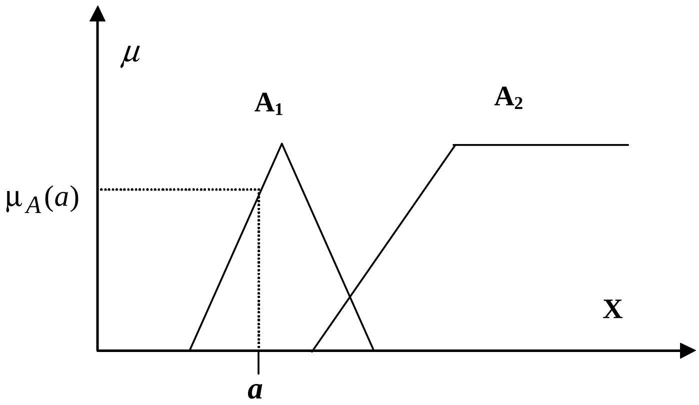


Figure 2-14. Fuzzification process example

The purpose of defuzzification process is to obtain a crisp value from inferred membership function $\mu_C(z),\ z\in Z$.

So, a **defuzzifier** *performs a mapping from fuzzy set in* Z *to a crisp value* z. There are a few possible choices of that:

- *maximum defuzzifier* defined as

$$z=\arg\sup\ \mu_C(z).$$

- *center of gravity defuzzifier* defined as

$z=\dfrac{\int_z \mu_C(z)z\,dz}{\int_z \mu_C(z)dz}$ for continuous case; $z=\dfrac{\sum_z \mu_C(z)z}{\sum_z \mu_C(z)}$ for discrete case, where $\mu_C(z)$ is a final output fuzzy set after applying all rules.

- *center average defuzzifier* defined as

$$y = \frac{\sum_{l=1}^{M} \bar{y}^l \mu_{out}^l(\bar{y}^l)}{\sum_{l=1}^{M} \mu_{out}^l(\bar{y}^l)},$$

where $\mu_{out}^l(y)$ is an output fuzzy set after applying the rule l, $\bar{y}^l$ is the center of the fuzzy set $\mu_{out}^l(y)$ and M is number of rules.

Fuzzy systems as universal approximators

Fuzzy modeling methodology was supplemented by important theorems (necessary and sufficient conditions) according to which fuzzy systems have properties of *universal approximators* [2, 3, 4].

Necessary condition theorem:

For any given real continuous function f on a compact set $U \subseteq R^n$ and arbitrary ε, there exists a fuzzy logic system F with product fuzzy implication , singleton fuzzifier, center-of-gravity deffuzifier, and Gaussian membership functions such that

$$|F(x) - f(x)| < \varepsilon$$

The theorem was proved by Wang L.-X. [3] , and Kosko B. [2].

Sufficient condition theorem (Buckley J.J. [4]):
A fuzzy logic system can approximate any real continuous function.

These two theorems explain why fuzzy systems are very attractive in control application. A fuzzy system (fuzzy controller) can be considered as universal approximator of systems with unknown dynamic and structure.

2.2.5 Typical Fuzzy Models

Consider three types of the most commonly used fuzzy models. They use different types of fuzzy rules, fuzzy inference techniques and defuzzification methods [13, 14, 15].

Mamdani Fuzzy Model

The general form of fuzzy rules in Mamdani fuzzy model represented as follows:

IF x_1 is $\mu_{j_1}^{(l)}(x_1)$ AND x_2 is $\mu_{j_2}^{(l)}(x_2)$ AND … AND x_n is $\mu_{j_n}^{(l)}(x_n)$

THEN y is $\mu_k^{(l)}(y)$,

where $x_1, x_2, ..., x_n$ are *input variables* of a fuzzy model and y is *output variable* of the fuzzy model; l is a rule index and $l = 1, 2, ..., M$ (number of fuzzy rules); $j_1 \in I_{m_1}$ - the set of membership functions (MF) describing linguistic values of x_1 input variable; $j_2 \in I_{m_2}$ - the set of MF describing linguistic values of x_2 input variable; and so on ,

$j_n \in I_{m_n}$ - the set of MF describing linguistic values of x_n input variable; and
$k \in O$ - the set of MF describing linguistic values of y output variable.

In general form, the crisp output of Mamdani fuzzy model with fuzzy AND operation as a product and center average defuziffier calculated as:

$$F(x_1,...,x_n) = \frac{\sum_{l=1}^{M} \bar{y}^l \prod_{i=1}^{n} \mu_{j_i}^l(x_i)}{\sum_{l=1}^{M} \prod_{i=1}^{n} \mu_{j_i}^l(x_i)} = \frac{\sum_{l=1}^{M} \bar{y}^l \omega^l}{\sum_{l=1}^{M} \omega^l}, \text{where } \omega^l = \prod_{i=1}^{n} \mu_{j_i}^l(x_i)$$

$\bar{y}^l$ is the point of maximum value (called also as a central value) of $\mu_k{}^l(y)$.

A simple graphical interpretation of Mamdani inference scheme for multiple rules with multiple antecedents is shown in Fig.2-15.

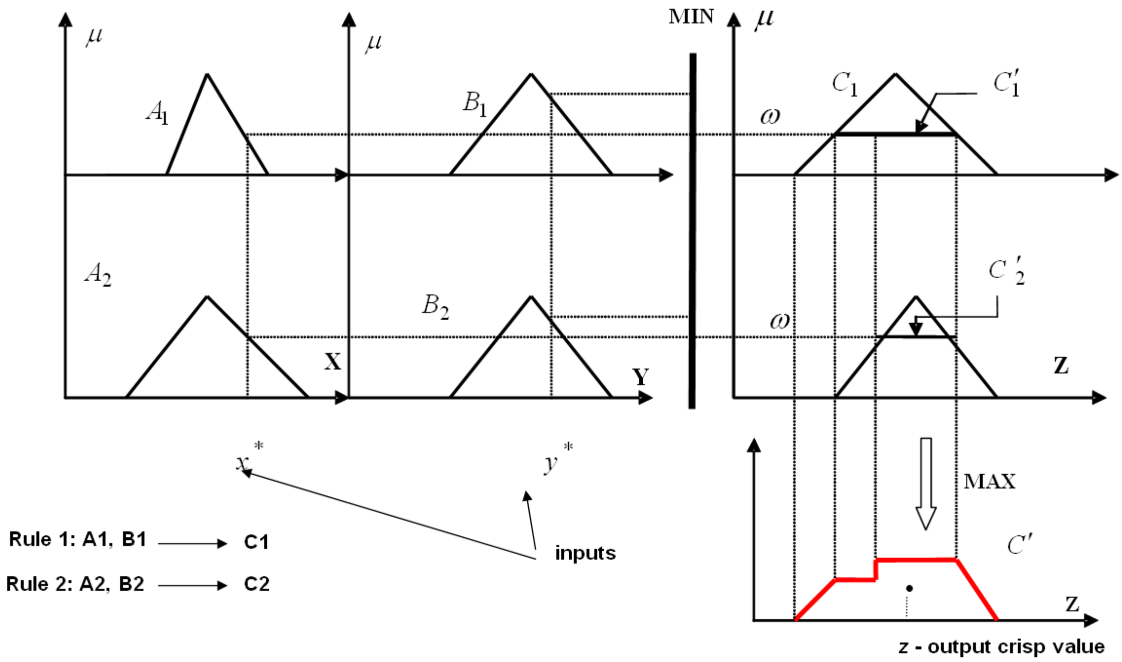


Figure 2-15. A simple graphical interpretation of Mamdani inference scheme
(for multiple rules with multiple antecedents)

Sugeno Fuzzy Model

General form of fuzzy riles in Sugeno fuzzy model can be expressed as follows:

IF x_1 is $\mu_{j_1}^{(l)}(x_1)$ AND x_2 is $\mu_{j_2}^{(l)}(x_2)$ AND … AND x_n is $\mu_{j_n}^{(l)}(x_n)$

THEN $y = f^l(x_1,...,x_n)$,

where $x_1, x_2,..., x_n$ are *input variables* of a fuzzy model and y is *output variable* of the fuzzy model; l is a rule index and $l = 1, 2,..., M$ (number of fuzzy rules), and

$y = f^l(x_1,...,x_n)$ is a crisp polynomial function.

$j_1 \in I_{m_1}$ - a set of MF describing linguistic values of x_1 input variable;

$j_2 \in I_{m_2}$ - a set of MF describing linguistic values of x_2 input variable; and so on,

$j_n \in I_{m_n}$ - a set of MF describing linguistic values of x_n input variable.

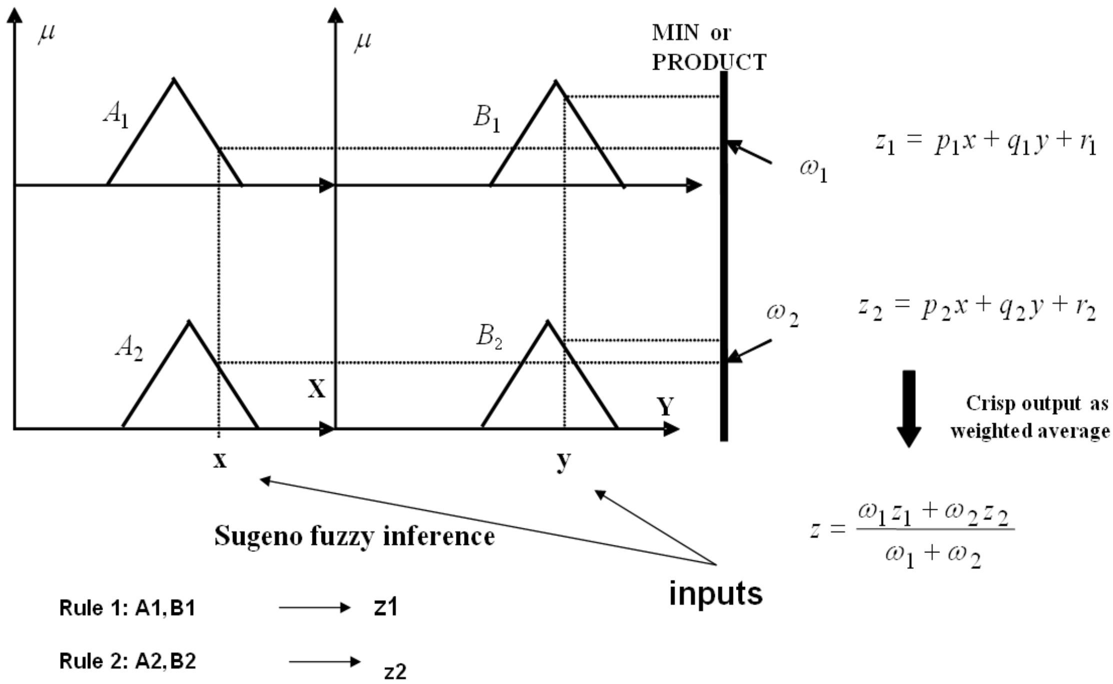


Figure 2-17. A simple graphical interpretation of Sugeno inference scheme

When $y = f^l(x_1,...,x_n)$ is a constant, then fuzzy model is called *zero-order Sugeno fuzzy model*. This model is used for FC-PID control. Output crisp value is defined as a weighted average. For the case of *max* and *product* for fuzzy OR and fuzzy AND, we can write general form of Sugeno model output as follows:

$$\boxed{F(x_1, x_2,..., x_n) = \frac{\sum_{l=1}^{M} f^l(x_1, x_2,..., x_n) \prod_{i=1}^{n} \mu_{j_i}^l(x_i)}{\sum_{l=1}^{M} \prod_{i=1}^{n} \mu_{j_i}^l(x_i)}.}$$

A simple graphical interpretation of Sugeno inference scheme for multiple rules with multiple antecedents is shown in Fig.2-17.

Tsukamoto Fuzzy Model

Typical rules in the Tsukamoto Fuzzy Inference System look as follows:

IF x_1 is $\mu^{(l)}_{j_1}(x_1)$ AND x_2 is $\mu^{(l)}_{j_2}(x_2)$ AND … AND x_n is $\mu^{(l)}_{j_n}(x_n)$

THEN y is $\mu^{(l)}_{k}(y)$,

where $x_1, x_2, ..., x_n$ are input variables , y is an output variable of a fuzzy model, and $\mu^{(l)}_{k}(y)$ is a *monotonic* membership function. *M* is the number of rules.

Output crisp value is defined as a weighted average. For the case of *max* and *product* for fuzzy OR and fuzzy AND operations, we can write general form of Tsukamoto model output as follows:

$$\boxed{F(x_1,...,x_n) = \frac{\sum_{l=1}^{M} z^l \prod_{i=1}^{n} \mu^l_{j_i}(x_i)}{\sum_{l=1}^{M}\prod_{i=1}^{n} \mu^l_{j_i}(x_i)} = \frac{\sum_{l=1}^{M} z^l \omega^l}{\sum_{l=1}^{M} \omega^l}, \text{ where } \omega^l = \prod_{i=1}^{n} \mu^l_{j_i}(x_i) \;;\; z^l = arg[\mu^{(l)}_k = \omega^l].}$$

A simple graphical interpretation of Sugeno inference scheme for multiple rules with multiple antecedents is shown in Fig.2-18.

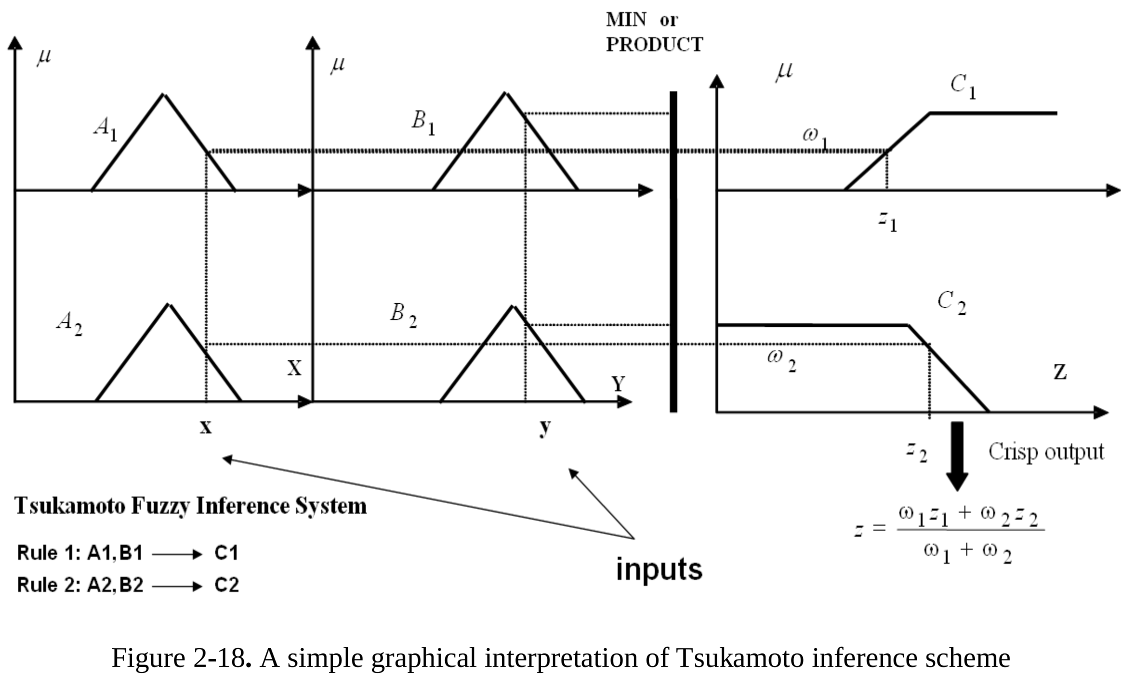


Figure 2-18. A simple graphical interpretation of Tsukamoto inference scheme

All mentioned above fuzzy models are widely used in Fuzzy Control Engineering.

2.3 Fuzzy PID Control

Fuzzy controllers allow for a simpler, more human approach to control design and provide reasonable, effective alternative to classical controller. The structure of a Fuzzy PID Controller (FC) is shown in Fig.2-19.

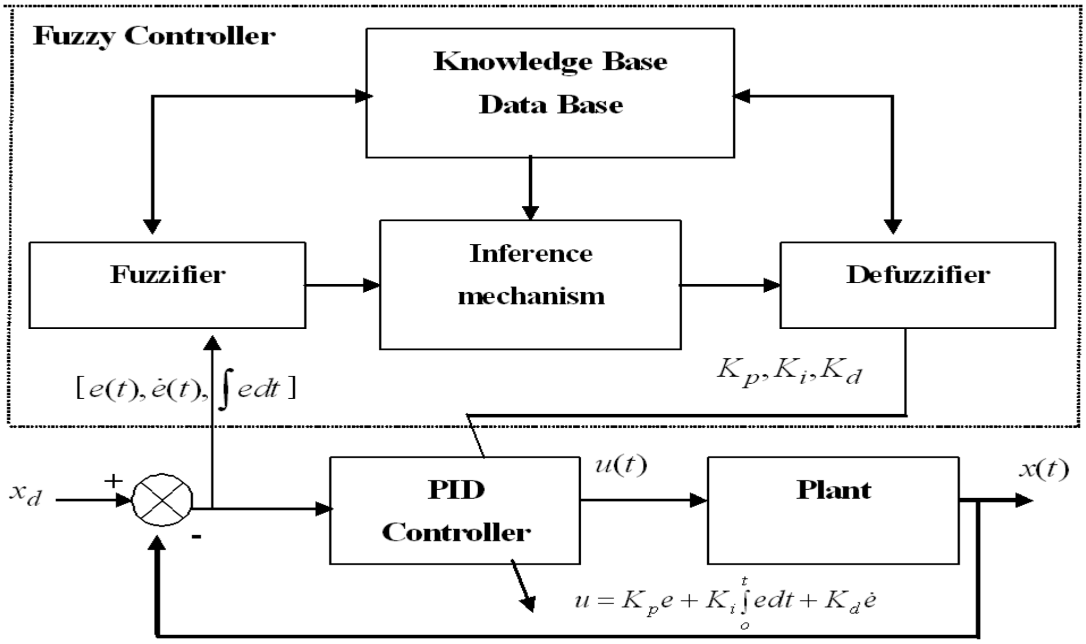


Figure 2-19**.** The structure of a Fuzzy PID Controller

In Fig.2-19, $x(t)$ is a state vector describing behavior of a controlled object (a plant), x_d is desired state of the controlled object, called also as a *reference signal*; $e(t)$ is an *error*, calculated as $e(t) = x_d - x(t)$, $\dot{e}(t)$ is a derivative of error, and $ie = \int edt$ is an integral error; $u(t)$ is a control vector; $e \in E, \dot{e} \in dE, ie \in iE \quad u \in U$, where E, dE, iE and U are universes of discourse for error, derivative of error, integral error and control, correspondingly.

The input-output relation of the PID controller is expressed as

$$u = k_p e + k_i \int_o^t edt + k_d \dot{e}$$

where k_p, k_d, k_i denote proportional gain, derivative gain, and integral gain respectively.

If different values of k_p, k_d, k_i are chosen, then it is obvious that various responses of the plant will be obtained. The performance of the controlled plant is greatly influenced by the selected PID parameters set. It is desirable for designers to propose effective way of selecting these parameters in order to provide a better control response.

How to define this effective way of PID-parameters selecting?

To enhance the capabilities of traditional PID controllers we use a fuzzy logic approach.

Fuzzy set U' of control in FC is defined by a fuzzy inference algorithm represented in general form as the following computation procedure

$$U' = (E' \times dE' \times iE') \circ R\,,$$

where "$\circ$" is a composition operator (see Eq.2-7) and U', E', dE', iE' are fuzzy sets defined on the corresponding universes of discourse, and $R = \bigcup_{i=1}^{M} R_i$ is the union of fuzzy relations (fuzzy rules) contained in the knowledge base of FC.

Once we establish fuzzy rules, we can realize the control strategy by fuzzy reasoning. Therefore, the structure of the Fuzzy Controller is the structure of fuzzy reasoning itself.

Input variables for fuzzy reasoning are the *error, derivative of error and integral error* and *output variables* of fuzzy reasoning are k_p, k_i, k_d *parameters* (see Fig.2-19). Knowledge Base (KB) of the fuzzy PID control system contains a set of fuzzy rules represented as follows.

If n_1, n_2, n_3 are the numbers of membership functions for fuzzy description of error, its derivative and integral values , then there will be $n_1 \times n_2 \times n_3$ fuzzy rules expressed as:

If e is A_1 and $\dot{e}$ is B_1 and $\int_o^t edt$ is S_1, then $K_p = C_{111}, K_i = D_{111}, K_d = E_{111}$

$\vdots$

If e is A_1 and $\dot{e}$ is B_1 , and $\int_o^t edt$ is S_{n_3}, then $K_p = C_{11n_3}, K_i = D_{11n_3}, K_d = E_{11n_3}$

$\vdots$

If e is A_1 and $\dot{e}$ is B_2 and $\int_o^t edt$ is S_1, then $K_p = C_{121}, K_i = D_{121}, K_d = E_{121}$

$\vdots$

If e is A_1 and $\dot{e}$ is B_{n_2} and $\int_o^t edt$ is S_1, then $K_p = C_{1n_21}, K_i = D_{1n_21}, K_d = E_{1n_21}$

$\vdots$

If e is A_{n_1} and $\dot{e}$ is B_{n_2} , and $\int_o^t edt$ is S_{n_3} then $K_p = C_{n_1n_2n_3}, K_i = D_{n_1n_2n_3}, K_d = E_{n_1n_2n_3}$,

where $A_1, A_2, ..., A_{n_1}$, $B_1, B_2, ..., B_{n_2}$ and $S_1, S_2, ..., S_{n_3}$ are membership functions of e , $\dot{e}$, and $\int_o^t edt$, correspondingly, and $C_{11}, ..., C_{n_1n_2}$, $D_{11}, ..., D_{n_1n_2}$ and $E_{11}, ..., E_{n_1n_2}$ are real numbers that satisfy:

$$K_{p,\min} \le C_{ij} \le K_{p,\max}, 1 \le i \le n_1, 1 \le j \le n_2, 1 \le k \le n_3$$

$$K_{i,\min} \le D_{ij} \le K_{i,\max}, 1 \le i \le n_1, 1 \le j \le n_2, 1 \le k \le n_3$$

$$K_{d,\min} \le E_{ij} \le K_{d,\max}, 1 \le i \le n_1, 1 \le j \le n_2, 1 \le k \le n_3.$$

According to Sugeno fuzzy inference model, the outputs will be calculated as follows:

$$K_p = \frac{\left(\sum_{i=1}^{n_1}\sum_{j=1}^{n_2}\sum_{k=1}^{n_3} \omega_{ijk} C_{ijk}\right)}{\left(\sum_{i=1}^{n_1}\sum_{j=1}^{n_2}\sum_{k=1}^{n_3} \omega_{ijk}\right)}, \quad K_i = \frac{\left(\sum_{i=1}^{n_1}\sum_{j=1}^{n_2}\sum_{k=1}^{n_3} \omega_{ijk} D_{ijk}\right)}{\left(\sum_{i=1}^{n_1}\sum_{j=1}^{n_2}\sum_{k=1}^{n_3} \omega_{ijk}\right)}, \quad K_d = \frac{\left(\sum_{i=1}^{n_1}\sum_{j=1}^{n_2}\sum_{k=1}^{n_3} \omega_{ijk} E_{ijk}\right)}{\left(\sum_{i=1}^{n_1}\sum_{j=1}^{n_2}\sum_{k=1}^{n_3} \omega_{ijk}\right)},$$

where $\omega_{ijk} = A_i(e) \cdot B_j(\dot{e}) \cdot S_k(\int edt)$.

Remark. More simple fuzzy rules can be used in FC-KB, for example instead of three input parameters (*error, derivative of error and integral error*) may be used two input parameters (*error*, *derivative of error*) or two input and two output parameters and so on. It's depends upon a chosen structure of classical controller (P, PD, PI, or PID).

3. Implementation of adaptation and optimization capabilities in intelligent control systems

Fuzzy logic approach enables us to translate qualitative knowledge about the problem into a reasoning system capable of performing approximate pattern matching and interpolation. But, in fuzzy logic based technology the generation of membership functions (MF) and fuzzy rules (FR) is a task mainly done by a human expert. Human expert also solves the task of refining (or tuning) of knowledge base.

Fuzzy logic approach does not have adaptation and learning capabilities for self-constructing (automatic constructing) of MFs and FRs of fuzzy controller.

These tasks can be successfully solved by using *Soft Computing technology* including FC KB optimization based on *Genetic Algorithms* (GA) and membership function's parameters tuning based on *Artificial Neural Networks* (ANN) learning algorithms.

3.1 GA theoretical backgrounds for global optimization processes

John Holland was the first researcher who in 70's developed and analyzed *genetic algorithms* [16]. GA represent global optimum search procedures with a probabilistic component based on the *mechanism of natural evolution.*

Natural evolution is a process that operates on *chromosomes*, which are special organic devices. Chromosomes encode the structure of living beings (individuals). The set of individuals is called a *population*. The population evolves over time through competition: *survival of the fittest*. This is the biological law of evolution.

Three main processes characterize the natural evolution: *selection, recombination and mutation.*

Natural selection processes work as follows:

1) link chromosomes with the performance (*fitness*) of their decoded structure ;
2) define the chromosomes that encode *successful structures* (which have a maximum values of fitness) and
3) cause those better chromosomes to reproduce more often than a chromosomes encoding not successful structures.

Recombination process creates different chromosomes in children by combining material from the chromosomes of the two parents.

Mutation process causes the chromosomes of children to be different from the chromosomes of their parents.

GA incorporates these features of natural evolution in computer algorithms to solve difficult problems of optimization through evolution.

Main paradigm of GA:

GA investigates a search space, finds and maintains a population of individual's structures that represent candidates of optimal solution to the current problem.

GA can be applied to a wide range of optimization and learning problems, including routing and scheduling, machine vision, control systems design and so on.

We consider the using of GA for the task of fuzzy rules and members hip function design for fuzzy controllers. Let us discuss the basic structure and mechanism of GA [17].

3.1.1 Basic Structure and Mechanisms of GA

Step 0: *Coding*

GA operates on a *coding* of the parameters of the problem. Thus, at first, the parameters of the problem must be encoded in finite length strings (like chromosomes).

A chromosome can be considered as a vector x consisting of l genes:

$x = (a_1 a_2 ... a_l),\ a_i \in A_i$,

where l is the length of the chromosome , A_i is the alphabet . Commonly, all A_i is the same, that is $A_1 = A_2 = ... = A_l = A$.

If A = {0,1} then chromosome is represented by *binary genes*.

If $A = R$ {real numbers} then chromosome is represented by *real-valued genes*. Further the following steps are performed by GA.

Step 1: *Initial population construction*

The initial population can be initialized using whatever knowledge is available about possible solutions. In the absence of such knowledge, the initial population should represent a random sample of search space.

Randomly generate initial population $X(0) := (x_1, x_2, ..., x_n)$.

Step 2: *Fitness evaluation*

Compute a fitness $f(x_i)$ of each individual x_i in the current population $X(t)$.

In the step 2 each member of population is evaluated and assigned a measure of its *fitness* as a solution. Fitness can be measured by using some fitness function (called also as objective function, or evaluation function).

When each individual in the population has been evaluated, a new population of individuals is formed in two steps (step3 and step4).

Step 3: *Selection*

Generate an intermediate population $X_r(t)$ (called also as a *Mating pool* or a set of possible parents) applying the *reproduction (selection) operator*.

In this step individuals in the current population are selected for replication (reproduction, copy) based on their relative fitness. Individuals with high relative fitness ("good" individuals) might be chosen several times for replication, while individuals with low relative fitness ("bad" individuals) might not be chosen at all.

The probability $p(x_i)$ that an individual x_i will be copied into the next generation depends upon of the ratio of its fitness value $f(x_i)$ to the total fitness, *F*, of all individual in the population. This ratio ($f(x_i)$ / *F*) is called a *relative fitness*.

The reproduction is done by conducting a series of random trials in which each string is copied to the intermediate population a number of times that is proportional to the value of its relative fitness. This random procedure can be, for example, like a *Monte Carlo random procedure*, called also as "wheel of fortune" (or Roulette wheel) (see Fig.3-1).

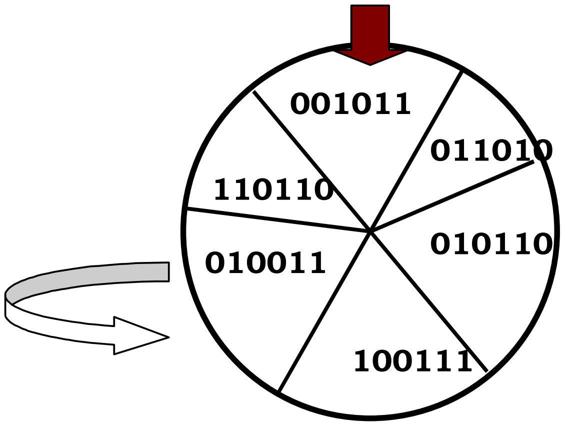


Figure 3-1. Roulette wheel procedure (or "Wheel of fortune")

Each chromosome (in the roulette procedure) occupies an area that proportional to its relative fitness.

Expected number of times ($E_{select}(x)$) that chromosome x will be selected is given as follows:

$$E_{select}(x) = NP_{select}(x), \tag{3-1}$$

where *N* is the population size; and $P_{select}(x) = \dfrac{f(x)}{\sum_{i=1}^{N} f(x_i)}$. (3-2)

(We approximate the value in Eq.(3-1) to the nearest integer.)

Remark. Besides the "roulette wheel" procedure, may be another methods of selections. For example:

- *uniform selection*: each chromosome has an equal chance of selection regardless of its fitness;
- *tournament selection*: small number of chromosomes is uniformly chosen, after they compete with each other on the basis of their fitness.
- *Selection with elitism*: the fittest chromosome is transferred to the next generation without change. Then random selection is performed over remaining chromosomes.

In the absence of any other mechanism, the resulting selective procedure would cause the best individuals to occupy a larger and larger proportion in the population over time.

Step 4: *Crossover and Mutation*
Generate the population $X(t+1)$ applying genetic operators to the $X_r(t)$.
In this step the selected individuals (from $X_r(t)$)) are altered using a *genetic operators* to form a new set of individuals for evaluation. Consider two main genetic operators: *crossover* and *mutation*.

Crossover Operation

The primary genetic search operator is the *crossover operator*, which performs the following functions: 1) selects two parent's individuals, 2) mates (or combines) the features of these two parents and 3) forms two similar *offspring (children)*.

There are many possible forms of crossover. The simplest is the following. Pairs of parents are selected randomly. For each pair of parents a *point of crossover* (one or two, or a few) is selected also randomly. This point indicates how many bits on the right end of each string should be interchanged.
For example, if the parents are represented by the lists

$$(a_1 a_2 \updownarrow a_3 a_4 a_5) \text{ and } (b_1 b_2 \updownarrow b_3 b_4 b_5),$$

and the point of crossover is shown by the symbol "$\updownarrow$ ", then the crossover-operator produces the following *offspring*:

$$(a_1 a_2 b_3 b_4 b_5) \quad \text{and} \quad (b_1 b_2 a_3 a_4 a_5).$$

You can see that the <u>*contiguous groups of bits at the right end of two strings are interchanged their values*</u>.

Remark. There are another types of crossover, for example, those shown below.

a) *Uniform crossover*:

$$\begin{array}{ccc} \textit{parents}: & & \textit{offspring}: \\ (A_1 B_1 \updownarrow C_1 D_1 \updownarrow E_1 \updownarrow F_1) & & (A_1 B_1 C_2 D_2 E_1 F_2) \\ & (001101)\text{ - } \textit{a crossover mask}) & \\ (A_2 B_2 \updownarrow C_2 D_2 \updownarrow E_2 \updownarrow F_2) & & (A_2 B_2 C_1 D_1 E_2 F_1) \end{array}$$

In the uniform crossover, a bit string called a *crossover mask* is used to generalize the crossover process. **1** bit in this mask indicates that corresponding bits in the parents are to be interchanged; bit **0** indicates no bit interchange.

b) *Linear interpolation 2-point crossover*:

parents: offspring:

$$(A_1 B_1 C_1 \updownarrow D_1 E_1 \updownarrow F_1) \Rightarrow$$

$$\left(\frac{2A_1 + A_2}{3} \quad \frac{2B_1 + B_2}{3} \quad \frac{2C_1 + C_2}{3} \quad \frac{D_1 + 2D_2}{3} \quad \frac{E_1 + 2E_2}{3} \quad \frac{2F_1 + F_2}{3}\right)$$

$$(A_2 B_2 C_2 \updownarrow D_2 E_2 \updownarrow F_2) \;\Rightarrow$$

$$(\frac{A_1 + 2A_2}{3} \quad \frac{B_1 + 2B_2}{3} \quad \frac{C_1 + 2C_2}{3} \quad \frac{2D_1 + D_2}{3}$$

$$\frac{2E_1 + E_2}{3} \quad \frac{F_1 + 2F_2}{3}).$$

In generating new individuals for testing, the crossover operator usually based only on the information present in the structures of the current individuals. If specific information is missing, due to storage limitations or loss incurred during the selection process of a previous iteration, then *crossover can not produce new structures* that contain it.

Mutation

A *mutation operator* which alters one or more components of a selected structure, provides the way to introduce new information into the population.

A wide range of mutation operators have been proposed, ranging from completely random alterations to more heuristically motivated local search operators. Mutation operator serves as secondary search operator that allows GA to investigate all points in the search space.

The resulting offspring are then evaluated and inserted back into population, replacing older members.

Specific decisions about how many members are replaced during each iteration, and how members are selected for replacement, define a range of alternative implementation.

Step 5: *Checking of End_Test*

t = *t*+1; IF NOT (*End_Test*) THEN go to Step2 ; else stop.

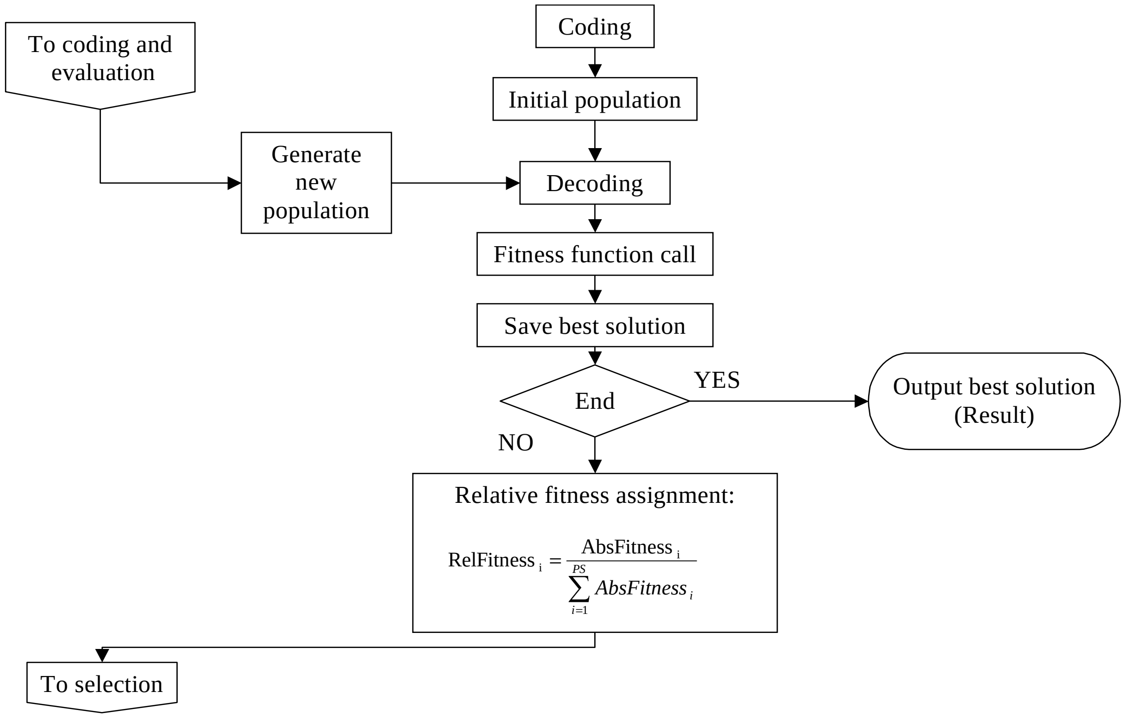
The "End_Test" describes the condition of finishing (termination) of GA. It is a stopping criteria. The "End_Test" is usually given by the number of generation (for example, t = 1, 2,...,100), or by the time-length of work of GA (for example, 3 hours) , or may be some special convergence criteria.

So, final generation of individuals represents a solution of a given optimization task. In this case we say that GA converges to the optimal solution. *Convergence of GA means the situation when all of the chromosomes in the final population have the same gene values.*

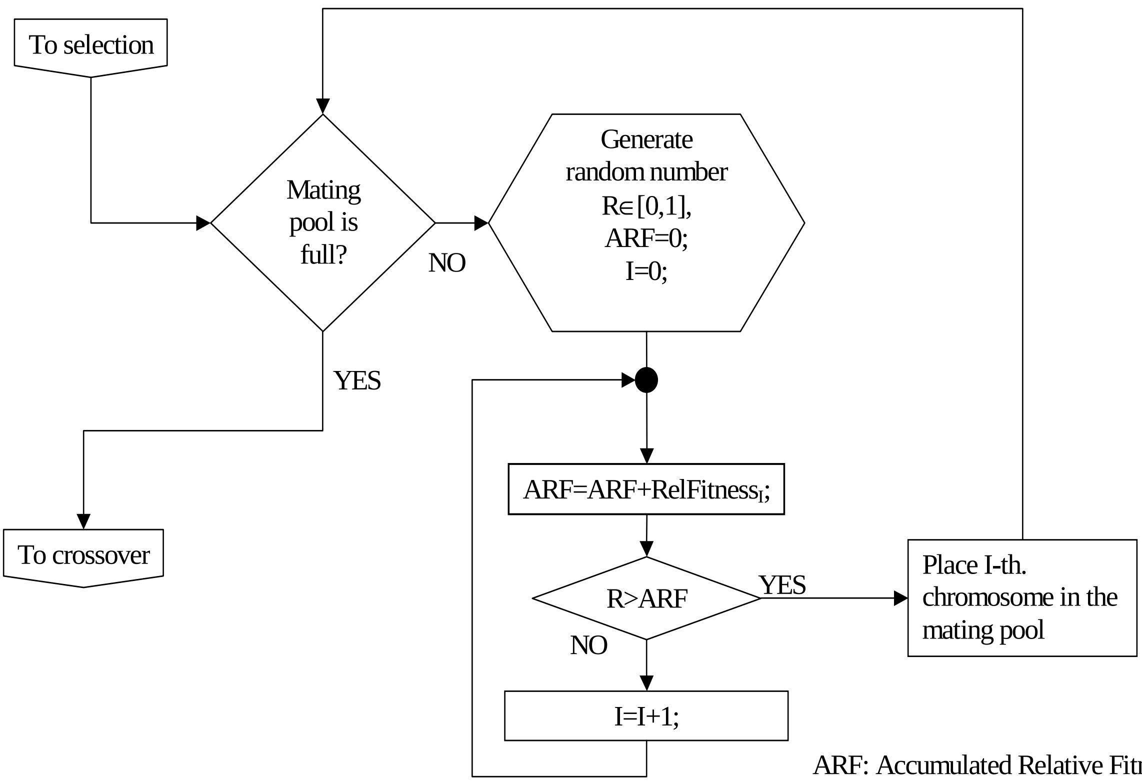
We have seen that even simple GA exhibits a sophisticated information processing capabilities.

Block-diagrams of GA [2,18] are shown below in Fig.3-2 (a,b,c,d).

(a) Coding and Evaluation

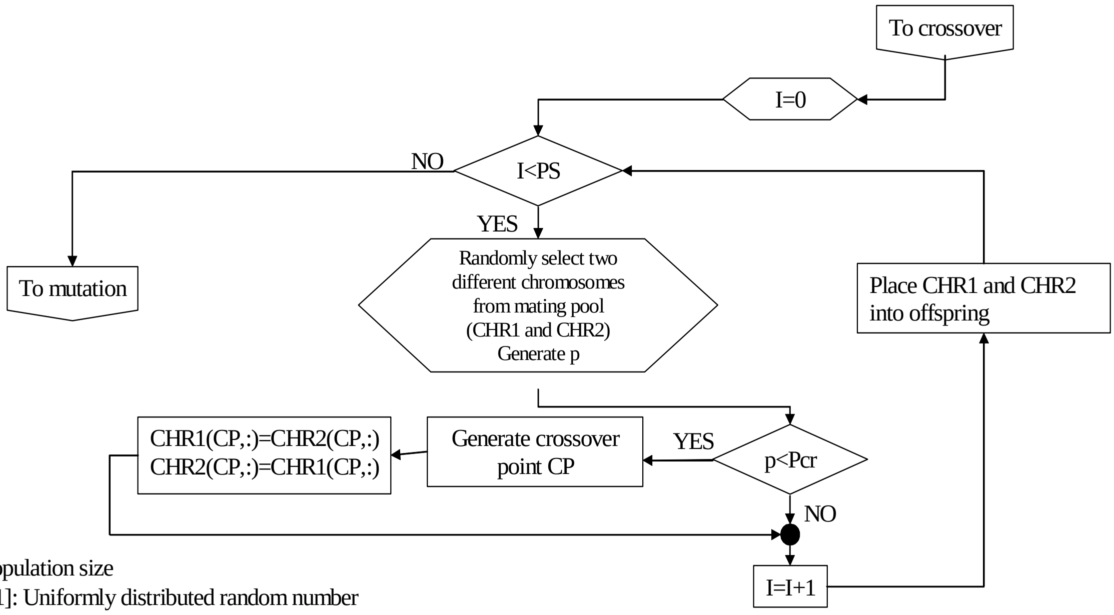


(b) Selection



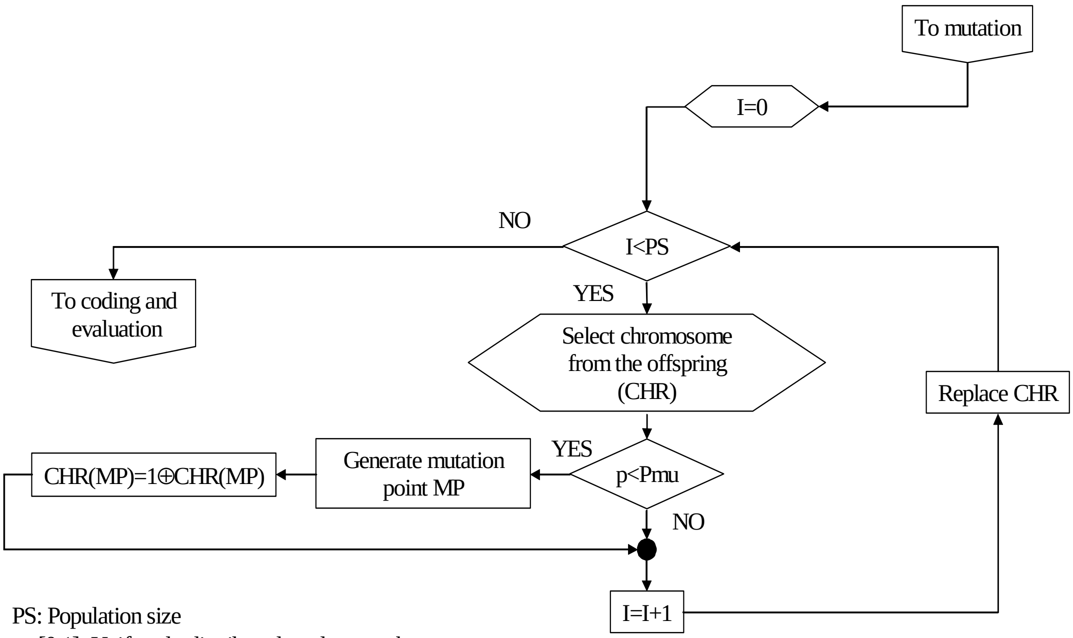
ARF: Accumulated Relative Fitness

(c) Crossover



PS: Population size
p∈[0,1]: Uniformly distributed random number
CP∈[1,L]:Uniformly distributed random number, crossover point, L- length of chromosome

(d) Mutation



PS: Population size
p∈[0,1]: Uniformly distributed random number
MP∈[1,L]:Uniformly distributed random number, mutation point, L- length of chromosome

Figure 3-2. Block-diagrams of GA

3.1.2 Theoretical Foundation of GA

GA based optimization supplemented by the Fundamental Theorem of GA (Holland, [16]). Let us give some necessary definitions in order to introduce the Holland theorem [17].

Simple GA usually apply a binary coded strings. We will say, that such strings are produced from the alphabet A = {0,1}. Each symbol in a string is identified by its bit position, where *bit position equal* 1 *means the first left symbol in a string*.

Definition: (a **schema**)

A schema S (the plural is schemata) is a string that is constructed from alphabet A' = {0,1,*} , where * indicates " don't care" .

For example, the following string is the schema:

$$S = * 1 * 0 * * 0 0. \qquad (3\text{-}3)$$

Definition: (a **representative of a schema**)

A string which matches a schema in all of its definite bit position (0 or 1) is called the representative of given schema.

The following strings A, B, and C represent schema S given in (3-3):

A = 1 1 0 0 1 0 0 0
B = 0 1 1 0 0 0 0 0
C = 1 1 1 0 1 1 0 0.

Definition: (an **order of a schema**)

The order of a schema , *o*(S), is the number of definite bit positions .
For the schema given in (3-3), *o*(S) = 4.

Definition: (a **defining length of a schema**)

The defining length of schema S, δ (S) is the distance between the leftmost (b_{left}) and rightmost (b_{right}) bit positions holding either a 0 or a 1. δ (S) is calculated as

$$b_{right} - b_{left} .$$

For our schema example (3-3) , δ (S) = $b_{right} - b_{left}$ = 8-2= 6.

Implicit Parallelism of GA

Suppose that the strings are length l . Each string represents 2^l schemata. For example, the string 1101 has length equal 4, and therefore represents 2^4 =16 following schemata:

1100, 110*, 11*0, 11**, 1*00, 1*0*, 1**0, 1***, *100, *10*, *1*0, *1**, **00, **0*, ***0, 0***, ****.

If there is a population of *n* string, the total number of schemata, N_S is satisfy the following boundaries:

$$2^l \le N_S \le n 2^l .$$

Since each string can represent many schemata, it means that GA operations defined on a population of strings process a much larger number of schemata in parallel. This property is called *implicit parallelism of GA*.

Consider the following task.

Let some schema S has $n(S,t)$ representative strings in a population at time t. We will calculate how many representatives of the given schema will appear in the next generation, i.e.:

$$n(S,t+1) = ?$$

This number depends upon the operations of reproduction, crossover, and mutation. The answer on the given question is given by the following fundamental theorem proved by Holland.

Schema Theorem: Fundamental Theorem of GA.

Schemata grow exponentially if they have high fitness values, short defining length, and low order, i.e.

$$n(S,t+1) \geq \left[n(S,t) f(S) / f(P)\right]\left[1 - p_c \delta(S)/(l-1)\right]\left[1 - o(S) p_m\right]$$

where:

$n(S,t)$ is the number of representatives of schema S at time t;

$f(S)$ is the average fitness function for schema *S*;

$f(P)$ is the average fitness function over the population;

p_c is the crossover probability; l is a length of a string;

$\delta(S)$ is the defining length of schema *S*;

$o(S)$ is the order of *S*;

p_m is the probability of mutation at any bit position.

Main conclusion of the theorem: *population representatives with higher fitness values grow exponentially in a time. The theorem proves the convergence of GA solution to global optimum.*

Differences between Classical Methods of Optimization and GA

1) Classical optimization techniques represent a class of *derivative - based methods*. This means that for optimal solution finding a fitness function's derivative information is needed.
   GA represents derivative-free optimization. This means that for optimal solution's searching the fitness function's derivative information is not needed. In this case GA can be used both for continuous and discrete optimization problems.
2) Classical methods operate with parameters of a given problem. *GA operates on a coding of the parameters*. Thus, the parameters of the problem must be encoded in finite length strings. The string may be a sequence of any symbols. The binary symbols "0" and "1" are often used in GA. A selection (choice) of a coding method is very important.
3) GA optimization is performed on a set of string by using probabilistic mechanism and fitness measure for a string.

4) GA can operate without any knowledge of a search space.
5) GA provides a means to search an optimal solution in poorly understood and irregular spaces.

3.1.3 GA-based FC design

Consider the task of GA-based FC-PID design as follows: by using GA optimization select k_p, k_i, k_d parameters of a PID controller *such that the response of the plant will be desired* [5,18,19,20]. The block diagram of GA-based PID gains optimization is shown in Fig. 3-3.

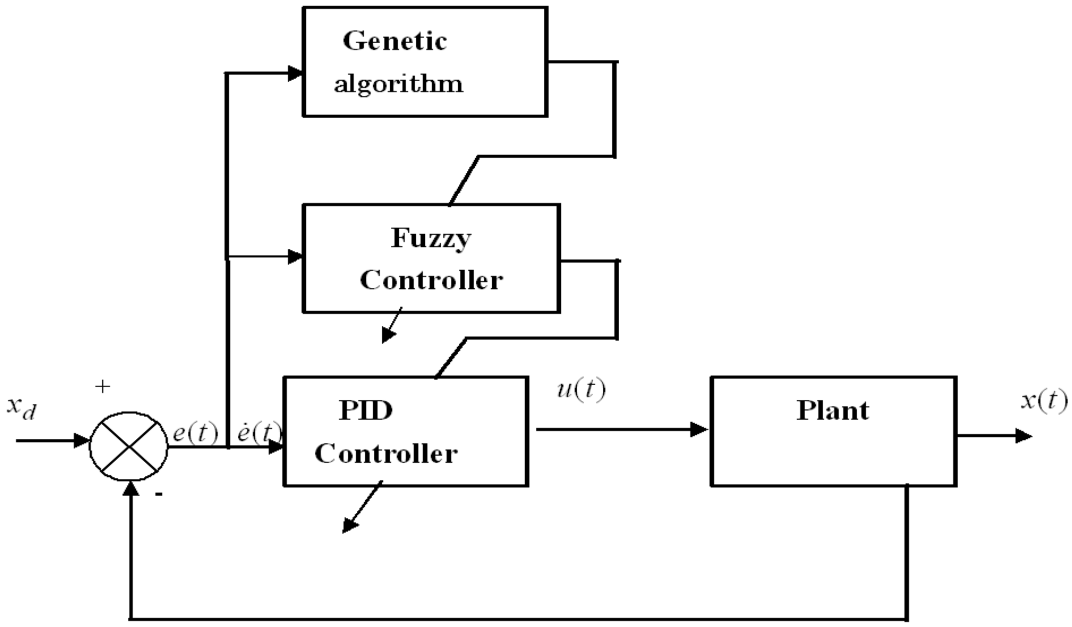


Figure 3-3. Block-diagram of GA-based optimization of a fuzzy-PID controller

In order to apply GA we must also define fitness function for our task. We introduce the following fitness function *F* as the reciprocal of the squared control error:

$$F = \frac{1}{1+e^2}.$$

In this case larger values of the fitness will correspond to a lower control error and, hence, a better performance of PID. Since GA works with a coding of parameters, let us describe a coding method. Without any loss of generality, we can assume that there are N_1, N_2, N_3 bits for each value of k_p, k_i, k_d. In this case, the size of chromosome S representing set of PID-parameters is defined as $(N_1 + N_2 + N_3)$ bits. Let us express *S* as $S = S_1 S_2 S_3$, where $S_i, i = 1,2,3$ are the following strings:

$$S_1 = \underbrace{00...001}_{N_1 \ bits},\ S_2 = \underbrace{00...010}_{N_2 \ bits}\ \text{, and }\ S_3 = \underbrace{00...011}_{N_3 \ bits}\ ,$$

Consider how to define the number N_i of bits for coding one PID-parameter.

Let $k_p \in \left[k_p^{\min}, k_p^{\max}\right] \subset R$, $k_d \in \left[k_d^{\min}, k_d^{\max}\right] \subset R$, $k_i \in \left[k_i^{\min}, k_i^{\max}\right] \subset R$, where R is a set of all real numbers.

For each interval $\left[k_p^{\min}, k_p^{\max}\right]$ let us obtain the corresponding *grid interval* by the following way. Define the distance $\Delta x_i, i = 1,2,3$, between two neighbor points in the grid. This distance is called GA discretization step. Then the total number of points in this grid is calculated as:

$$\frac{k_{p(i,d)}^{\max} - k_{p(i,d)}^{\min}}{\Delta x_i}.$$

To select best value (from the given fitness function point of view) from interval, for example, $k_p \in \left[k_p^{\min}, k_p^{\max}\right] \subset R$, GA investigates all N_1 points in the grid obtained by dividing the interval $\left[k_p^{\min}, k_p^{\min}\right]$ on Δx_1. The length of a binary chromosome for coding this number is calculated as:

$$N_i = \left\{ \underbrace{Fix}_{\text{integer-part}} (\log \frac{k_{p(i,d)}^{\max} - k_{p(i,d)}^{\min}}{\Delta x_i} \right\} + 1.$$

Simple decoding of binary (i.e. with base two) numbers $a_{i1},...,a_{il_x}$ to decimal (with base ten) number takes the following form:

$$Y^i(a_{i1},...,a_{il_x}) = k_i^{\min} + \frac{k_i^{\max} - k_i^{\max}}{2^{l_x} - 1} \cdot \left(\sum_{j=0}^{l_x - 1} a_{i(l_x - j)} \cdot 2^j \right),\ \text{where}\ l_x = N_i\ (i = 1,2,3).$$

After coding is defined, we can apply GA to solve our control task as follows. We design the block called SSCQ (simulation system of control quality) based on GA and put this block into our stochastic simulation system based on Matlab (see Chapter 6 below). The Matlab/Simulink structure of GA-based optimal control searching is shown in Fig.3-4.

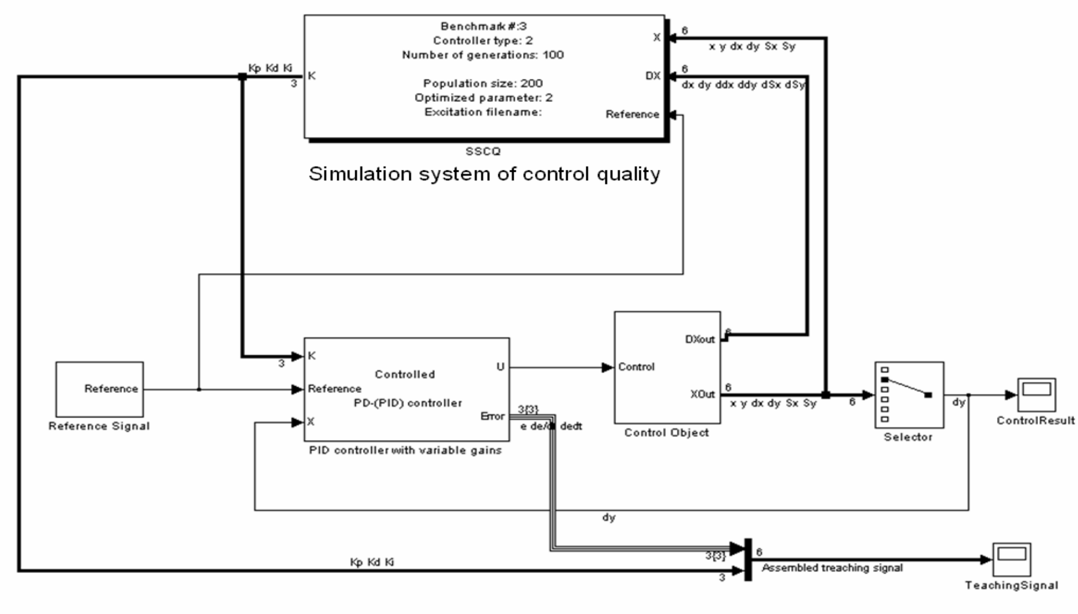


Figure 3-4. Simulink structure of GA-based control optimization

Obtained by GA optimal control laws $\{k_P(t_i),\ k_D(t_i),\ k_I(t_i)\}$, where $t_i = 1,...,n$ are time moments, and corresponding error's values in the form of a teaching signal (TS) are applied as input to Fuzzy Neural Network (FNN). Example of TS is shown in Fig.3-5.

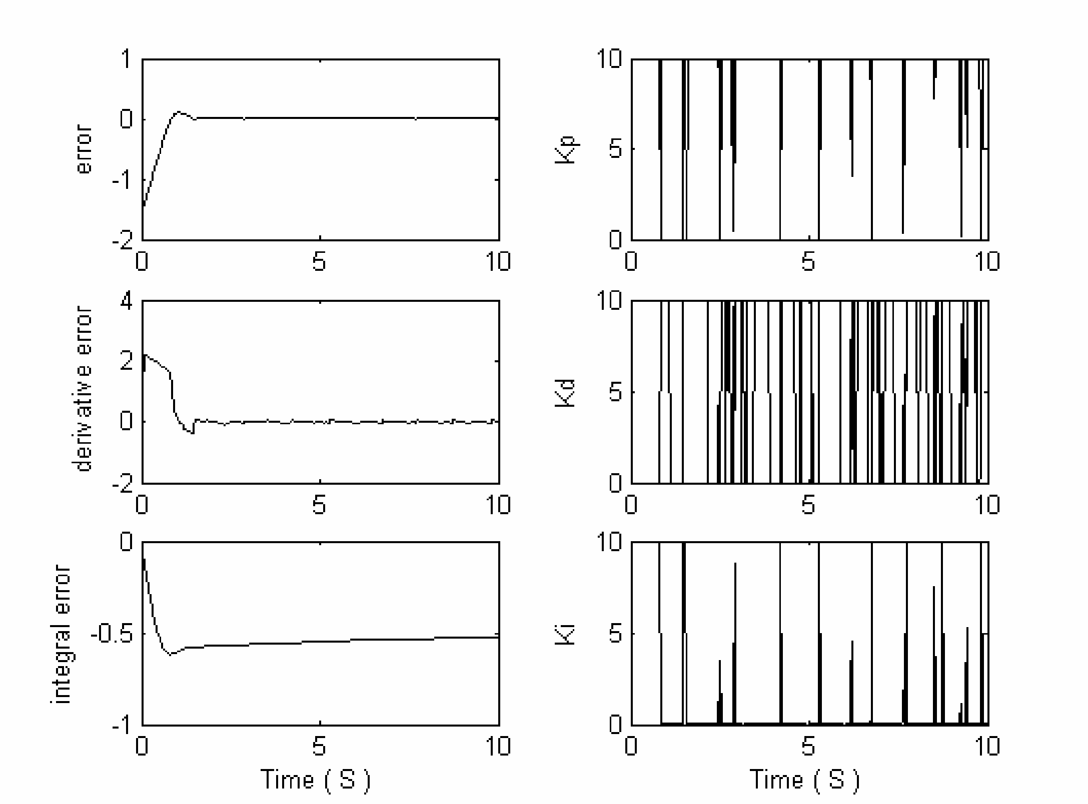


Figure 3-5. Teaching signal example

4. Fuzzy Neural Networks for physical implementation of FC and its local optimization

In practice, it is difficult to realize physically control laws obtained by GA. Moreover, TS doesn't contain knowledge information in direct form. Therefore, we need in one more step: an extraction of KB FC from TS obtained by GA.

For this aim two approaches are used:

- Traditional soft computing approach based on Fuzzy Neural Networks (FNN) tuning with error back propagation algorithm (step 1 technology);

and

- SC Optimizer tools for TS approximation and KB FC optimization (step 2 technology).

In next Chapters steps 1 and 2 of ICS design technology are described, simulation results are shown and analyzed.

4.1 FNN implementation of fuzzy control

For the task of physical implementation of FC performance and its local optimization (fine tuning), we will use the methodology of Fuzzy Neural Networks (FNN) [21-24].

FNN belong to a more general class of networks called adaptive networks. A general structure of FNN as adaptive network is shown in Fig. 4-1. This network has L layers and every layer $l(l = 0,1,2,...,L)$ has $N(l)$ nodes. 0-layer represents the input layer. The output of a node depends on the input signals and the parameters set of the node. Denote the output of node i in layer l as $y_{l,i}$:

$$y_{l,i} = f_{l,i}(y_{l-1,1},...,y_{l-1,N(l-1)},\alpha,\beta,\gamma),$$

where α,β,γ are the parameters of $f_{l,i}$ activation function of the node.
The nodes in each layer are independent.

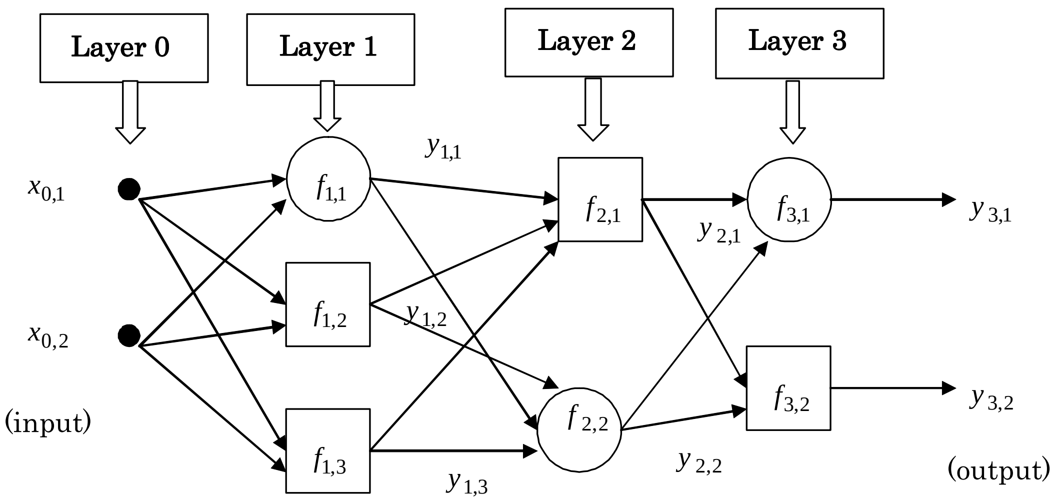


Figure 4-1.General architecture of FNN as adaptive network

In Step 1 technology, we use supervised method of FNN tuning by using TS. Consider briefly main ideas of this process.
Introduce one example of FNN structure for Sugeno fuzzy inference system (FIS) as shown in Fig.4-2 and describe its layers structure.
Denote the output of node i in layer l as $y_{l,i}$.

Layer 1
Every node i in this layer has output defined by $y_{1,i} = \mu_{A_i}(x)$, for $i =$ 1,2, and $y_{1,i} = \mu_{B_{i-2}}(y)$ for $i =$ 3,4, where x and y is the input to the node, $\mu_{A_i}(x)$ and $\mu_{B_{i-2}}(y)$ are membership functions .

Layer 2
Every node i in this layer has output (a product of two fuzzy values) defined by:

$y_{2,i} = w_i = \mu_{A_i}(x) \times \mu_{B_i}(y)$, i =1,2.
Thus, each node in layer 2 represents the firing strength of the rule.

Remark: In this example we use the product operation for fuzzy AND.

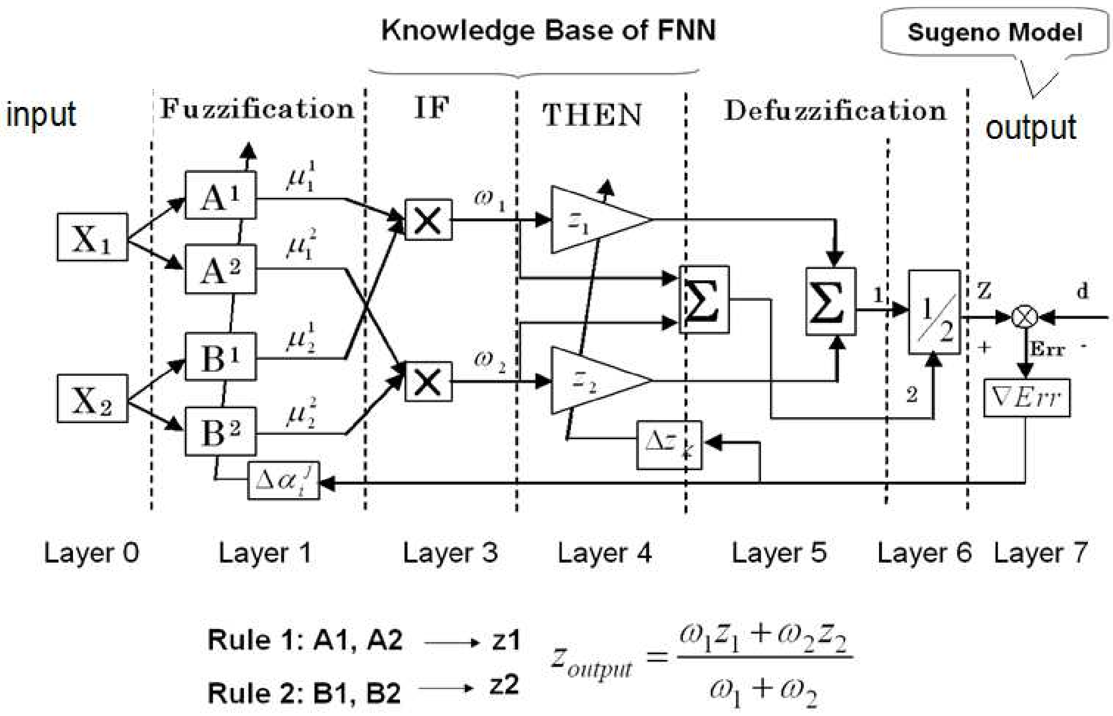


Figure 4-2. FNN structure for Sugeno FIS

Layer 3
Every node i in this layer has the following output:

$$y_{3,i} = w_i z_i, \ i = 1,2.$$

Layer 5
Every node i ($i = 1,2$) in this layer computes the overall output and has the following output:

$$y_{5,1} = w_1 + w_2; \quad y_{5,2} = \sum_i w_i f_i = w_1 z_1 + w_2 z_2 \ .$$

Layer 6
One node $i, i = 1$ in this layer computes the overall output and has the following output:

$$y_{5,1} = overal - output = \frac{\sum_i w_i f_i}{\sum_i w_i}.$$

Layer 7 is output layer**.** Here also is shown the level of error approximation calculation for error-back propagation algorithm (see below).

4.2 **Supervised FNN learning**

A learning (or training, tuning) process in FNN is a process of mapping between output and input data. There are two main learning paradigms: *supervised* and *unsupervised.*
In *supervised learning*, the network is provided with a correct output for every input pattern, and then node's parameters are determined as close as possible to the known correct output (Fig.4-3).

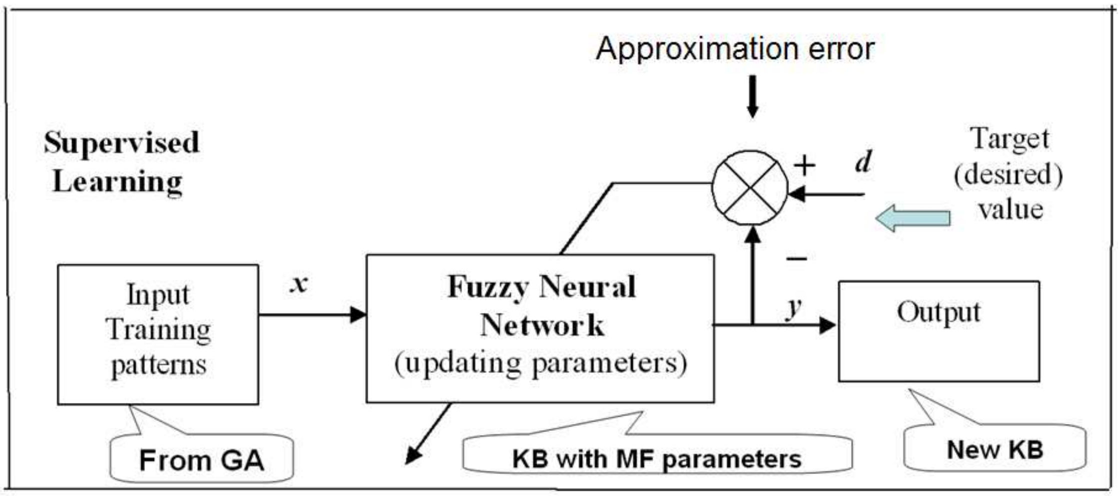


Figure 4-3. Supervised learning scheme in fuzzy neural networks

Let us formulate *FNN supervised learning task* as follows.

By using training input-output patterns (TS), update FNN-parameters (membership functions parameters of left-rights parts of rules) so that the total *approximation error measure*

$$E = \sum_k \frac{1}{2}\left(d^k - f(\vec{x}(t_k))\right)^2 \qquad (4\text{-}1)$$

is *minimal.*
The following designations are used in Eq.(4-1): t_k is a time moment; d^k is desired output value at time t_k; $\vec{x} = (x_1, x_2, ..., x_n)$ is a vector of input variables to FNN; $\vec{x}(t_k)$ inputs values at time t_k; $f(\vec{x}(t_k))$ are real outputs values at time t_k; $f(\vec{x}(t_k))$ is calculated according to a chosen FIS.
The central idea of a learning rule for FNN concerns how to obtain learning algorithm in which the error measure is minimal with respect to FNN parameters. For the construction of FNN updating (learning) algorithm we will use a *gradient descent- based learning rule*.

The update (learning) rule for the parameter α is expressed as follows:

$$\Delta\alpha = -\eta \frac{\partial E}{\partial \alpha} \qquad (4\text{-}2)$$

where η is the learning rate coefficient.

A so called *error back-propagation algorithm* applies given above learning rule (Eq.4-2). By using this learning algorithm, parameters of membership functions of given FNN are adjusted. Describe briefly this tuning process.

4.2.1. FNN parameters: example

Consider, for example, Sugeno-0 FIS with fuzzy rules as follows:

IF x_1 is $\mu_{j_1}^{(l)}(x_1)$ AND x_2 is $\mu_{j_2}^{(l)}(x_2)$ AND … AND x_n is $\mu_{j_n}^{(l)}(x_n)$ THEN $y = z^l$,

where l is a rule index, $l = 1, 2, ..., M$ (number of fuzzy rules).

Let input-output fuzzy variables are described by Gaussian membership functions as shown in Fig. 4-4:

$$\mu_{j_i}(x_i) = 1 \cdot e^{-\frac{(x-\overline{x}_i)^2}{\sigma_i^2}}.$$

We can calculate real output values of FNN by using Sugeno-0 fuzzy model as follows (see chapter 2.2.5):

$$y = f(\vec{x}) = f(x_1, ..., x_i, ..., x_n) = \frac{\sum_{l=1}^{M} z^l \prod_{i=1}^{n} \mu_{j_i}^l(x_i)}{\sum_{l=1}^{M} \prod_{i=1}^{n} \mu_{j_i}^l(x_i)}.$$

Thus, for the chosen FIS, we can define the following FNN parameters for tuning: $\overline{x}_i^l, \sigma_i^l$ (parameters of membership functions in left part of a fuzzy rule), and z^l (crisp parameter (real number) in right part of a fuzzy rule).

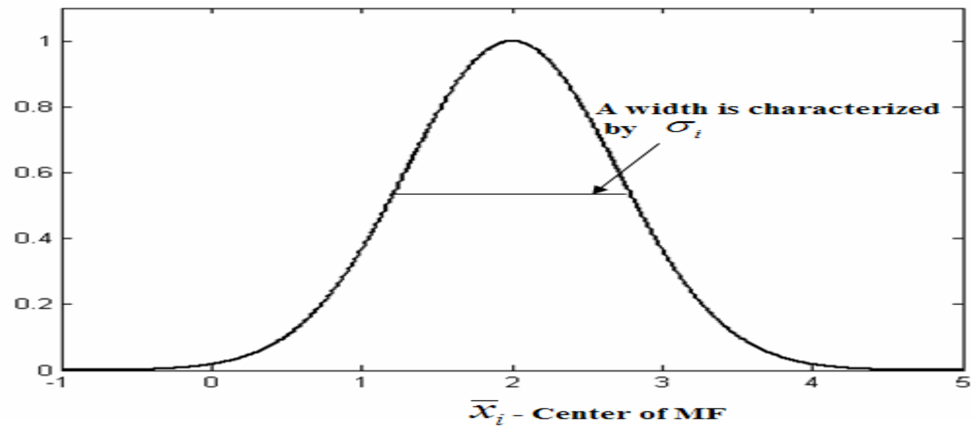


Figure 4-4: *Example of MF parameters for tuning task*

4.2.2. Error back propagation algorithm for FNN parameters tuning

The block-scheme of error back propagation algorithm for FNN tuning is shown in Fig.4-5, where *N* is a number of points in TS, *M* is a number of learning iteration steps.

Thus, finally, the output of the FNN learning process is a new knowledge base of FC with modified parameters of membership functions in left-right parts of fuzzy rules. It means that by error back propagation FNN tuning we realized a local optimization of KB FC.

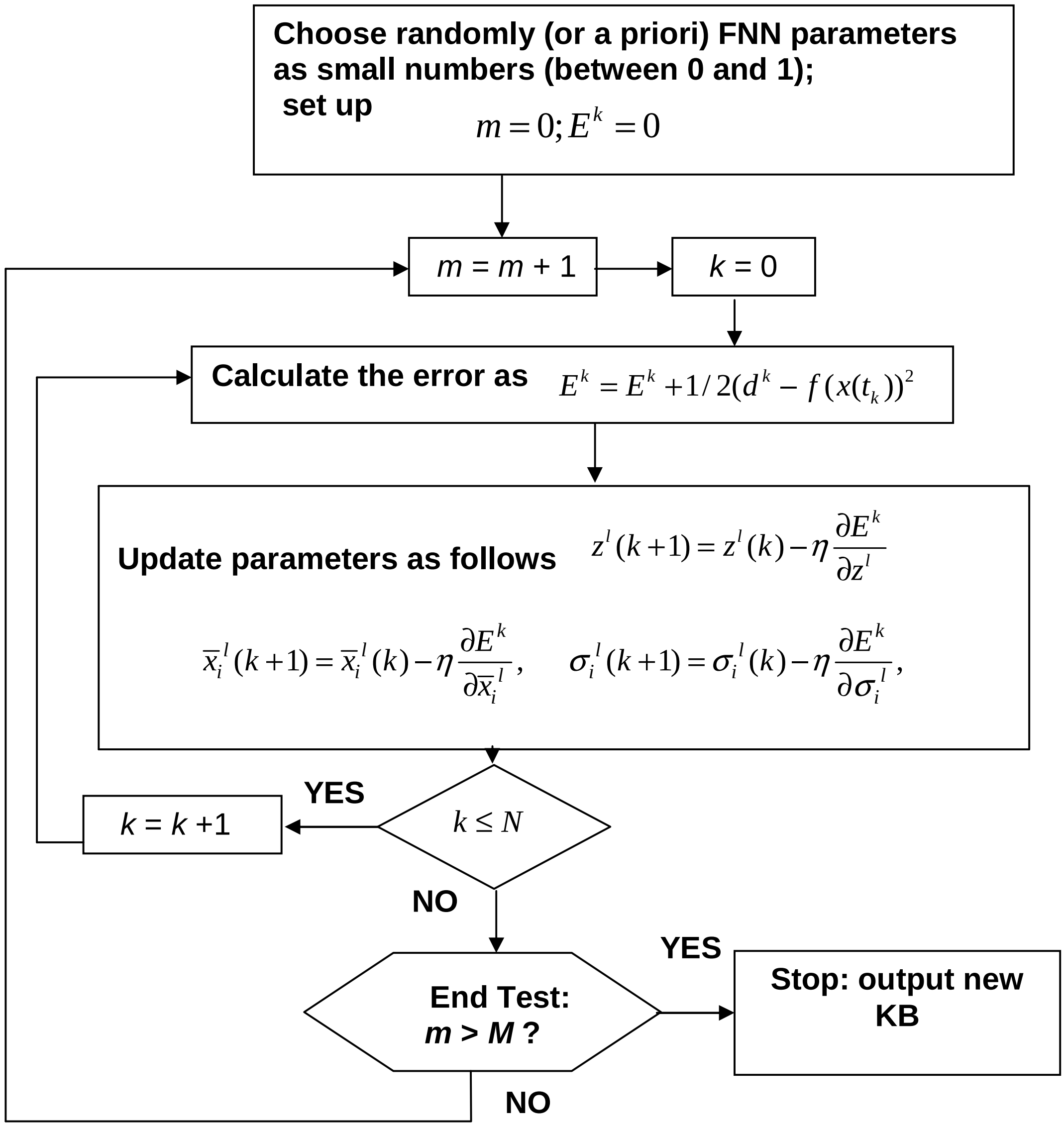


Figure 4-5. The block-scheme of error back propagation algorithm

Then we must test obtained KB FC by using CO model or in real experiment. If the test result of FC performance is sufficient, then obtained FC can be used online to control real control object. If the simulation result of FC fails, repeat simulations with new fitness function of GA for teaching signal design or new steps of design technology must be applied.

5. Physical criterion of optimization based on thermodynamic approach

For design GA fitness functions we use an information-thermodynamic approach based on the analysis of control object (CO) dynamic behavior and fuzzy PID-controller (FC) [9,25,26]. Principle of minimum of entropy production in CO and in FC is the background for design of intelligent robust control. Robustness of control means that the minimum of initial information about uncertainty of external environments or structure's disturbances of control object is required.

In our approach we consider thermodynamic criterion (the positive value of entropy production rate) as a physical measure for the realization of mathematical model of a dynamic system. This criterion indicates the necessity to put extra (thermodynamic) limitations on the parameters of differential equations and on qualitative properties that describes the dynamic evolution of systems.

Let us consider the interrelation between a Lyapunov stability (function V) and entropy production of dynamic system for two cases: closed dynamic system and open dynamic system.

From classical mechanics standpoint, a description of CO is based on two approaches: 1*)* *Lagrange's equations*; 2) *Hamilton's equations*.

We consider first approach for the problem of entropy production rate definition and calculation.

The *Lagrange's Approach.* Let us consider the Lagrangian equations

$$\frac{d}{dt}\left(\frac{\partial K}{\partial \dot{q}_i}\right)-\frac{\partial U}{\partial q_i}+\frac{\partial Q_i}{\partial \dot{q}_i}=F_i\left(t\right) \tag{5-1}$$

where $L=K-U$ is Lagrangian of the dynamic system, $K=\frac{1}{2}\sum_{i,k=1}^{n}a_{ik}\dot{q}_i\dot{q}_k$ is the kinetic energy, and $U=\frac{1}{2}\sum_{i,k=1}^{n}b_{ik}q_iq_k$ is the potential energy of dynamic system, q_i is a generalized coordinate.

In linear algebra it is possible to define an operator A such that $q=A\xi$ or $q_i=A_{11}\xi_1+...+A_{in}\xi_n \ \left(i=1,2,...,n\right)$ and $T=\frac{1}{2}\sum_{i=1}^{n}a'_i\,\xi_i^2$; $U=\frac{1}{2}\sum_{j=1}^{n}b'_j\,\xi_j^2$.

From Eq. (5-1) for closed system we can obtain

$$\ddot{\xi}_i+f_i\left(\dot{\xi}_1,...,\dot{\xi}_n\right)+\omega_i^2\xi_i=0,(i=1,2,...,n). \tag{5-2}$$

The Newton's Eqs (5-2) include additive non-conservative friction forces $f_i\left(\dot{\xi}_1,...,\dot{\xi}_n\right)$.

Let us consider the Lyapunov function V (for $a'_i=1$ and $b'_j=\omega_j^2$) as the full energy ($V\equiv E$)

$$V = \frac{1}{2}\sum_{i=1}^{n}\dot{\xi}_i^2 + \frac{1}{2}\sum_{i=1}^{n}\omega_i\,\xi_i^2 = T + U = E \tag{5-3}$$

and

$$\frac{dV}{dt} = \sum_{i=1}^{n}\dot{\xi}_i\ddot{\xi}_i + \sum_{i=1}^{n}\omega_i^2\xi_i\dot{\xi}_i \tag{5-4}$$

After multiplication the Eq. (5-2) on $\dot{\xi}_i$ and summing up on index i from 1 to n we obtain the following equation:

$$\sum_{i=1}^{n}\dot{\xi}_i\ddot{\xi}_i + \sum_{i=1}^{n}\omega_i^2\dot{\xi}_i\xi_i = -\sum_{i=1}^{n}\dot{\xi}_i\,f_i\left(\dot{\xi}_1,...,\dot{\xi}_n\right) \tag{5-5}$$

From Eqs (5-4) and (5-5) it is follows that

$$\frac{dV}{dt} = -\sum_{i=1}^{n}\dot{\xi}_i\,f_i\left(\dot{\xi}_1,...,\dot{\xi}_n\right) < 0. \tag{5-6}$$

The entropy production rate (for closed system) according to 2nd law of thermodynamics

$$\frac{d_i S}{dt} = \frac{1}{T}\sum_{i=1}^{n}\dot{\xi}_i\,f_i\left(\dot{\xi}_1,...,\dot{\xi}_n\right) > 0. \tag{5-7}$$

From Eqs (5-6) and (5-7) we can obtain

$$\boxed{\frac{dV}{dt} = -\frac{1}{T}\frac{d_i S}{dt} < 0}. \tag{5-8}$$

Thus, we obtain the general interrelation between the Lyapunov function V (stability), the entropy production rate and the full energy of dynamic system.

This interrelation is one of general relations in vibration theory of dynamic systems.

From Eq.(5-8) it is follows that an infringement of the thermodynamic criterion of physical realization in right side of Eq. (5-8) inclines to instability of dynamic system and vice versa.

Interrelations between thermodynamic criterion of Lyapunov stability and robustness of intelligent control system. Let us consider the open dynamic system (under control). In this case dynamic control process can be described as follows:

$$\dot{q}_i = \varphi(q,t,u)\,, \tag{5-9}$$

where $\varphi(q,t,u)$ are equation of motion of controlled object, q is a vector of generalized coordinates and u is a control force.

According to generalized thermodynamic approach [3, 5], we can choose Lyapunov function V for this process as following:

$$V = \frac{1}{2}\sum_{i=1}^{n}q_i^2 + \frac{1}{2}S^2. \tag{5-9a}$$

S in Eq.(5-9a) is entropy production of an open system described by $\dot{q}_i$ and represented as

$$S = S_p - S_c\,, \tag{5-10}$$

where S_p is the entropy production of a plant (control object) and S_c is the entropy production of controller (fuzzy PID-controller).

Remark. According to the second Lyapunov method (dynamic systems stability conditions) a function V must be as following: (i) $V > 0$; and (ii) $\frac{dV}{dt} \le 0$ for $t \to \infty$.
It is easy check the first requirement for Eq. (5-9a).
According to the second requirement $\frac{dV}{dt} \le 0$, and by differentiation of V in Eq. (5-9) we can receive the following

$$\frac{dV}{dt} = \sum_{i=1}^{n} q_i \frac{dq_i}{dt} + S\frac{dS}{dt}.$$

While $\frac{dq_i}{dt} = \varphi(q,t,u)$ (from Eq. (5-9)) and $\frac{dS}{dt} = \frac{dS_P}{dt} - \frac{dS_C}{dt}$ (from Eq. (5-10)), as a result we have the following

$$\frac{dV}{dt} = \sum_{i=1}^{n} q_i \varphi(q,t,u) + \left(S_P - S_C\right)\left(\frac{dS_P}{dt} - \frac{dS_C}{dt}\right). \tag{5-11}$$

By using dynamic systems stability conditions as $\frac{dV}{dt} \le 0$, we have

$$\frac{dV}{dt} = \sum_{i=1}^{n} q_i \varphi(q,t,u) + (S_p - S_c)\left(\frac{dS_p}{dt} - \frac{dS_c}{dt}\right) \le 0. \tag{5-12}$$

The interrelation between Lyapunov stability (*V*) and entropy of open dynamic system $(S \cdot \dot{S})$ described by Eq.(5-12) is the general physical law for design of intelligent control systems [8, 9 ,25, 26].

From Eq. (5-12) follows that $\min\,(S \cdot \dot{S})$ supplies Lyapunov criterion of dynamic systems stability. We will call $(S_p - S_c)(\dot{S}_p - \dot{S}_c)$ as a g*eneralized entropy production* and will consider $\min\,(S \cdot \dot{S})$ as a thermodynamic criterion of dynamic stability of dynamic system.

Remark. For the general class of dynamic control systems, described by Hamilton-Jacobi-Bellman equations, the optimal solution of the variational fixed-end problem for the maximum work *W* is equivalent to the solution of variational fixed-end problem for the minimum entropy production. Thus, the analytical formalism, which is strongly analogous to those in analytical mechanics and control theory, is effective in thermodynamic optimization too [6, 7].
The problem of maximum of released work, i.e., $\max\limits_{q_i,u} W$, where q_i, u are generalized coordinate and control force correspondingly, is equivalent to the associated problem of the minimum of entropy production [8, 9], i.e. $\min\limits_{q_i,u} S$. Thus, the principle of minimum of

entropy production in control object and fuzzy PID-controllers is the background for design of intelligent robust control.

Finally, in Table 5-1 interrelation between Lyapunov stability (function V) and entropy production of a system is shown for two cases: open dynamic system and closed dynamic system.

Table 5-1.Global Stability, Entropy Production & Controllability.

Closed System	**Open System**
$\frac{dV}{dt} = -\frac{1}{T}\frac{dS_P}{dt}$ $\dot{q}_i = \varphi(q_i, u, t)$ ***Equation of CO Motion***	$\underbrace{0 > \frac{dV}{dt}}_{\text{Stability Condition}} = \underbrace{\sum_i q_i \varphi(q_i, u, t)}_{\text{Mechanical Motion Part}} + \underbrace{(S_P - S_C)\left(\frac{dS_P}{dt} - \frac{dS_C}{dt}\right)}_{\text{Thermodynamic Behavior Part}}$ **Stability Condition** $\equiv$ **Mechanical Motion Part** $\oplus$ **Thermodynamic Behavior Part**

Thus, based on thermodynamic criterion of dynamic stability, we can introduce the following control quality criteria:

- Minimum of control error [control criterion]
- Minimum of $(S_p - S_c)(\dot{S}_p - \dot{S}_c)$ [thermodynamic criterion]
- Minimal complexity of control laws (i.e., minimum of control force and simplicity of k_p, k_d, k_i gain parameters). [control realization criterion].

We will use these control quality criteria in the process of robust smart KB FC design based on our tools.

6. Stages of SC technology development for Intelligent Control Systems Design

Figures 6-1 and 6-2 show main stages of our design technology of robust intelligent control systems (ICS) [6, 7, 8, 9].

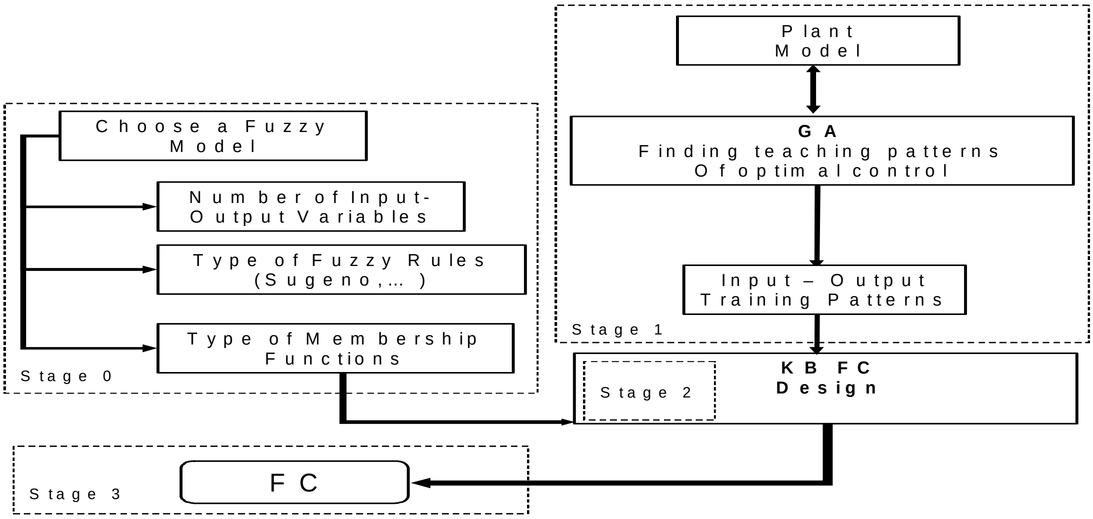


Figure 6-1. Flow chart of ICS design technology

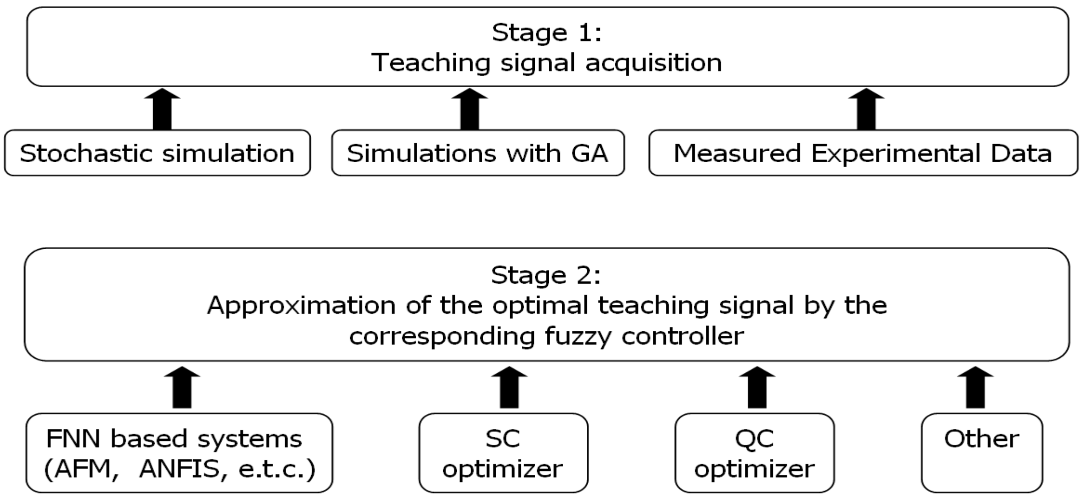


Figure 6-2. Main steps of ICS design technology

Our approach includes the following steps.

Stage 0: *Choice of a fuzzy inference system for the given control problem*

At this stage a user makes the selection of a fuzzy model with the following initial parameters:

1. Type of a fuzzy inference system (FIS);
2. Number of input and output variables.
3. Type of membership functions for input linguistic variables description.

Stage 1: *Teaching signal acquisition*
This step may be done by using measured experimental data from physical model of control object. Or it can be done by using our *stochastic simulation system* based on mathematical model of control object. The core of stochastic simulation system is GA based optimization of control.

By using GA, we obtain a set of optimal control values $\left(k_p, k_d, k_i\right)$, which minimize the selected physical characteristics of the stochastic model of control object. One of the characteristics can be control error, or the minimum of the entropy production rate of the control system and of the control object. In some complicated cases, the fitness function may include a weighted sum of different motion characteristics of the control object like accelerations, velocities, spectral characteristics. Thus the resulted motion under control will tend to reduce all of them simultaneously.
At this stage we find a solution $\{k_p(t), k_d(t), k_i(t)\}$ which is close to a global optimum. By joining optimal control values $\{k_p(t), k_d(t), k_i(t)\}$ with corresponding values of control error, we obtain a *teaching signal* (TS) of optimal control which is applied at the next stage.

Stage 2-1 (step 1 technology): *KB FC design based on traditional FNN approach*
At this stage, obtained by GA TS is applied as input to FNN. At the stage of FNN tuning, the *structure of FNN is specified by a user.* FNN approximates the given control by some fuzzy inference system. Then, by using error back propagation-based learning algorithm, we can extract information from the TS in the form of fuzzy rules and membership functions of input FC variables (FC knowledge base). Thus, the output from FNN is the FC KB.

For the Stage 2-1 realization, *Adaptive Fuzzy Modeler (AFM)* developed by ST Microelectronics [30] was used.

Stage 2-2 (step 2 technology): *KB FC design based on SC Optimizer*
Main disadvantage of FNN-based approaches is that the FNN structure must be given *a priori* (i.e., the number and type of MF must be introduced by a user), but sometimes it is difficult to define optimal FNN structure manually, especially in the case of unstable and essentially non-linear control objects. To avoid all above mentioned disadvantages we developed *Soft Computing Optimizer* (SCO) [6].

At Stage 2-2, obtained at Stage 1 TS is applied as input to SC Optimizer. SC Optimizer at first selects an *optimal FNN structure* for the given FIS by using GA based optimization with information-thermodynamic criteria of optimization. Then SC Optimizer optimizes KB FC and designs robust KB FC.

Stage 3: *KB FC verification and robustness investigation*
Test the developed control system (with obtained KB FC) on the model of the control object. If the test result of the model output is sufficient obtained FC can be used online to control real control object. If the simulation result of FC fails, the stages 0-3 must be repeated with correction of the fitness function of GA or steps including QC Optimizer

must be applied. At stage 3 we check the performance of developed FC from *control quality and robustness* point of view. The result of the Stage 3 is a specification of a fuzzy inference structure, which approximates an optimal solution of a current control problem.

For program realization of each stage the special program tools are used:

- For Stage 1, our stochastic simulation system based on GA (with corresponding C++ realizations of GA) optimization of optimal control is used.
- For Stages 0 and 2, SC Optimizer is developed. For comparison with traditional SC approach based on FNN, Adaptive Fuzzy Modeler (AFM 1.0) is used.
- For Stages 1 and 3, Matlab with Simulink is used for control object simulation, and for FC control system simulation and optimization.

6. 1. Stochastic simulation system for TS design

The stochastic simulation system allows us to design *teaching signal* by extracting information from stochastic simulation of control object behavior [18].

Important peculiarities of our simulation system are:

- for stochastic noise generation we use methods of nonlinear forming filters based on Fokker-Plank-Kolmogorov equations developed in [25,27] (see also Appendix of the given report);
- for optimal control TS design we use GA based optimization based on mathematical model of given control object.

The Simullink-based structure of our stochastic simulation system for the Stage 1 is shown in Fig.6-3.

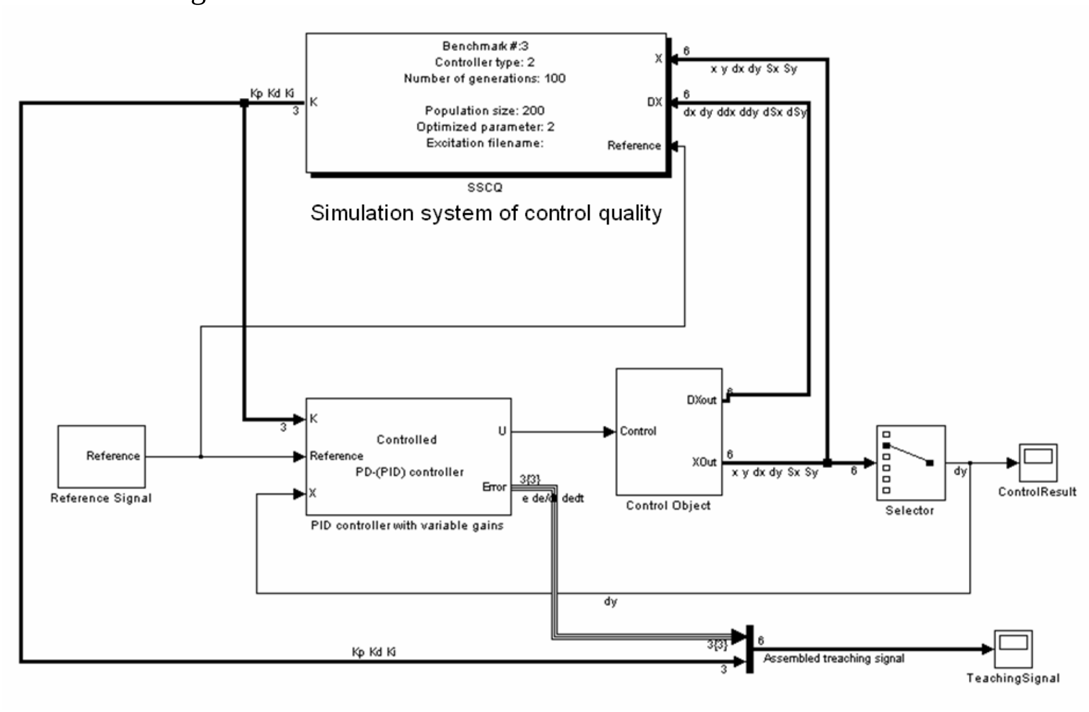


Figure 6-3. The Simulink structure of the simulation system for Stage 1

The system consists of the following basic components:

1. Subsystem with equations of motion for the given dynamic system (a control object);
2. Subsystem called "Simulation system of control quality" (SSCQ). SSCQ represents C++ implementation of GA with selector of fitness function, for example, it could be a "minimum of control error" or "minimum of the entropy production rate", or their combinations;
3. Reference signal source. Depending on the control task, this block can generate static signals, or some variable set points, which the control object state must follow;
4. PID controller with variable gain schedules (PID controller with variable gains) which generated control force as a first output and control errors as a second output;
5. Selector block for the selection from the output of the control object the variable to which the control force is applied;
6. Scope for the simulation result output;
7. Scope for teaching signal output, which outputs error signal multiplexed with the output of the genetic algorithm block SSCQ.

The core of the simulation system is the GA embedded into SSCQ block. It is a discrete time block, and there are following different sampling time depending on current step of GA performance:
T - moments of SSCQ calls; T^c - sampling time of control system; T^e - evaluation time of SSCQ.

SSCQ is realized as C++ Simulink S-Function, containing:

- Equations of motion of all benchmarks;
- Object for adaptive P(I)(D) controller simulation;
- Genetic algorithm;
- Fixed step Runge-Kutta (R-K) integrator;
- GA Search space definition;
- Evaluation block for different fitness function selection;
- Module for read < *.mat> excitation files for the simulation of the excited dynamics.

The output of stochastic simulation system is a *teaching signal* (TS) (or training patterns) representing a table of 'in-out' patterns as follows:

in	*out*
$E_1(t_1)$	$K_1(t_1)$
$\vdots$	$\vdots$
$E_n(t_n)$	$K_n(t_n)$

,

where $E_i(t_i) = \left\{ e(t_i), \dot{e}(t_i), \int e(t_i) \right\}$ and $K_i(t_i) = \left\{ k_p(t_i), k_d(t_i), k_I(t_i) \right\}$.

6.2. Stage 2-1 (Step 1 Technology): Demonstration of efficiency of Soft Computing approach to design intelligent control with respect to advanced classical control

Design of FC at this step is based on traditional SC approach which is founded on GA search algorithm of optimal control and approximation of obtained by GA optimal control by FNN tuning with error back propagation algorithm. We call this step as step1 technology. A general structure of intelligent control system in step 1 technology is shown in Fig.6-4 below.

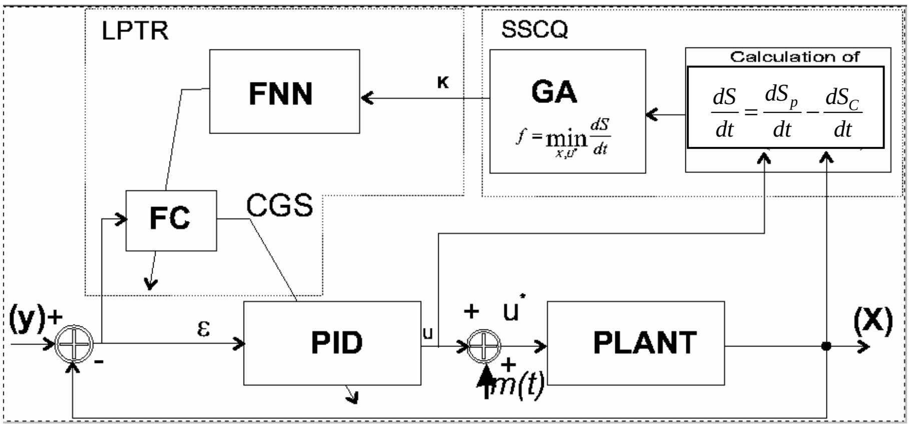


Figure 6-4. General structure of intelligent control system based on traditional soft computing

In Figure 6-4 we use the following designations: GA - Genetic Algorithm; f - Fitness Function of GA; S - Entropy production of System; S_c – Entropy production of Controller; S_p - Entropy production of Controlled Plant; ε - Error; u*- Optimal Control Signal; m(t) - Disturbance; FC - Fuzzy Controller; FNN - Fuzzy Neural Network; SSCQ - Simulation System of Control Quality; K - Global Optimum Solution of Coefficient Gain Schedule (Teaching Signal); LPTR - Look-up Table of Fuzzy Rules; CGS - Coefficient Gain Schedule $K=(k_p,k_d,k_i)$, (X) – current state of a plant, (y) – a reference signal (desired state of the plant).

In this chapter we show efficiency and limitations of traditional SC approach to design intelligent control systems [28]. Consider the following example.

Example 1: *Pendulum with variable length* (*Swing system*) *motion control problem* (Fig.6-5). The nonlinear equations of motion of the swing dynamic system are:

$$\begin{cases} \ddot{\theta}+2\frac{\dot{l}}{l}\dot{\theta}+\frac{g}{l}\sin\theta = k'_p\cdot e_\theta + k'_d\cdot\dot{e}_\theta + k'_i\cdot\int e_\theta dt+\xi_1(t)+u_1(t) \\ \ddot{l}+2k\dot{l}-l\dot{\theta}^2-g\cos\theta=\frac{1}{m}\left(k_p\cdot e_l+k_d\cdot\dot{e}_l+k_i\cdot\int e_l dt+\xi_2(t)\right)+u_2(t) \end{cases} \tag{6-1}$$

Here $\xi_{1,2}(t)$ are the given stochastic excitations with an appropriate probability density function, $u_1(t),u_2(t)$ are control forces. Equations of swing entropy production rate:

$$\frac{dS_\theta}{dt} = 2\frac{\dot{l}}{l}\dot{\theta}\cdot\dot{\theta}; \quad \frac{dS_l}{dt} = 2k\,\dot{l}\cdot\dot{l}\,.$$

Equation of Controller's Entropy Production rate: $\dfrac{dS_c}{dt} = k'_d\,\dot{e}_\theta^2 + k_d\dot{e}_l^2\,.$

Kinetic Energy: $T = \dfrac{\dot{\theta}^2}{2} + \dfrac{\dot{l}^2}{2}$,

Potential Energy: $U = \dfrac{g}{l}\cos\theta - \dfrac{l^2}{2}\dot{\theta}^2$,

Total Energy: $E = T + U$, $E = E_\theta + E_l$,

$$E_\theta = \frac{\dot{\theta}^2}{2} + \frac{g}{l}\cos\theta; \quad E_l = \frac{\dot{l}^2}{2} - \frac{l^2}{2}\dot{\theta}^2;$$

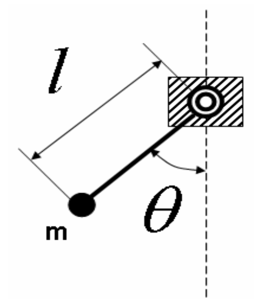


Figure 6-5. Physical model of pendulum with variable length

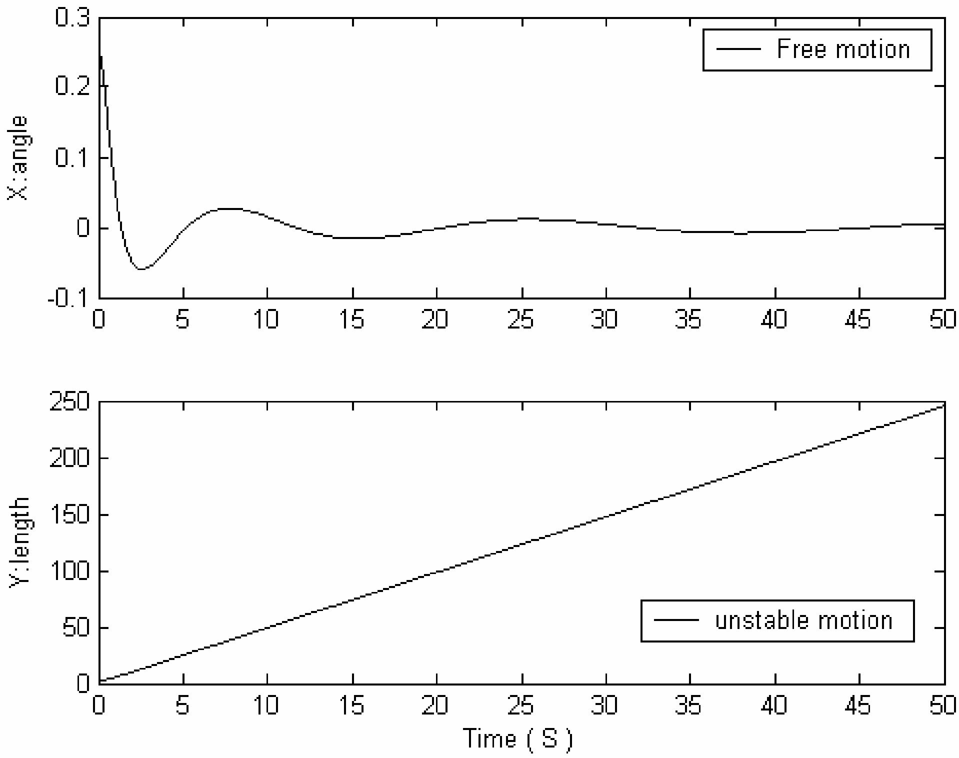


Figure 6-2. Free motion of the swing system

Simulation results in Fig.6-2 show that the system, described by Eq.(6-1), represents *globally unstable* (along generalized coordinate *l*) dynamic system.

Control task description

Consider excited motion of the swing system with following model parameters and initial conditions: m = 1, k = 1 and [$\theta_0 = 2.5$, $l_0 = 25$] [$\dot{\theta}_0 = 0$, $\dot{l}_0 = 0.01$]. The reference signals are as follows**:** $\theta_{ref} = \pi/4$; $l_{ref} = 30 + \sin(2\pi/50)t$. Let the system is disturbed by a stochastic harmonic noise (Fig.6-3):

$$\varsigma(t) = (A + \varsigma_A)\sin((\omega + \varsigma_\omega)2\pi t),$$

acting along l-axis, where A is a main amplitude, $\omega = 2\pi/50$ is a main frequency and $\varsigma_A, \varsigma_\omega$ are random parameters as «white noises».

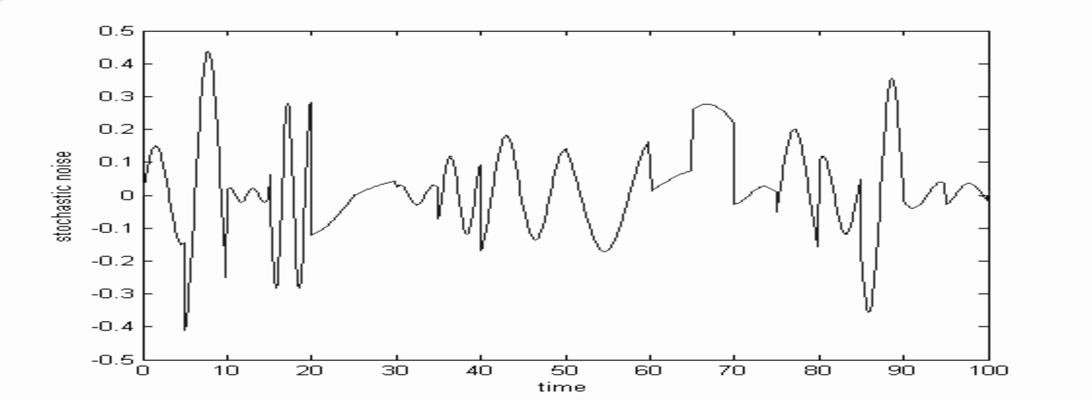


Figure 6-3. Stochastic harmonic noise

Consider swing motion under *three types of control*:

- two classical PID controllers (along θ and l-axes) with constant (k_p, k_d, k_i) - gains;
- GA-based control along θ and l-axes; and
- fuzzy control of two PID-controllers.

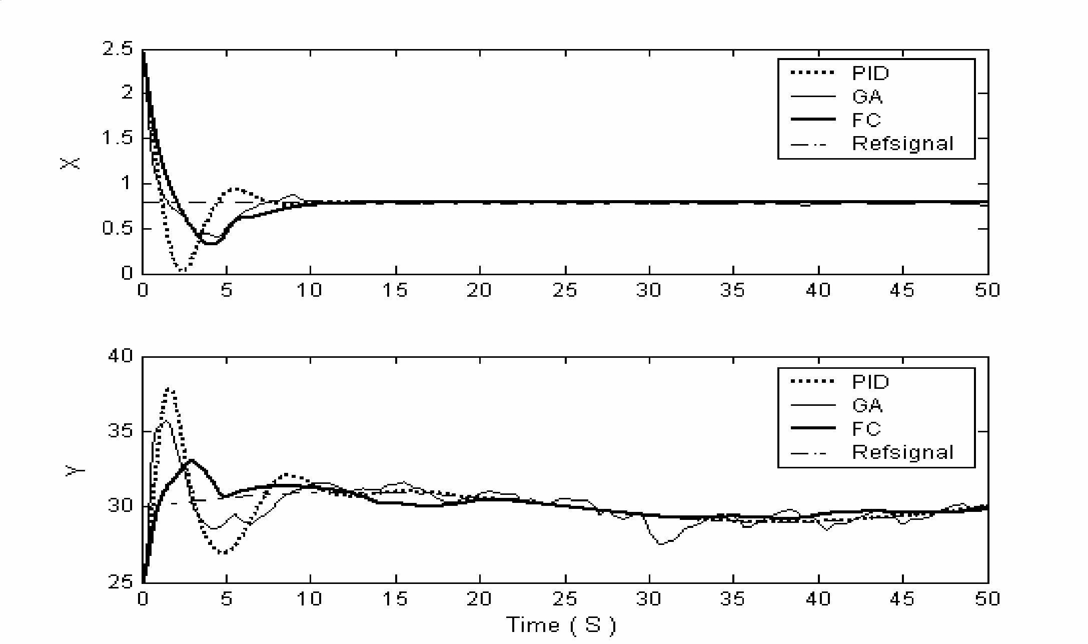


Figure 6-4. Coordinate motion of swing system

By using our stochastic simulation system of control quality based on GA, design for this control situation a teaching signal (TS). K-gains ranging area is [0,10] and GA fitness function is used to minimize control error. Then approximate TS by FNN using error back propagation algorithm. For this aim, in step 1 technology we have used Adaptive Fuzzy Modeler (AFM) developed by ST Microelectronics [8].

In Figures 6-4, 6-5, 6-6, 6-7 and 6-8 results of comparison of control quality obtained with advanced classical approach based on PID regulators, with GA-based control and with fuzzy control based on GA and FNN tuning with error back propagation algorithm (step 1 technology) are shown.

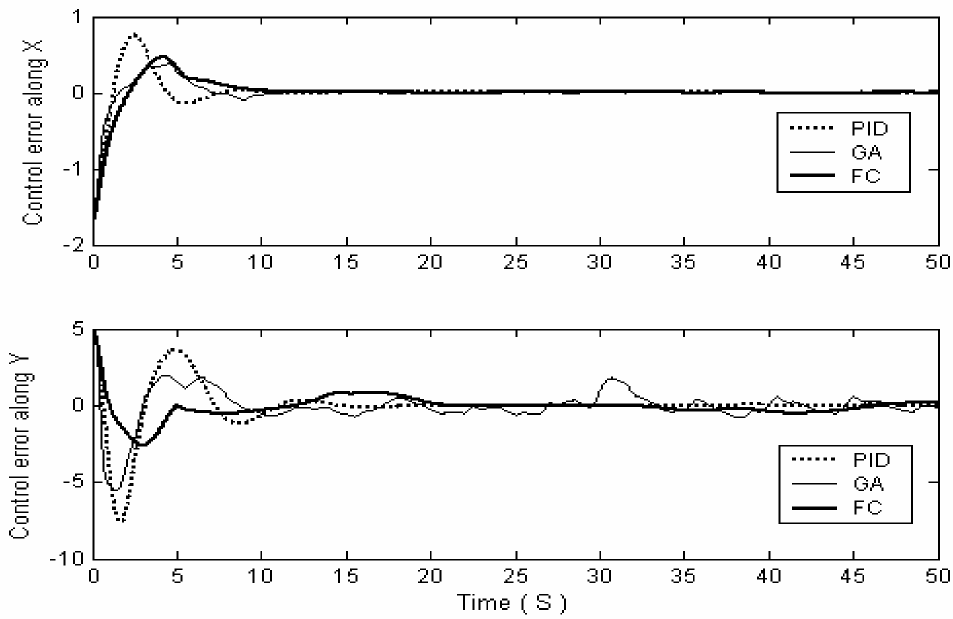


Figure 6-5. Swing system. Control error

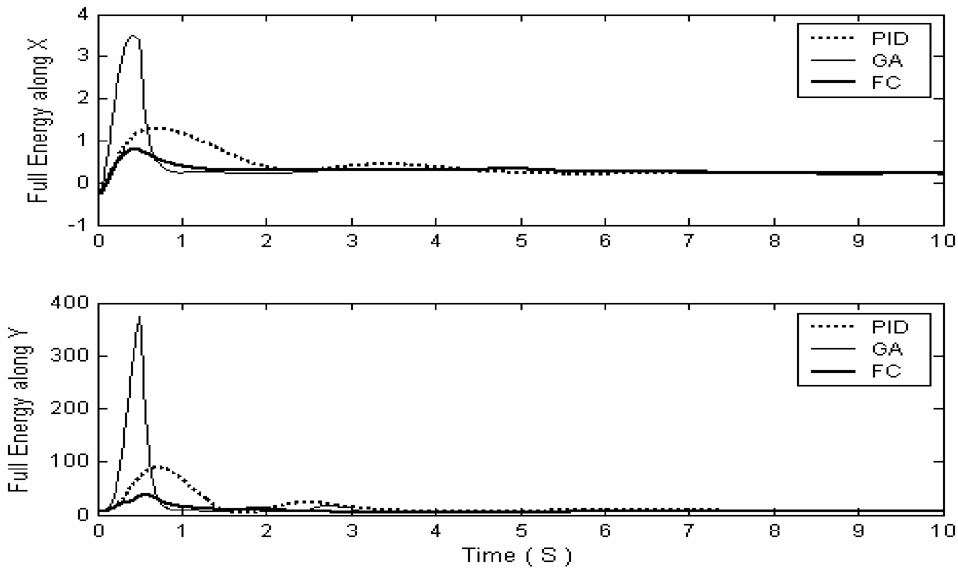


Figure 6- 6. Swing system. Full energy

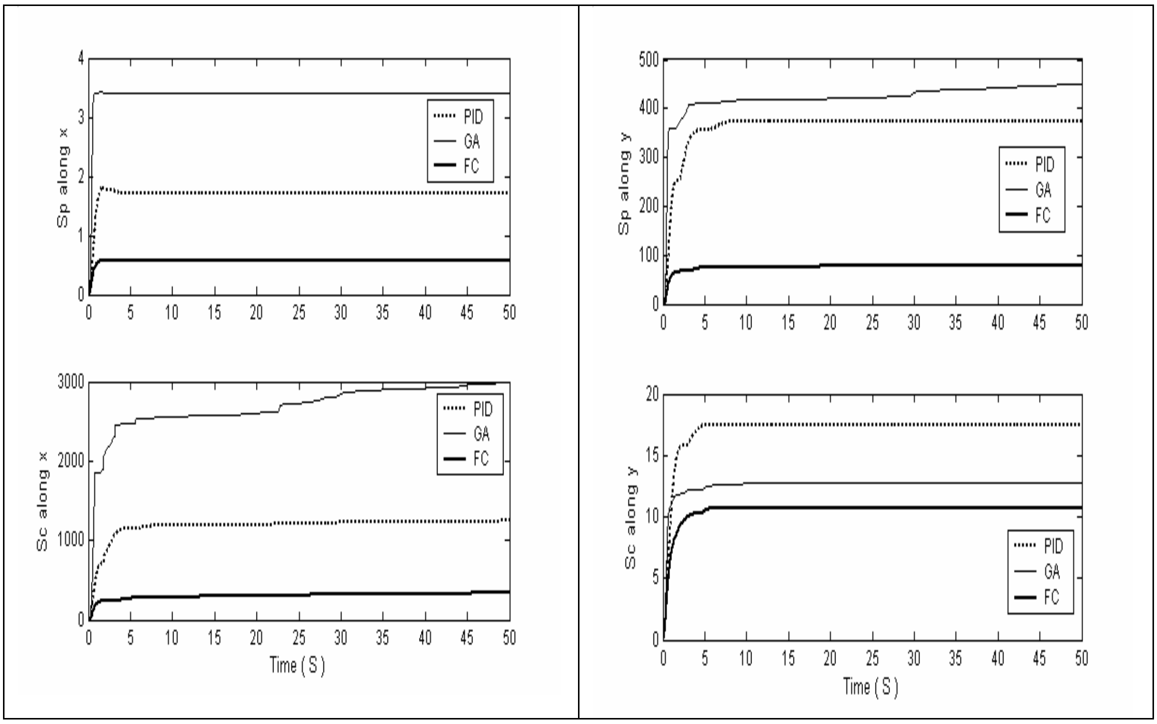


Figure 6-7. Swing system. Entropy production in a plant and in controllers along θ and l-axes

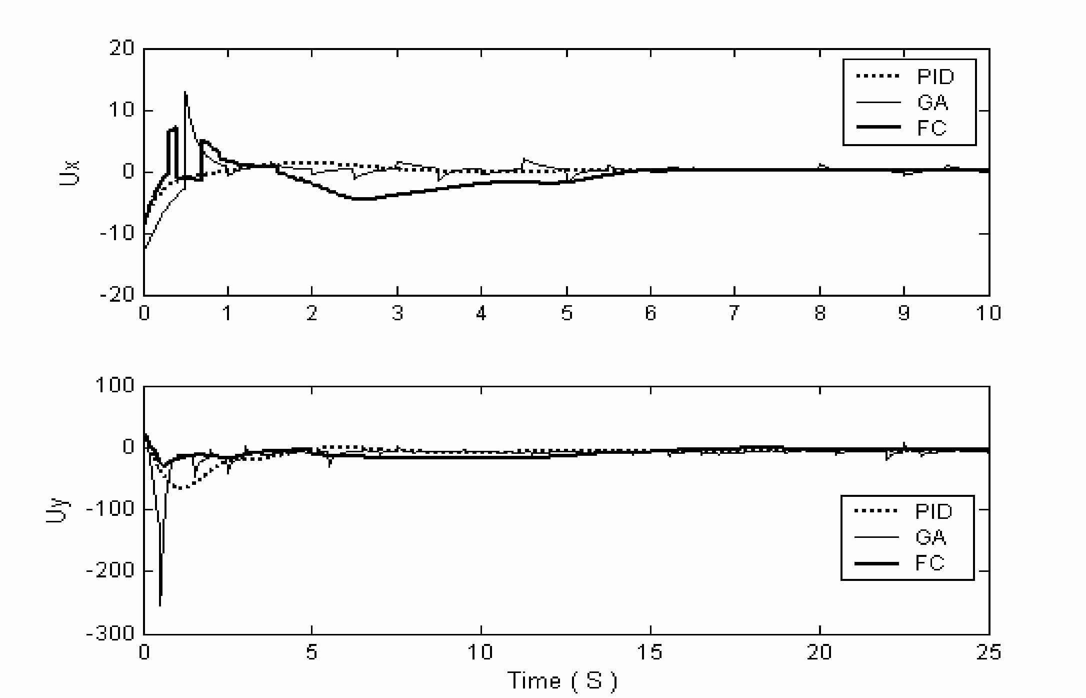
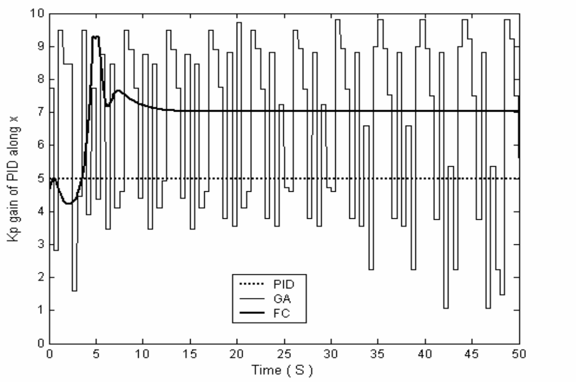
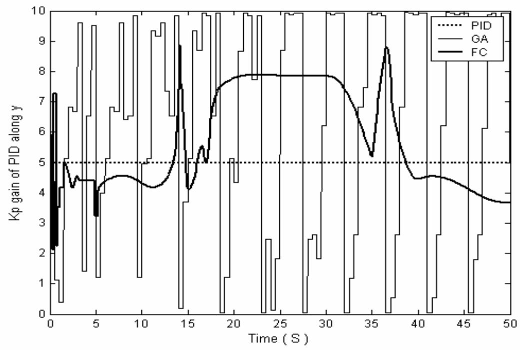
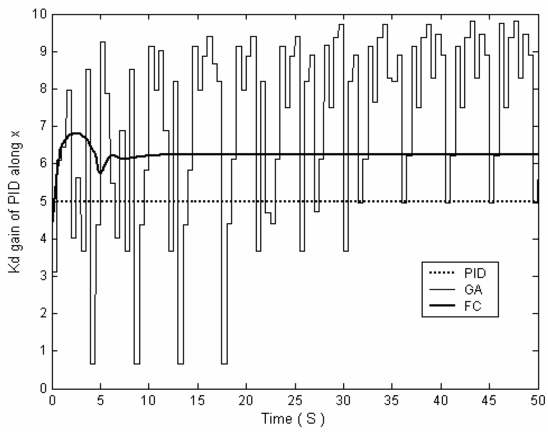
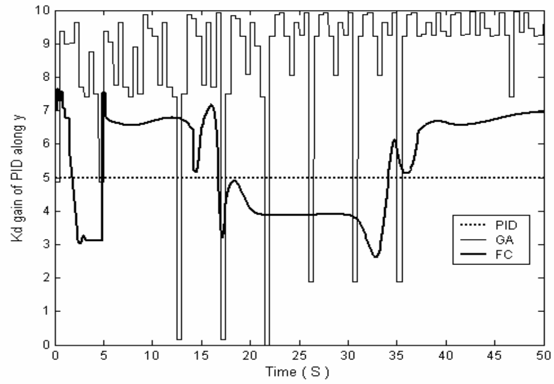
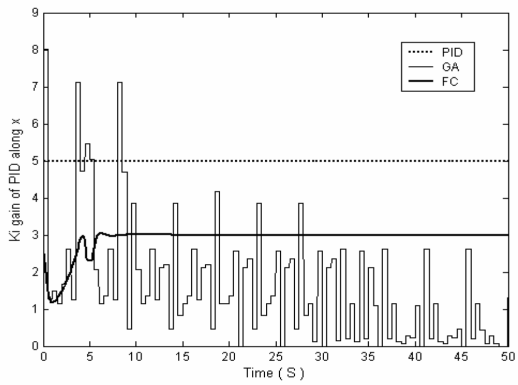
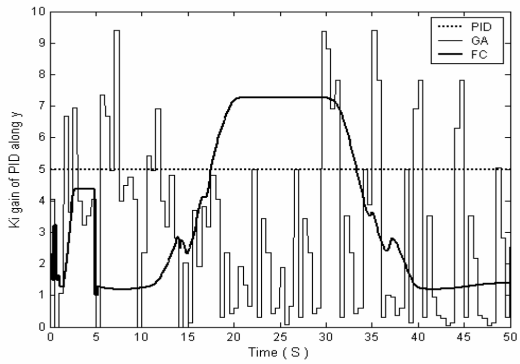


Figure 6-8: Swing system. Control force along θ and *l*-axes.

In Table 6-1 control laws along θ and *l*-axes are shown.

Table 6-1.

Swing system. Control laws along θ -axis. PID, GA and FC control	*Swing system*. Control laws along *l*-axis. PID, GA and FC control
	
	
	

Simulation results represented in Figures 6-4, 6-5, 6-6, 6-7 and 6-8 allow us to make the following *main conclusions*:

- **Control error** (along x and y axes) under FC control is *smaller* than under classical PID control;
- **Plant entropy production** along y-axis under FC control is 4 *times smaller* than under classical PID control; plant entropy production along x- axis is 3 *times*

smaller than in classical PID control; So, **total plant entropy** under FC control *is smaller* than under classical PID control;

- **Entropy production in FC-PID controller** (along y axis) is *2 times smaller* and FC-PID controller (along x axis) is 4 *times smaller* than in classical PID controller;
- **Total plant energy** under FC control is 2 times *smaller* than this energy in classical PID controller;
- **Control force in FC-PID controller** is *smaller* than in classical PID controller

Thus, according to the control quality criteria: minimum of control error, minimum of entropy productions in a plant and control system and minimum of control force - a *fuzzy control* with variable K-gains is more effective than advanced classical control with constant K-gains.

Main problem in the step 1 technology based on traditional SC approach lies in *robustness* of designed fuzzy controllers.

6.3 Robustness investigation of FNN based approach (step 1 technology)

Let us consider more complicated control problem, where swing system is disturbed by two different stochastic noises acting along θ and l-axes.
Let excitation along θ-axis is described by a *Gaussian* noise and excitation along l-axis is described by a *Rayleigh* noise (Fig.6-9). (Description of noises and noise modeler see in Appendix).

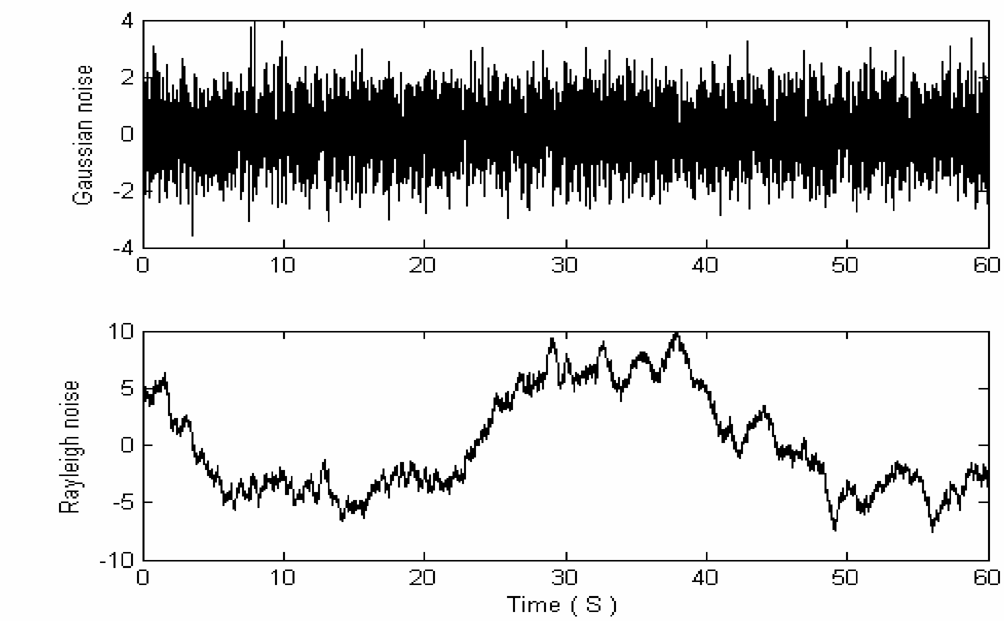


Figure 6-9. Stochastic noises: Gaussian and Rayleigh noises

Consider also the following swing parameters and initial conditions: m = 1, k = 1 and [$\theta_0 = 0.25,\ l_0 = 2.5$] [$\dot{\theta}_0 = 0,\ \dot{l}_0 = 0.01$].

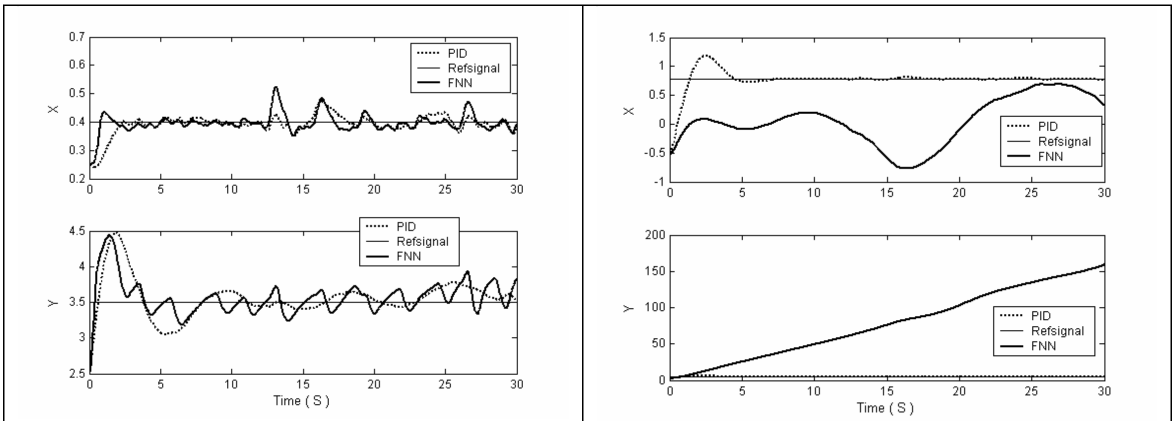
The reference signals are as follows: $\theta_{ref} = 0.4$; $l_{ref} = 3.5$.

By using simulation system of control quality based on GA, design for this control situation a teaching signal (TS). K-gains ranging area and GA fitness function are the same as in the case mentioned above. Then approximate TS by FNN tuning with AFM tools. As a result, we obtain FC developed for the given control task.

Let us investigate robustness properties of the developed FC. Consider two control situations: (a) TS control situation described above and (b) a new control situation.

The new control situation is described as follows: the same model parameters: m = 1, k = 1; *new initial conditions* [-0.52 (-30°), 2.5] [0.01, 0]; *new reference signals*: θ_{ref} = 0.78 (45^{0}); l_{ref} = 5; and *new noises amplitudes*: noise along θ: *Gaussian* noise with max amplitude A =1.5; and noise along length l: *Rayleigh* noise with max amplitude A = 1.
In case (b) max amplitudes of noises are 2.5 and 4 times smaller than in case (a).

In Fig.6-10 simulation results of swing motion under developed FC are shown in two control situations (a) and (b). You can see that in the case (b) CO dynamic motion under FC control is unstable along l-axis (y-axis).



(a) TS control situation

(b) new control situation

Figure 6-10. Swing system. FC performance in two control cases

Limitations of Step 1 technology based on traditional SC approach and main reasons of disadvantages

Simulation results represented in Fig.6-10 allow us to make the following *main conclusions* :

- In the case of more complicated random noises FC-FNN has not good control quality performance even in the TS control situation (a) ;
- FC-FNN incapable to control swing system when initial conditions, reference signals and random noises amplitudes are changed, i.e., *FC-FNN is not robust*.

Benchmarks simulations results show also main limitations of Step 1 approach. We can formulate them as follows. For globally unstable or essentially non-linear control objects in the following cases:

- changing of initial conditions and reference signals;
- changing of control object model parameters;
- presence of complicated stochastic excitations;
- presence of random noises in sensor's measurement systems –

Step 1 technology cannot guarantee robust and stable control achievement.

Main reasons of mentioned above disadvantages of traditional soft computing approach lie in the following:

- FNN structure must be given *a priori* (i.e., *the number and type of MF must be introduced by a user*), but sometimes it is difficult to define the optimal FNN structure manually.
- GA with fitness function as a minimum control error is not enough for design robust KB FC.

To avoid these disadvantages we developed Step 2 technology (for Stage 2-2) based on Soft Computing Optimizer [29].

6.4. Step 2 Technology based on SC Optimizer

SC Optimizer (SCO) is considered as a new flexible tool for design of optimal structure and robust KB of FC. SC Optimizer is based on a chain of GAs with information-thermodynamic criteria of optimization.

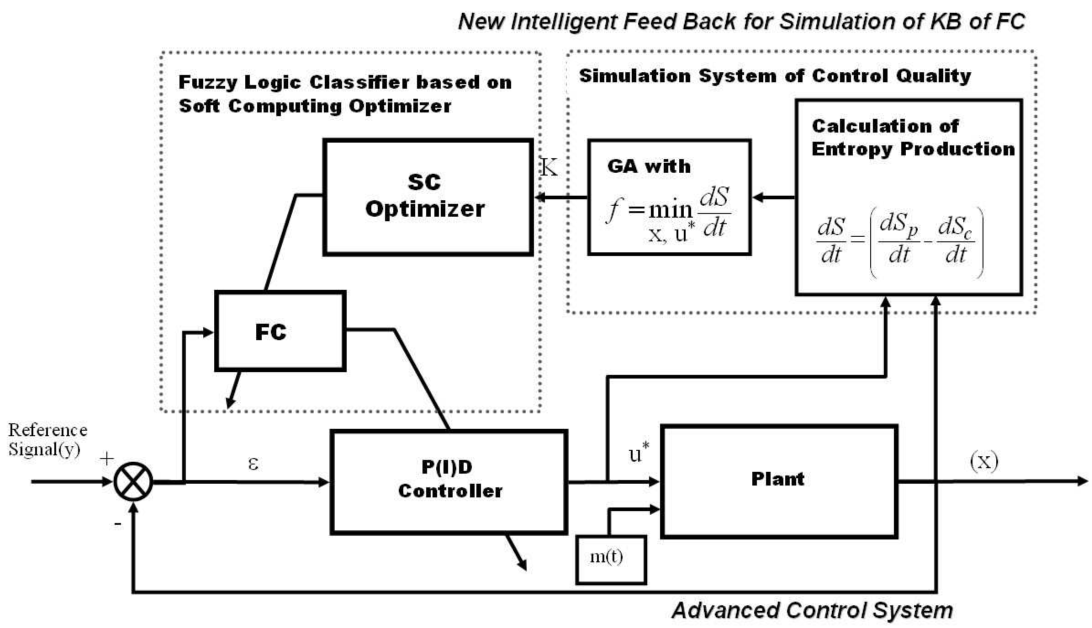


Figure 6-11. Structure of self-organizing robust intelligent control system based on SC Optimizer

Input to SCO can be some measured or simulated data (called as ‘teaching signal”) about the modeled system.
For TS design we use stochastic simulation system based on the model of control object and GA with possibility to choose different fitness function representing control quality criteria. Random trajectories of the chaotic behavior of control object with appropriate probability density function are generated by the *Stochastic Noise Modeler* according to the method of forming filters based of Fokker-Planck-Kolmogorov equations [27]. Design of KB for robust FC is based on the extraction of the value information about random dynamic behavior of control object using different GA fitness functions in stochastic and fuzzy simulation subsystems.

We will demonstrate SC Optimizer tool’s efficiency and robustness for design of new types of *self-organizing intelligent control systems* adapted to control of essentially nonlinear stable and unstable plants under different kinds of stochastic excitations.

The general structure of *self-organizing intelligent control systems* based on SCO is shown in Fig. 6-11, where we use the following designations: GA - Genetic Algorithm; *f* - Fitness Function of GA; S- Entropy Production of System; S_c - Entropy production of Controller; S_p - Entropy production of Controlled Plant; ε - Error; u*- Optimal Control Signal; m(t) – stochastic disturbance; FC - Fuzzy Controller.

The basic peculiarities of step 2 technology are: (1) SCO uses the chain of GAs to solve optimization problems connected with the optimal choice of number of membership functions (MFs) for input variables values description, their shapes and parameters and with optimal choice of fuzzy rules; (2) To design GA fitness functions we use an information-thermodynamic approach based on the analysis of dynamic behavior of control object and FC; (3) SCO works as a universal approximator, which extracts information from simulated (or measured) data about the modeled system.
SCO guarantees the robustness of FC, i.e. successful control performance in wide range of plant’s parameters, reference signals, and external disturbances (see below simulation results).

6.4.1 SC Optimizer structure and its main functions

SCO uses the chain of GAs (GA_1, GA_2, GA_3) and approximates measured or simulated data (TS) about the modeled system with desired accuracy. GA_1 solves optimization problem connected with the optimal choice of number of membership functions and their shapes. GA_2 searches optimal KB with given level of rules activation. Introduction of activation level of rules allows us to sort fuzzy rules in accordance with value information and design robust KB. GA_3 refines KB by using a few criteria (see below).
Figure 6-12 shows the flow chart of SCOptimizer operations on macro level and combines several stages.
Stage 1: *Fuzzy Inference System* (*FIS*) *Selection*. The user makes the selection of fuzzy inference model with the featuring of the following initial parameters: Number of input

and output variables; Type of fuzzy inference model (Mamdani, Sugeno, Tsukamoto, etc.); Preliminary type of MFs.

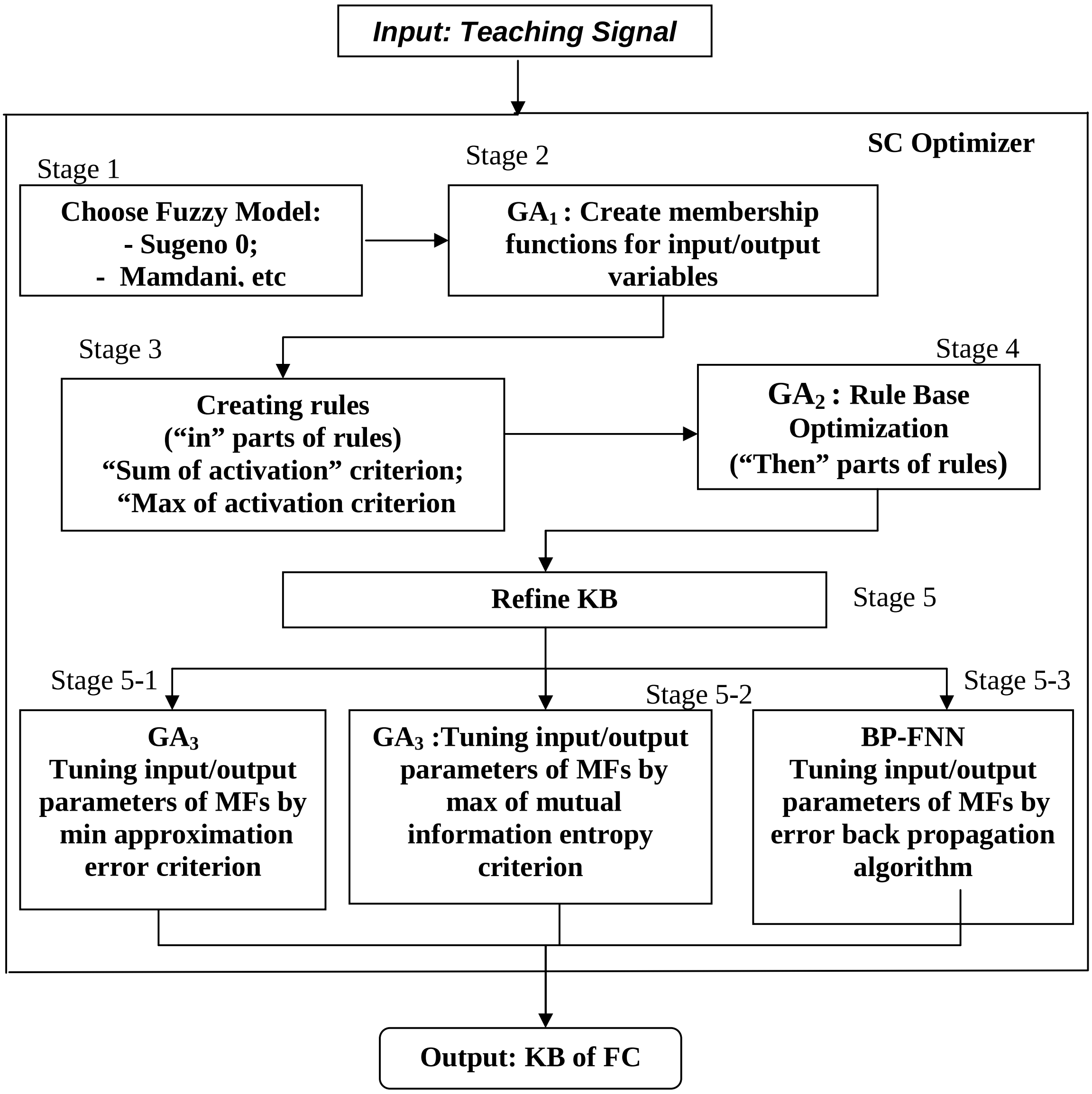


Figure 6-12. Flow chart of SC Optimizer

Stage 2: *Creation of linguistic values*. By using the information obtained on Stage 1, GA_1 optimizes membership functions number and their shapes, approximating TS, obtained from the in-out tables, or from dynamic response of control object (real or simulated in Matlab).

Stage 3: *Creation rules.* At this stage we use the rule rating algorithm for selection of certain number of selected rules prior to the selection of the index of the output membership function corresponding to the rules. For this case two criteria based on a

rule's activation parameter called as a "manual threshold level" (*TL*). This parameter is given by a user (or it can be introduced automatically).
Then the following criteria are used for the KB pre-selection:
(1) "*Sum of firing strength*" criterion: select KB, where rules satisfy to the following condition:

$$R^l_{total\_fs} \geq TL$$

(2) "*Max of firing strength*" criterion: select KB, where rules satisfy to the following condition:

$$\max_t R^l_{fs}(t) \geq TL .$$

At this stage the *total firing strength of each rule* is calculated as

$$R^l_{total\_fs} = \sum_{k=1}^{N} R^l_{fs}(t_k) \text{, where}$$

$$R^l_{fs}(t_k) = \prod \left[\mu^l_{j1}(x_1(t_k)), \mu^l_{j2}(x_1(t_k)), ..., \mu^l_{jn}(x_n(t_k)) \right], \qquad (6\text{-}2)$$

where t_k is a time moment, $k = 1,...,N$, N is a number of temporal points in TS, and l is a rule index.

Remark. The calculation of $R^l_{fs}(t_k)$ is performed according to chosen interpretation of fuzzy AND operation. In Eq.(6-2) this is considered as a "product" operation (see chapter 2.2.5).

Output of this stage is KB designed according to the chosen criterion and the given threshold level *TL*.

Stage 4: *Rule base optimization*. GA_2 optimizes the rule base obtained on the Stage 3, using the fuzzy model obtained on Stage 1, optimal linguistic variables, obtained on Stage 2, and the same teaching signal as it was used on Stage 1.

Stage 5: *Refine KB*. On this stage, the structure of KB is already specified and close to global optimum. In order to reach the optimal structure, a few methods can be used.
First method is based on GA_3 with fitness function as minimum of approximation error, and in this case KB refining is similar to classical derivative based optimization procedures (like error back propagation (BP) algorithm for FNN tuning).
Second method is also based on GA_3 with fitness function as maximum of mutual information entropy (see Table 6-1).
Third method is realized as pure error back propagation (BP) algorithm. BP algorithm may provide further improvement of output after genetic optimization.
As output results of the Stages 3, 4 and 5, we have a set of KB corresponding to chosen KB optimization criteria.
Finally, we must test all developed KB FC by using the model of the control object and choose best KB from control quality point of view.

In Table 6-1 types and the role of SC Optimizer GA's fitness functions (FF) are shown.

Table 6-1: Types and the role of GA fitness function in SCO

Type of GA	Criteria	Fitness Function	The Role of FF
GA$_1$: **Linguistic Variables Optimization**	**MAX** of mutual information entropy *AND* **MIN** of information amount in each signal	$H_{X_i}^{j} = -p^{j}{}_{X_i} \log(p^{j}{}_{X_i}) = -p\left(x_i \middle| x_i = \mu^{j}{}_{X_i}\right) \log\left[p\left(x_i \middle| x_i = \mu^{j}{}_{X_i}\right)\right] = -\frac{1}{N}\sum_{t=1}^{N} \mu^{j}{}_{X_i}(x_i(t)) \log\left[\mu^{j}{}_{X_i}(x_i(t))\right] \rightarrow \max$ $H^{(j,l)}{}_{X_i \mid X_k} = H\left(x_i \middle|_{x_i=\mu^{j}{}_{X_i}, x_k=\mu^{l}_{X_k}}\right) = -\frac{1}{N}\sum_{t=1}^{N}\left[\mu^{j}{}_{X_i}(x_i(t)) * \mu^{l}{}_{X_k}(x_k(t))\right] \log\left[\mu^{j}{}_{X_i}(x_i(t)) * \mu^{l}{}_{X_k}(x_k(t))\right] \rightarrow \min$ where * denotes selected T-norm (Fuzzy AND) operation.	*Data compressing;* *Choice of optimal number of MF approximating TS*
GA$_2$: **Rule Base Optimization**	**MIN** of total error (a difference between the FIS and TS outputs)	$E = \sum_{p} E^{p} \rightarrow \min$, where $E^{p} = \frac{1}{2}(F(x_1^{p}, x_2^{p}, \dots, x_n^{p}) - d^{p})^2$	*Choice of optimal number of rules and MF parameters*
GA$_3$: **Refine KB**	**MIN** of total error (a difference between FIS and TS utputs) **OR** **MAX** of mutual information entropy	$E = \sum_{p} E^{p} \rightarrow \min$ $H_{X_i}^{j} \rightarrow \max$	*Fine Tuning of MF parameters*

Consider now simulation results obtained for a given set of benchmarks in the following three cases of control:

- classical control based on P(I)(D) regulator with constant gains,
- fuzzy control based on traditional SC approach (step 1 technology) and
- fuzzy control based on new SC approach with SC Optimizer (step 2 technology).

7. Benchmarks of Smart Control Simulation for Nonlinear Dynamic Systems

We will consider the following benchmarks of non-linear control objects.

1. *Swing system* (pendulum with variable length):
Equations of motion:

$$\begin{cases} \ddot{\theta} + 2\dfrac{\dot{l}}{l}\dot{\theta} + \dfrac{g}{l}\sin\theta = k'_p \cdot e_\theta + k'_d \cdot \dot{e}_\theta + k'_i \cdot \int e_\theta dt + \xi_1(t) \\ \ddot{l} + 2k\dot{l} - l\dot{\theta}^2 - g\cos\theta = \dfrac{1}{m}\left(k_p \cdot e_l + k_d \cdot \dot{e}_l + k_i \cdot \int e_l dt + \xi_2(t)\right) \end{cases} \tag{7-1}$$

Equations of entropy production rate:

$$\frac{dS_\theta}{dt} = 2\frac{\dot{l}}{l}\dot{\theta}\cdot\dot{\theta}; \quad \frac{dS_l}{dt} = 2k\,\dot{l}\cdot\dot{l}$$

2. *Van der Pol oscillator*:

$$\ddot{x} + (x^2 - 1)\dot{x} + kx = u(t) + \xi(t); \quad \frac{dS}{dt} = (x^2 - 1)\dot{x}\dot{x} \tag{7-2}$$

3. *Oscillator with hysteresis effects*:

$$\begin{cases} \ddot{x} + \bar{\delta}\dot{x} + kx + z = u(t) \\ \dot{z} = A\dot{x} - \left\{\bar{\beta}\left|\dot{x}\right| z + \gamma\dot{x}\left|z\right|\right\} \end{cases}, \frac{dS_x}{dt} = \bar{\delta}\dot{x}\dot{x} \tag{7-3}$$

4. *Coupled nonlinear oscillators*:

$$\begin{cases} \ddot{x} + 2\beta_1\dot{x} + \omega^2{}_1\left[1 - k\cdot y\right]x = 0 \\ \ddot{y} + 2\beta_2\dot{y} + \omega^2{}_2 y + \dfrac{\pi^2}{2l}\left[x\ddot{x} + \dot{x}^2\right] = \frac{1}{M}u(t) \end{cases}; \quad \frac{dS_x}{dt} = 2\beta_1\dot{x}\cdot\dot{x}; \frac{dS_y}{dt} = 2\beta_2\dot{y}\cdot\dot{y} \tag{7-4}$$

5. *Nonlinear oscillator with sizable nonlinear dissipative components*:

$$\ddot{x} + \left[2\beta + a\dot{x}^2 + k_1x^2 - 1\right]\dot{x} + kx = u(t); \quad \frac{dS_x}{dt} = \left[2\beta + a\dot{x}^2 + k_1x^2 - 1\right]\dot{x}\cdot\dot{x} \tag{7-5}$$

6. *Nonlinear oscillator with sizable nonlinear dissipative components*:

$$\ddot{x} + 2\beta\left|\dot{x}\right|^5 sign\left(\dot{x}\right) + kx + \gamma x^3 = u(t); \quad \frac{dS_x}{dt} = 2\beta sign(\dot{x})\left|\dot{x}\right|^5\dot{x} \tag{7-6}$$

7. *Nose-Hoover oscillator*:

$$\begin{cases} \ddot{x} + \xi\dot{x} + x = u(t) \\ \dot{\xi} = \dot{x}^2 - 1 \end{cases}, \frac{dS_x}{dt} = \xi\dot{x}\dot{x} \tag{7-7}$$

Methodology of simulation

Simulations results are received in accordance with the following methodology of simulation.

We investigate dynamic and thermodynamic behavior properties of Control Object (CO) in presence of different kind of stochastic excitations and different kind of control by using our stochastic simulation and SC Optimizer tools.

1) Free Motion Investigation

At this step we identify dynamic and thermodynamic behavior of CO for different parameters and initial conditions. As a result we define type of CO behavior: stable or unstable.

2) Stochastic motion investigation

We consider CO motion under different stochastic excitation with different probability distribution density.

For the simulation of dynamic system motion under different stochastic excitation with different probability distribution density we use our stochastic simulation tools. The tools are based on the mathematical model of CO and methods of nonlinear forming filters using Fokker-Planck-Kolmogorov (FPK) equations (see stochastic backgrounds in Appendix) [27]. The tools have the following particularities:

- Random processes with given probability distribution density based on nonlinear forming filters by using FPK equation are formed. The solution of FPK equation gives us full information about random processes;
- Chosen trajectories, obtained on the basis of forming filters, are informative representative of the given class of random processes. It allows us to realize methodology of Soft Computing, because fuzzy sets theory "works" with individual trajectory, describing individual characteristic of process under investigation.

We consider different types of random noises: Gaussian excitations with symmetric probability distribution density function and non-Gaussian excitations such as Rayleigh noises with non-symmetric probability distribution density function (see Appendix).

3) Investigation of excited motion under classical control based on classical regulators (Advanced control system theory)

At this step we investigate dynamic and thermodynamic behavior of CO under classical control (P(I)(D)) regulators with fixed gain parameters and define gain parameters ranging area and physical limitations of classical regulators. As the result of this stage, we can conclude in what cases classical regulators cannot effectively control a plant in the presence of random noises. At this step we also define ranging area of PID K-gains for next step of GA based optimization.

FC KB design process

The following below stages represent a process of FC KB design.

4) GA-based optimization of control. Design of a teaching signal
At this step we design TS of optimal control based on the given control quality criterion as a GA fitness function.

5) FNN based approximation of TS (Step 1 Technology)
At this step we extract Knowledge Base (KB) of FC by using FNN tuning with error back propagation with AFM tools [30].

6) SC Optimizer based approximation of TS (Step 2 Technology)
At this step we extract Knowledge Base (KB) of FC by using SC Optimizer tools. In this case the FC-FNN structure is optimized by GA based on information theory criteria as fitness functions.

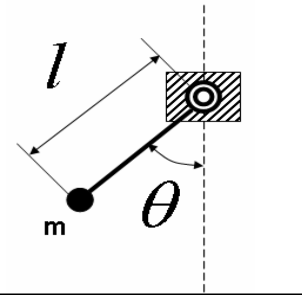
7) Comparison of control quality performance obtained by classical PID control, FC-FNN control and SCO control
At this step we show efficiency and robustness of step 2 technology comparative to traditional SC approach based on FNN-tuning and traditional PID Controller.
For the evaluation of developed FC performance, we use the following control quality criteria:

- Minimum of control error [control criterion]
- Minimum of $(S_p - S_c)(\dot{S}_p - \dot{S}_c)$ [thermodynamic criterion]
- Minimum of control force. [control realization criterion]

As a consequent of second criterion we will consider also minimum of entropy production in control object and minimum of entropy production in control system itself.

Simulations results description and analysis

7.1 Example 1: *Pendulum with variable length* (*Swing system*) *motion control problem*

Equation of motion:	*Physical Model*
$\begin{cases} \ddot{\theta} + 2\frac{\dot{l}}{l}\dot{\theta} + \frac{g}{l}\sin\theta = k'_p \cdot e_l + k'_d \cdot \dot{e}_l + k'_i \cdot \int e_l dt + \xi_1(t) \\ \ddot{l} + 2k\dot{l} - l\dot{\theta}^2 - g\cos\theta = \frac{1}{m}\left(k_p \cdot e_l + k_d \cdot \dot{e}_l + k_i \cdot \int e_l dt + \xi_2(t)\right), \end{cases}$ $\xi_{1,2}(t)$ are the given stochastic excitations with an appropriate probability density function.	

Equations of entropy production rate are the following: $\frac{dS_\theta}{dt} = 2\frac{\dot{l}}{l}\dot{\theta}\cdot\dot{\theta}$; $\frac{dS_l}{dt} = 2k\,\dot{l}\cdot\dot{l}$.
Consider the following model parameters and initial conditions:

$$m = 1,\ k = 1\ ;\ [\theta_0 = 0.25,\ l_0 = 2.5]\ [\dot{\theta}_0 = 0,\ \dot{l}_0 = 0.01].$$

In Fig.7.1-1, 2, 3 dynamic and thermodynamic behavior of the swing system is shown.

Simulation results in Fig.7.1-2 show that the dynamic system, described by Eq.(7-1), represents a *globally unstable* (along generalized coordinate *l*) dynamic system.

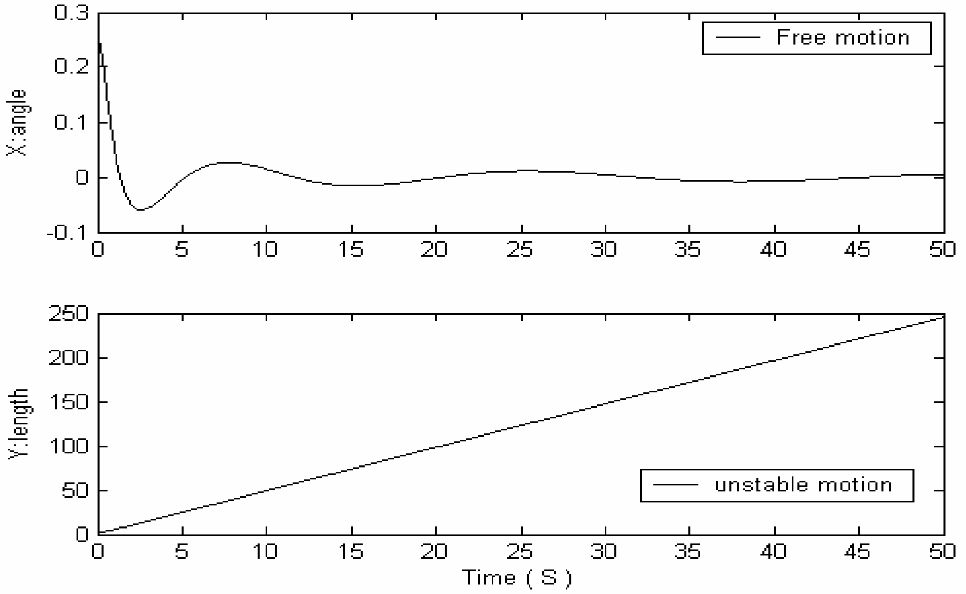


Figure 7.1-1. Free motion of the swing system

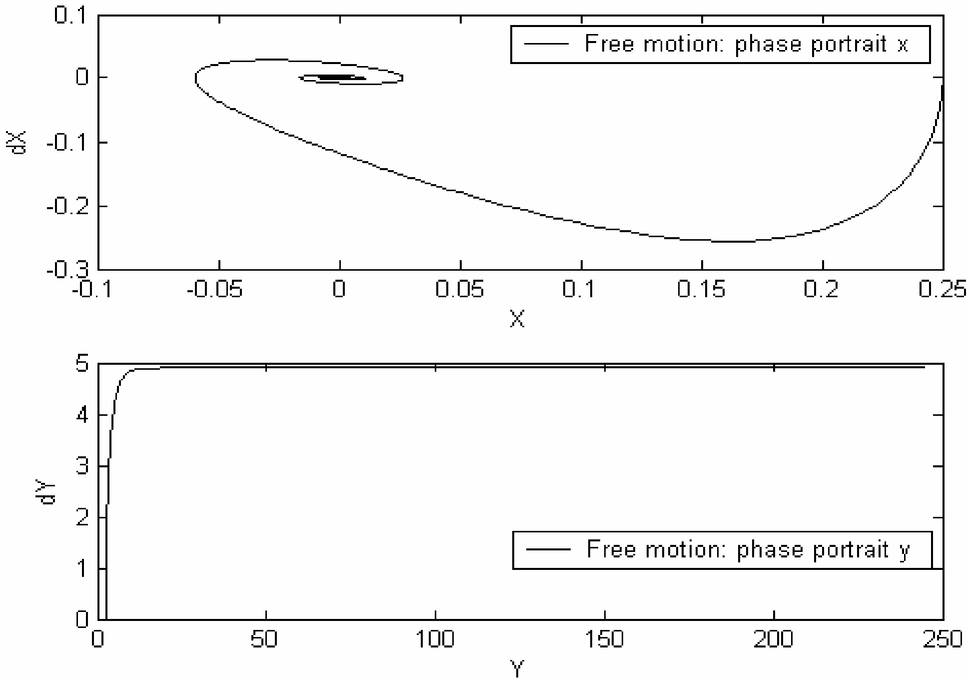


Figure 7.1-2. Free motion of the swing system. Phase portraits

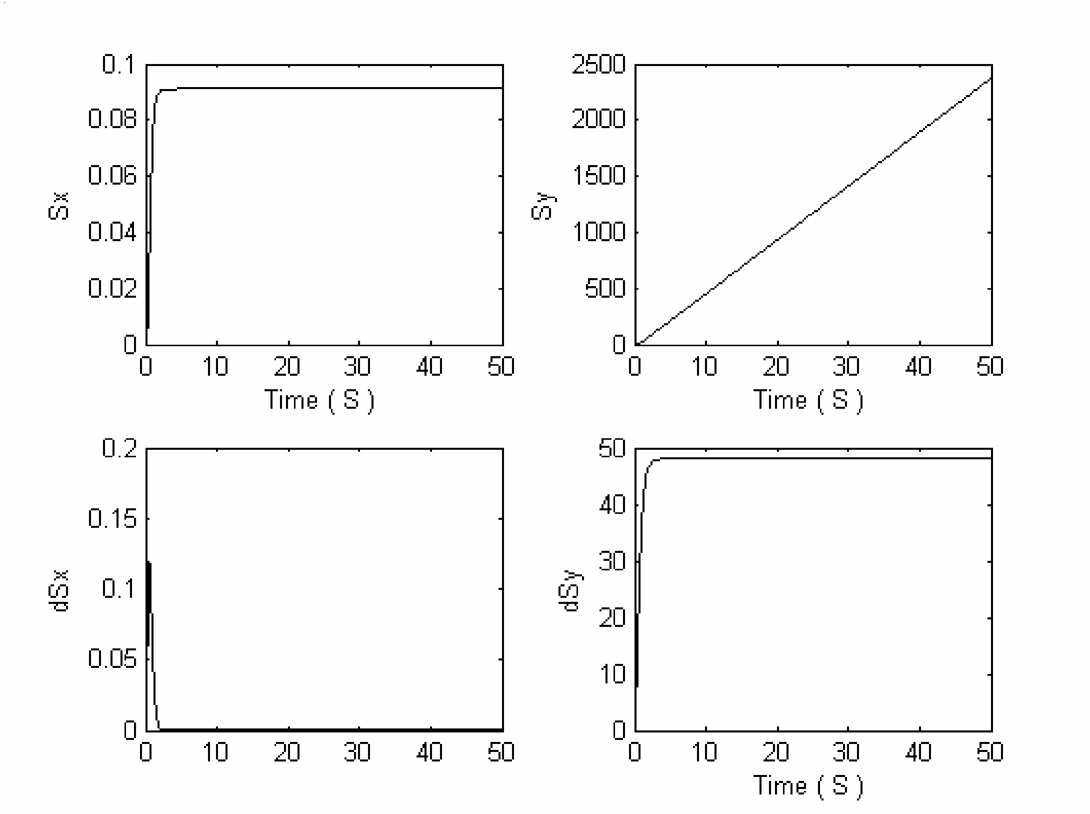


Figure 7.1-3. Free motion of the swing system. Thermodynamic characteristics

Consider behaviour of this control object under two different types of stochastic excitations: Gaussian noise acting along *x* (angle) axis and Rayleigh noise acting along *y* (length) axis (Fig.7.1-4)

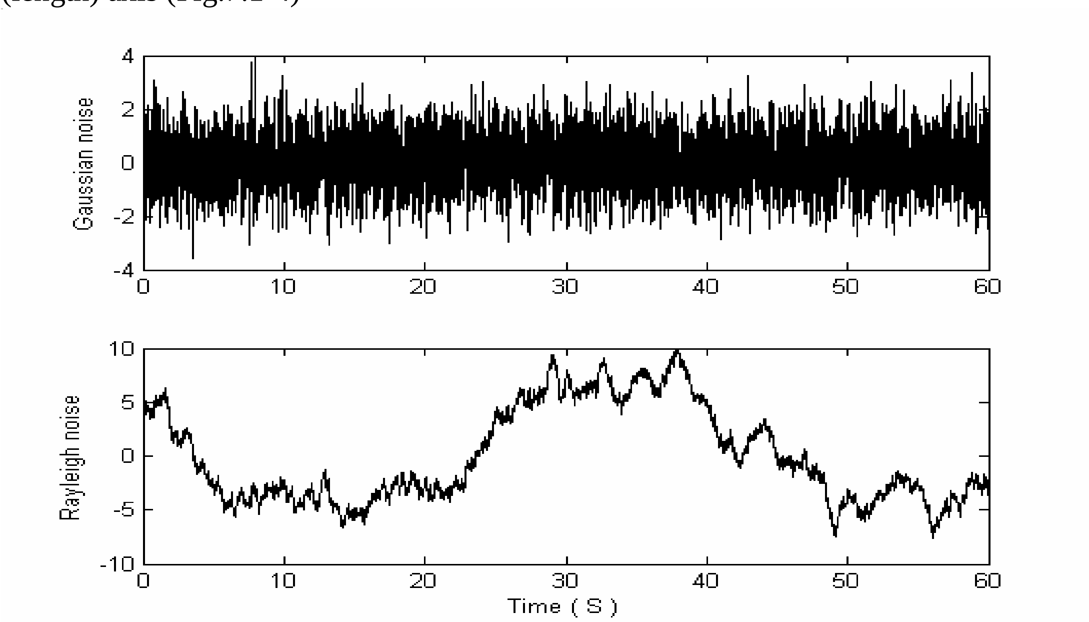


Figure 7.1-4. Gaussian and Rayleigh noises

In Fig. 7.1-4, 5, 6 stochastic motion (dynamic and thermodynamic behavior) of control object is shown. Dynamic behavior of control object under two stochastic excitations is more complicated than free motion.

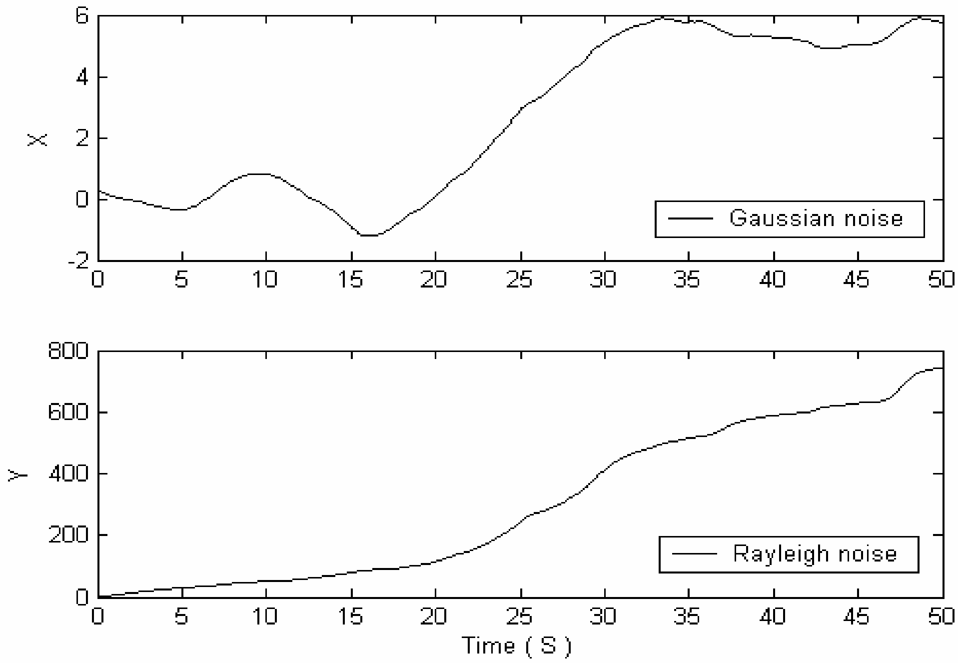


Figure 7.1-4. Swing oscillator. Stochastic motion

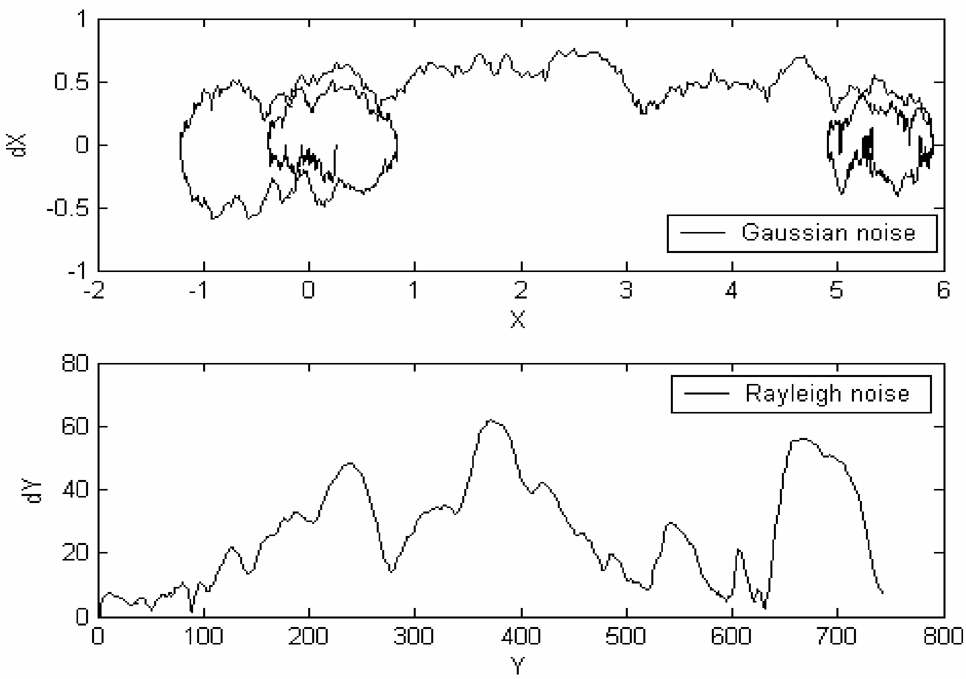


Figure 7.1-5. Swing oscillator. Stochastic motion. Phase portraits

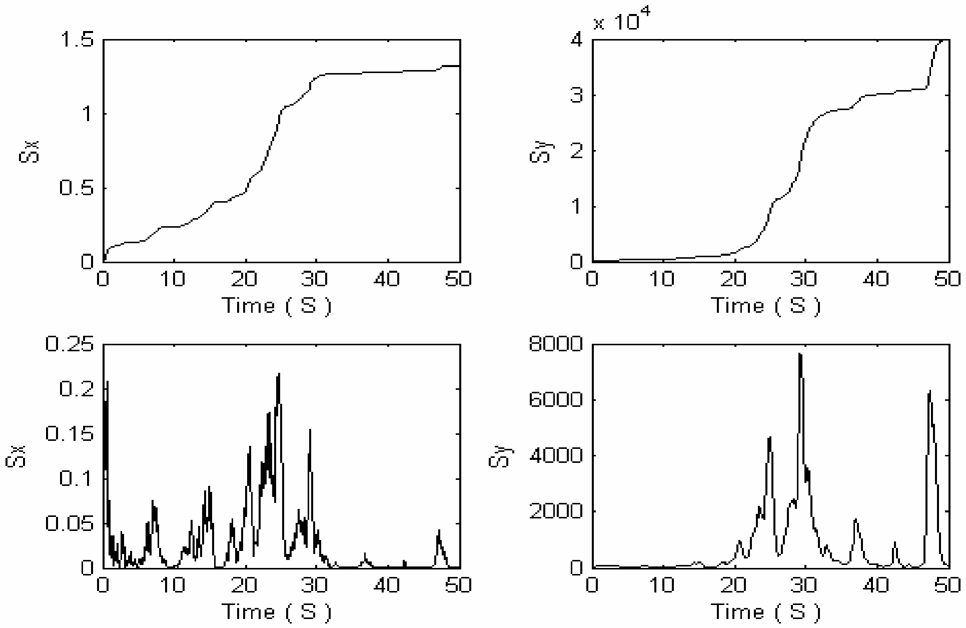


Figure 7.4-6. Swing oscillator. Stochastic motion. Thermodynamic behavior

Consider the following *control task*:
in the presence of Gaussian noise (with maximum amplitude A= 4) along θ-axis and in the presence of Rayleigh noise (with maximum amplitude A= 10) along l-axis, maintain motion of CO at the given reference signals: θ_{ref} = 0.4 ; l_{ref} = 3.5.

For control a stochastic motion of the control object under complicated noises mentioned above we will use two PID controllers with constant K-gains (see Fig.7.1-7).

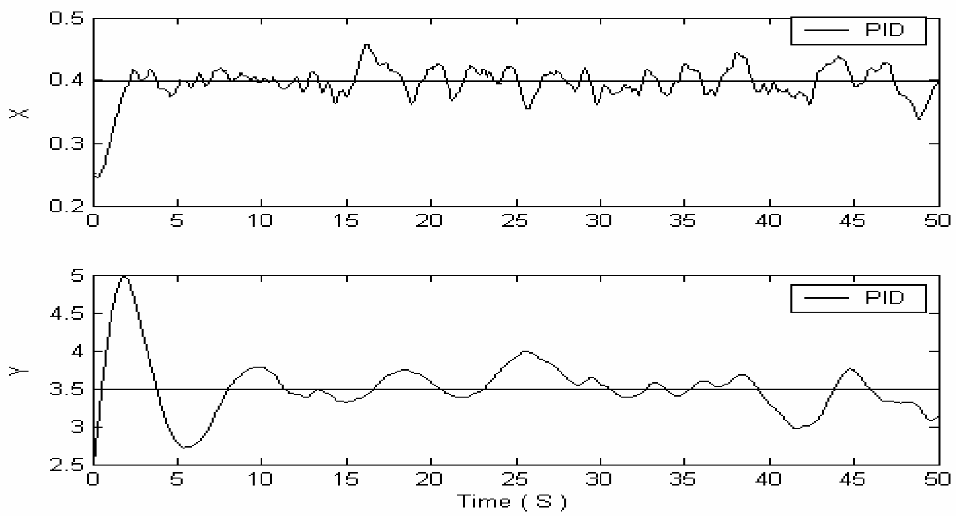


Figure 7.1-7. Swing oscillator. Stochastic motion. Classical PID control with constant k-gains.

The simulation results in Fig. 7.1-7 show limitations of traditional PID control with constant gain coefficients.

Let us design intelligent control system for the given above control problem by using our KB FC design tools and compare results obtained with traditional PID Controllers.

The process of KB FC design is based on a Teaching Signal, TS, obtained for the following control situation (called as TS control situation):

- **Gaussian noise** along θ-axis (max amplitude A = 4);
- **Rayleigh noise** along l-axis (max amplitude A = 10);
- Model parameters: m = 1, k = 1; Initial conditions: [0.25 2.5] [0 0.01].
- Reference signals: θ_{ref} = 0.4, l_{ref} = 3.5.
- GA fitness function is a minimum of control error; GA search space for K gains: [0,10] .

We will design one FC for two PID controllers (along *theta* and *length* axes) with 4 input variables to FC $\{e_x, \dot{e}_x, e_y, \dot{e}_y\}$ and 6 output variables of FC: $\{(k_p, k_d, k_i)_x (k_p, k_d, k_i)_y\}$.

FNN-based KB FC design process (step 1 technology)

FNN based KB design process (with AFM tools) is described as follows:

- Manual choice of numbers of membership functions for each input variables: 3;
- Complete number of fuzzy rules: 3x3x3x3= 81 rules;
- Number of rules in KB**: 81 rules.**

Remark: For the given case, if we choose more than 3 membership functions for each input variables, AFM error back propagation algorithm is failed.
In Fig. 7.1-8 membership functions representation in AFM is shown.

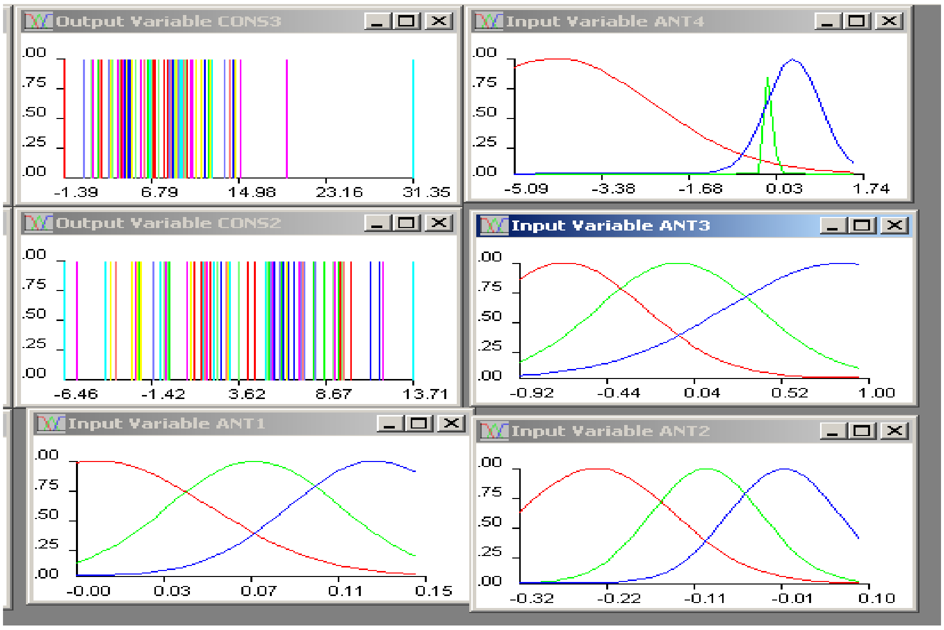


Figure 7.1-8. Swing system. Membership functions representation by AFM.

SC Optimizer-based KB FC design process (step 2 technology)

SC Optimizer-based KB design process is described as follows:

- *Creation of linguistic values* by GA1 : number of membership functions for each input variables is: 9,8,9,9;
- Complete number of fuzzy rules: 9x8x9x9= 5832 rules;
- *Rules selection by*: Max of firing strength criterion (manual threshold level = 0.45);
- *KB optimization by* GA2. Optimized KB contains **143 rules.**

In Fig.7.1-9 example of membership function representation for second FC input is shown.

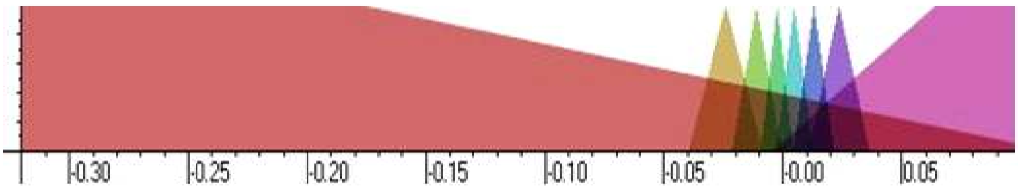


Figure 7.1-9. Example of membership function representation in SC Optimizer

Remark. In AFM based representation number and MF shapes are selected manually by a user. In SC optimizer for each input variables optimal MF shapes and MF number are defined by GA.

Compare control quality of FC_SCO obtained by SC Optimizer, FC_FNN obtained by traditional SC approach based on FNN-tuning (with AFM tools) and traditional PID Controller with constant gains K = (6 6 6 6 6 6).

In Figures 7.1-10, 7.1-11, 7.1-13,7.1-14, and 7.1-16 results of comparison are shown.

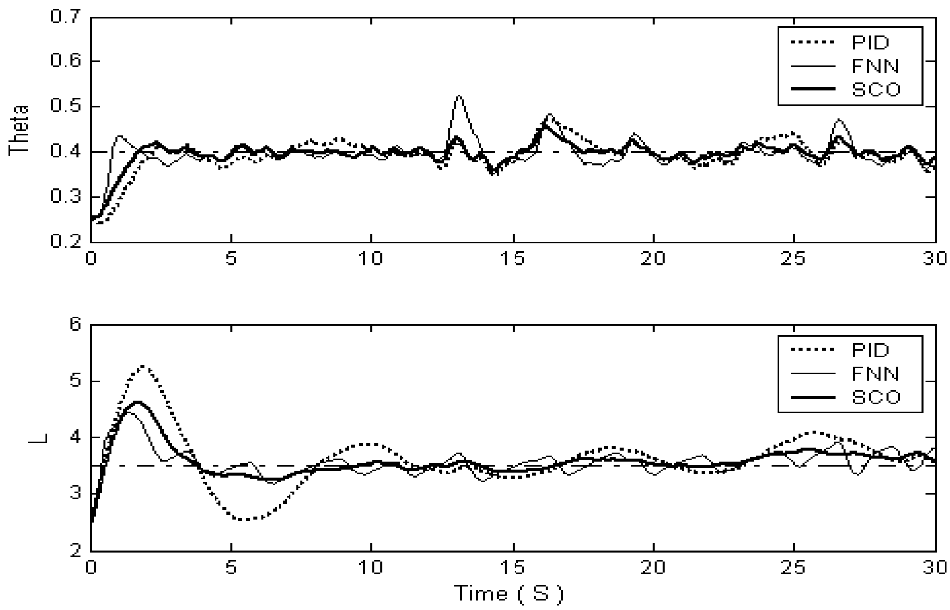


Figure 7.1-10. Swing system motion under 3 types of control. TS control situation

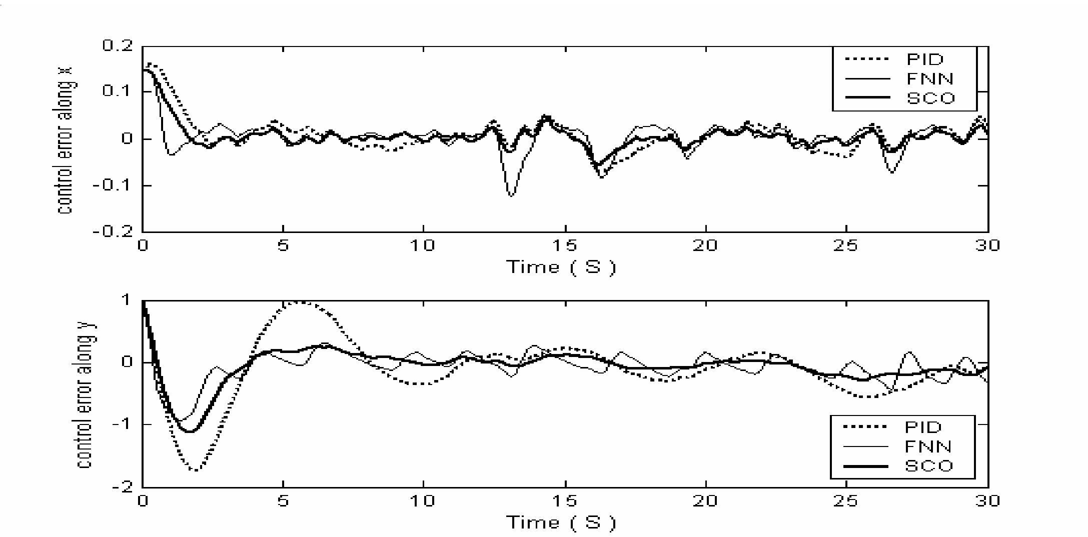


Figure 7.1-11. Swing system under 3 types of control. Control error. TS control situation.

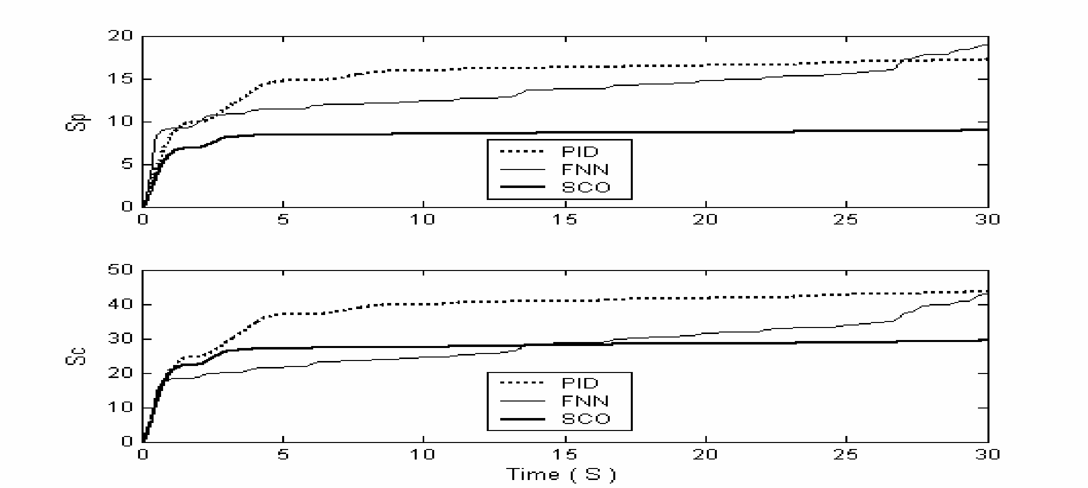


Figure 7.1-12. Swing system under 3 types of control. Entropy production in plant and in controllers. TS control situation.

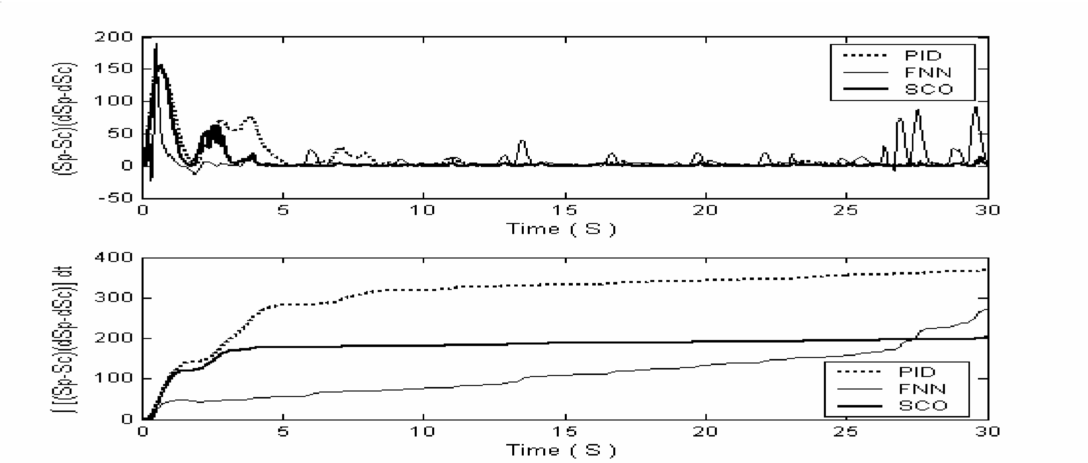


Figure 7.1-13. Swing system under 3 types of control. Generalized entropy production. TS control situation

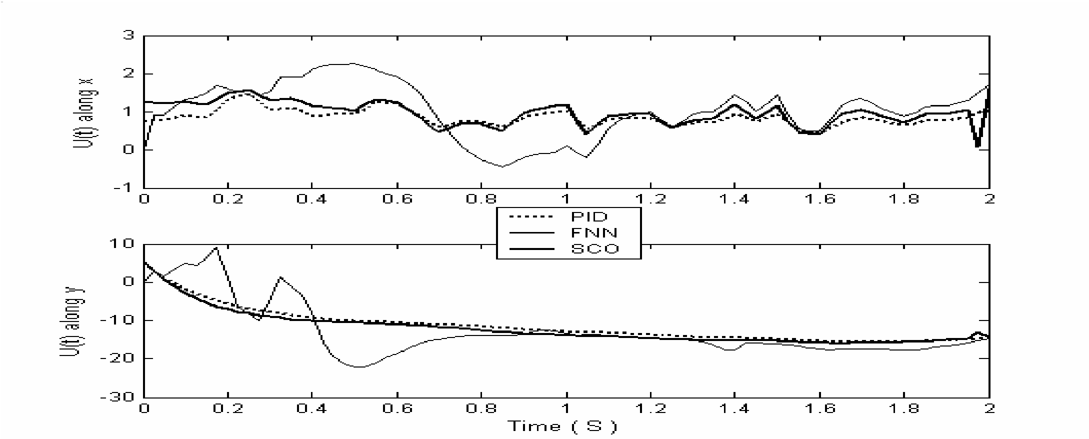


Figure 7.1-14. Swing system under 3 types of control. Control force. TS control situation

Control laws comparison

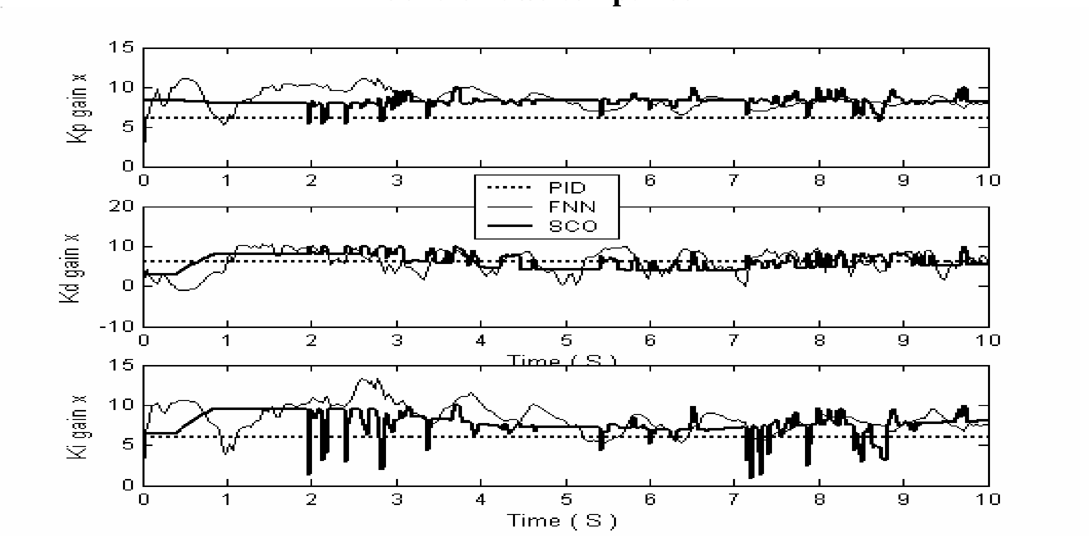


Figure 7.1-15. Swing system. Control laws for PID along theta axis. TS control

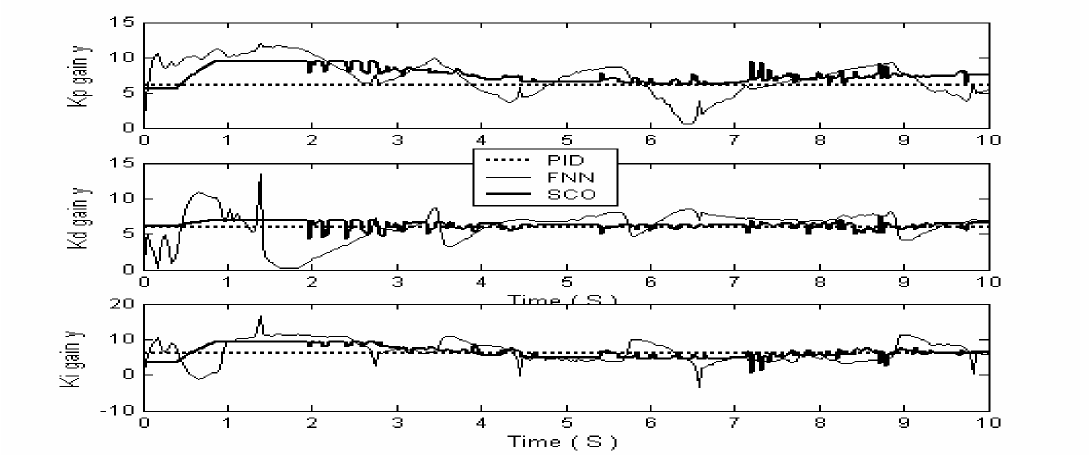


Figure 7.1-16. Swing system. Control laws for PID along *l*-axis. TS control

Conclusion

- From control quality point of view (minimum of control error, minimum of generalized entropy production, minimum of entropy productions in a plant and in controllers, and minimum of control force) Fuzzy PID-controller designed by SC Optimizer realizes more effective control in comparison to FC-FNN and traditional PID-controller.

Robustness investigation of KB FC designed for the given TS control situation

Let us take FC_SCO, FC_FNN developed for the TS control situation and use them in new control situations.

New control situation 1:

- *new initial conditions only* [-0.52 (-30^o), 2.5] [0.01, 0];
- Rest is the same as in TS control situation.

New control situation 2:

- *new initial conditions* [-0.52 (-30^o), 2.5] [0.01, 0];
- *new reference signals*: *theta* = 0.78 (45^o); *length* = 5;
- Rest is the same as in TS control situation

New control situation 3:

- *new initial conditions* [-0.52 (-30^o), 2.5] [0.01, 0];
- *new reference signals*: *theta* = 0.78 (45^o); *length* = 5;
- *new noises amplitudes (smaller)*:
  noise along theta-axis: *Gaussian* noise with max amplitude *A* =1.5; and noise along length-axis: *Rayleigh* noise with max amplitude *A* = 1.

Compare control quality of FC_SCO obtained by SC Optimizer, FC_FNN obtained by traditional SC approach based on FNN-tuning (with AFM tools) and traditional PID controller with constant gains K = (6 6 6 6 6 6). In Figures 7.1-17, 7.1-18, and 7.1-19 results of comparison for three new control situations are shown.

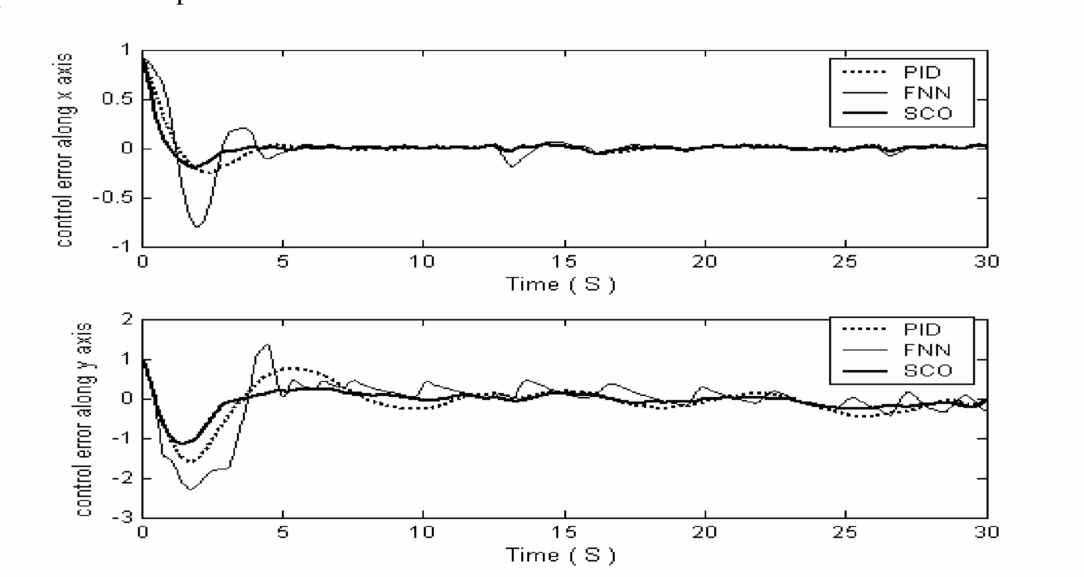


Figure 7.1-17. Swing system. Control error. New control situation 1.

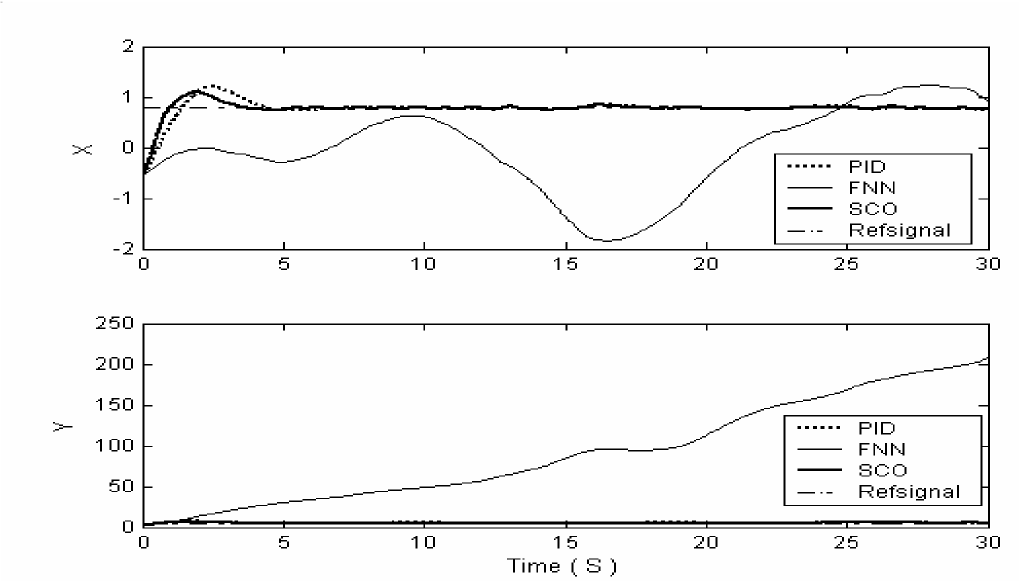


Figure 7.1-18. Swing system. Control error. New control situation 2.

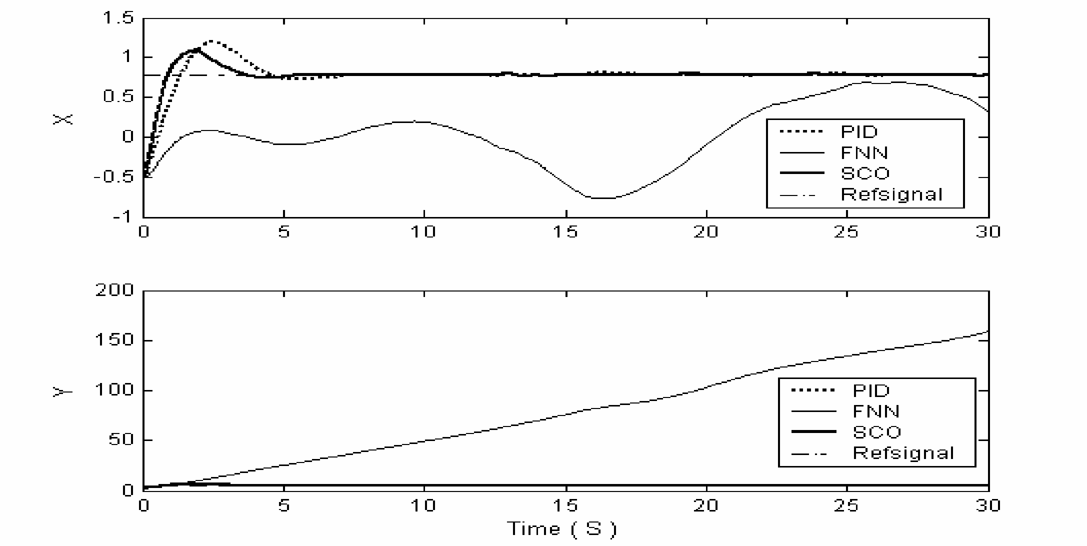


Figure 7.1-19. Swing system. Control error. New control situation 3.

Conclusions

- SC Optimizer control is robust, but FNN control is failed (unstable), i.e. it is not robust in all 3 new control situations (when initial conditions and reference signal are changing and disturbance amplitudes are much smaller).
- From control quality point of view (minimum of control error, etc.) FC_SCO is more effective than PID in given three new control situations.

7.2 Example 2: *Van der Pol oscillator*

Equations of motion and entropy production rate are:

$$\ddot{x} + (x^2 - 1)\dot{x} + kx = u(t) + \xi(t); \quad \frac{dS}{dt} = (x^2 - 1)\dot{x}\dot{x}.$$

Here $\xi(t)$ is a given stochastic excitations with an appropriate probability density function, and $u(t)$ is a control force.

Consider the following model parameters and initial conditions. Model parameters: $k = 1$; Initial conditions: $[x_0][\dot{x}_0]$ = [0.5] [0.1].

In Fig. 7.2-1 and 7.2-2 are shown free motion (dynamic and thermodynamic behavior) of control object with the given above parameters.

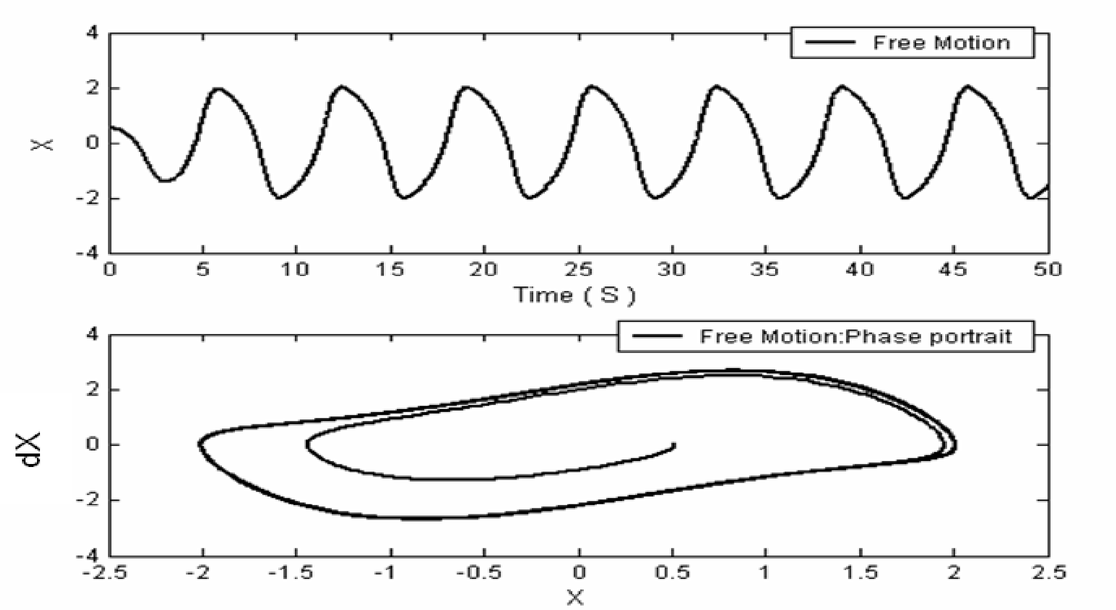


Figure 7.2-1. Van der Pol oscillator. Free motion

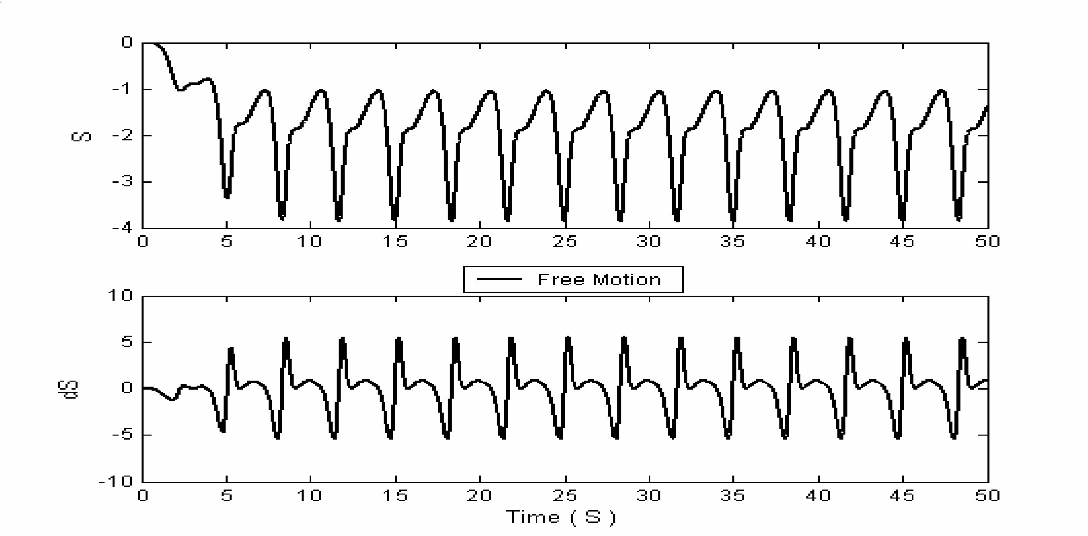


Figure 7.2-2. Van der Pol oscillator. Free motion. Thermodynamic behavior

Simulation results show that: $\exists t, \frac{dS}{dt} < 0, S < 0$. It means that the system described by Eq.(7-2) is a *locally unstable dynamic system* in Lyapunov sense.

Consider behaviour of this control object under two different types of stochastic excitations (Gaussian and Rayleigh noises): case 1 shown in Fig. 7.2-3 and case 2 shown in Fig.7.2-7 (see stochastic backgrounds in Appendix).

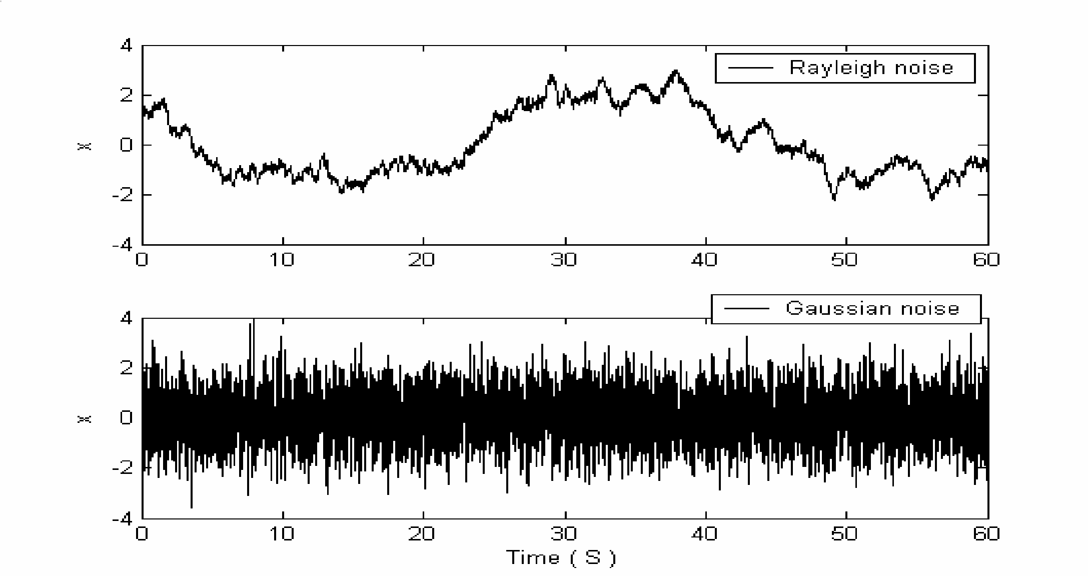


Figure 7.2-3. Stochastic noises. Case 1

In Figures 7.2-4, 7.2-5 and 7.2-6 dynamic and thermodynamic behaviour of CO motion for the case 1 of noises is shown.

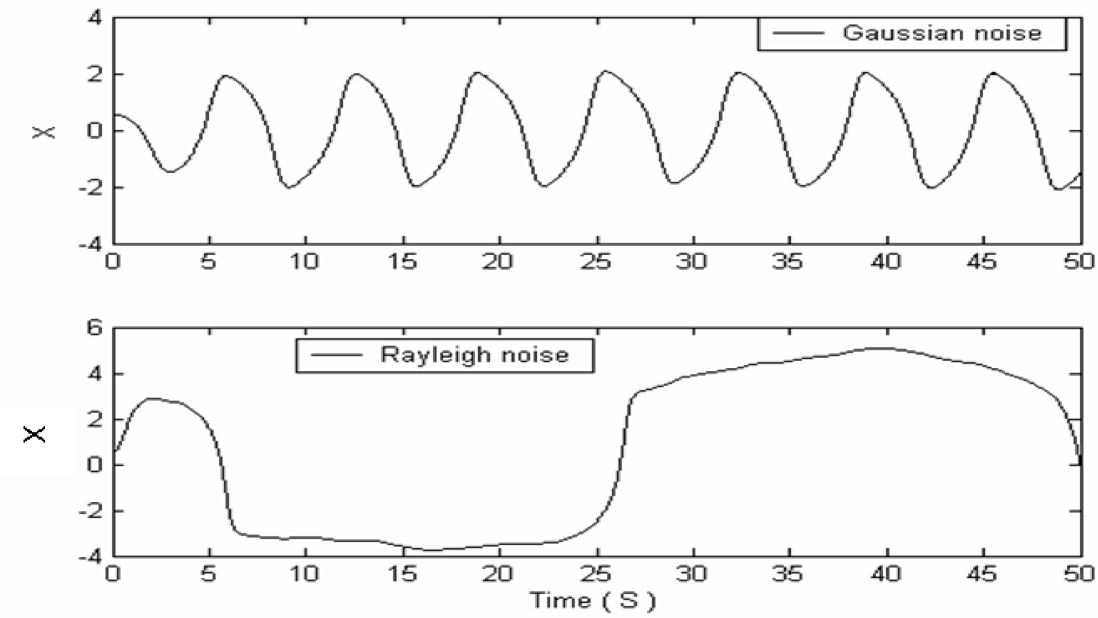


Figure 7.2-4. Van der Pol oscillator. Stochastic motion. Case1 noises.

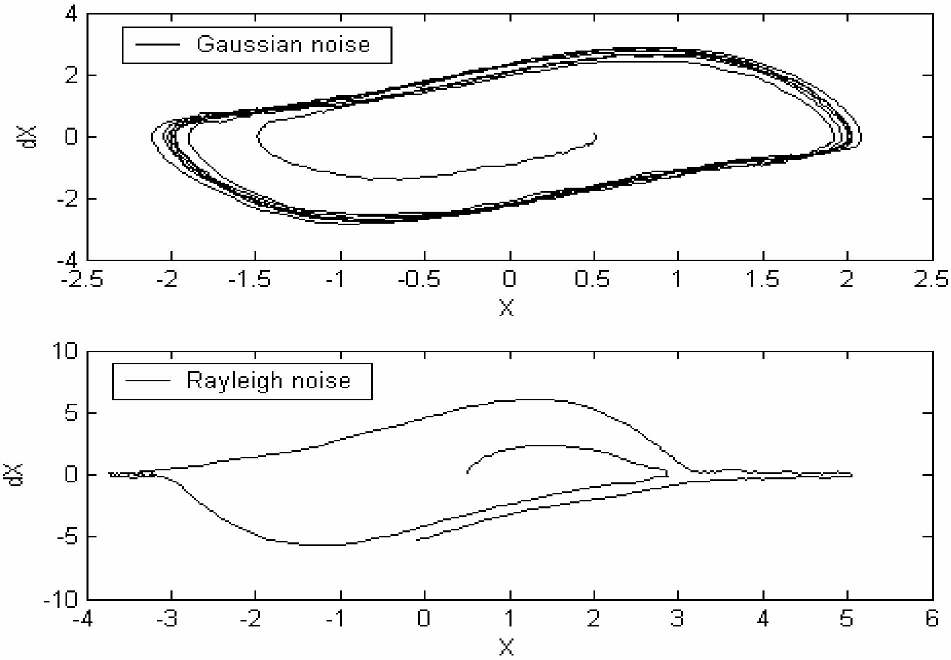


Figure 7.2-5: Van der Pol oscillator. Stochastic motion. Case1 noises. Phase portraits

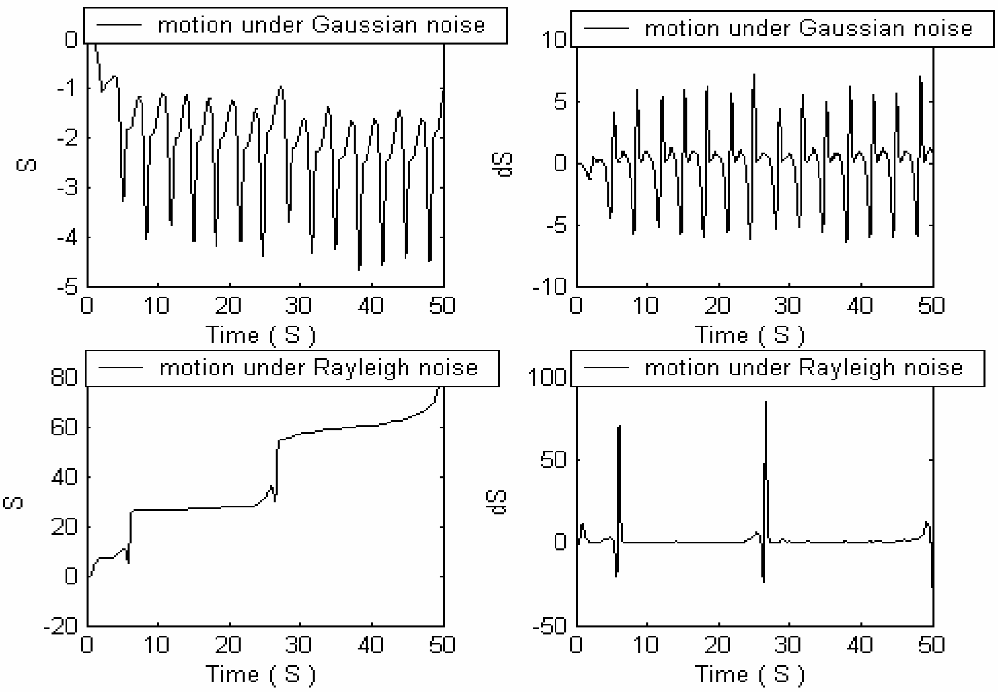


Figure 7.2-6. Van der Pol oscillator. Stochastic motion. Case1 noises. Thermodynamic behavior

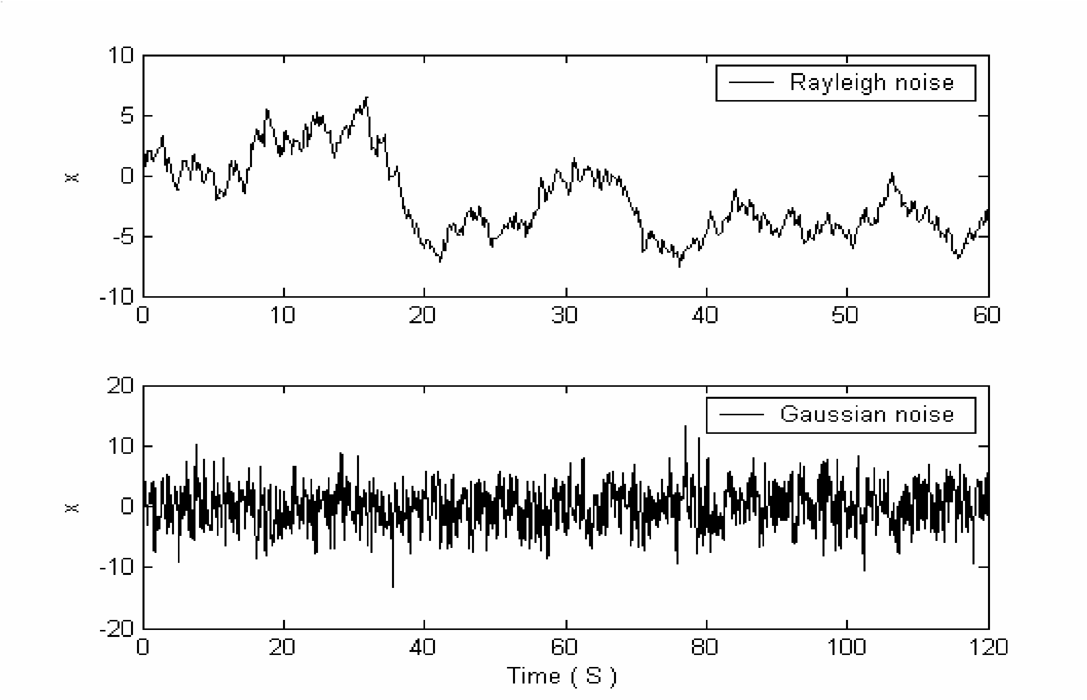


Figure 7.2-7. Stochastic noises. Case 2.

In Figures 7.2-8, 7.2-9 and 7.2-10 dynamic and thermodynamic behaviour of CO motion for the case 2 of noises is shown.

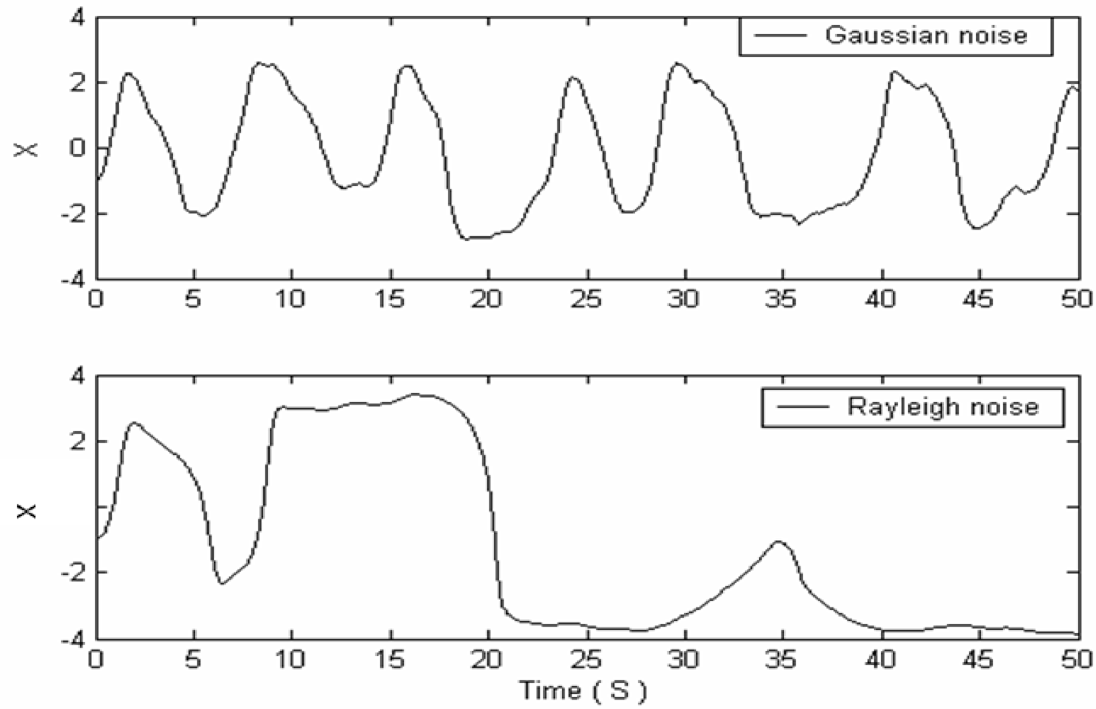


Figure 7.2-8. Van der Pol oscillator. Stochastic motion. Case 2 noises.

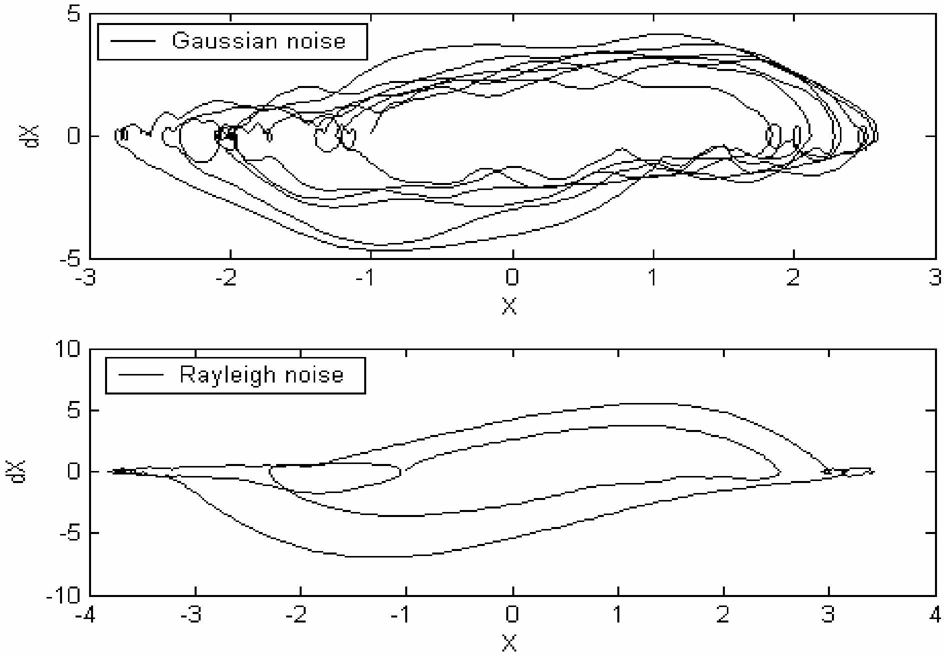


Figure 7.2-9. Van der Pol oscillator. Stochastic motion. Case 2 noises. Phase portraits

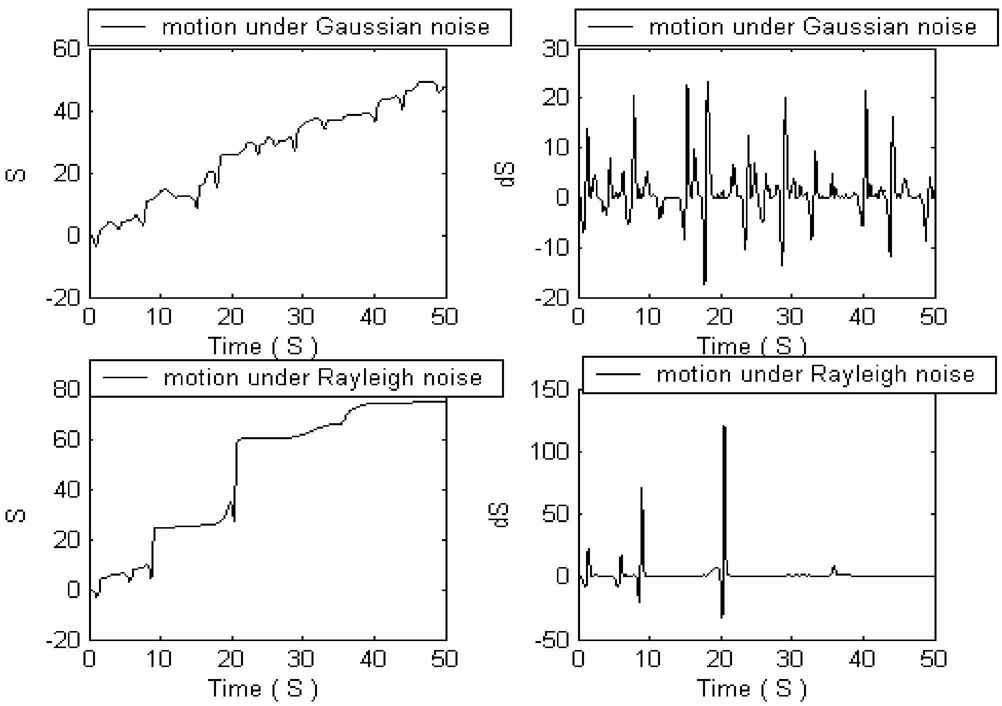


Figure 7.2-10. Van der Pol oscillator. Stochastic motion. Case 2 noises. Thermodynamic behavior

Simulation results show that CO dynamic behavior under Rayleigh excitation is more complicated.

Consider the following three *control tasks* for this example.

Control task 1: in the presence of Rayleigh noise maintain motion of CO at the given reference signal $x_{ref} = 0$;

Control task 2: in the presence of Rayleigh noise maintain motion of CO at the given reference signal shown in Fig.7.2-21 (step reference signal)

Control task 3: stochastic noise compensation:

in the presence of Rayleigh noise maintain motion of CO at a reference signal as the given free motion of CO (Fig.7.2-29).

Let us design intelligent control system for the given above control problems by using our KB FC design tools and compare results with traditional PID Controller.

Control task 1

We have the following TS control situation:

- Model Parameters: k = 1; Initial conditions: [0.5] [0.1];
- Reference signals: $x_{ref} = 0$
- Rayleigh noise (case 1, max amplitude = 3).

GA-based TS design

At this step we design TS of optimal control based on the given control quality criterion as a GA fitness function. For this case we will use the following GA parameters:

- GA search space for K-gains coefficients: (0,10);
- GA FF : minimum of " control error".

In Fig.7.2-11 a comparison of CO motion under GA and PID control is shown.

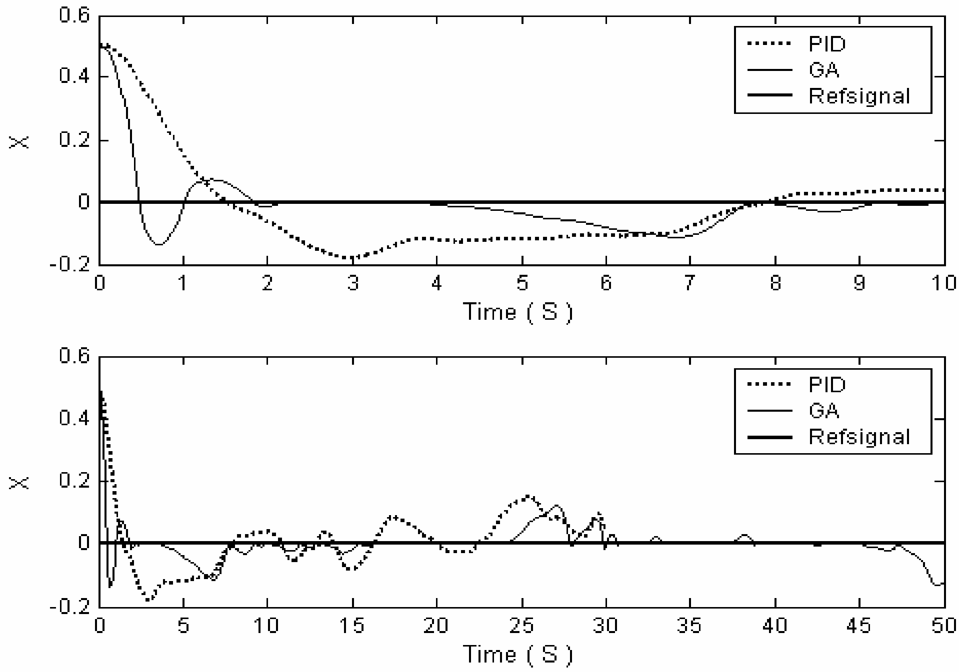


Figure 7.2-11. Van der Pol oscillator. GA- PID control. *Control task* 1.

FNN-based KB FC design process (step 1 technology)

For the given control tasks we will design FC-PID controller with 3 input variables to FC as $\left\{e, \dot{e}, \int edt\right\}$ and 3 output variables of FC as $\left\{k_p, k_d, k_i\right\}$.

AFM based KB design process is described as follows:

- Manual design of numbers of membership functions for each input variables: 5;
- Complete number of fuzzy rules: 5x5x5= 125 rules;
- Number of activated rules in KB**: 125 rules.**

SC Optimizer-based KB FC design process (step 2 technology)

KB FC design process based on SC Optimizer is described as follows.

- *Creation of linguistic values* by GA1 : number of membership functions for each input variables: 5,9,9 ;
- *Rule selection* : SUM of firing strength criterion with automatic manual threshold;
- *KB optimization by* GA2: optimized KB contains **27 rules.**

Compare control quality of FC_SCO obtained by SC Optimizer with 27 rules, FC_FNN obtained by traditional SC approach based on FNN-tuning (with AFM tools) with 125 rules and traditional PID controller with constant gains K = (7 7 7).

In Figures 7.2-12, 7.2-13, 7.2-14 and 7.2-15 results of comparison of CO stochastic motion under 3 types of control are shown.

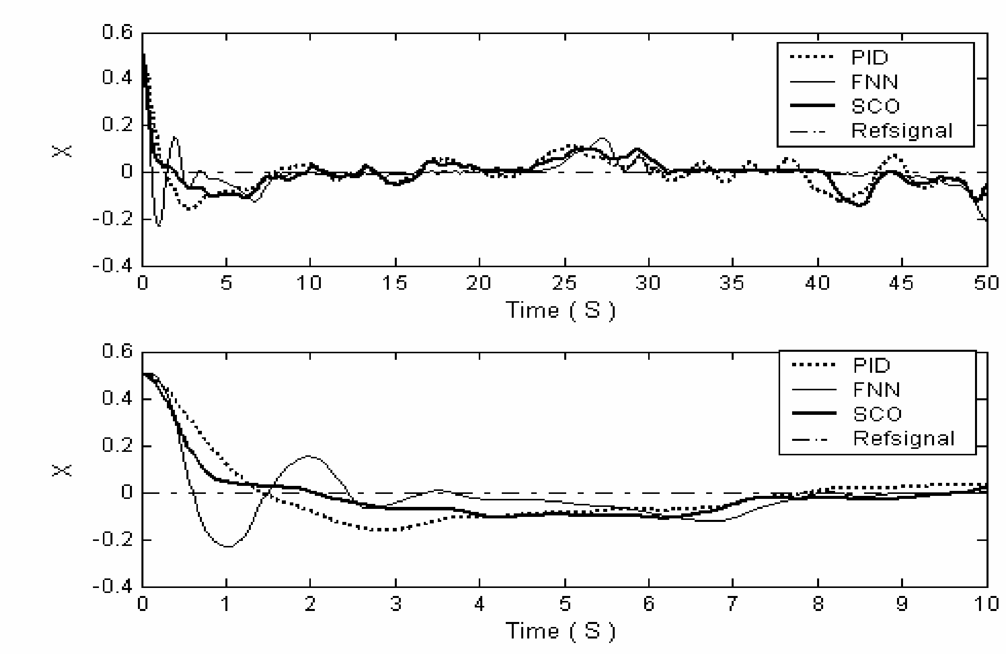


Figure 7.2-12. Van der Pol oscillator. TS control situation . *Control task* 1

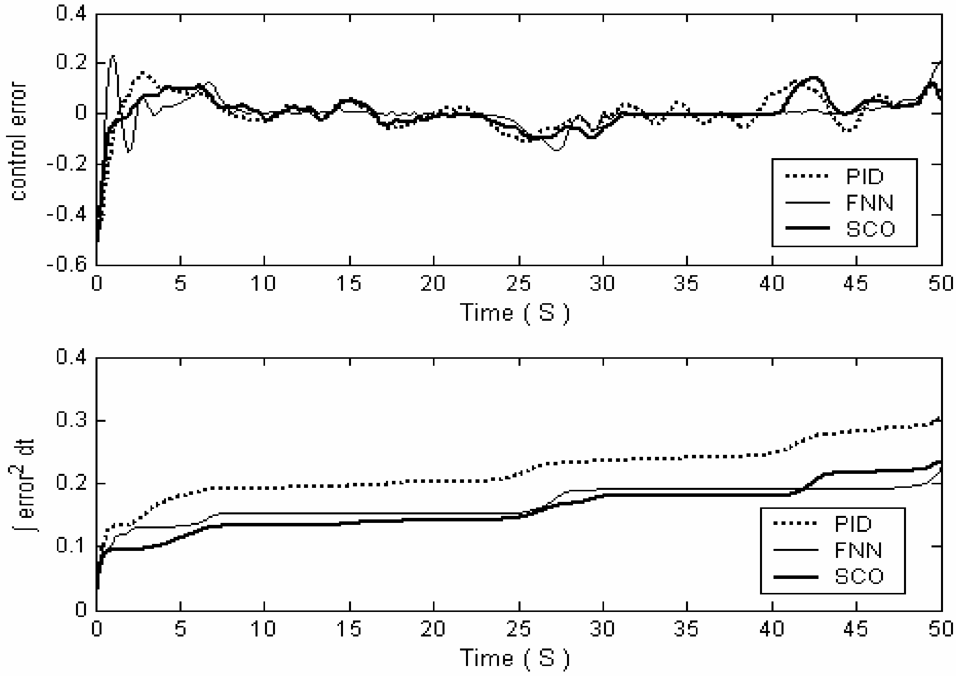


Figure 7.2-13. Van der Pol oscillator. Control error .TS control situation. *Control task* 1

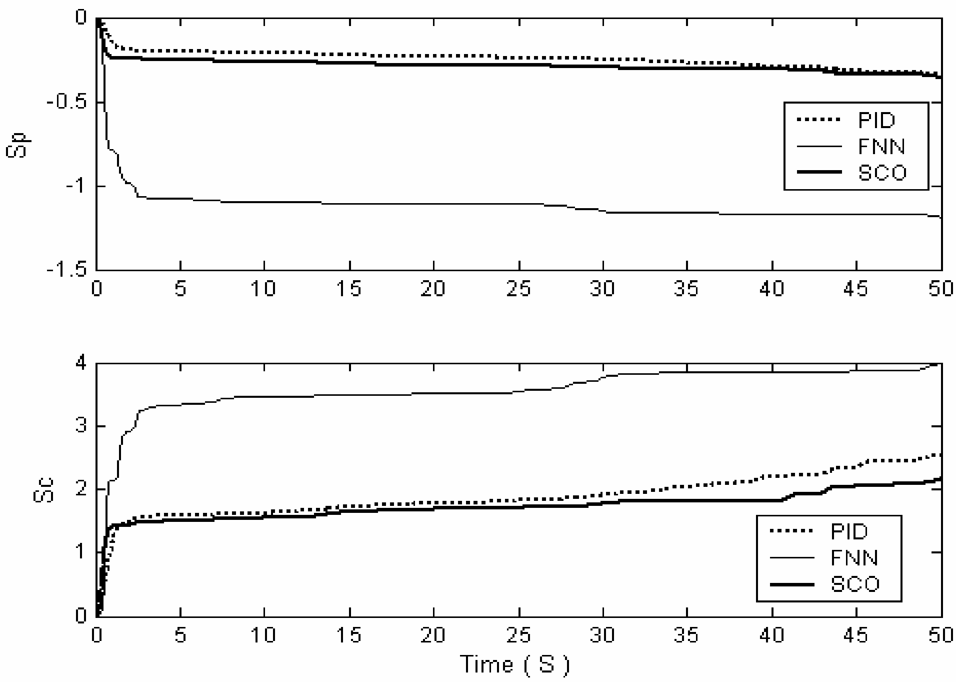


Figure 7.2-14. Van der Pol oscillator. Entropy productions of plant and controller .TS control situation . *Control task* 1

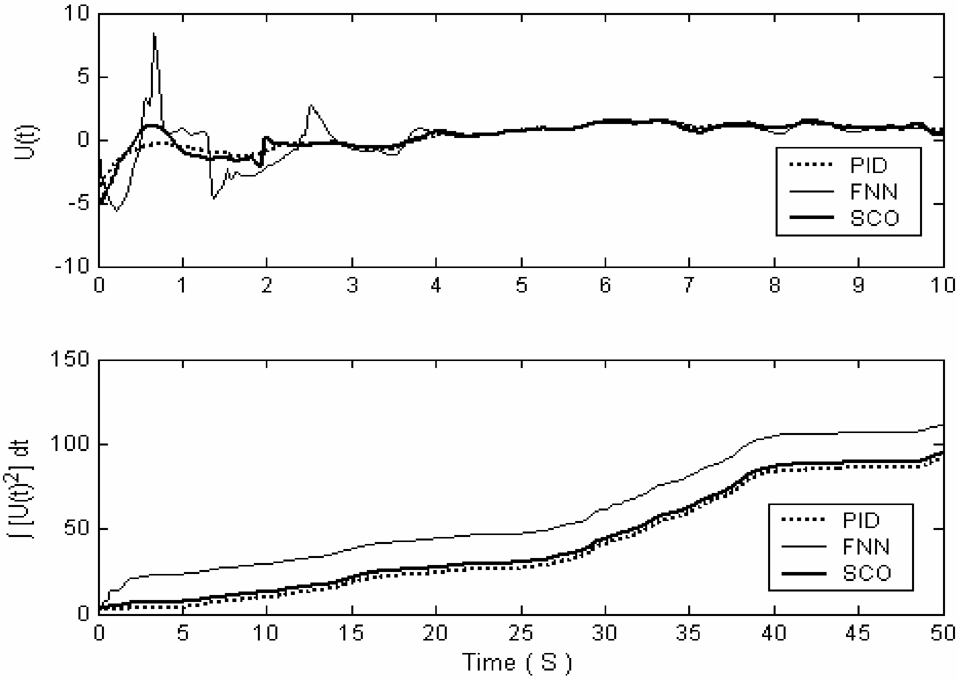


Figure 7.2-15. Van der Pol oscillator. Control force .TS control situation. *Control task* 1

Control laws for TS control situation are shown in Fig.7.2-16.

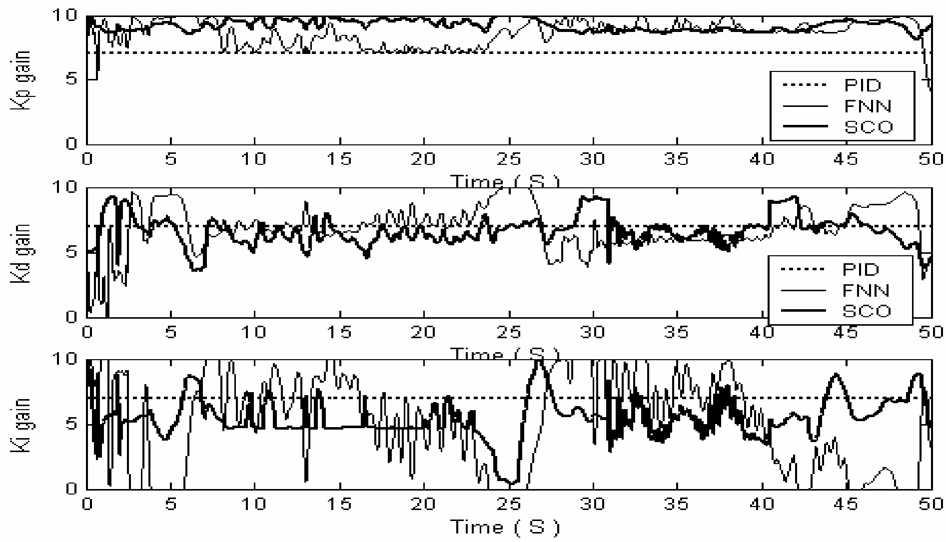


Figure 7.2-16.Van der Pol oscillator. Control laws. TS control situation. *Control task* 1

Conclusions

- FC_SCO has smaller transit time and control error than classical PID and FC_FNN.
- From control quality point of view (minimum of control error, minimum of entropy productions in a plant and in controller, and minimum of control force)

Fuzzy PID-controller designed by SC Optimizer with *27 rules* is more effective than FC-FNN *with 125* rules.

- At time 10-50 sec, FC_SCO, FC_FNN and PID performance in the given TS control situation are compatible.

Robustness investigation of KB FC designed for the given TS control situation

Let us take FC_SCO, FC_FNN developed for the TS control situation and use them in new control situation, where new type of noise, Gaussian (case 1) is considered.
In Fig.7.2-17, 7.2-18, 7.2-19 and 7.2-20 results of comparison of CO stochastic motion under three types of control in the new control situation are shown.

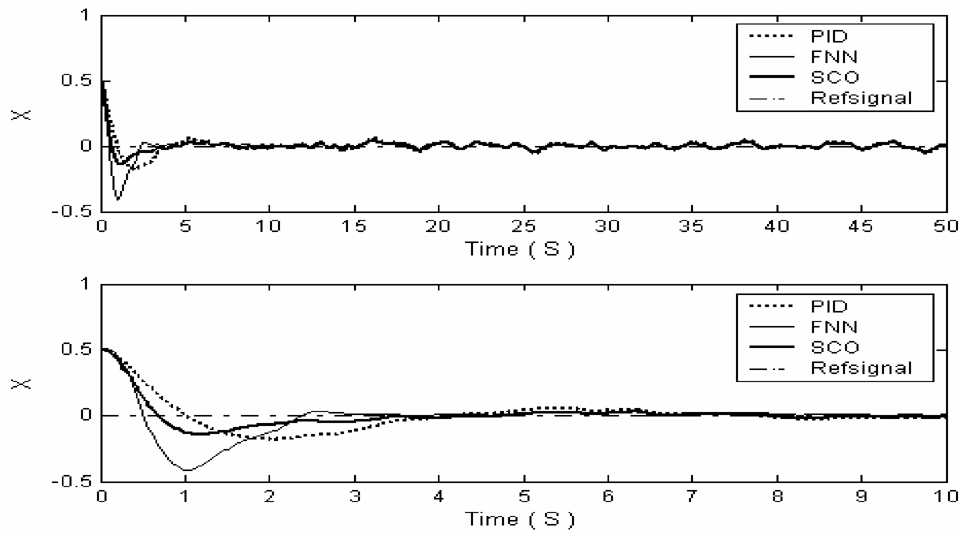


Figure 7.2-17. Van der Pol oscillator. New control situation . *Control task* 1

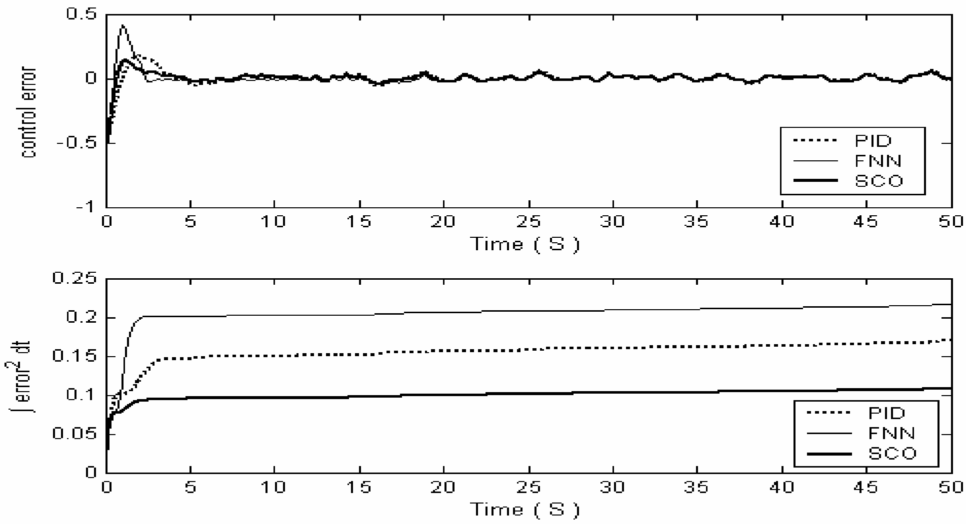


Figure 7.2-18. Van der Pol oscillator. Control error. New control situation. *Control task* 1

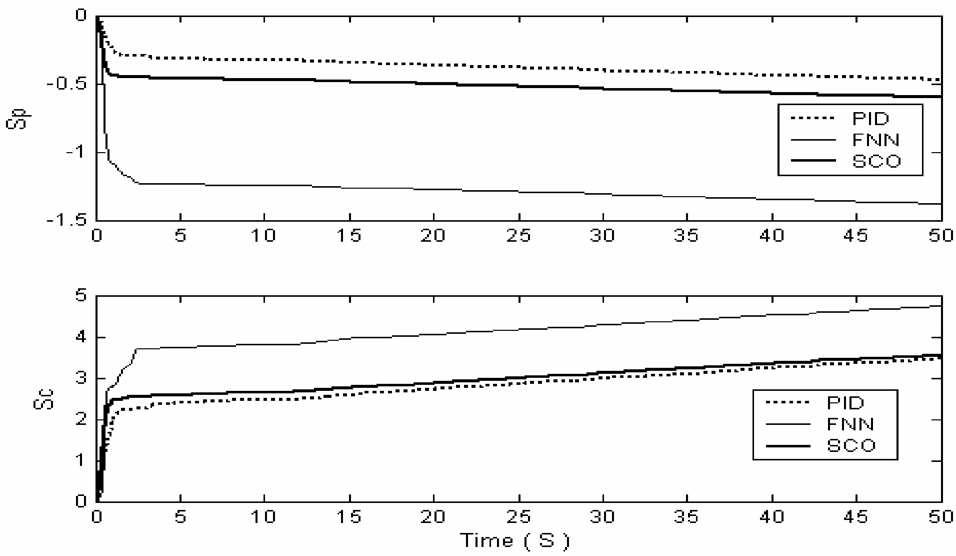


Figure 7.2-19. Van der Pol oscillator. Entropy productions of plant and Controller. New control situation. *Control task* 1

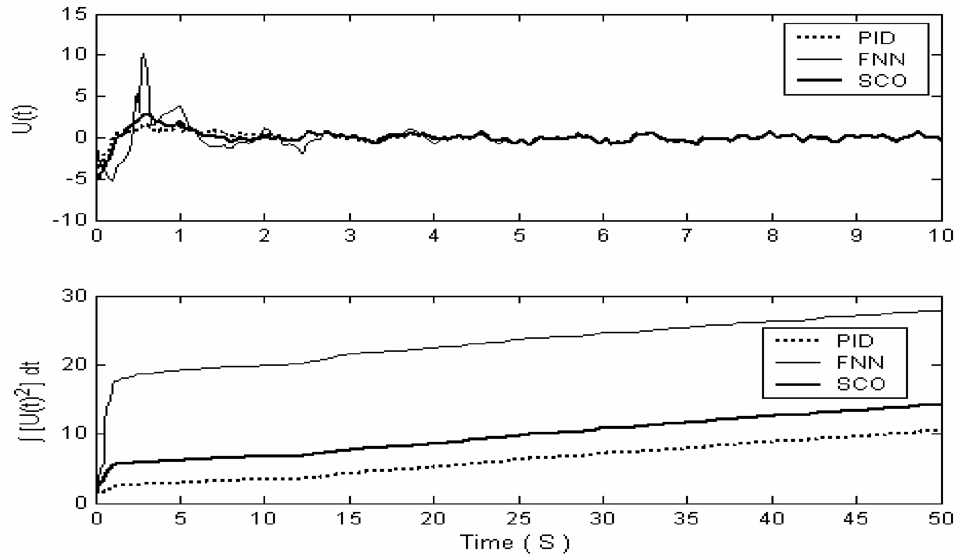


Figure 7.2-20. Van der Pol oscillator. Control force. New control situation. *Control task* 1

Conclusions

- FC_SCO and FC_FNN are robust in new control situation.
- FC_SCO has smaller transit time and control error than classical PID and FC_FNN.
- At time 5-50 sec, FC_SCO, FC_FNN and PID performance in the given control situation are compatible.
- From control quality point of view including a minimum of control error, minimum of entropy production in a plant and in controller, and minimum of

control force, FC_SCO with 27 rules realizes more effective (optimal) control than FC_FNN with 125 rules.

Control task 2
In this case we have the following TS control situation:

- Model Parameters: k = 1; Initial conditions: [-1] [0.1];
- Rayleigh excitation (case 2 noise, max amplitude = 6);
- Step reference signal as shown in Fig.7.2-21.

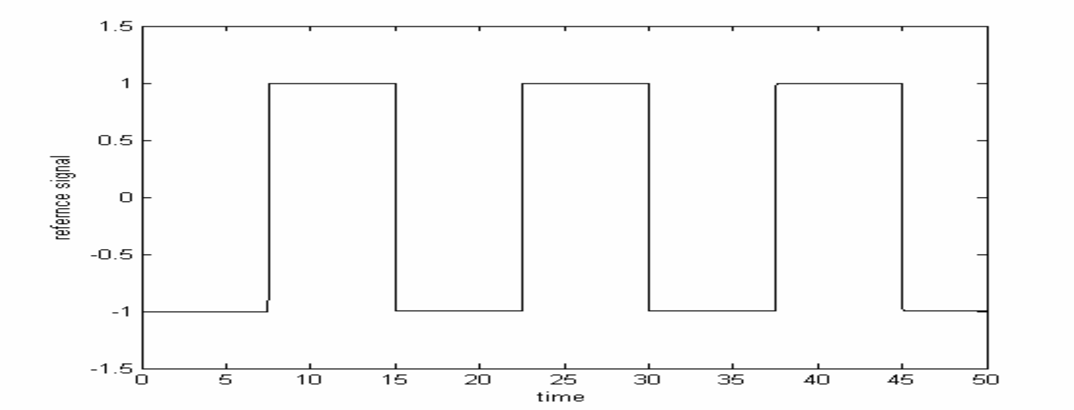


Figure 7.2-21. Van derPol oscillator. Reference signal for control task 2

GA-based TS design

At this step we design TS of optimal control based on the given control quality criterion as a GA fitness function. For this case we will use the following GA parameters:

- GA parameters: search space for K-gains parameters: (0,10);
- GA FF: “minimum control error”.

In Fig.7.2-22 comparison of motion under GA and PID control is shown.

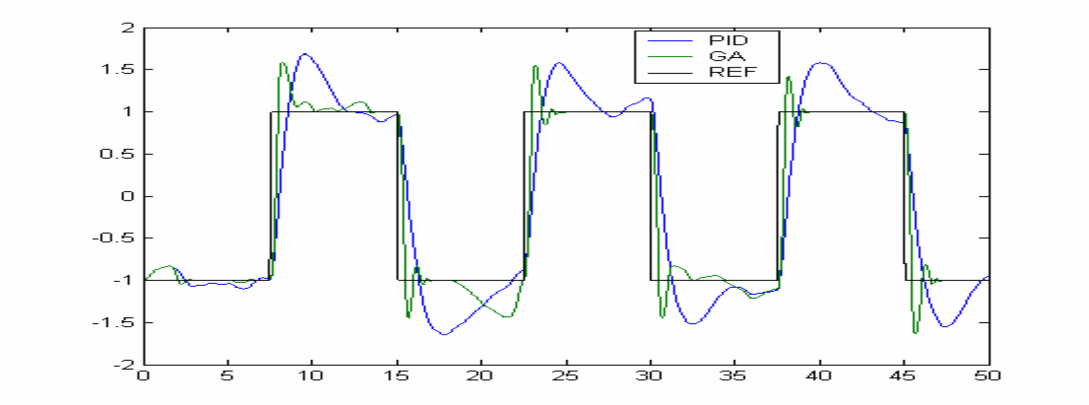


Figure 7.2-22. Van derPol oscillator. Motion under GA control. Control task 2

FNN-based KB FC design process (step 1 technology)

We will design FC-PID controller with three input variables to FC: $\left\{ e, \dot{e}, \int edt \right\}$ and three output variables of FC: $\left\{ k_p, k_d, k_i \right\}$.

AFM based KB design process is described as follows:

- Manual choice of numbers and shapes of membership functions for each input variables: 5;
- Complete number of fuzzy rules: 5x5x5= 125 rules;
- Number of activated rules in KB**: 125 rules.**

SC Optimizer-based KB FC design process (step 2 technology)

KB FC design process based on SC Optimizer is characterized as follows:

- *Creation of linguistic values* by GA1 : number of membership functions for each input variables: 9,9,7 ;
- *Rule selection* : SUM of firing strength criterion with automatic manual threshold;
- *KB optimization by* GA2: optimized KB contains **42 rules.**

Compare control quality of FC_SCO obtained by SC Optimizer with 42 rules, FC_FNN obtained by traditional SC approach based on FNN-tuning (with AFM tools) with 125 rules and traditional PID controller with constant gains K = (10 10 10).

In Figures 7.2-23, 7.2-24, 7.2-25 and 7.2-26 results of comparison of CO stochastic motion under three types of control are shown.

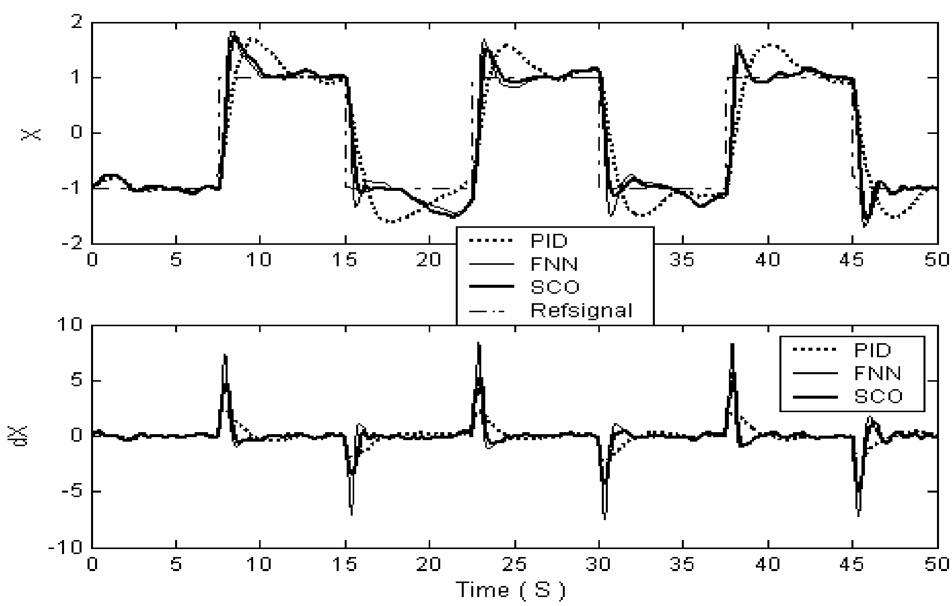


Figure 7.2-23. Van der Pol oscillator. TS control situation. *Control task* 2

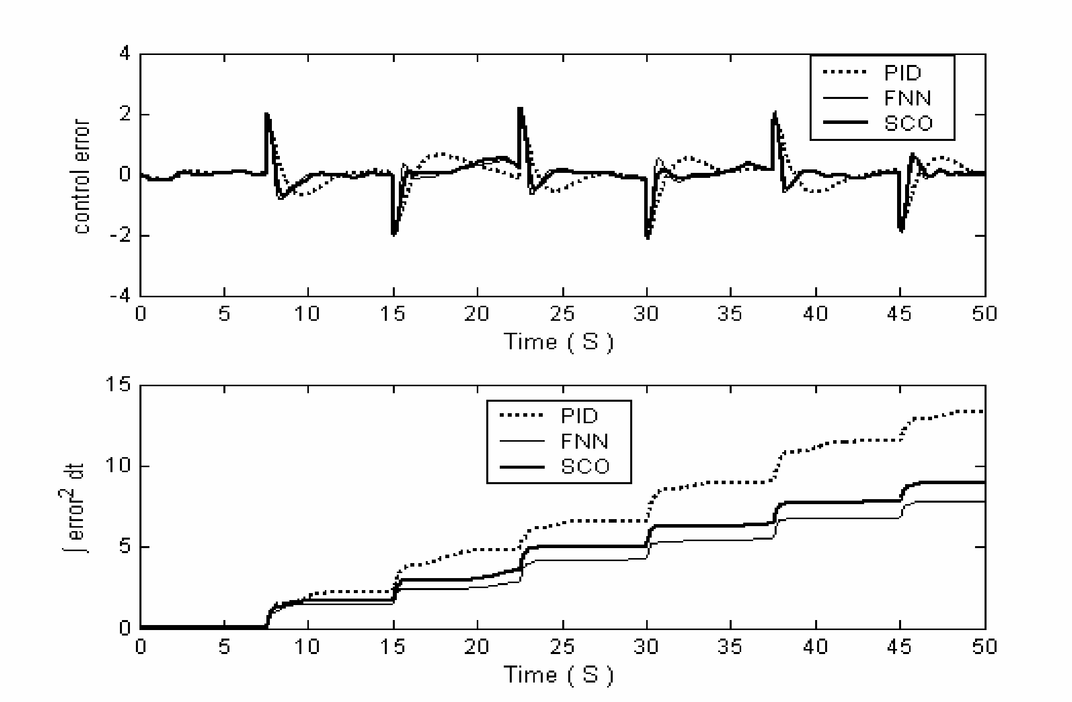


Figure 7.2-24. Van der Pol oscillator. Control error.
TS control situation. *Control task* 2

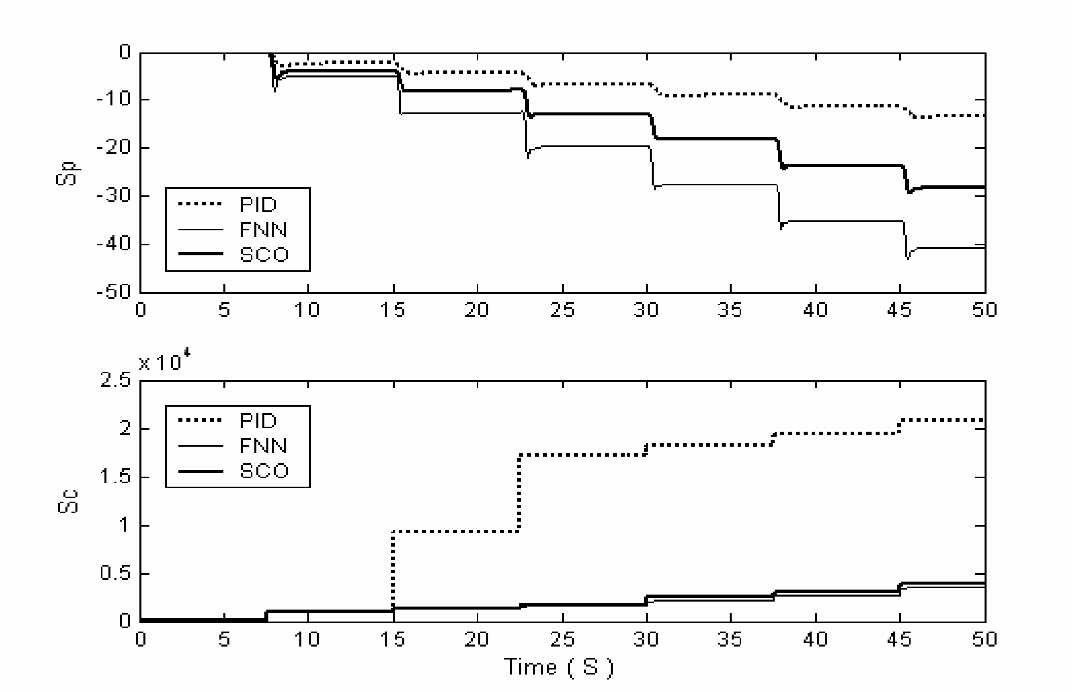


Figure 7.2-25. Van der Pol oscillator. Entropy productions of plant and Controller.
TS control situation. *Control task* 2

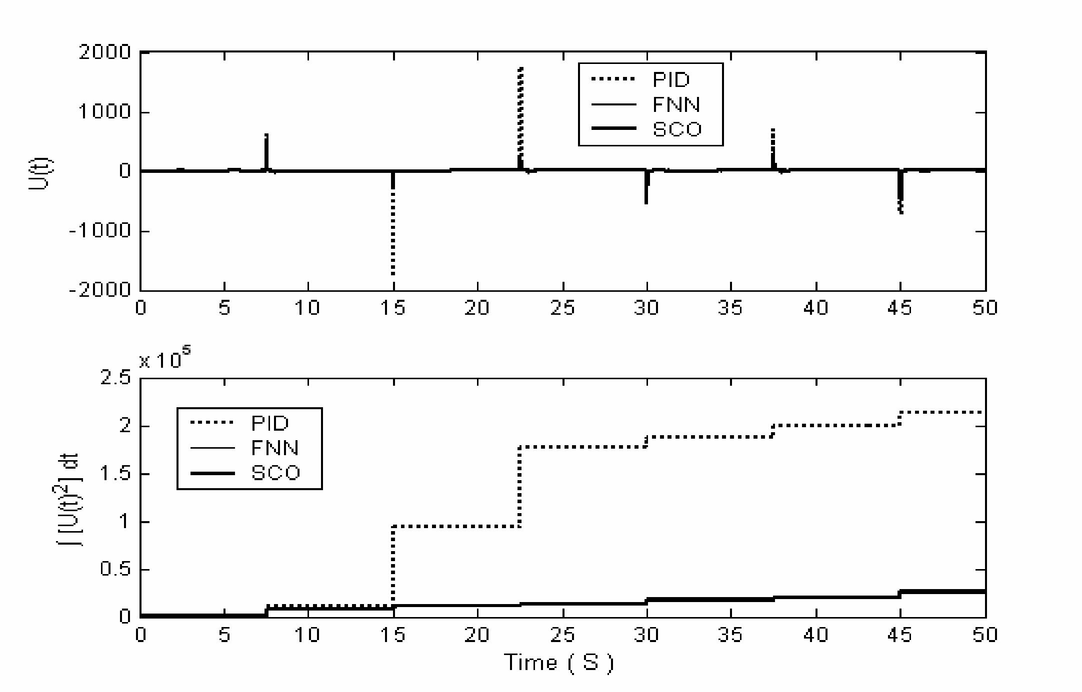


Figure 7.2-26. Van der Pol oscillator. Control force.
TS control situation. *Control task* 2

Control laws for TS control situation are shown in Fig.7.2-27.

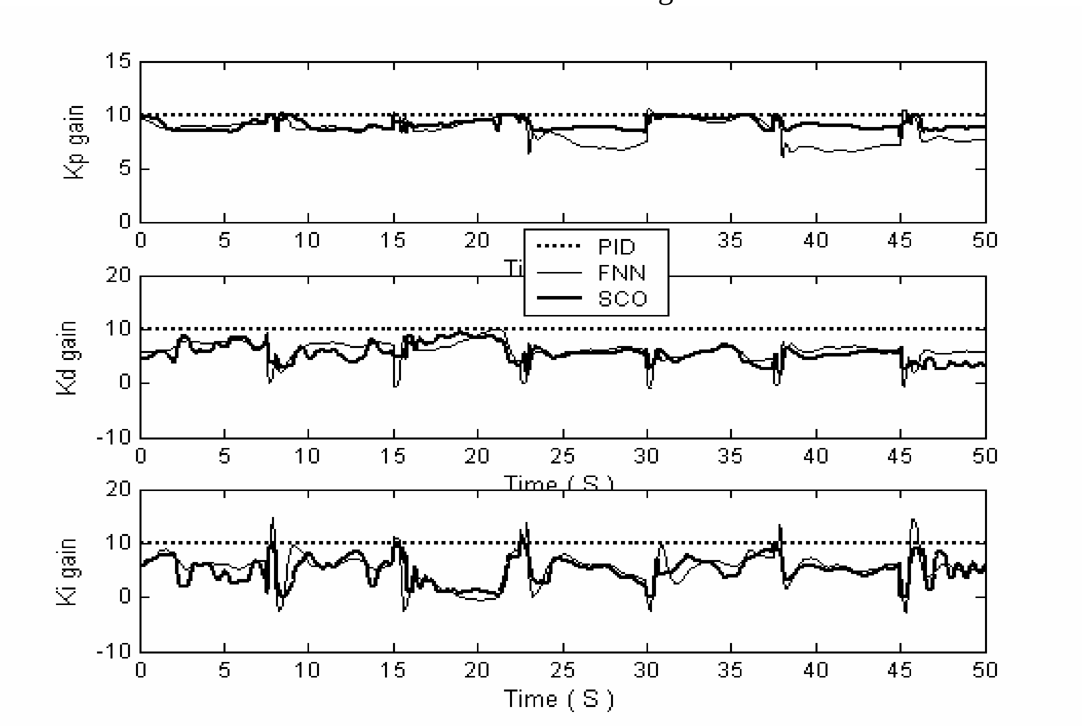


Figure 7.2-27. Van der Pol oscillator. Control laws.
TS control situation. *Control task* 2

Conclusions

- From control quality point of view (minimum of control error, minimum of entropy productions in a plant and in controller, and minimum of control force) FC_SCO and FC_FNN are more effective than classical PID.
- With respect to control error and control force FC_SCO and FC_FNN are compatible, but FC_SCO performance is more precise.
- FC_SCO has smaller KB (42 rules) than FC_FNN (125 rules).

Robustness investigation of KB FC designed for the given TS control situation

Let us take FC_SCO, FC_FNN developed for the TS control situation and use them in new control situation, where new type of noise (Gaussian, case 2) is considered.
In Fig.7.2-28 results of comparison of CO stochastic motion under three types of control in the new control situation are shown.

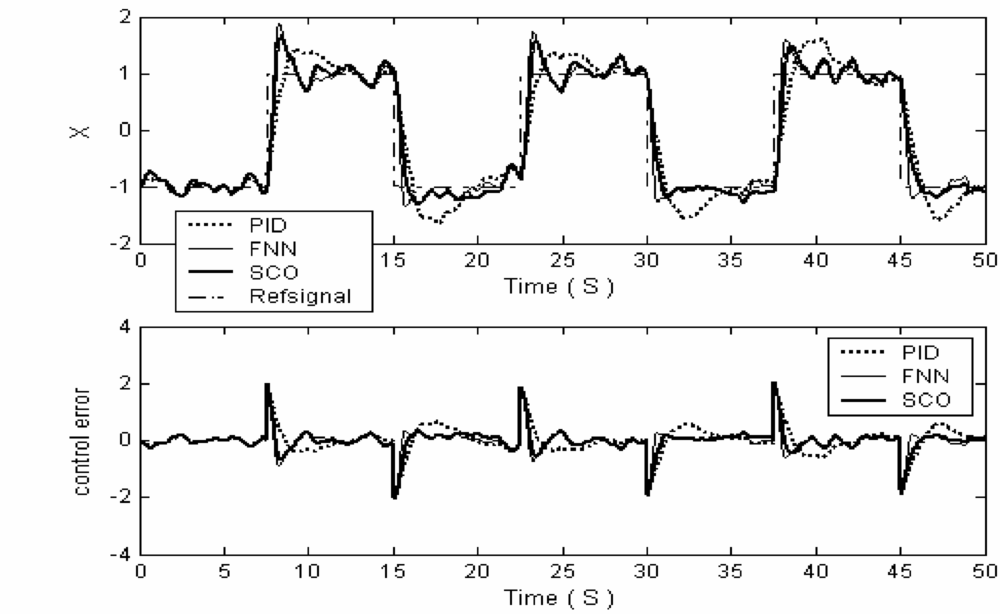


Figure 7.2-28. Van der Pol oscillator. New control situation. *Control task* 1

Conclusions

- FC_SCO and FC_FNN are robust relative to the given new control situation.
- From control quality point of view (minimum of control error, minimum of entropy productions in a plant and in controller, and minimum of control force) FC_SCO and FC_FNN are more effective than classical PID.
- With respect to control error and control force FC_SCO and FC_FNN are compatible.

Control task 3 (*stochastic noise compensation*)
In this case we have the following TS control situation:

- Motion under Rayleigh excitation (case 2, max amplitude = 6);
- Model Parameters: $k = 1$;

- Initial conditions: [-1] [0.1];
- Reference signal as free motion shown in Fig.7.2-29.

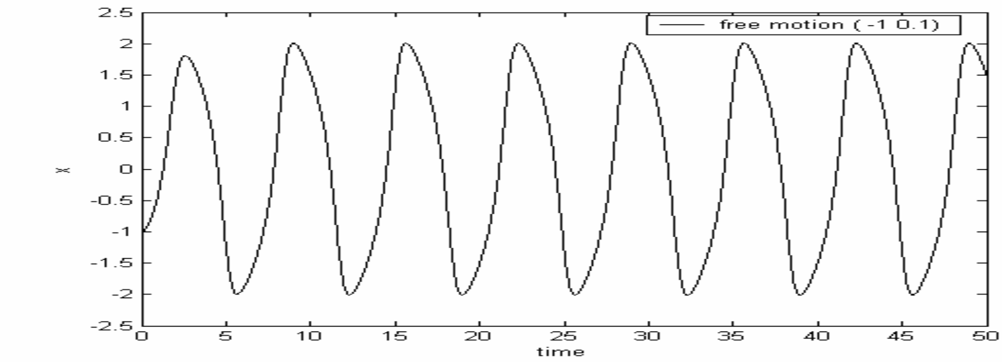


Figure 7.2-29. Van der Pol oscillator. Reference signal for control task 3

GA-based TS design

At this step we design TS of optimal control based on the given control quality criterion as a GA fitness function. For this case we will use the following GA parameters:

- GA parameters: search space for K-gains parameters: (0,10);
- GA FF: “minimum control error”.

In Fig.7.2-30 a comparison of CO motion under GA and PID control is shown.

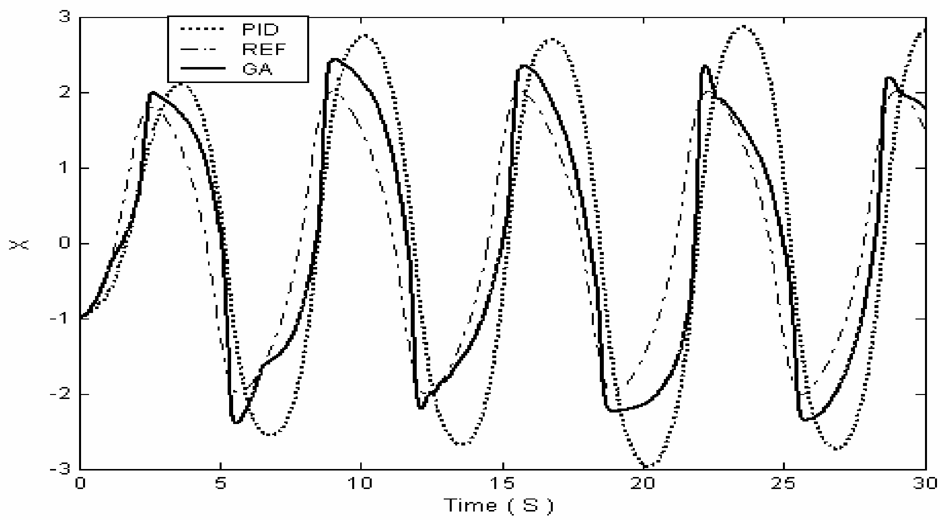


Figure 7.2-30. Van der Pol oscillator. GA- PID control. Control task 3

FNN-based KB FC design process (step 1 technology)

We will design FC-PID controller with 3 input variables to FC: $\left\{e, \dot{e}, \int edt\right\}$ and 3 output variables of FC: $\left\{k_p, k_d, k_i\right\}$. AFM based KB design process is described as follows:

- Manual design : Number of membership functions for each input variables: 5;
- Complete number of fuzzy rules: 5x5x5= 125 rules;

- Number of activated rules in KB**: 125 rules.**

AFM based MF representation is shown in Fig.7.2-31.

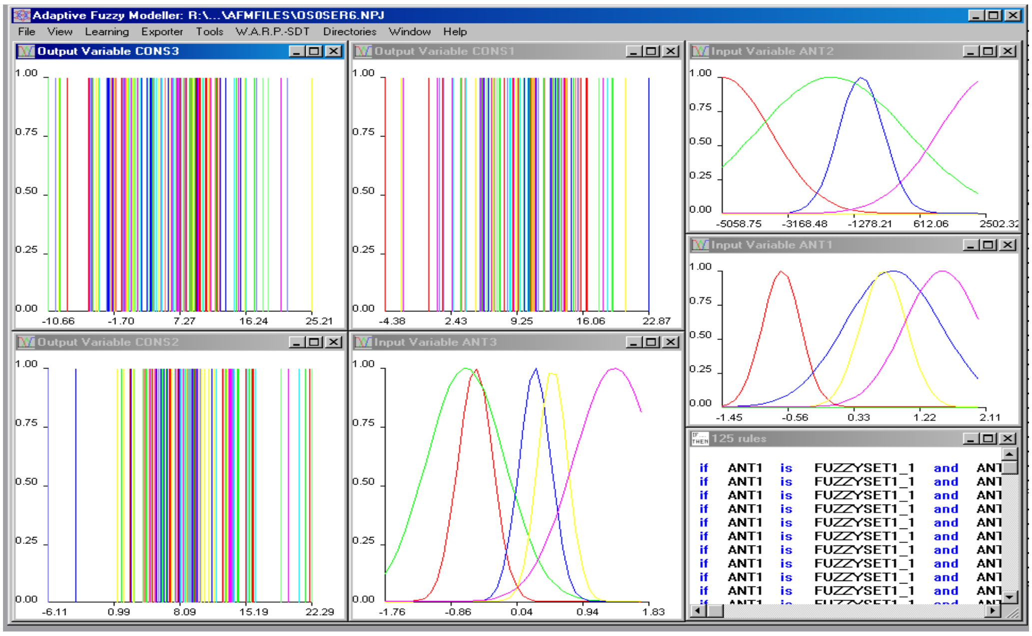


Figure 7.2-31**.** AFM based MF representation. Control task 3.

SC Optimizer-based KB FC design process (step 2 technology)

In this case KB design process is described as follows:

- *Creation of linguistic values* by GA1 : number of membership functions for each input variables: 7,9,9 ;
- Complete number of fuzzy rules: 7x9x9 = 567 rules;
- *Rule selection by*: SUM criterion with automatic manual threshold;
- GA2: optimized KB contains **57 rules.**

SC optimizer based MF representation is shown in Fig.7.2-32.

SCO based MF representation

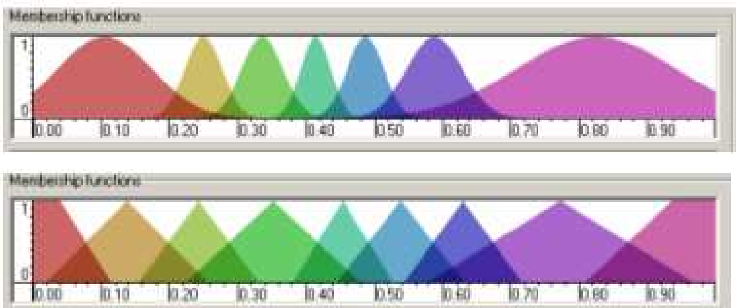


Figure7.2-32**.** SCO based MF representation. Control task 3.

Remark. You can see the difference between AFM and SCO MF representations for given example: in AFM only one type of MF shape for all input variable is allowed. In SC optimizer for each input variables optimal MF shapes are defined by GA.

Compare control quality of FC_SCO obtained by SC Optimizer with 42 rules, FC_FNN obtained by traditional SC approach based on FNN-tuning (with AFM tools) with 125 rules and traditional PID controller with constant gains K = (10 10 10).

In Figures 7.2-33, 7.2-34, 7.2-35, 7.2-36 and 7.2-37 results of comparison of CO motion under stochastic excitation and 3 types of control are shown.

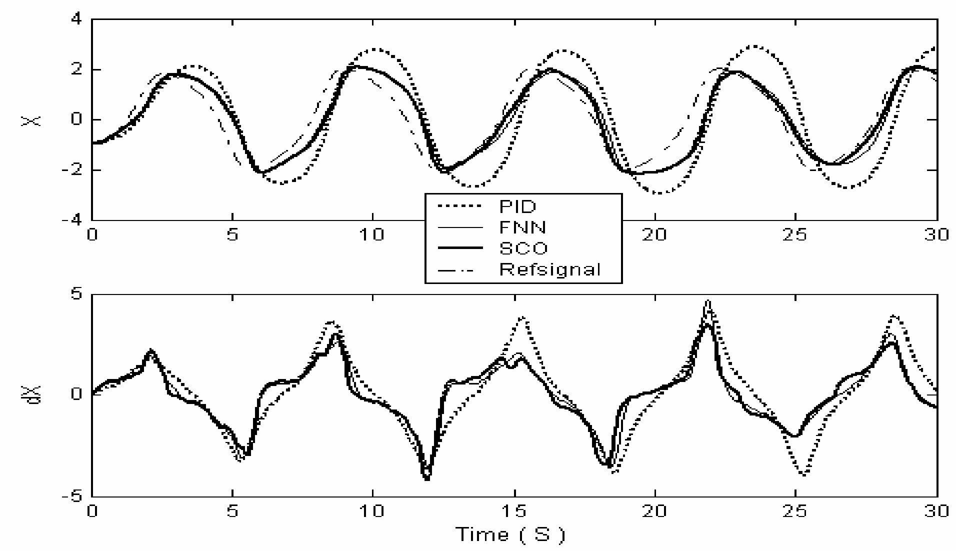


Figure 7.2-33. Van der Pol oscillator motion. TS control situation.. *Control task* 3

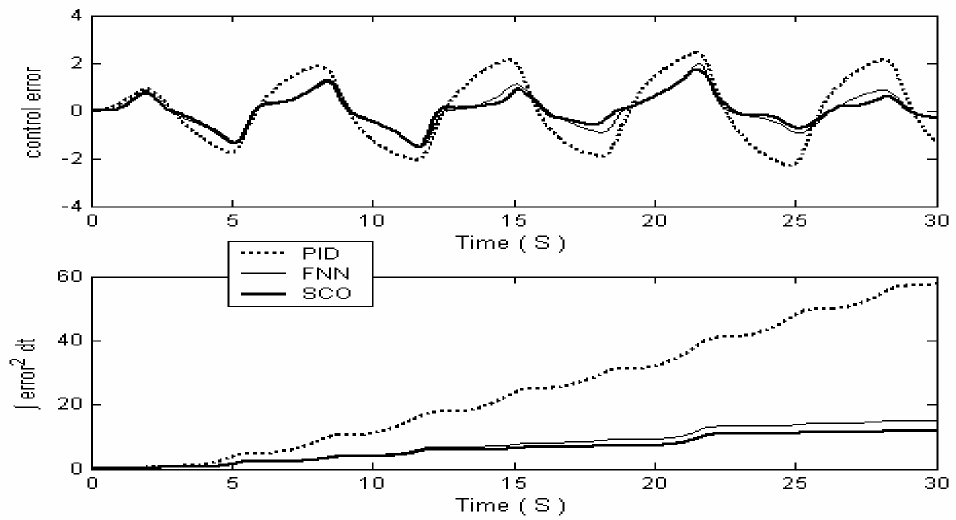


Figure 7.2-34. Van der Pol oscillator. Control error. TS control situation. *Control task* 3

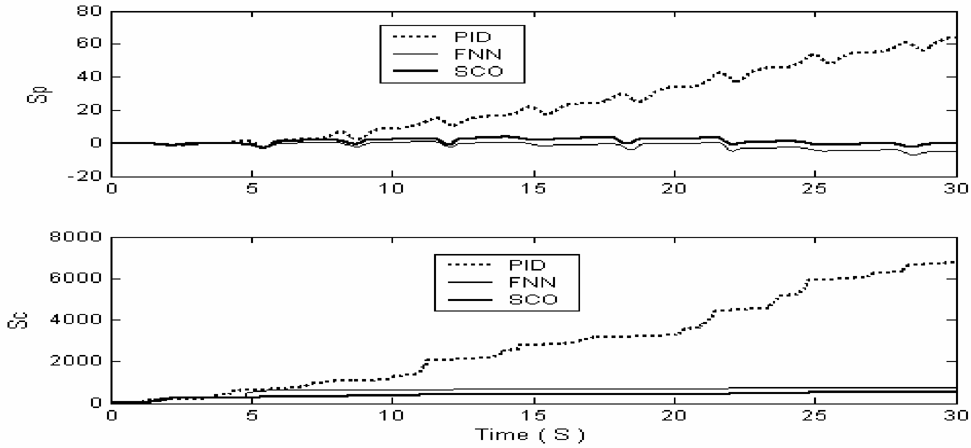


Figure 7.2-35. Van der Pol oscillator. Entropy productions of plant and Controller.
TS control situation. *Control task* 3

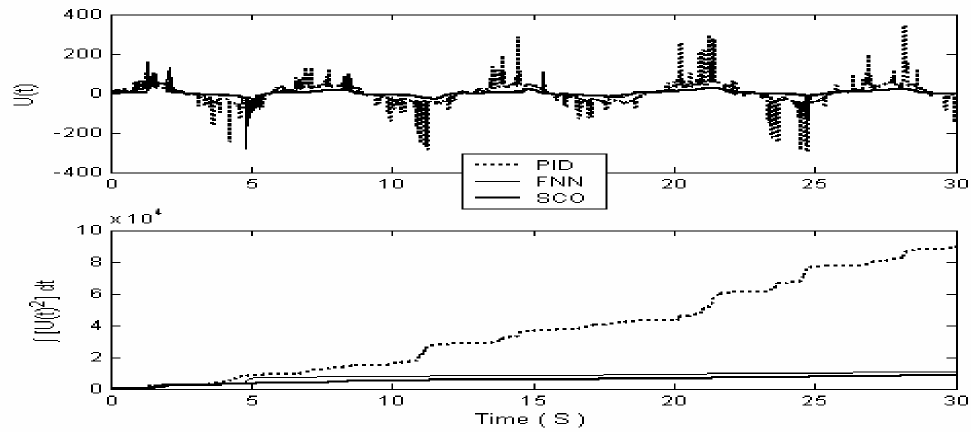


Figure 7.2-36. Van der Pol oscillator. Control force .TS control situation. *Control task* 3

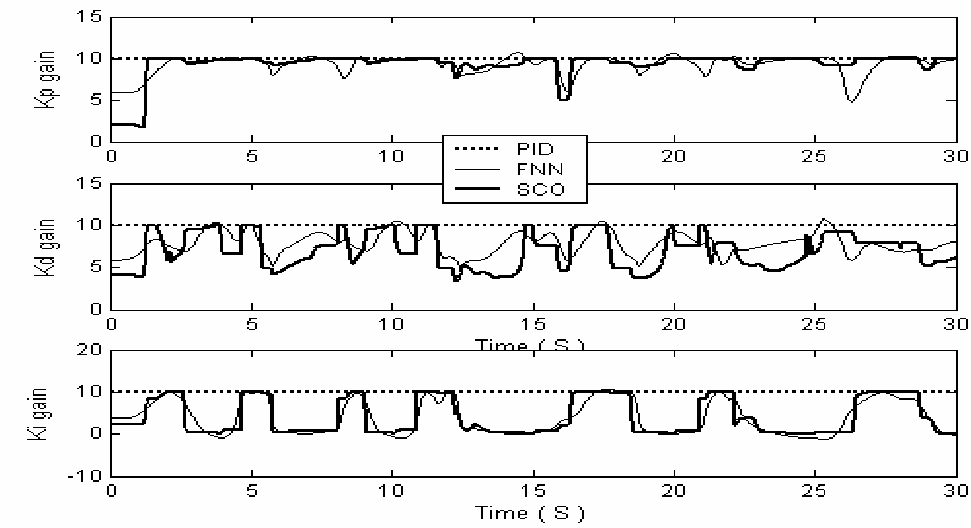


Figure 7.2-37. Van der Pol oscillator. Control laws .TS control situation. *Control task* 3.

Conclusions

- FC_SCO and FC_FNN are robust relative to the given new control situation.
- From control quality point of view (minimum of control error, minimum of entropy productions in a plant and in controller, and minimum of control force) FC_SCO and FC_FNN are more effective than classical PID.
- With respect to control error and control force FC_SCO and FC_FNN are compatible.

Robustness investigation of KB FC designed for the given TS control situation

Let us take FC_SCO, FC_FNN developed for the TS control situation and use them in new control situation, where new type of noise (Gaussian, case 2) is considered.
In Fig.7.2-38, 7.2-39, 7.2-40 and 7.2-41 results of comparison of CO motion under stochastic excitation and 3 types of control in the new control situation are shown.

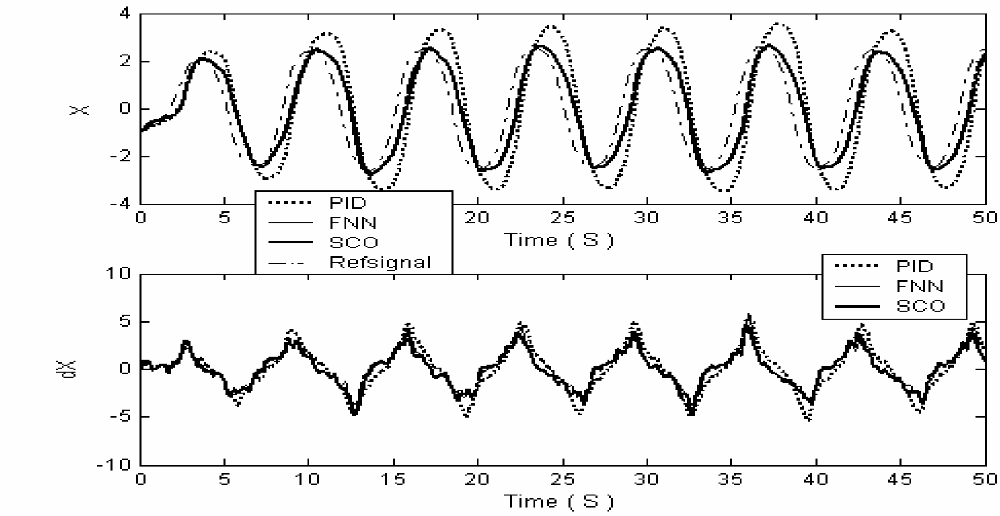


Figure 7.2-38. Van der Pol oscillator motion. New control situation. Control task 3

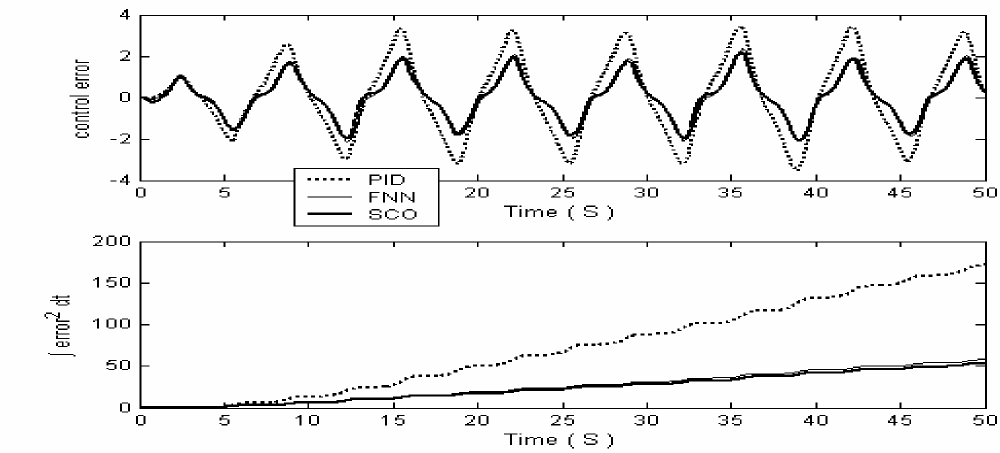


Figure 7.2-39**.** Van der Pol oscillator. Control error. New control situation. Control task 3.

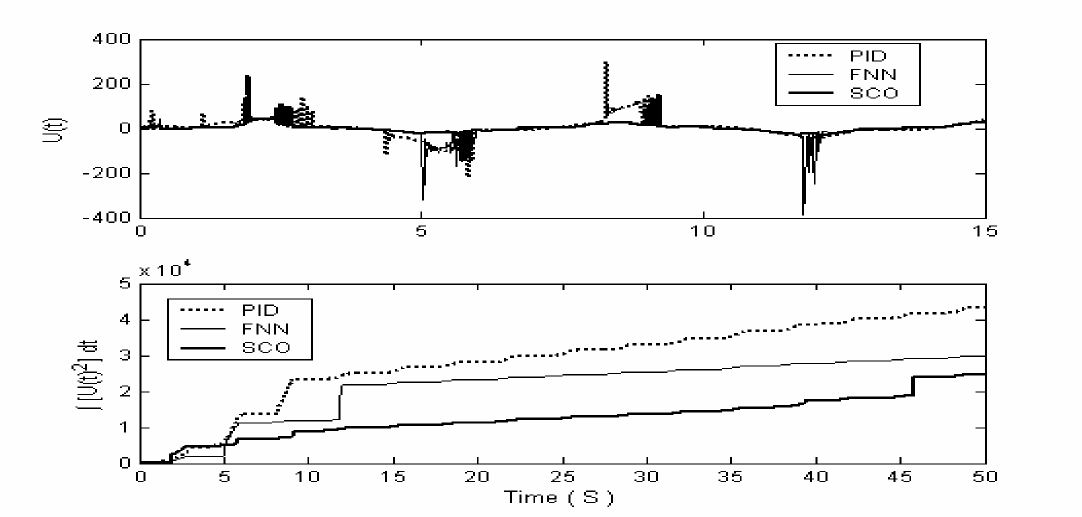


Figure 7.2-40. Van der Pol oscillator. Control force. New control situation. Control task 3.

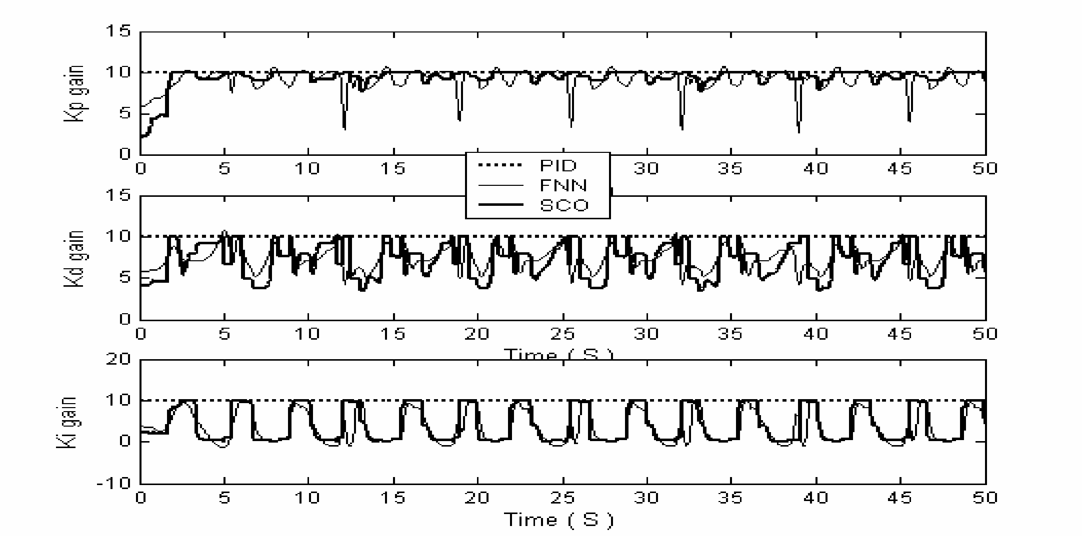


Figure 7.2-41. Van der Pol oscillator. Control laws. New control situation . Control task 3

Conclusions

- KB obtained from TS with Rayleigh noise can control new situations with Gaussian noise, i.e. designed KB FC (as by SC Optimizer and by AFM) are robust.
- From control quality point of view (minimum of control error, minimum of entropy productions in a plant and in controller, and minimum of control force) FC_SCO and FC_FNN are more effective than classical PID.
- With respect to control error FC_SCO and FC_FNN are compatible.
- With respect to control force FC_SCO is more effective.

7.3 Example 3: *Oscillator with hysteresis effects*

Equation of motion:

$$\begin{cases} \ddot{x} + \delta\dot{x} + kx + z = \xi(t) + u(t) \\ \dot{z} = A\dot{x} - \{\beta|\dot{x}|z + \gamma\dot{x}|z|\} \end{cases}, \frac{dS_x}{dt} = \delta\dot{x}\dot{x}$$

Here $\xi(t)$ is a given stochastic excitations with an appropriate probability density function, and $u(t)$ is a control force.

Investigate free motion behavior of given CO under the following model parameters and initial conditions:

- model parameters: $\delta = 0.1; A = 0.95; \beta = 0.1; k = 4; \gamma = 0.255$;
- initial conditions: $[x_0\ z_0][\dot{x}_0\ \dot{z}_0]$ = [10 5] [1 0].

In Fig. 7.3-1, 7.3-2 and 7.3-3 free motion (dynamic and thermodynamic behavior) of control object with the given above parameters are shown.

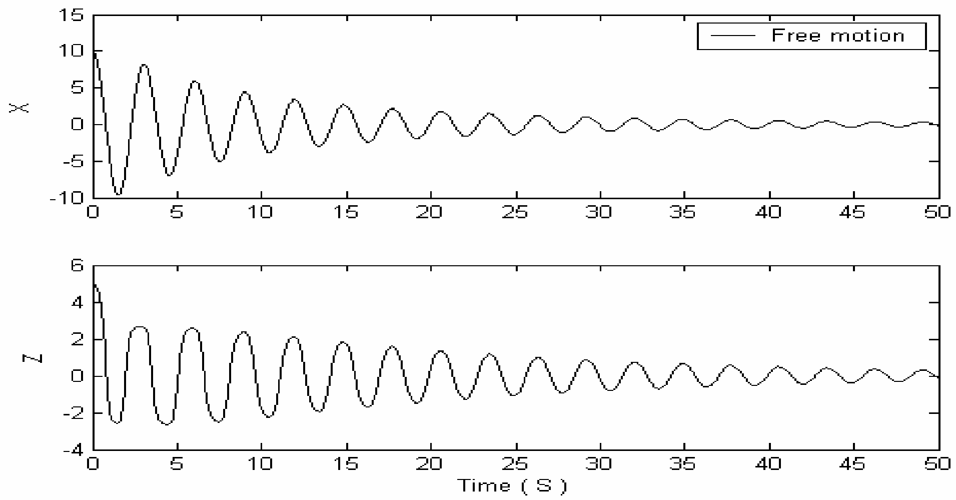


Figure 7.3-1. Example 3 oscillator. Free motion

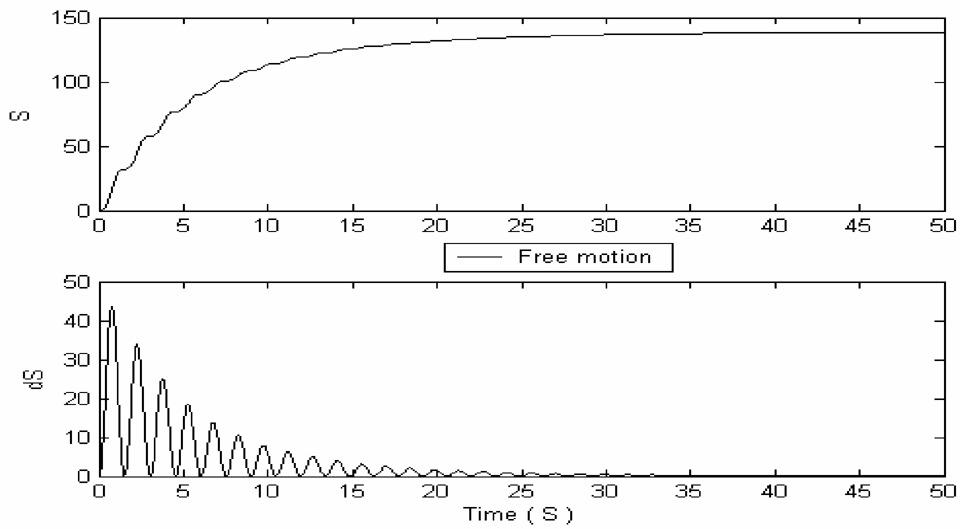


Figure 7.3-2. Example 3 oscillator. Thermodynamic characteristics

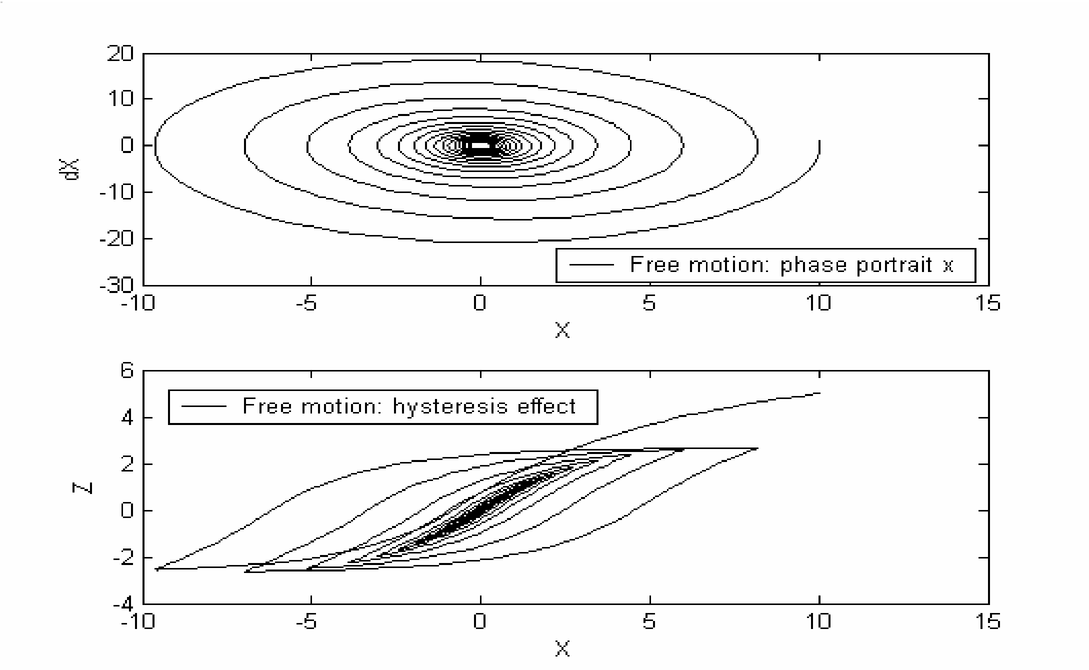


Figure 7.3-3. Example 3 oscillator. Phase portrait and hysteresis effect

Simulation results show that considered CO is a *stable* dynamic system.
Consider CO behaviour under two different types of stochastic excitations: Gaussian and Rayleigh noises shown in Fig. 7.3-4.

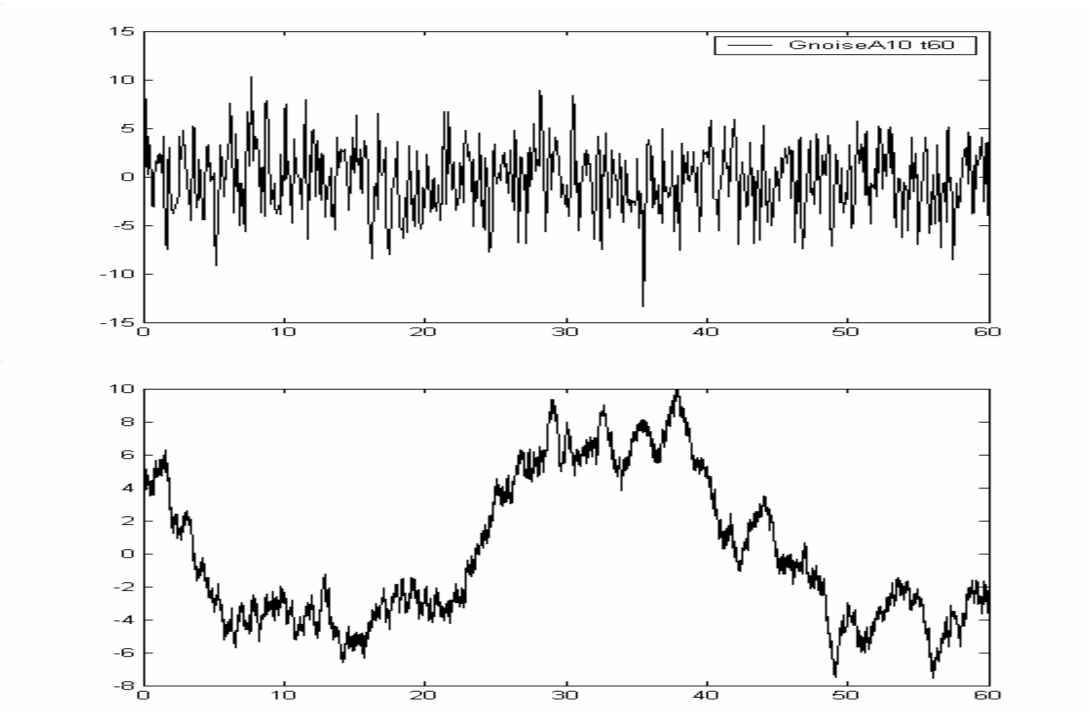


Figure 7.3-4. Two different stochastic noises: Gaussian (top) and Rayleigh (below)

In Fig. 7.3-5, 7.3-5 and 7.3-7 stochastic motion (dynamic and thermodynamic behavior) of CO under two stochastic noises is shown.

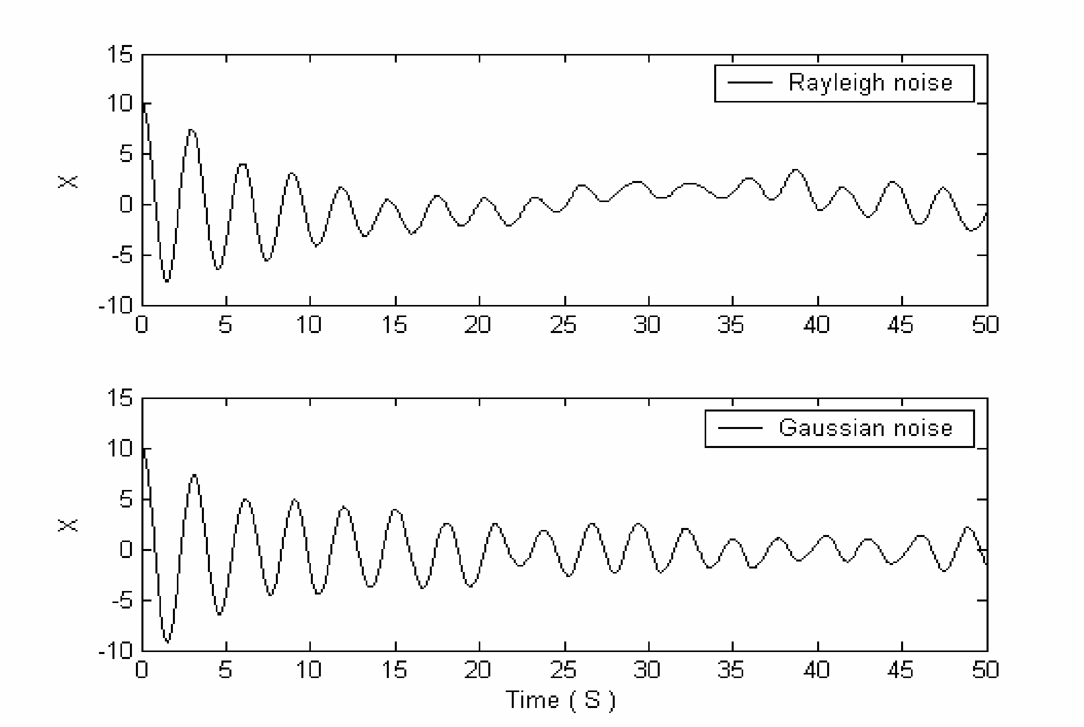


Figure 7.3-5. Example 3 oscillator motion under two types of noises

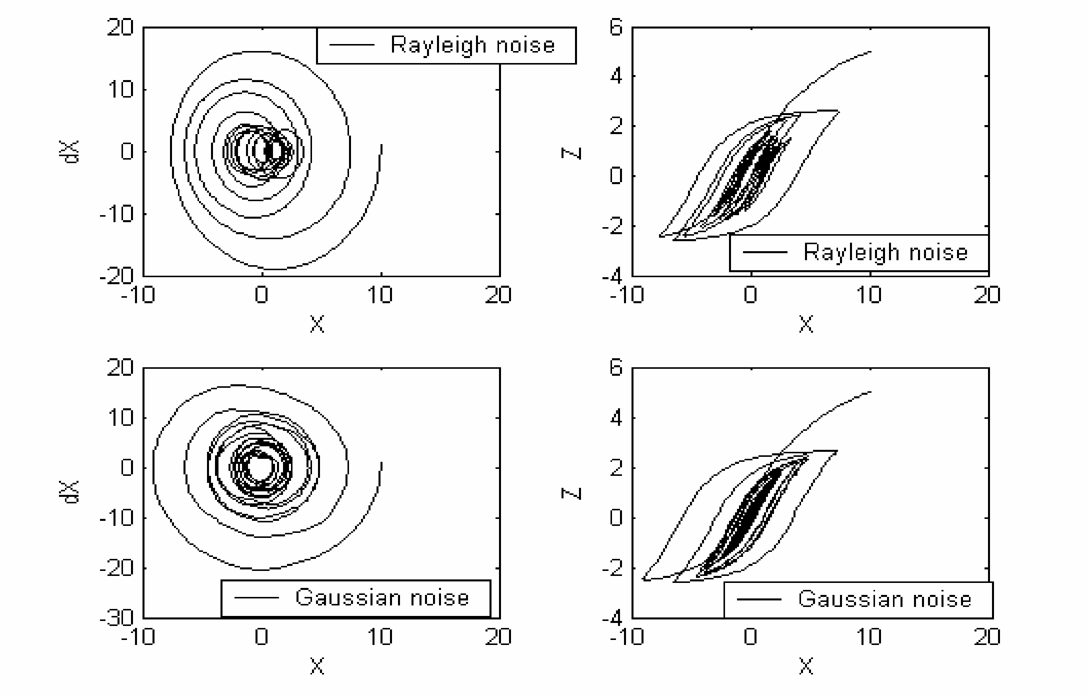


Figure 7.3-6. Example 3 oscillator. Phase portraits

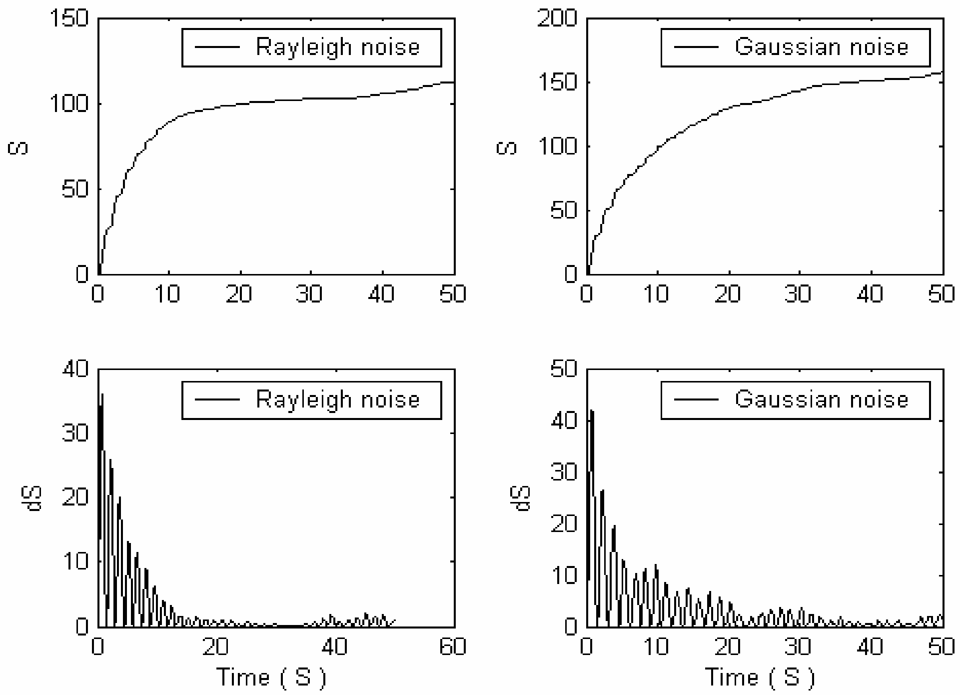


Figure 7.3-7. Example 3 oscillator. Thermodynamic characteristics

Dynamic behavior of CO under Rayleigh excitation is more complicated.

Consider the following *control task*: in the presence of Rayleigh (Gaussian) noise maintain a motion of CO at the given reference signal $x_{ref} = -2$.

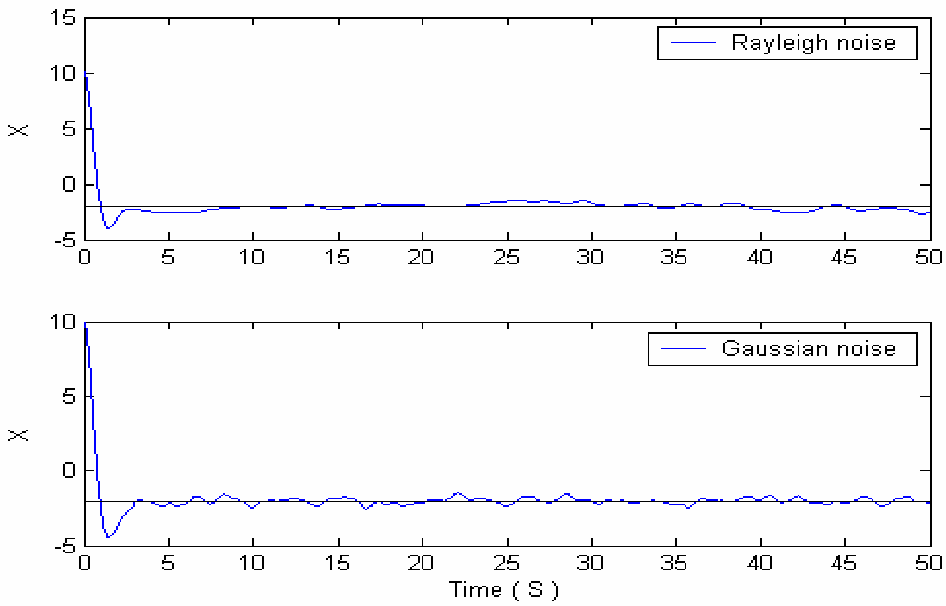


Figure 7.3-8. Example 3 oscillator. Motion under classical PID control.

Consider a PID controller (with K-gains $K = \{k_p, k_d, k_i\} = \{3\,3\,3\}$) for control a stochastic motion of CO under complicated noises mentioned above. Figures 7.3-8, 7.3-9, 7.3-10 and 7.3-11 show dynamic and thermodynamic characteristics of CO behavior under classical PID control.

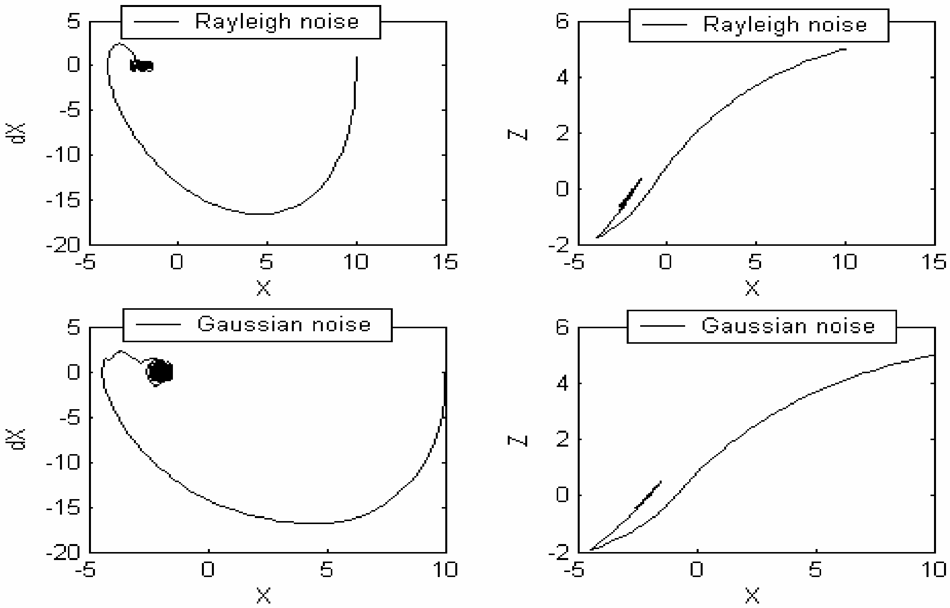


Figure 7.3-9. Example 3 oscillator. Phase portraits of stochastic motion

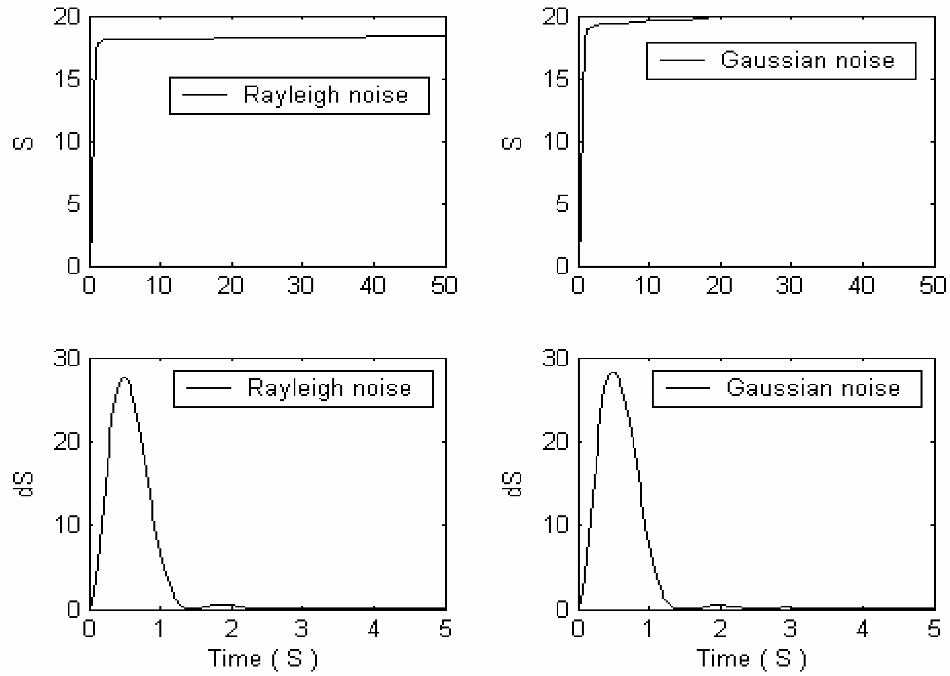


Figure 7.3-10. Example 3 oscillator. Thermodynamic characteristics

Let us design intelligent control system for the given above control problem by using our KB FC design tools and compare results with traditional PID Controller.

We have the following TS control situation:

- Model Parameters: $\delta = 0.1; A = 0.95; \beta = 0.1; k = 4; \gamma = 0.255$;
- Initial conditions: [10 5] [1 0];
- *Gaussian* excitation with max amplitude = 10 (see Fig.7-3.4, top);
- Reference signals: $x_{ref} = -2$.

GA-based TS design

At this step we design TS of optimal control based on the given control quality criterion as a GA fitness function. For this case we will use the following GA parameters:

- GA search space for K-gains coefficients: (0,3);
- GA FF as a minimum of “control error”.

In Fig. 7.3-11 simulation results with GA-PID control and PID control are shown. By using simulation results of GA-PID control we design a Teaching signal (TS) will be applied in step 1 and 2 technologies shown below.

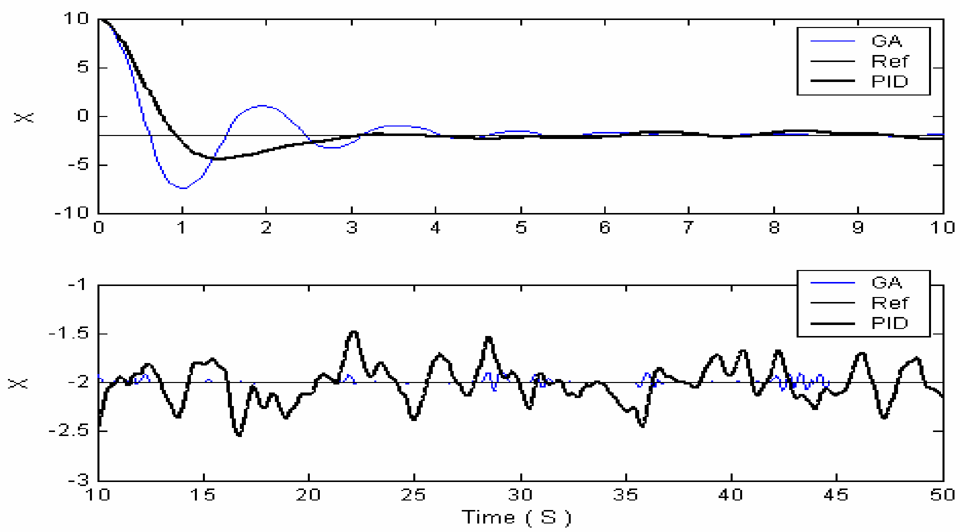


Figure 7.3-11. Example 3 oscillator. Stochastic motion under GA-PID control

Simulation results in Fig. 7.3-11 show that at 10 to 50 seconds of control GA-PID control has smaller control error than traditional PID control error.

FNN-based KB FC design process (step 1 technology)

We will design FC-PID controller which has three input variables to FC: $\left\{e, \dot{e}, \int edt\right\}$ and three output variables of FC: $\left\{k_p, k_d, k_i\right\}$.

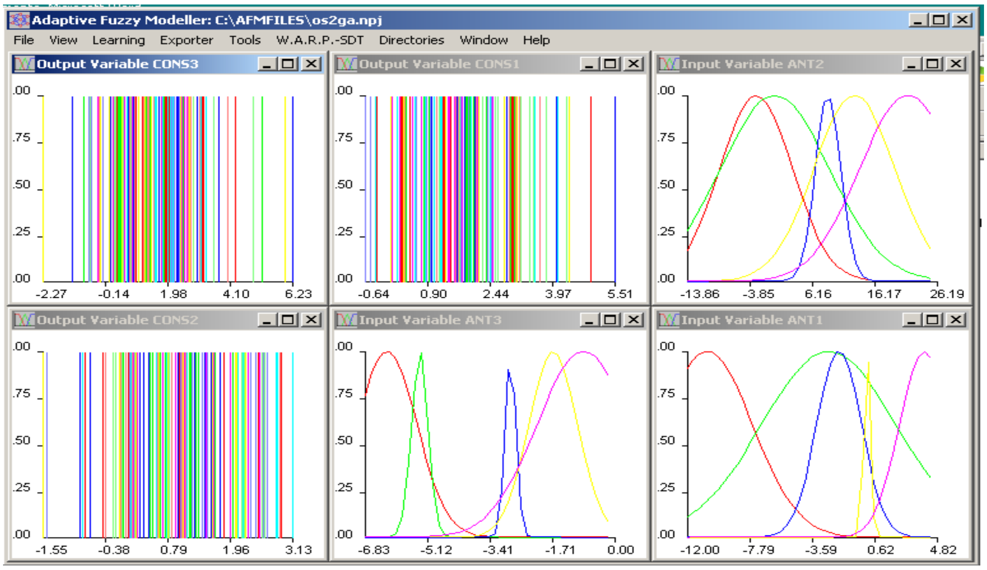


Figure 7.3-12. Example 3 oscillator. AFM representation of membership functions

At the stage of FNN tuning with error back propagation algorithm, we will use the AFM tools developed by ST Microelectronics. AFM based KB design process is described as follows:

- Manual design of numbers and shapes of membership functions for each input variables: 5;
- Complete number of fuzzy rules: 5x5x5= 125 rules;
- Number of activated rules in KB**: 125 rules.**

In Fig. 7.3-12 AFM representation of membership functions for input FC variables is shown.

SC Optimizer-based KB FC design process (step 2 technology)

KB FC design process based on SC optimizer is described as follows:

- *Creation of linguistic values* by GA1 : number of membership functions for each input variables: 7,6,8 ;
- Complete number of fuzzy rules: 7x6x8 = 336 rules;
- *Rule selection* : SUM of firing strength criterion with manual threshold value = 0.02;
- *KB optimization by* GA2 : optimized KB consists of **23 rules.**

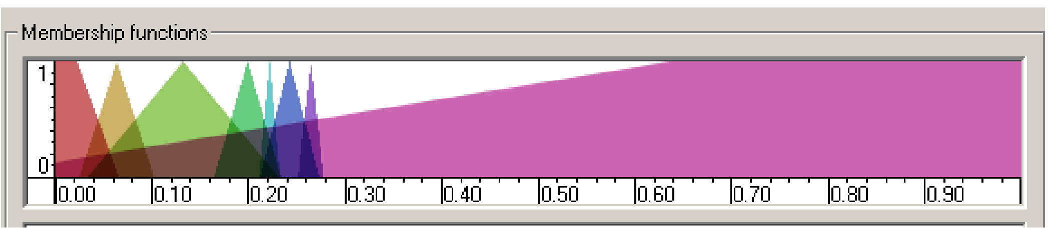


Figure 7.4-13. Example of membership function representation in SC Optimizer

In Fig. 7.3-13, SCO representation of membership functions for the third input variable of FC is shown.
Remark. In AFM number and MF shape are selected manually by a user. In SC optimizer for each input variables optimal MF shapes and MF number are defined by GA.
Compare control quality of FC_SCO obtained by SC Optimizer with 23 rules, FC_FNN obtained by traditional SC approach based on FNN-tuning (with AFM tools) with 125 rules and traditional PID controller with constant gains K = (3 3 3).
In Figures 7.3-14, 7.3-15, 7.3-16, 7.3-17 and 7.4-18 results of comparison of CO stochastic motion under 3 types of control are shown.

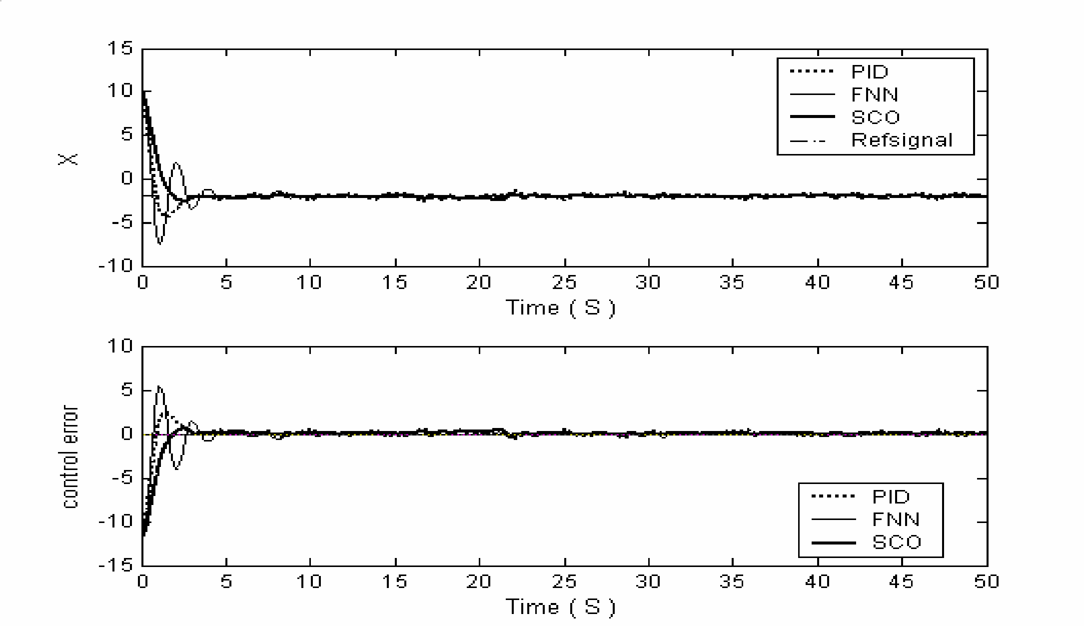


Figure 7.3-14. Example 3 oscillator. Coordinate motion and control error. TS control situation

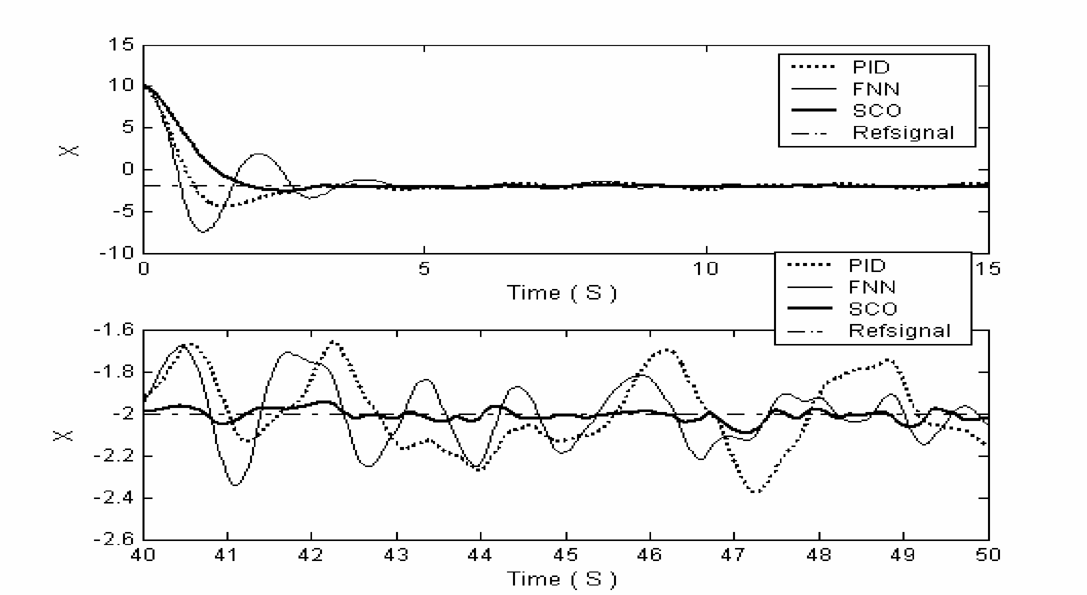


Figure 7.3-15. Example 3. Motion at fist and end intervals of time. TS control situation

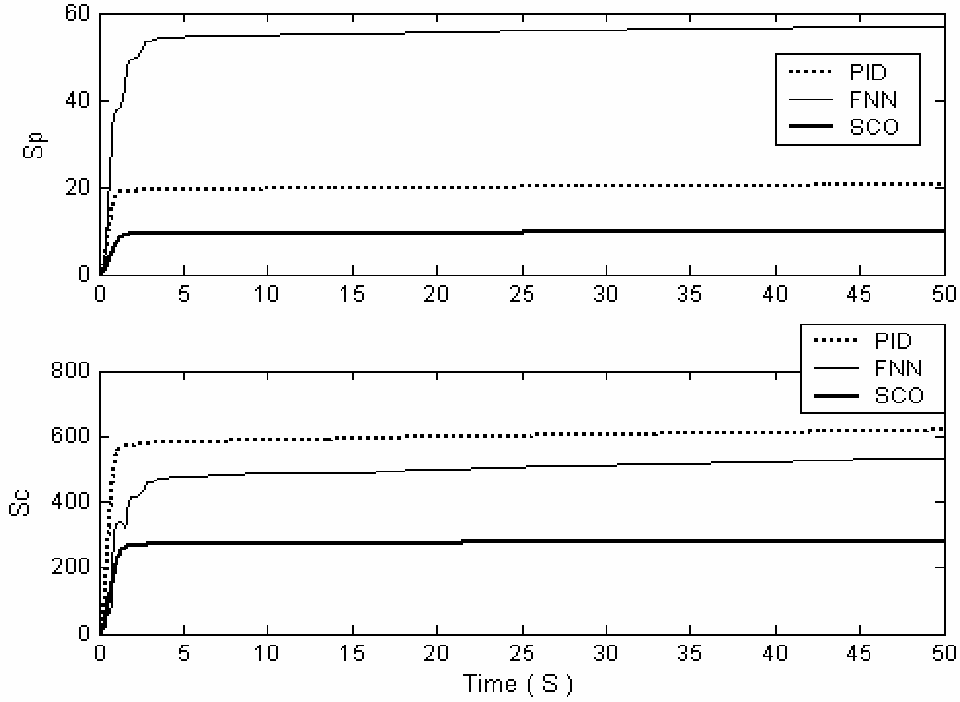


Figure 7.3-16. Example 3 oscillator. Entropy production in plant and in controller. TS control situation

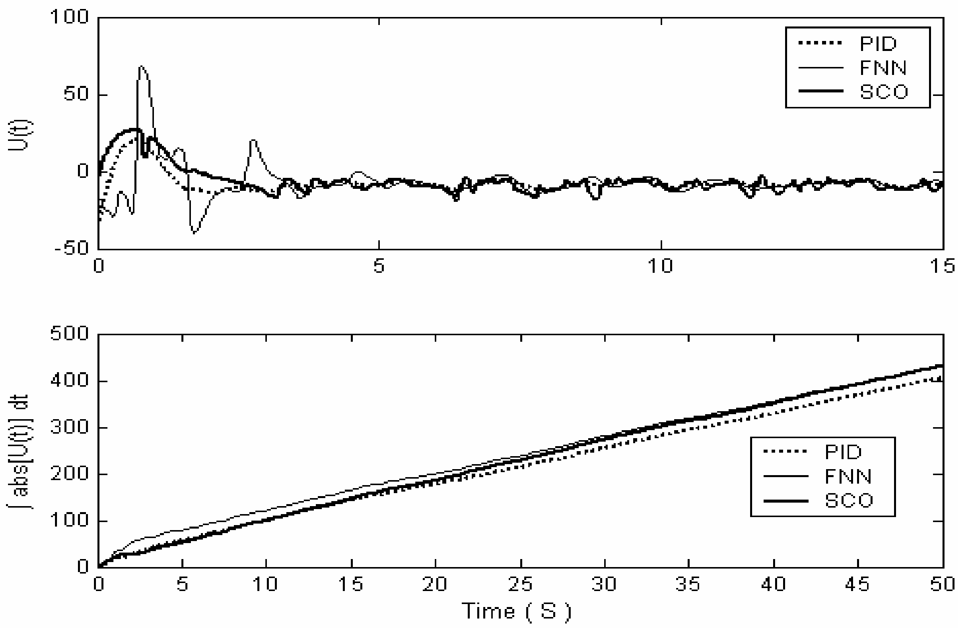


Figure 7.3-16. Example 3 oscillator. Control force and its integral value. TS control situation

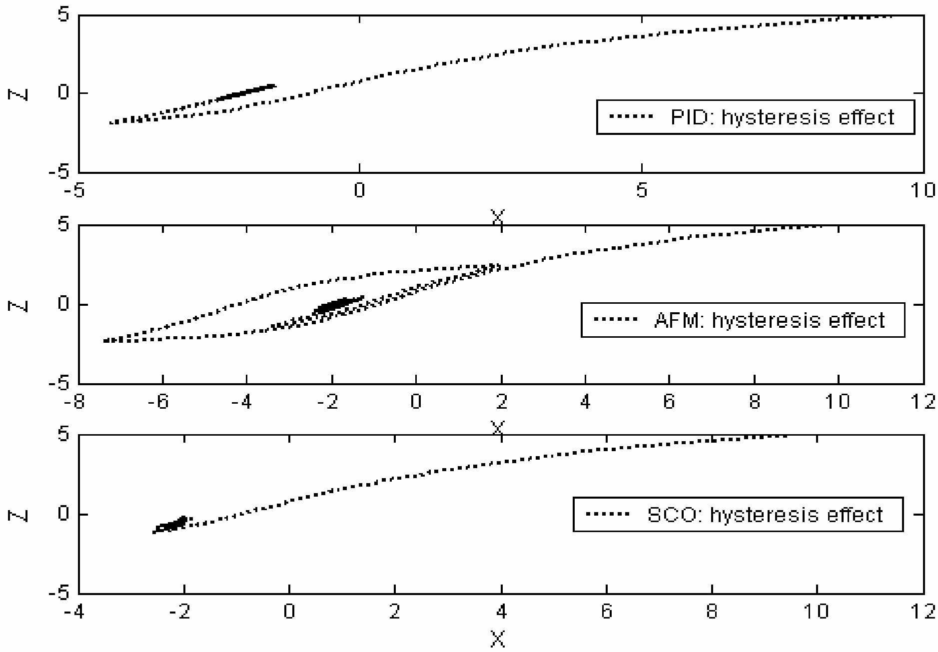


Figure 7.3-17. Example 3 oscillator. Hysteresis effect. TS control situation

Control laws for TS control situation are shown in Fig.7.3-18.

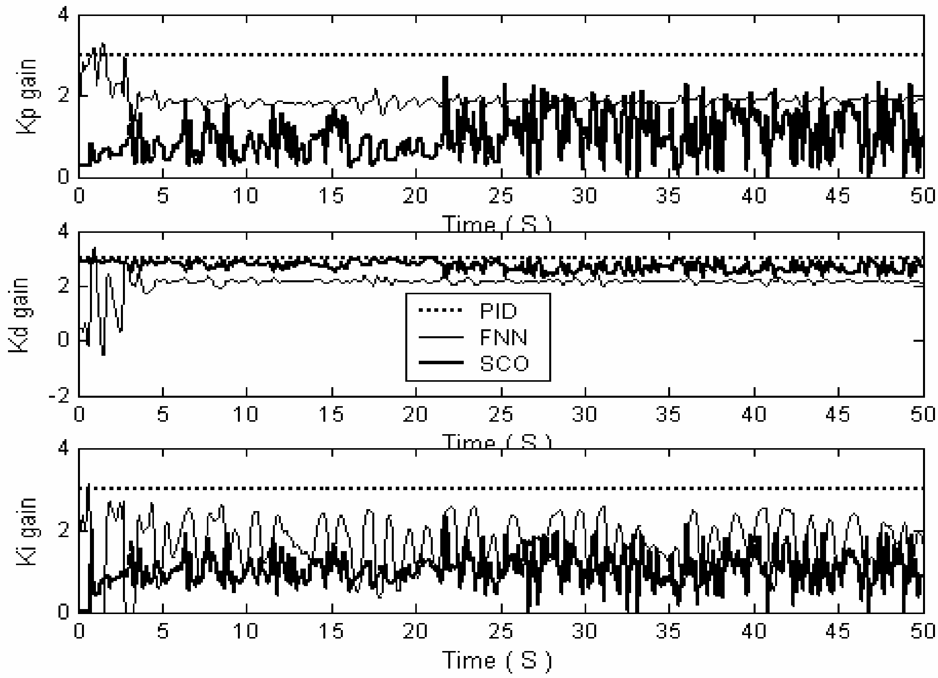


Figure 7.3-18. Example 3 oscillator. Control laws. TS control situation

Conclusions

- From control quality point of view including a minimum of control error, a minimum of entropy productions in a plant and in controllers, and a minimum of control force, FC_SCO is more effective than FC_FNN and classical PID.
- KB FC designed by SC Optimizer has smaller number of rules (**23** rules) than KB FC_FNN (**125** rules).

Robustness investigation of KB FC designed for the given TS control situation

Let us take FC_SCO, FC_FNN developed for the TS control situation and use them in new control situation, where

- new initial conditions [5 5] [0.1 0.1];
- new reference signal $x_{ref} = 0$, and
- new type of noise (Rayleigh) are considered.

In Fig.7.3-19 results of comparison in the new control situation are shown.

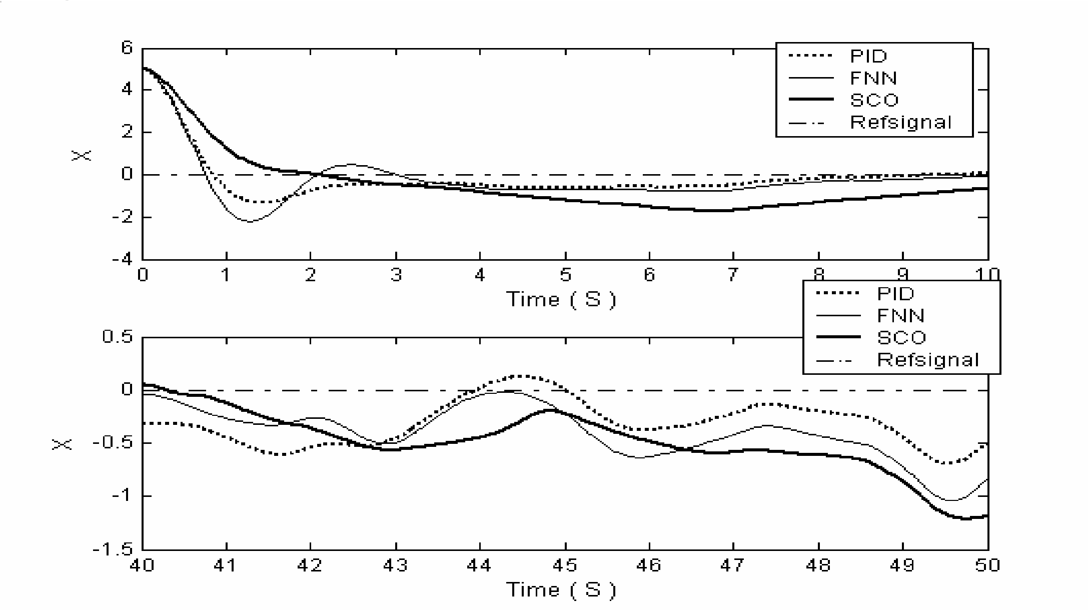


Figure 7.3-19. Example 3 oscillator. Motion at fist and end intervals of time. New control situation.

Conclusion

- Simulation results show that designed FC-KB applied in the new control situation has big control error, i.e. it's not robust. *Designed KB was obtained from TS control situation where Gaussian noise is used*.

Let us change TS control situation where Rayleigh noise will be considered.

Consider now FC KB design from GA simulations with Rayleigh noise. In this case TS control situation is as follows:

- Rayleigh excitation with max amplitude = 10 (see Fig.7.3-4);
- Model Parameters: $\delta = 0.1; A = 0.95; \beta = 0.1; k = 4; \gamma = 0.255$;

- Initial conditions: [10 5] [1 0].
- Reference signal: $x_{ref} = -2$.

GA-based TS design

At this step we design TS of optimal control based on the given control quality criterion as a GA fitness function. For this case we will use the following GA parameters:

- GA search space for K-gains coefficients: (0,3);
- GA FF as a minimum of "control error".

FNN-based KB FC design process (step 1 technology)

AFM based KB design process is described as follows:

- Manual design : Number of membership functions for each input variables: 5;
- Complete number of fuzzy rules: 5x5x5= 125 rules;
- Number of activated rules in KB**: 125 rules.**

SC Optimizer-based KB FC design process (step 2 technology)

The process of KB FC design based on SC optimizer is described as follows:

- *Creation of linguistic values* by GA1 : number of membership functions for each input variables: 8,8,8 ;
- Complete number of fuzzy rules: 8x8x8 = 512 rules;
- *Rule selection* : SUM of firing strength criterion with automatic manual threshold;
- *KB optimization by* GA2: optimized KB consists of **52 rules.**

Compare control quality of FC_SCO obtained by SC Optimizer with 52 rules, FC_FNN obtained by traditional SC approach based on FNN-tuning (with AFM tools) with 125 rules and traditional PID controller with constant gains K = (3 3 3).
In Figures 7.3-20, 7.3-21, 7.3-22, 7.3-23 and 7.3-24 results of comparison of CO motion under stochastic excitation and three types of control are shown. Control laws for TS control situation are shown in Fig.7.3-25.

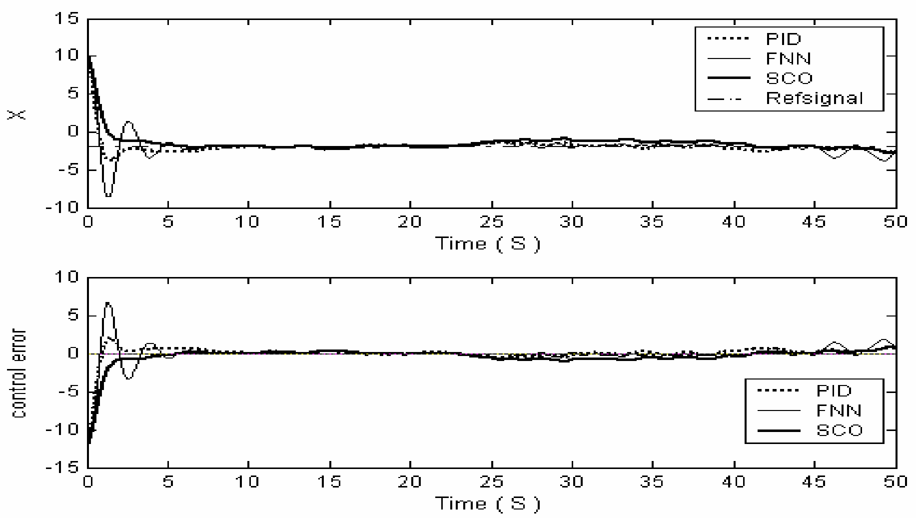


Figure 7.3-20. Example 3 oscillator. Motion and control error. TS control situation

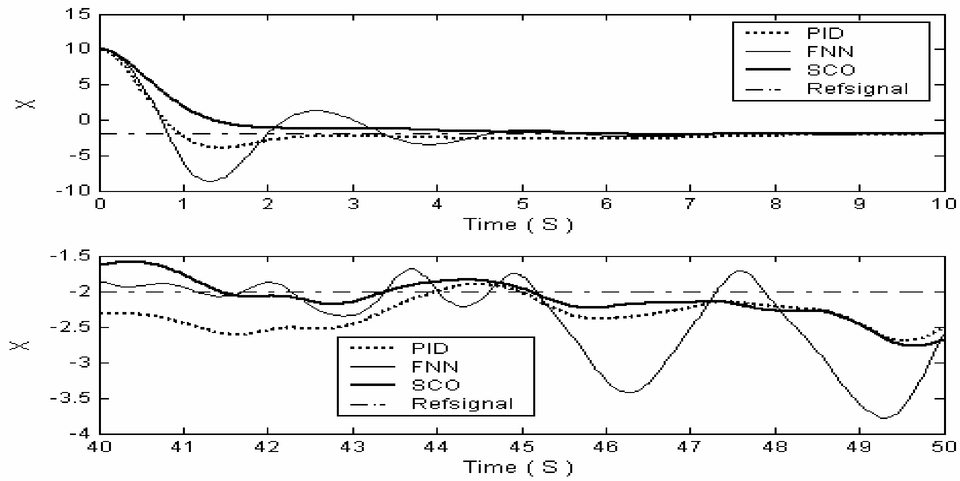


Figure 7.3-21. Example3. Motion at fist and end intervals of time. TS control situation

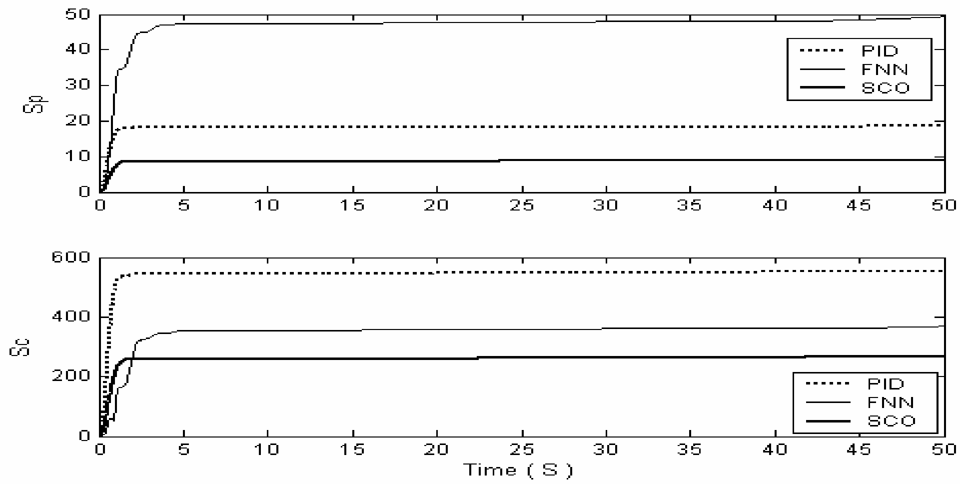


Figure 7.3-22. Example 3. Entropy production in plant and in controller. TS control situation

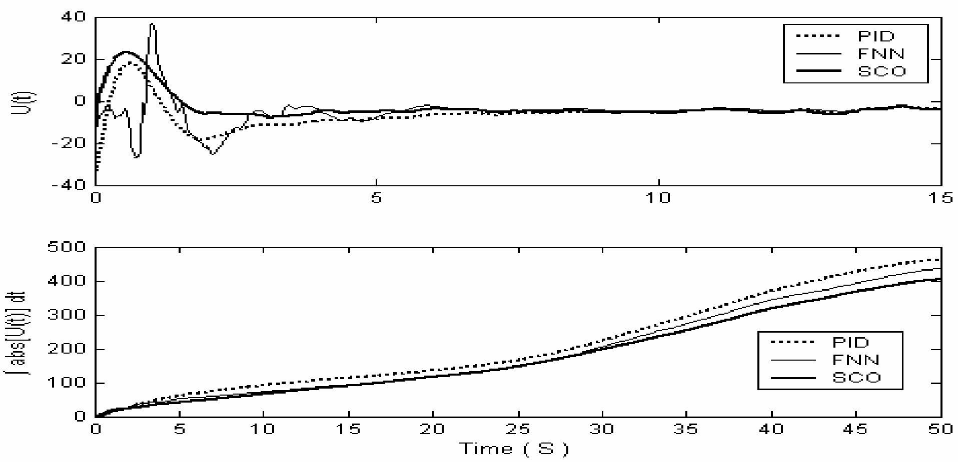


Figure 7.3-23. Example 3 oscillator. Control force. TS control situation

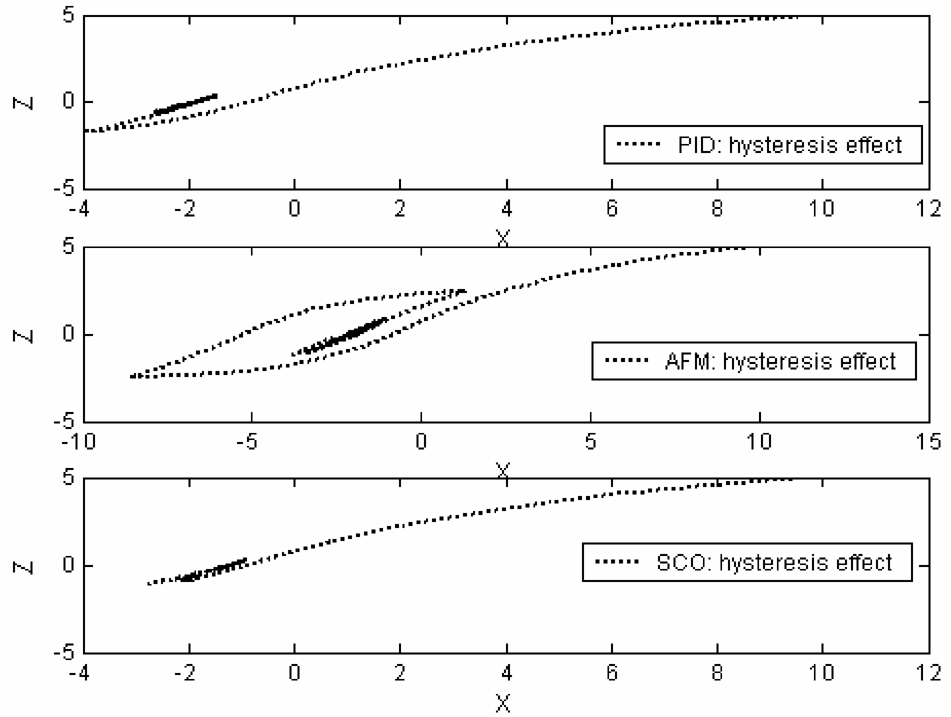


Figure 7.3-24. Example 3 oscillator. Hysteresis effect. TS control situation

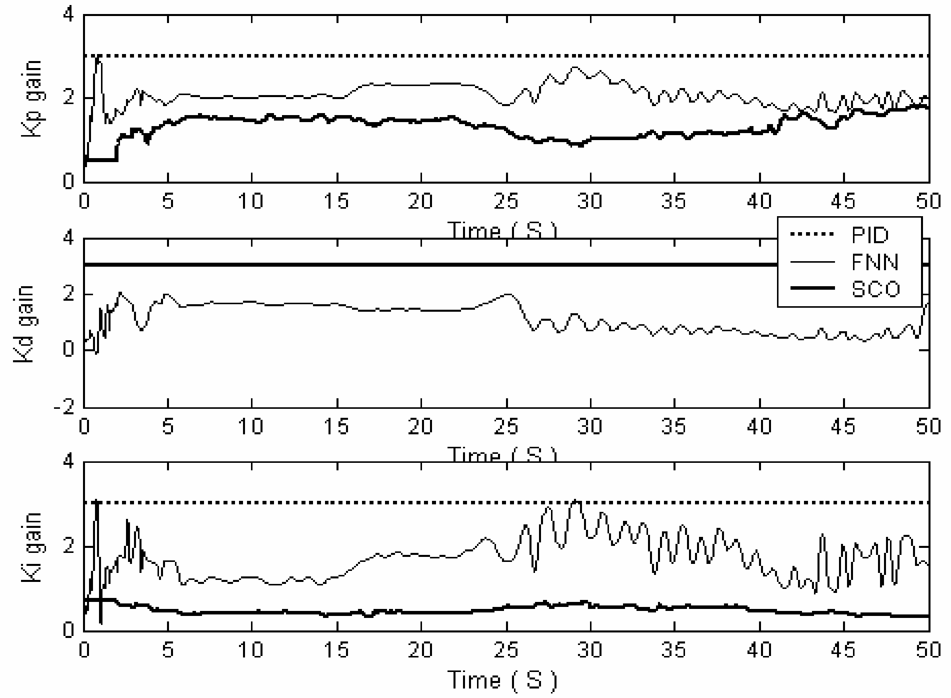


Figure 7.3-25. Example 3 oscillator. Control laws. TS control situation

Conclusion

- From control quality point of view (minimum of control error, minimum of entropy productions in a plant and in controllers, and minimum of control force) Fuzzy PID-controller designed by SC Optimizer with *52 rules* realizes more effective control in comparison to FC-FNN with *125 rules* and traditional PID-controller.

Robustness investigation of KB FC designed for the given TS control situation

Let us take FC_SCO, FC_FNN developed for the TS control situation and use them in new control situation, where new type of noise (Gaussian) are considered. Rest parameters are the same as in TS control situation.

Compare control quality of FC_SCO obtained by SC Optimizer, FC_FNN obtained by traditional SC approach based on FNN-tuning (with AFM tools) and traditional PID controller with constant gains K = (3 3 3).

In Fig.7.3-26, 7.3-27, 7.3-28, 7.3-29 and 7.3-30 results of comparison of CO motion under stochastic excitation and three types of control in the new control situation are shown.

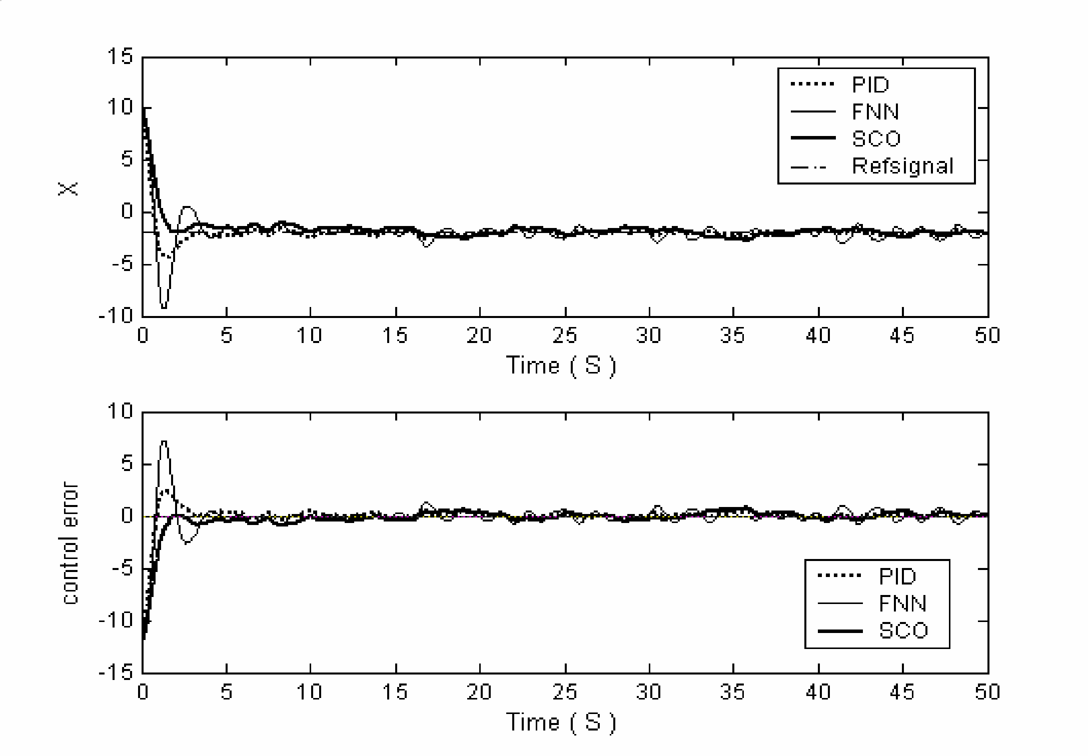


Figure 7.3-26. Example 3 oscillator. Motion and control error. New control situation

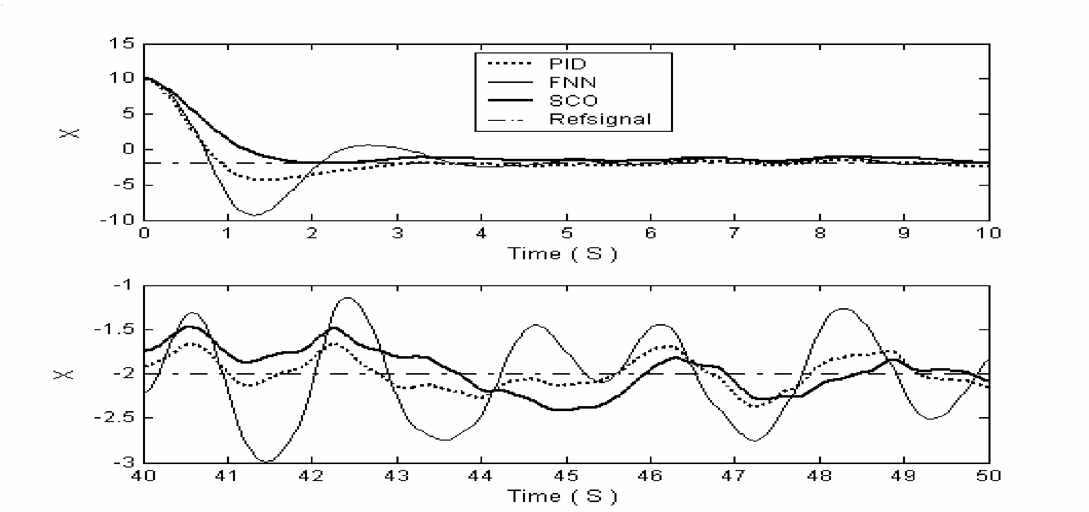


Figure 7.3-27. Example 3. Motion at fist and end intervals of time. New control situation

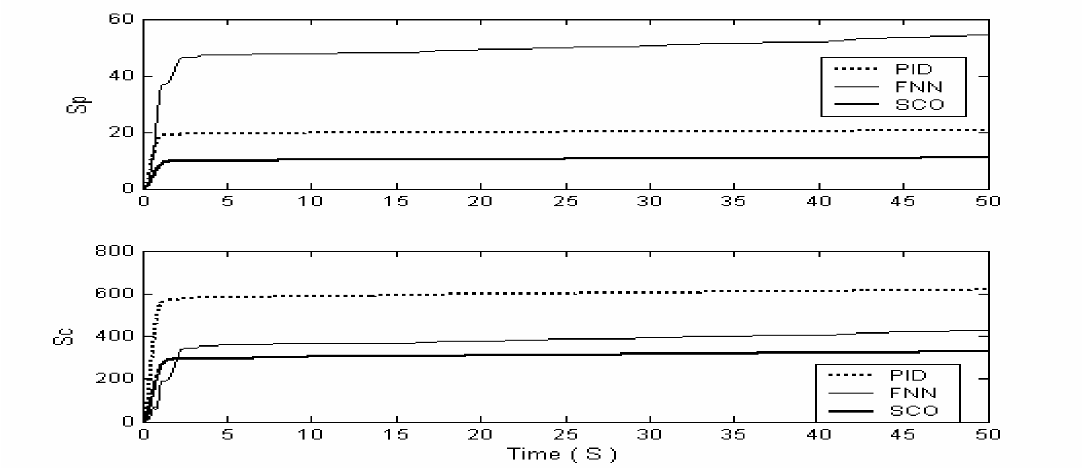


Figure 7.3-28. Example 3. Entropy production in plant and in controller. New control situation

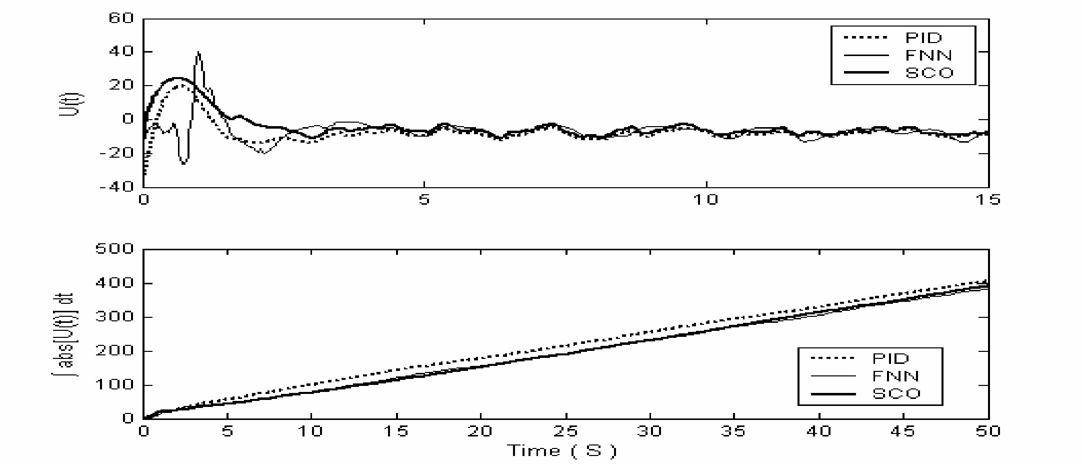


Figure 7.3-29. Example 3 oscillator. Control force. New control situation

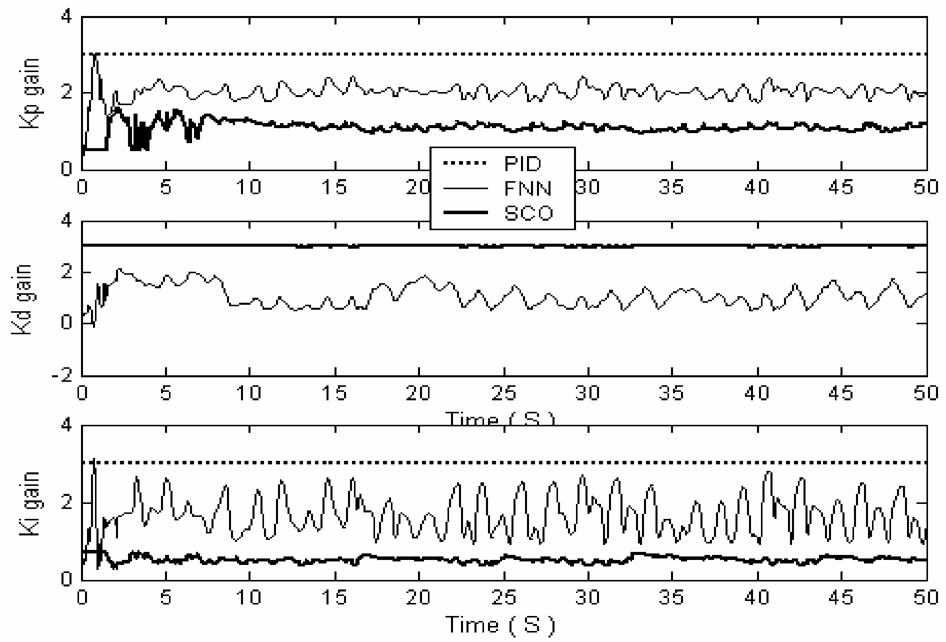


Figure 7.3-30. Example 3 oscillator. Control laws. New control situation

Conclusions

- From control quality point of view including a minimum of control error, a minimum of entropy productions in a plant and in controllers, and a minimum of control force, Fuzzy PID-controller designed by SC Optimizer with *52 rules* is more effective than FNN based controller with *125 rules* and traditional PID-controller.

- Simulation results show that KB FC designed from TS with Rayleigh noise can control new situations with Gaussian noise, i.e. designed by SC Optimizer KB FC is robust. FC_FNN has a comparatively big control error, i.e. FC_FNN is not robust.

7.4 Example 4: *Coupled non-linear oscillator motion control problem*

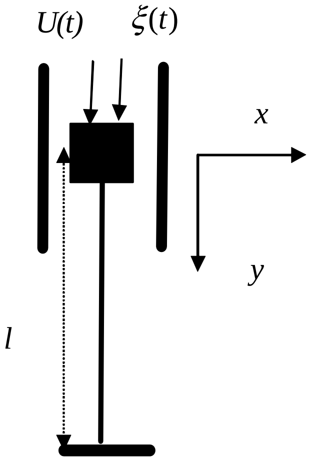
Equation of motion:

$$\begin{cases} \ddot{x} + 2\beta_1 \dot{x} + \omega_1^2 \left[1 - k \cdot y\right] x = \xi_1(t) + u_1(t) \\ \ddot{y} + 2\beta_2 \dot{y} + \omega_2^2 y + \dfrac{\pi^2}{2l}\left[x\ddot{x} + \dot{x}^2\right] = \dfrac{1}{m}\{\xi_2(t) + u(t)\} \end{cases}$$

Here $\xi_{1,2}(t)$ are the given stochastic excitations with an appropriate probability density function. $u_{1,2}(t)$ are control forces.

Equation for entropy production:

$$\frac{dS_x}{dt} = 2\beta_1 \dot{x} \cdot \dot{x}; \frac{dS_y}{dt} = 2\beta_2 \dot{y} \cdot \dot{y}$$



Consider the following model parameters and initial conditions.
Model parameters: $\beta_1 = 0.03;\ \beta_2 = 0.3;\ \omega_1 = 1.5;\ \omega_2 = 4;\ k = 5;\ l = 1;\ m = 5$

Initial conditions: $[x_0\ y_0][\dot{x}_0\ \dot{y}_0]$ = [0.5 0.1] [0.01 0.01].

In Fig. 7.4-1, 7.4-2 and 7.4-3 a dynamic and thermodynamic behavior of free CO motion with the given above parameters are shown.

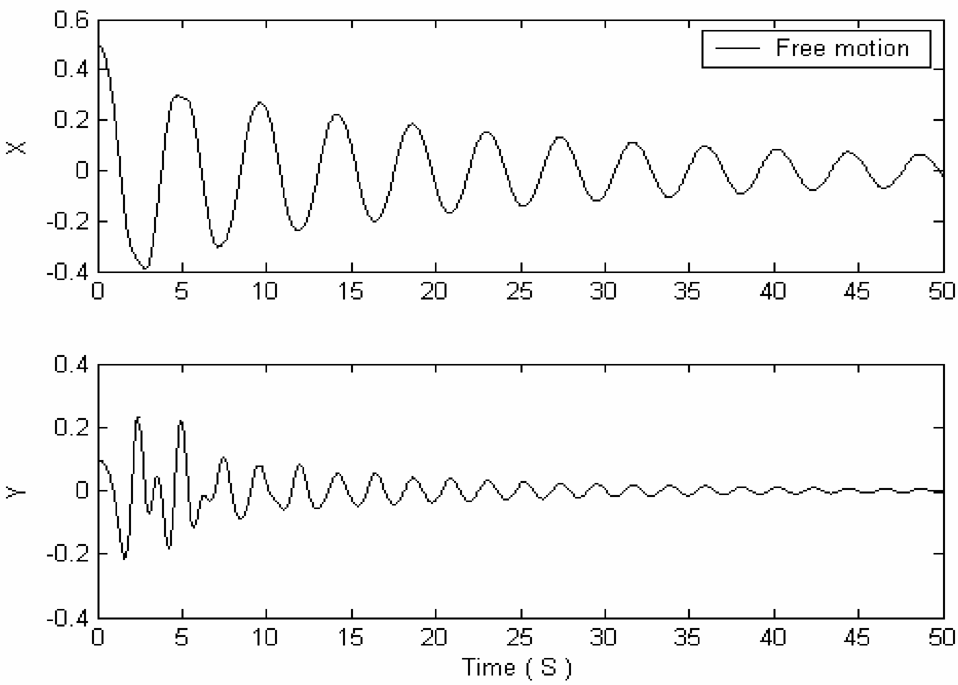


Figure 7.4-1. Coupled non-linear oscillator. Free motion

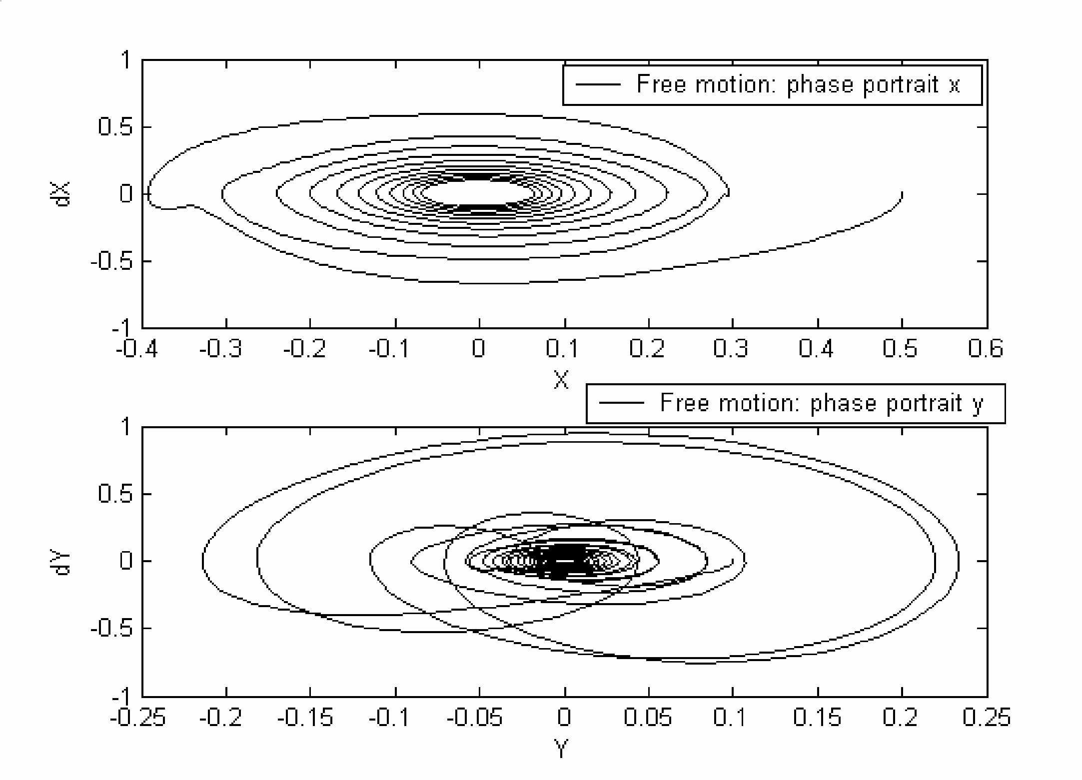


Figure 7.4-2. Coupled non-linear oscillator. Free motion. Phase portraits

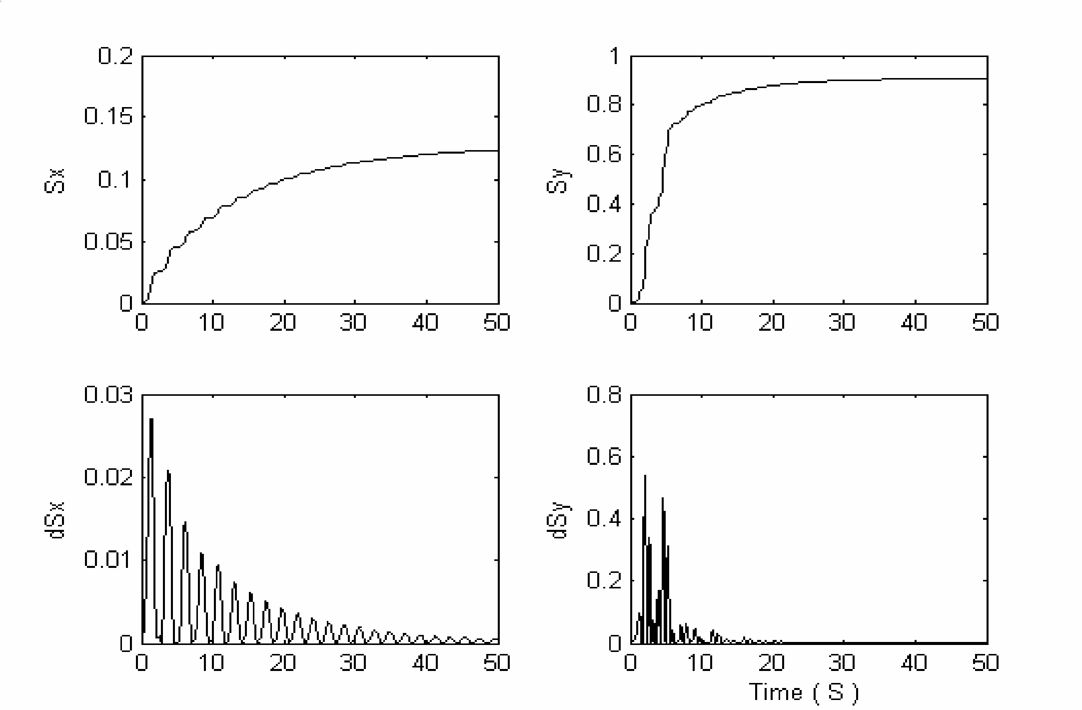


Figure 7.4-3. Coupled non-linear oscillator. Free motion. Thermodynamic behavior

Simulation results show that the given CO represents a *stable dynamic system*.

Consider behaviour of this control object under two different types of stochastic excitations: Gaussian and Rayleigh noises shown in Fig. 7.4.-4.

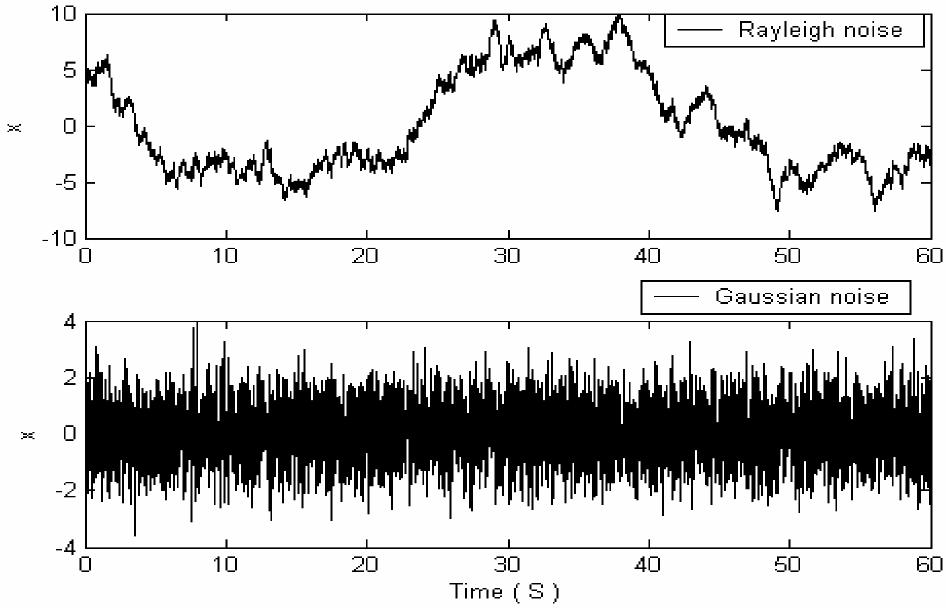


Figure 7.4-4. Stochastic noises

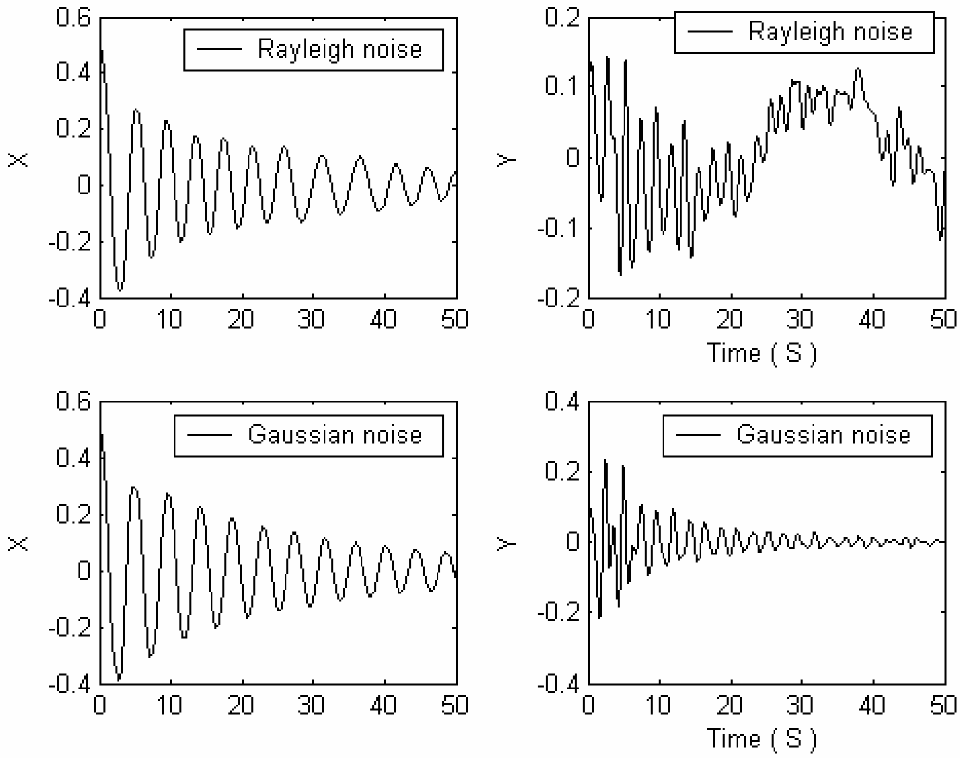


Figure 7.4-5. Coupled non-linear oscillator. Stochastic motion

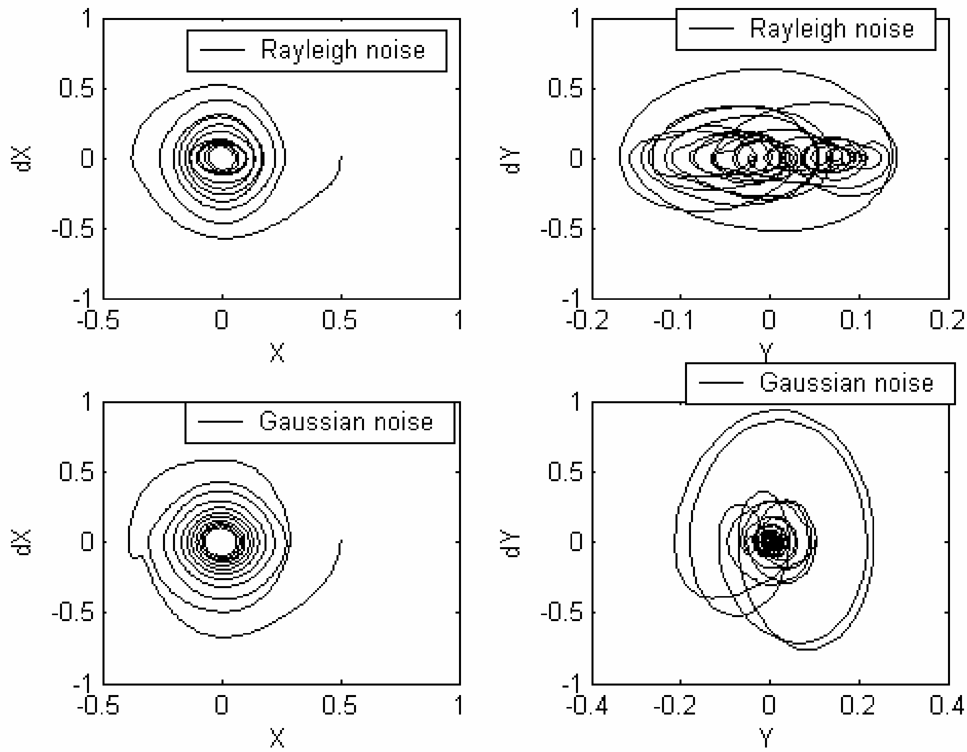


Figure 7.4-6. Coupled non-linear oscillator. Stochastic motion. Phase portraits

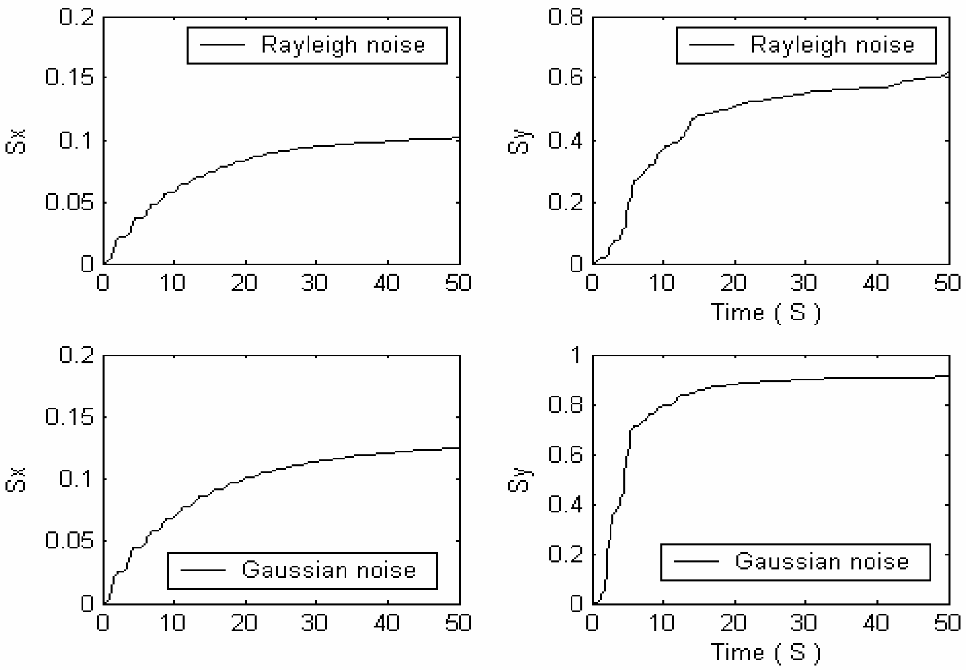


Figure 7.4-7. Coupled non-linear oscillator. Stochastic motion.
Thermodynamic behavior

In Fig. 7.4-5, 7.4-6 and 7.4-7 CO stochastic motion (dynamic and thermodynamic behavior) is shown. Dynamic behavior of control object under Rayleigh excitation is more complicated.
Consider the following *control task* for this example: in the presence of Rayleigh noise (with maximum amplitude A= 10) along *y*-axis and in the presence of Gaussian noise (with maximum amplitude A= 4) along *x*-axis stabilize motion of CO at the given reference signals $x_{ref} = 0; y_{ref} = 0$.

Consider two PD-controllers (with K-gains $K = \{k_p, k_d\} = \{20\ 20\}$) for control a stochastic motion of the control object under complicated noises mentioned above.

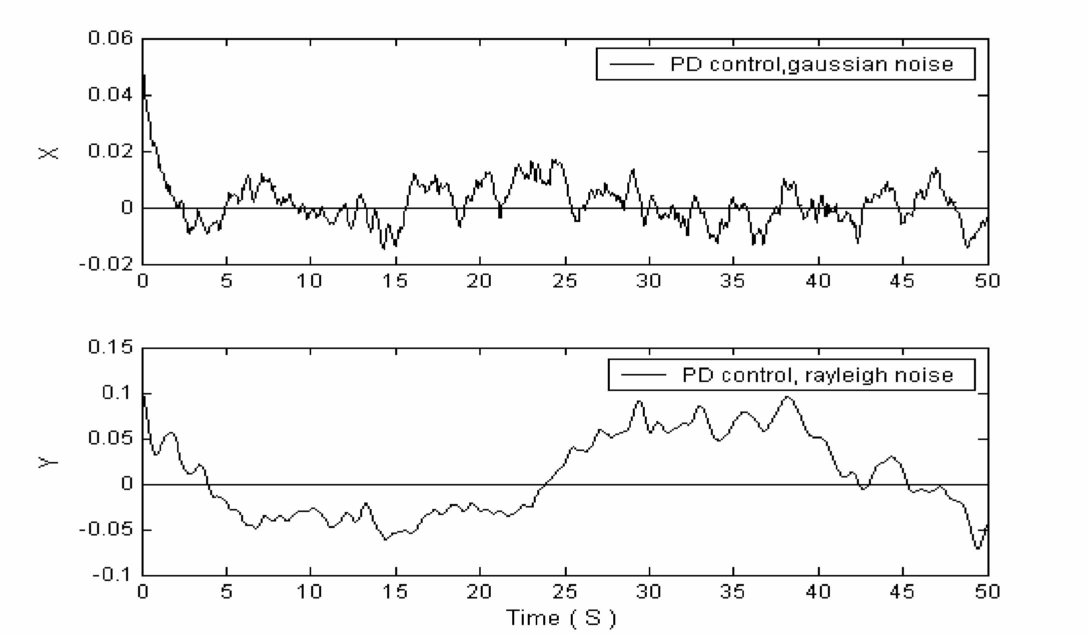


Figure 7.4-8. Coupled non-linear oscillator. Stochastic motion. Classical PD control with constant k-gains

The simulation results in Fig. 7.4-8 show CO motion under two traditional PD control with constant gain coefficients.
Let us design intelligent control system for the given above control problem by using our KB FC design tools and compare results with traditional PD Controllers.

Consider the process of KB FC design for the following TS control situation:

- Motion under Gaussian (qwert.mat) excitation along *x* axis;
- Motion under Rayleigh (qwert1.mat) excitation along *y* axis;
- Model Parameters: $\beta_1 = 0.03; \beta_2 = 0.3; \omega_1 = 1.5; \omega_2 = 4; k = 5; l = 1; m = 5$;
- Initial conditions: [0.5 0.1] [0.01 0.01];
- Reference signals: $x_{ref} = 0, y_{ref} = 0$

GA-based TS design

At this step we design TS of optimal control based on the given control quality criterion as a GA fitness function. For this case we will use the following GA parameters:

- GA parameters**:** search space for K-gains parameters: (0,20);
- GA FF : minimum of “ control error and control object entropy production rate”.

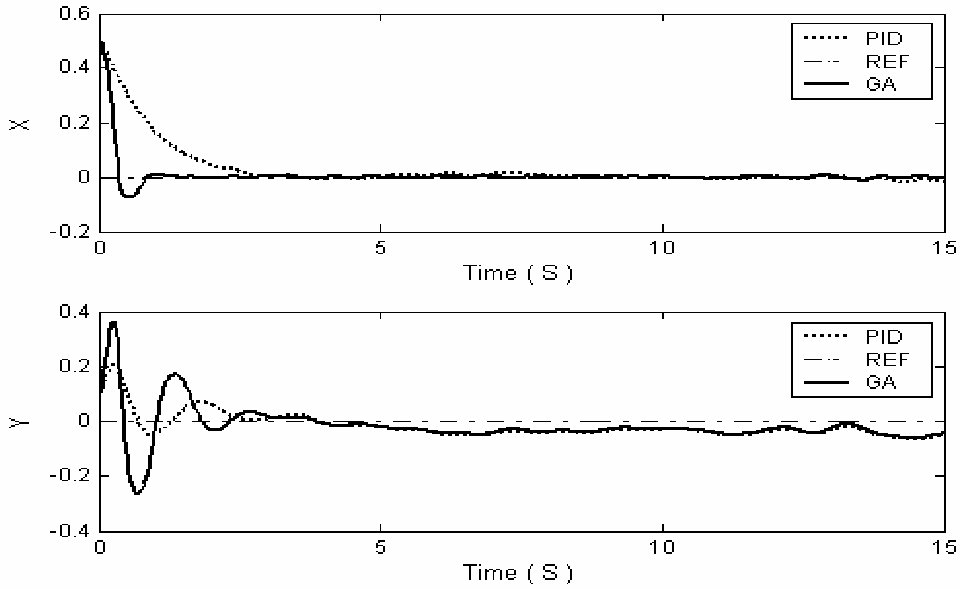


Figure 7.4-9. Coupled non-linear oscillator. Stochastic motion under 2 GA- PD control with variable k-gains

In Fig. 7.4-9, simulation results with GA-PID control are shown.

FNN-based KB FC design process (step 1 technology)

We will design one FC controller for two PD controllers along x and y axes. Thus we will have 4 input variables to FC: $\{e_x, \dot{e}_x, e_y, \dot{e}_y\}$ and 4 output variables of FC: $\{(k_p, k_d)_x, (k_p, k_d)_y\}$. AFM based KB design process is described as follows:

- Manual design of numbers and shapes of membership functions for each input variables: 5;
- Complete number of fuzzy rules: 5x5x5x5= 625 rules;
- Number of activated rules in KB**: 625 rules.**

In Fig. 7.4-10 membership functions representation in AFM is shown.

SC Optimizer-based KB FC design process (step 2 technology)

SC Optimizer based KB design process is described as follows:

- *Creation of linguistic values* by GA1 : number of membership functions for each input variables: 5,8,9,9;
- Complete number of fuzzy rules: 5x8x9x9 = 3240 rules;
- *Rules selection by*: MAX criterion with manual threshold value = 0.5;
- *KB optimization* by GA2: optimized KB consists of **34 rules.**

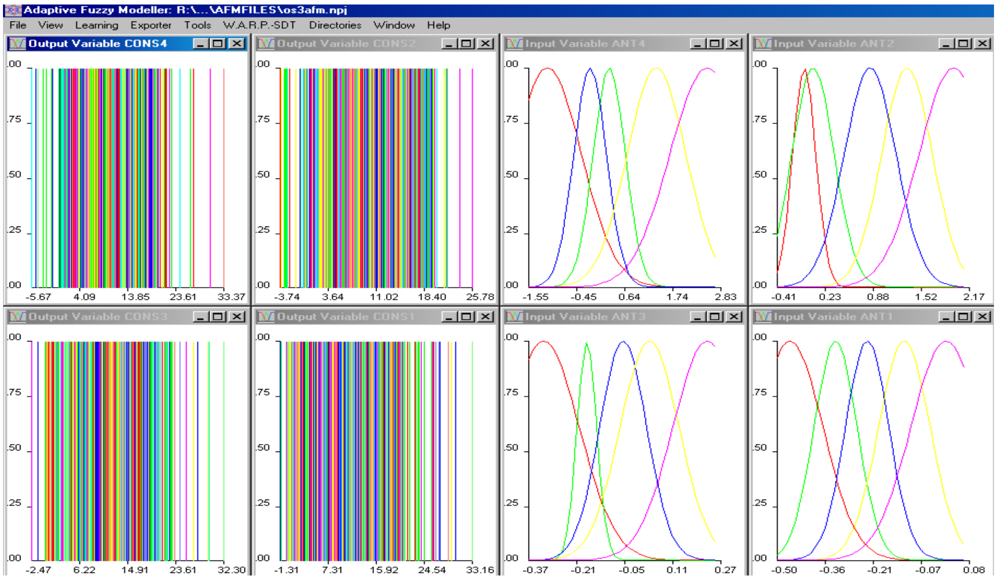


Figure 7.4-10. AFM based representation of membership functions

In Fig. 7.4-11 SC Optimizer-based membership functions representation is shown.

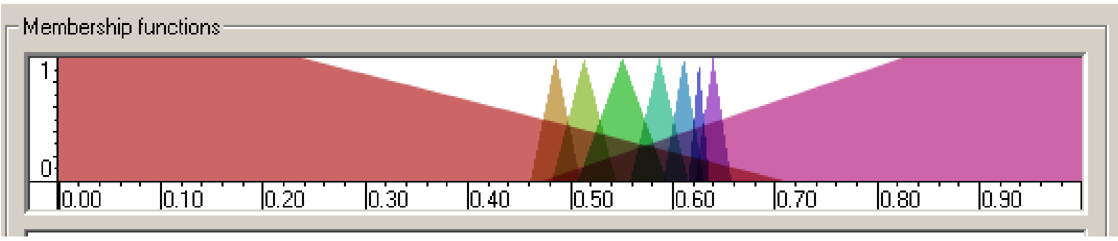


Figure 7.4-11. Coupled non-linear oscillator. Example of membership function representation in SC Optimizer (for third FC input)

Remark. In AFM number and MF shape are selected manually by a user. In SC optimizer for each input variables optimal MF shapes and MF numbers are defined by GA.

Compare control quality of FC_SCO obtained by SC Optimizer, FC_FNN obtained by traditional SC approach based on FNN-tuning (with AFM tools) and traditional PD controllers with constant gains K = (20 20).

In Figures 7.4-12, 7.4-13, 7.4-14, 7.4-15 and 7.4-16 results of comparison of CO motion under stochastic excitation and three types of control are shown.

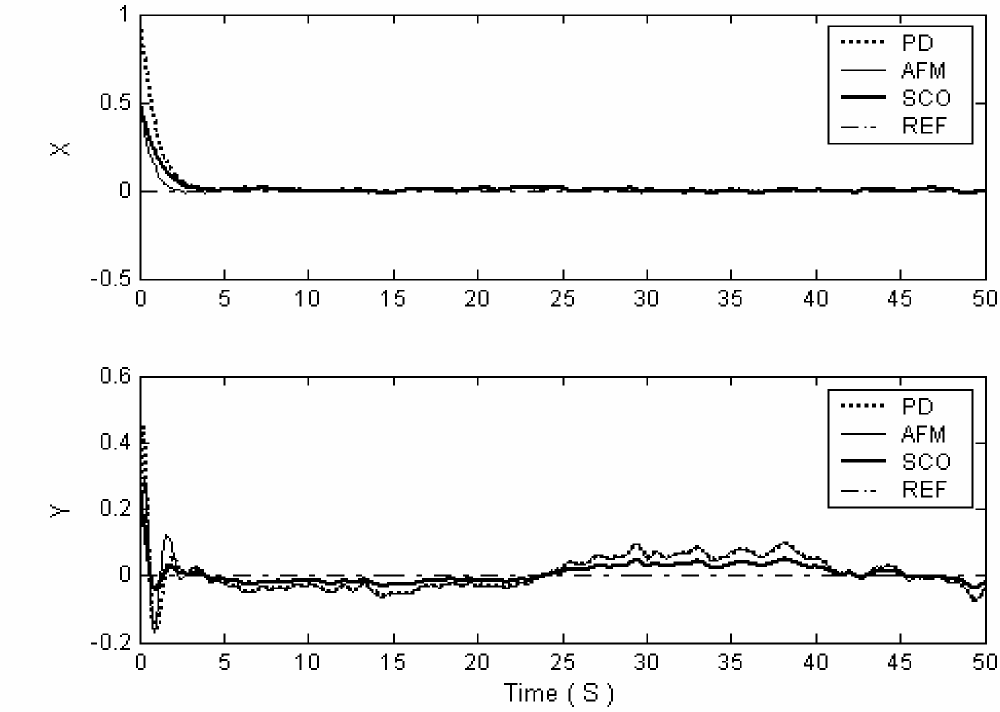


Figure 7.4-12. Coupled non-linear oscillator motion. TS control situation

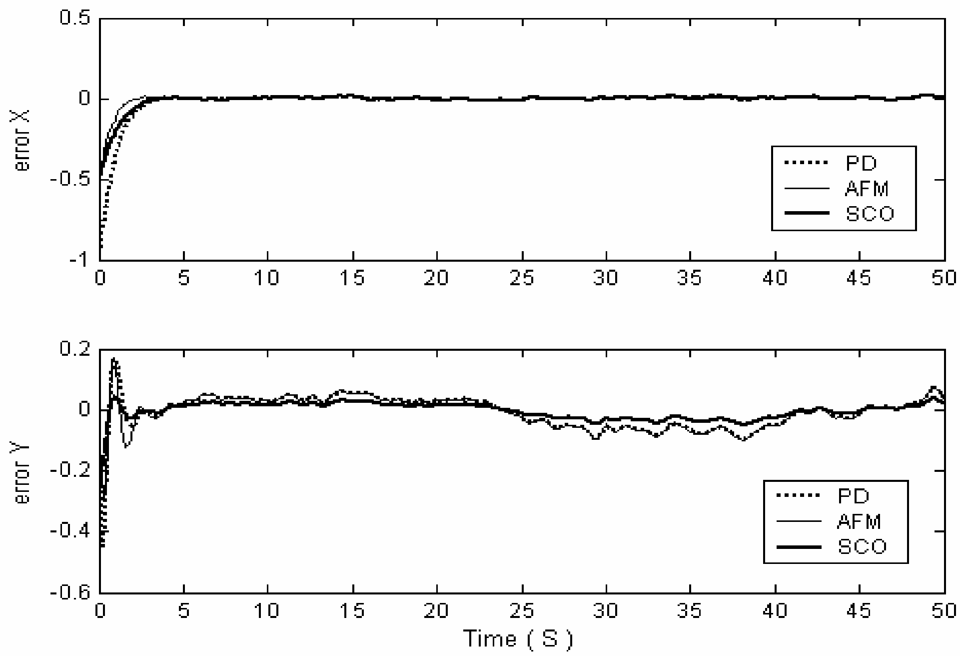


Figure 7.4-13. Coupled non-linear oscillator. Control error. TS control situation

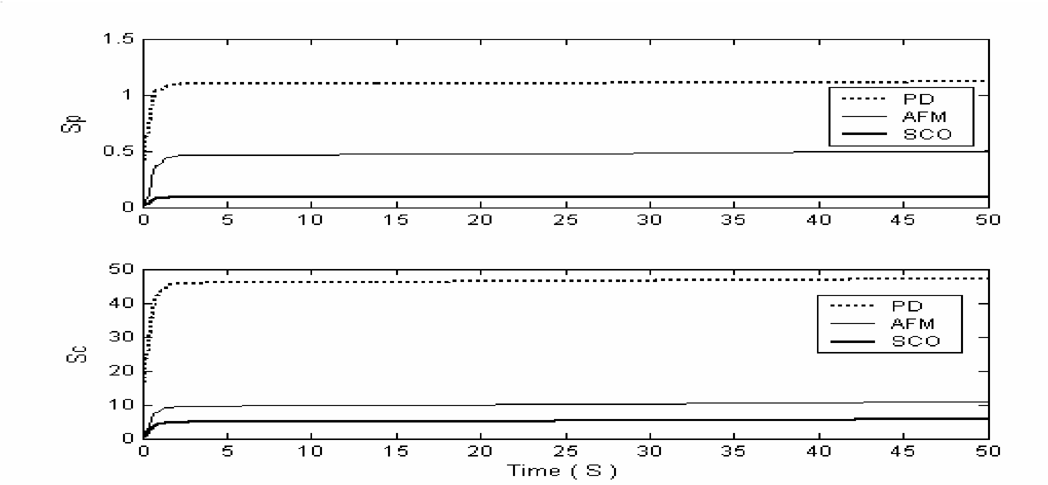


Figure 7.4-14. Example 4. Entropy production in plant and in controller. TS control situation

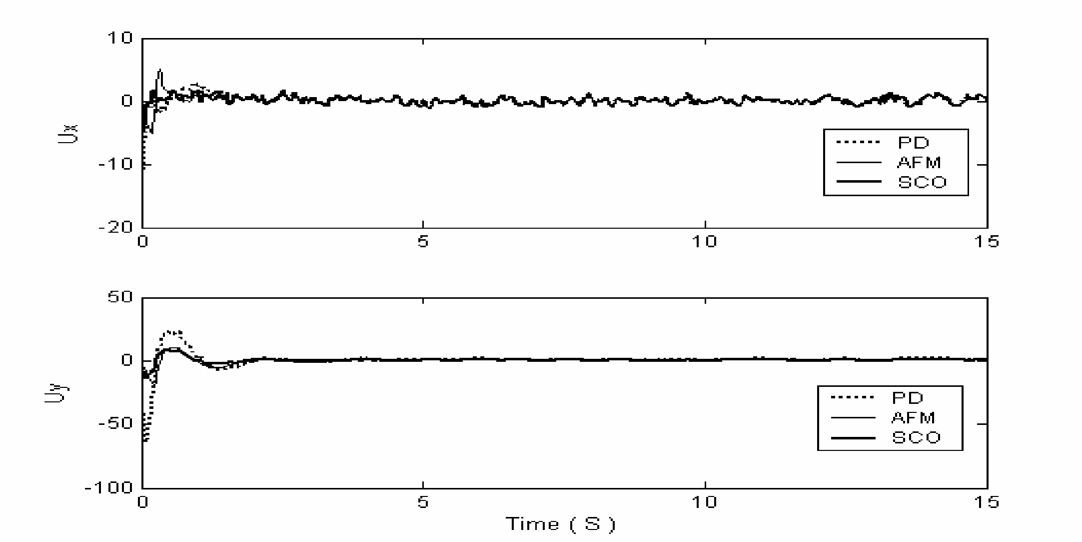


Figure 7.4-15. Coupled non-linear oscillator. Control force. TS control situation

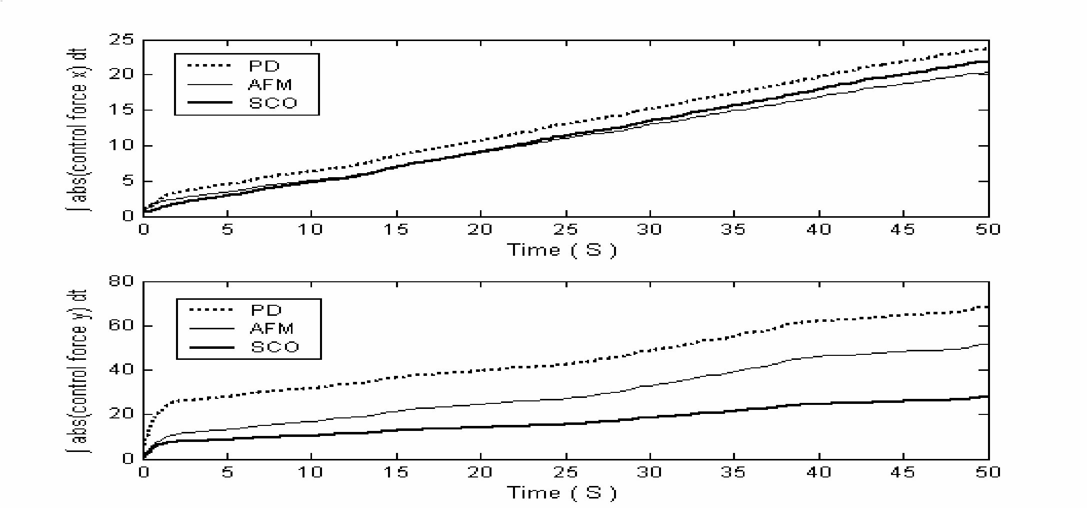


Figure 7.4-16. Example 4. Control force integral value. TS control situation

Control laws comparison is given in Fig. 7.4-17 below.

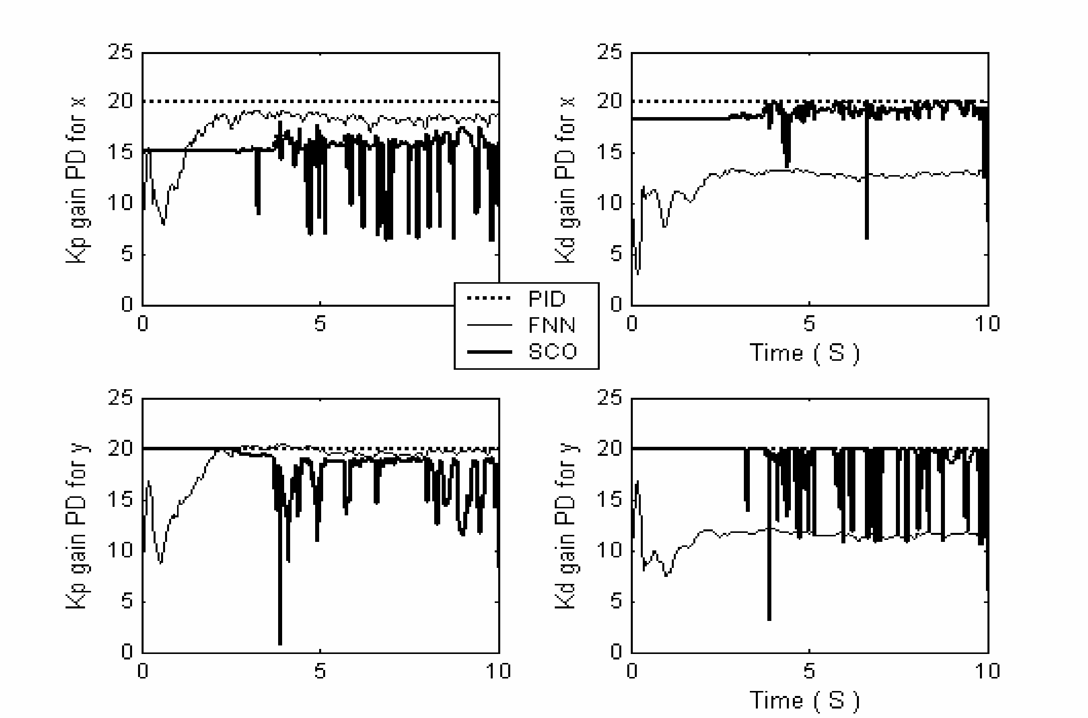


Figure 7.4-17. Coupled non-linear oscillator. Control laws. TS control situation

Conclusion

- From control quality point of view (minimum of control error, minimum of entropy productions in a plant and in controllers, and minimum of control force) Fuzzy PD-controller designed by SC Optimizer with *34 rules* gives more effective control in comparison to FC-FNN with *625 rules* and two traditional PD-controllers.

Robustness investigation of KB FC designed for the given TS control situation

Let us take FC_SCO, FC_FNN developed for the TS control situation and use them in new control situations.

New control situation:

- new model parameters:
  $\beta_1 = 0.5(old: 0.03);\ \beta_2 = 0.05\ (old: 0.3);$
  $\omega_1 = 5\ (old: 1.5);\ \omega_2 = 1\ (old: 4);\ k = 3\ (old: 5);\ l = 1;\ M = 5$
- new initial conditions [1 1] [0.01 0.01];
- new references: $x_{ref} = 0.1 \quad y_{ref} = 0.05$;
- noise along x axis = 0;
- noise along y axis: Rayleigh noise with max amplitude A=15.

Compare control quality of FC_SCO obtained by SC Optimizer, FC_FNN obtained by traditional SC approach based on FNN-tuning (with AFM tools) and control quality obtained by traditional PD controller with constant gains K = (20 20).

In Figures 7.4-18, 7.4-19, and 7.4-20, results of comparison of CO motion under stochastic excitation and three types of control in the new control situation are shown.

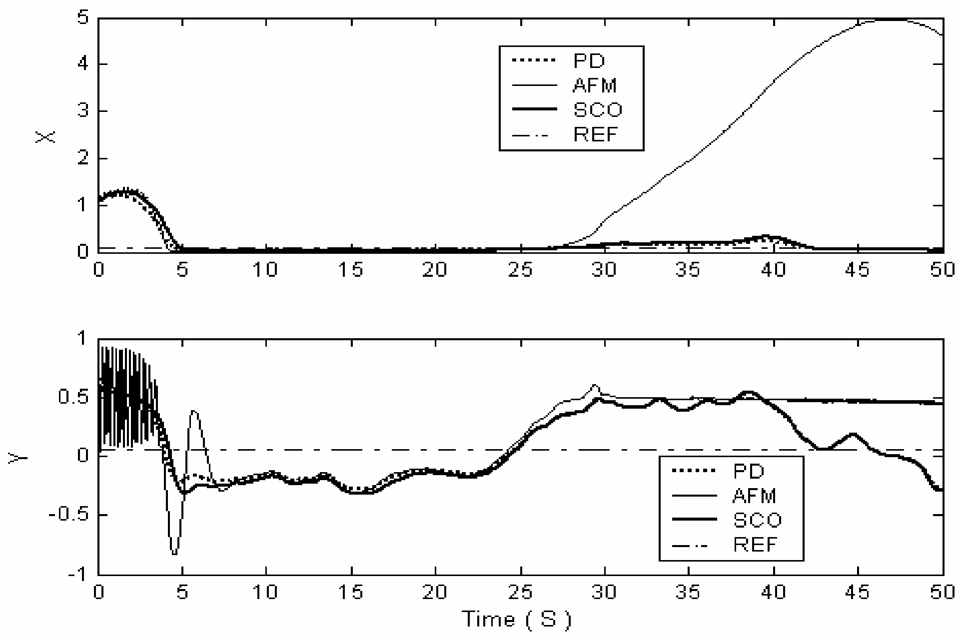


Figure 7.4-17. Coupled non-linear oscillator motion. New control situation.

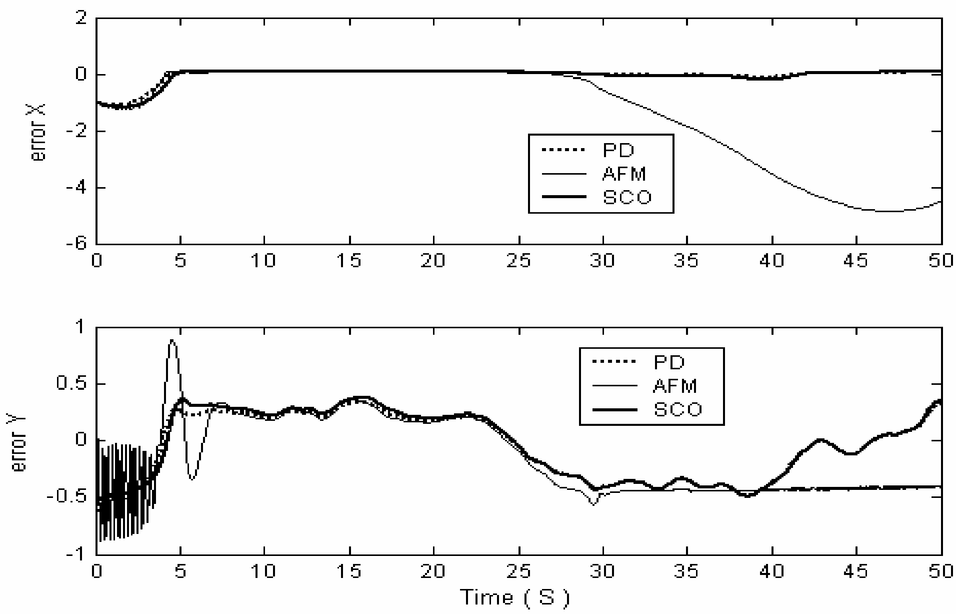


Figure 7.4-18. Coupled non-linear oscillator. Control error. New control situation

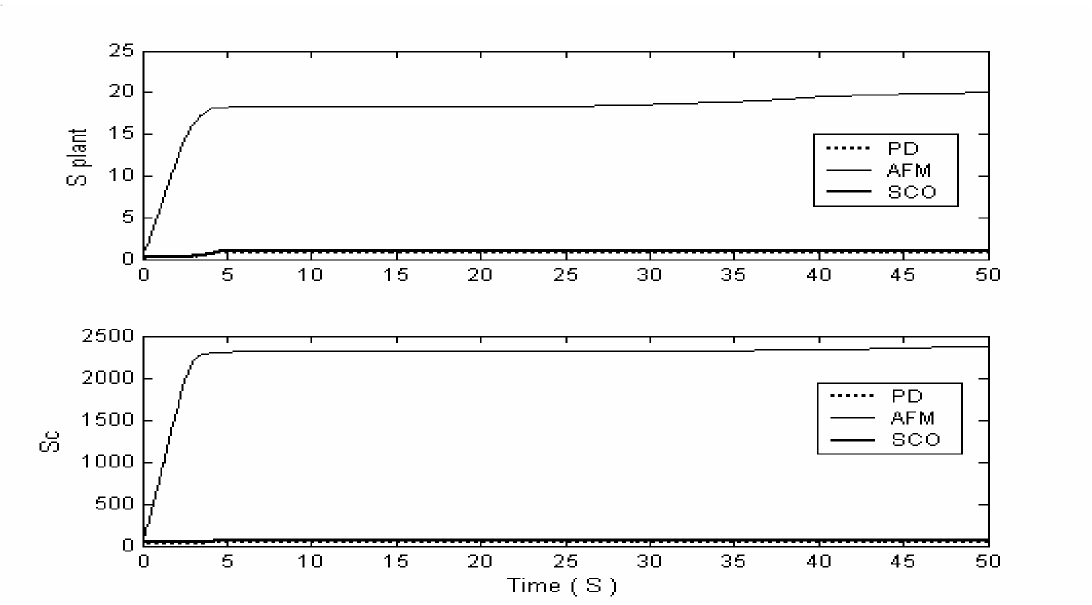


Figure 7.4-19. Coupled non-linear oscillator. Entropy production in a plant and in Controllers. New control situation

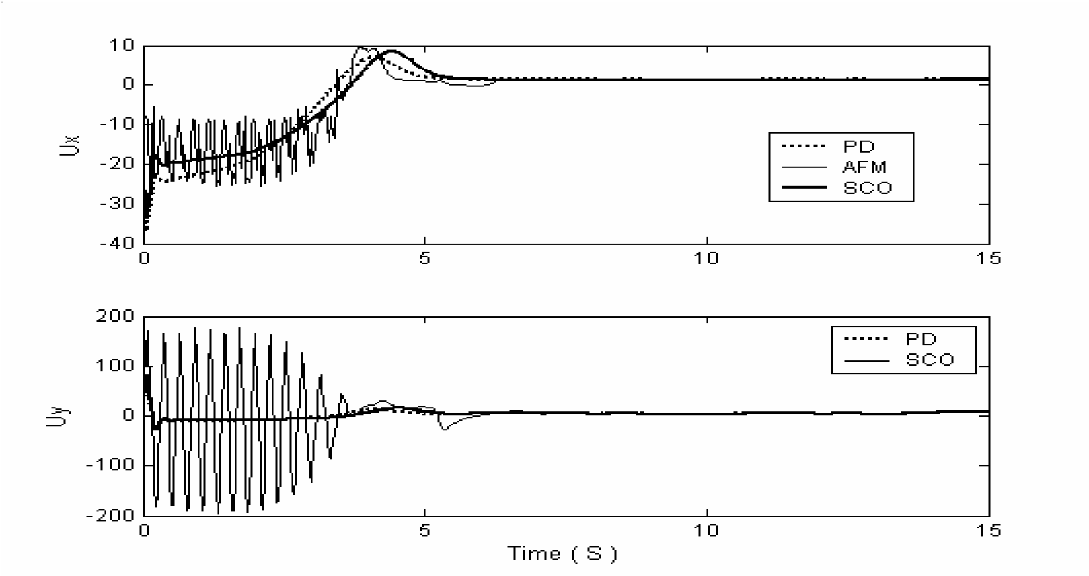


Figure 7.4-20. Coupled non-linear oscillator. Control force. New control situation

Conclusion

- SCO control is robust, but FNN control is failed (unstable), i.e. it is not robust when new initial conditions, reference signals, model parameters and external disturbances are changed.

7.5 Example 5: *Nonlinear oscillator with sizable nonlinear dissipative components*
Equations of motion and entropy production rate are:

$$\ddot{x}+\left[2\beta+a\dot{x}^2+k_1x^2-1\right]\dot{x}+kx=\xi(t)+u(t);$$

$$\frac{dS_x}{dt}=\left[2\beta+a\dot{x}^2+k_1x^2-1\right]\dot{x}\cdot\dot{x},$$

where $\xi(t)$ is a given stochastic excitations with an appropriate probability density function, and $u(t)$ is a control force.

Model parameters: $\beta=0.5;\alpha=0.3;k_1=0.2;k=5.$ Initial conditions: $[x_0][\dot{x}_0]$ = [2.5] [0.1].

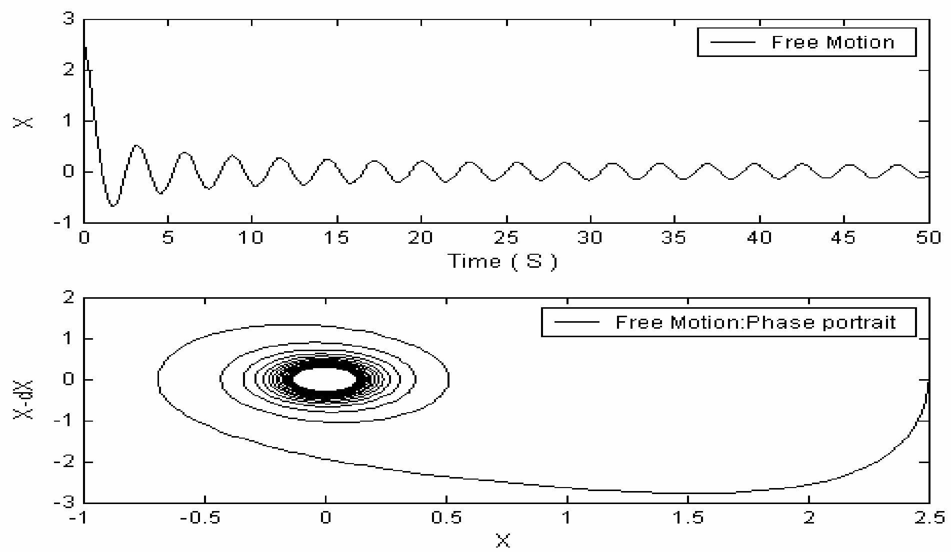


Figure 7.5-1. Example 5 oscillator. Free motion

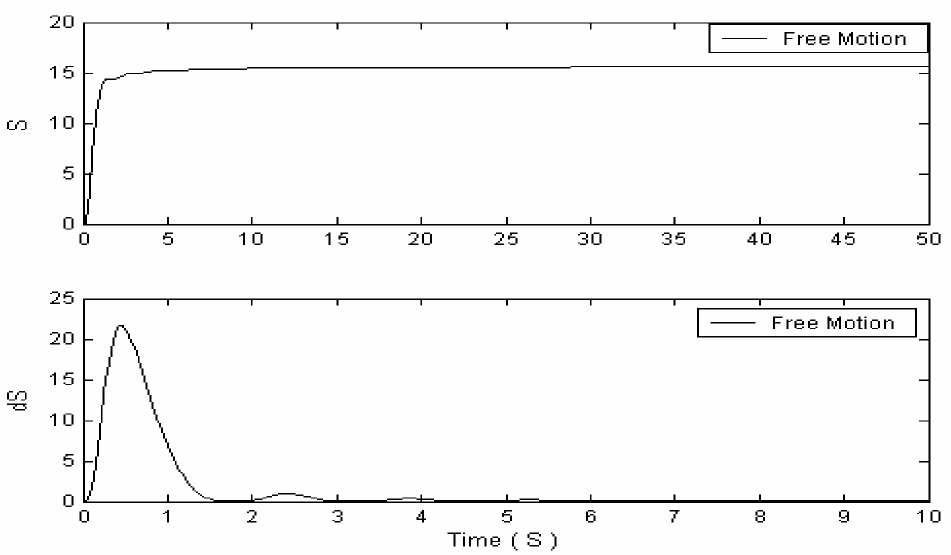


Figure 7.5-2: Example 5 oscillator. Free motion. Thermodynamic behavior

In Fig. 7.5-1 and 7.5-2 free motion (dynamic and thermodynamic behavior) of control object with the given above parameters are shown.
Simulation results show that the given CO is a *stable dynamic system*.

Consider behaviour of this control object under two different types of stochastic excitations (Gaussian and Rayleigh noises) shown in Fig.7.5 -3.

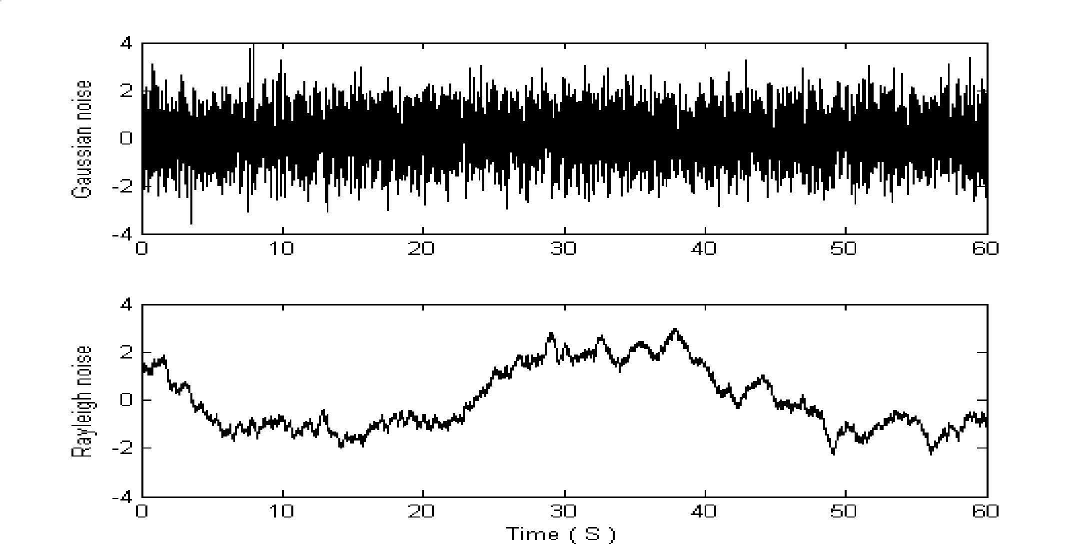


Figure 7.5-3. Stochastic noises: Gaussian (top) and Rayleigh (below)

In Figures 7.5-4, 7.5-5 and 7.5-6 dynamic and thermodynamic motion of CO under stochastic noises is shown.

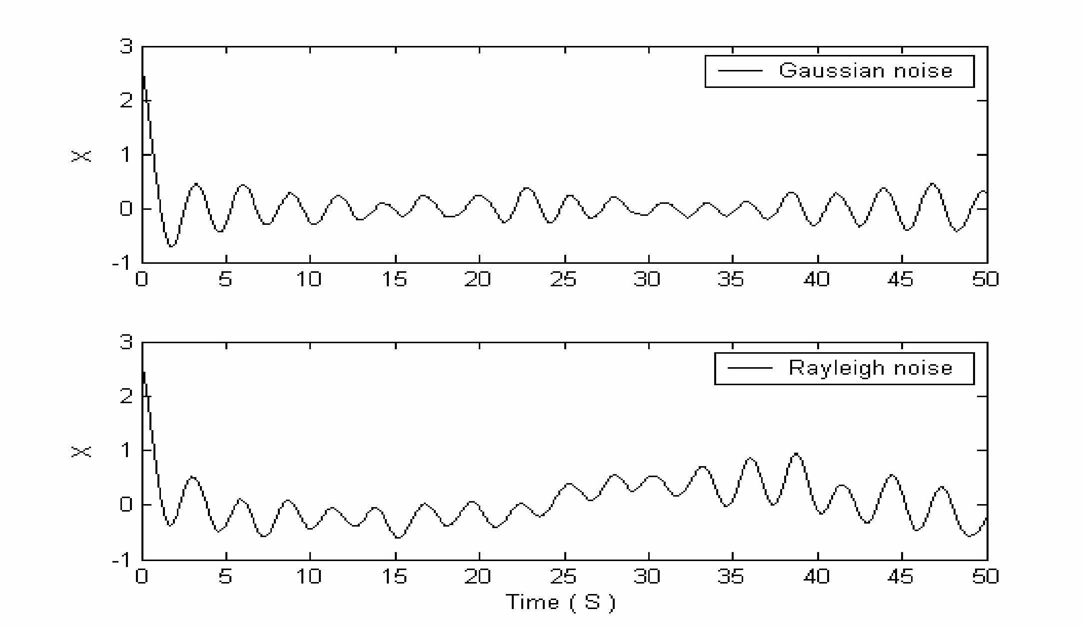


Figure 7.5-4. Example 5 oscillator. Stochastic motion

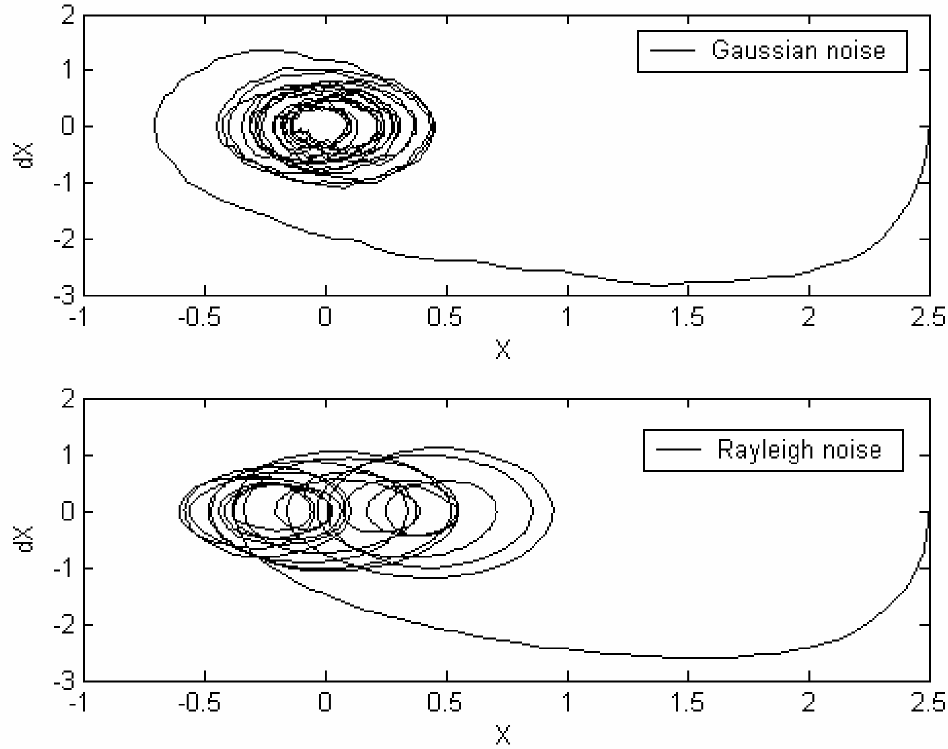


Figure 7.5-5. Example 5 oscillator. Stochastic motion. Phase portraits

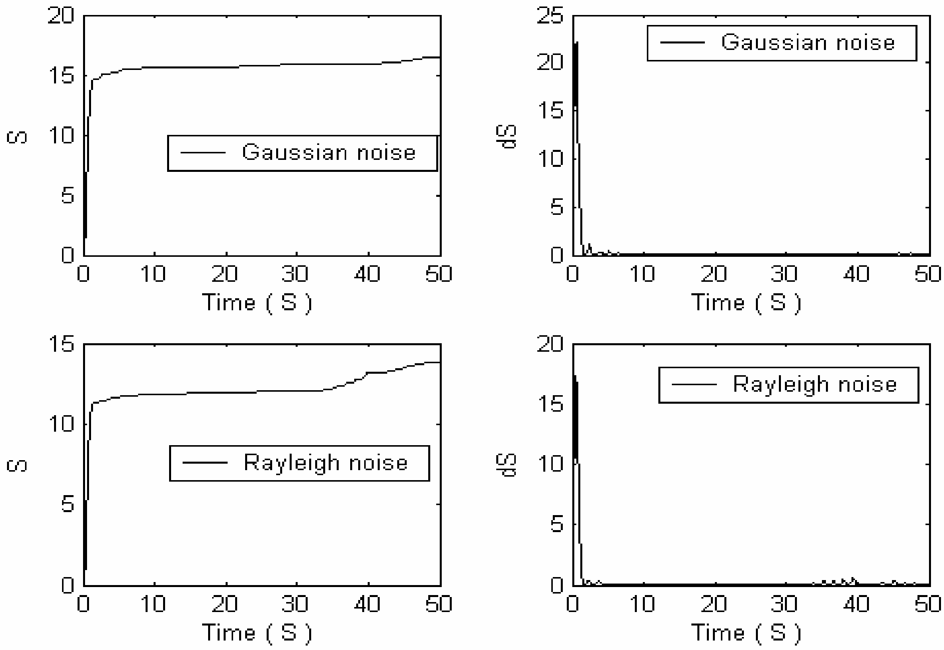


Figure 7.5-6. Example 5 oscillator. Stochastic motion. Thermodynamic behavior

Simulation results show that CO dynamic behavior under Rayleigh excitation is more complicated.
Consider the following *control task* for this example: in the presence of Rayleigh noise maintain motion of CO at the given reference signal $x_{ref} = 0$.
Let us design intelligent control system for the given above control problems by using our KB FC design tools and compare results with traditional PID Controller.

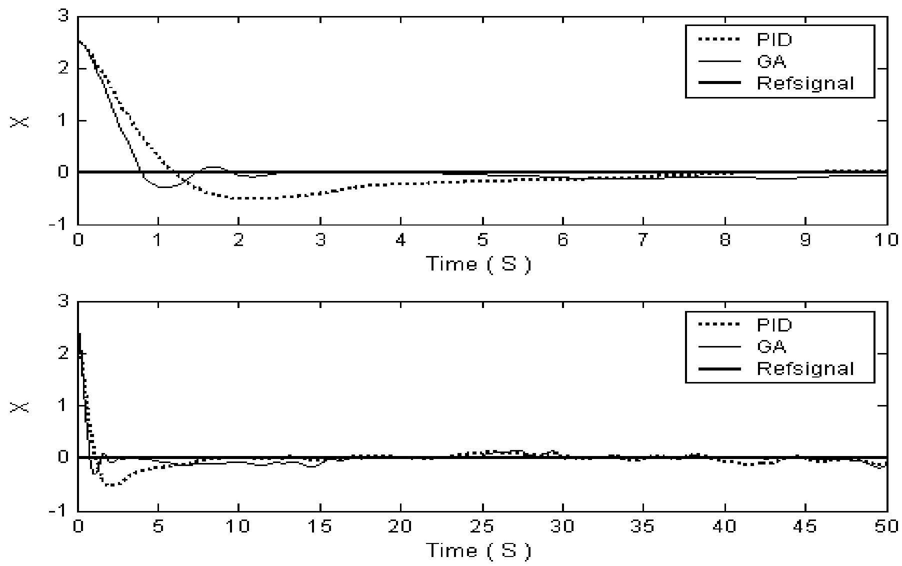


Figure 7.5-7. Example 5 oscillator. GA- PID control

GA-based TS design

At this step we design TS of optimal control based on the given control quality criterion as a GA fitness function. For this case we will use the following GA parameters:

- GA parameters: search space for K-gains parameters: (0,5);
- GA FF : minimum of " control error".

We have the following TS control situation:

- Model Parameters: $\beta = 0.5; \alpha = 0.3; k_1 = 0.2; k = 5.$
- Initial conditions: [2.5] [0.1]; Reference signals: $x_{ref} = 0$
- Rayleigh noise (max amplitude = 3);

In Fig.7.5-7 the comparison of motion under GA and PID control is shown. You can see limitations of classical PID controller.

FNN-based KB FC design process (step 1 technology)

For the given control tasks we will design FC-PID controller with three input variables to FC as $\left\{e, \dot{e}, \int edt\right\}$ and three output variables of FC as $\left\{k_p, k_d, k_i\right\}$.
AFM based KB design process is described as follows:

- Manual design of numbers of membership functions for each input variables: 5;
- Complete number of fuzzy rules: 5x5x5= 125 rules;
- Number of activated rules in KB**: 125 rules.**

In Fig. 7.5-8 AFM representation of membership functions for input FC variables is shown.

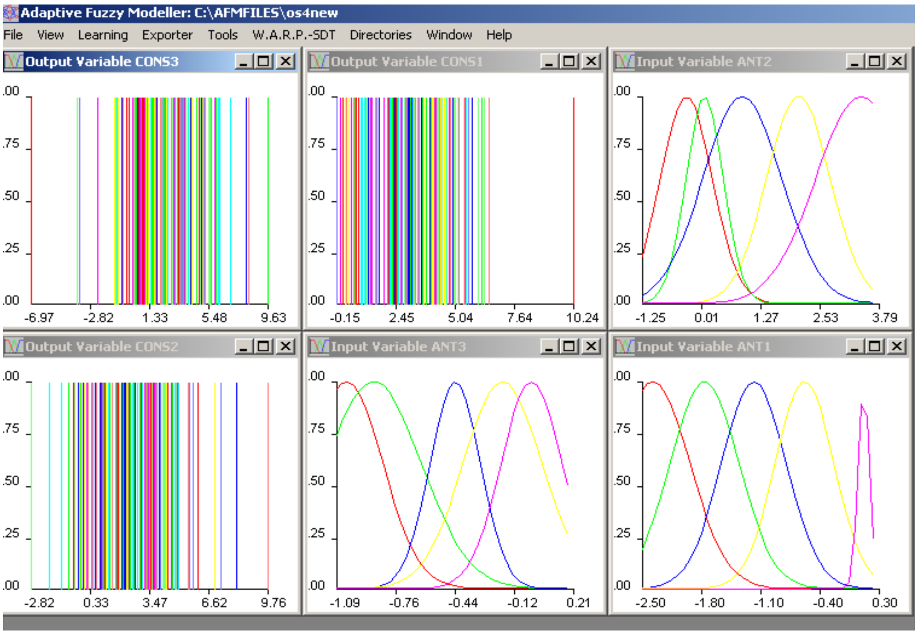


Figure 7.5-8. Example 5. AFM based membership functions representation

SC Optimizer-based KB FC design process (step 2 technology)

The process of KB FC design is described as follows:

- *Creation of linguistic values* by GA1 : number of membership functions for each input variables: 7,9,9 ;
- Complete number of fuzzy rules: 7x9x9 = 567 rules;
- *Rule selection* : SUM of firing strength criterion with limited number of rules = 20;
- *KB optimization by* GA2: optimized KB contains **20 rules.**

In Fig. 7.5-9 SC Optimizer representation of membership functions and their shapes for third input FC variables is shown.

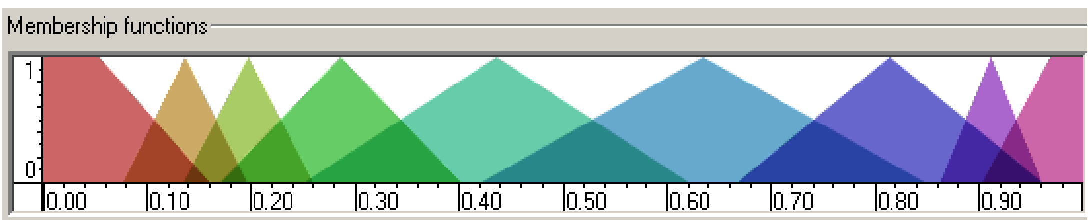


Figure 7.5-9. Example 5. SC Optimizer based membership functions representation

Remark. In AFM based representation number and MF shapes are selected manually by a user. In SC optimizer for each input variables optimal MF shapes and MF number are defined by GA.

Compare control quality of FC_SCO obtained by SC Optimizer with 20 rules, FC_FNN obtained by traditional SC approach based on FNN-tuning with 125 rules and traditional PID controller with constant gains K = (5 5 5).

In Figures 7.5-10, 7.5-11, 7.5-12, 7.5-13 and 7.5-14 results of comparison of CO stochastic motion under three types of control are shown. Control laws for TS control situation are shown in Fig. 7.5-15.

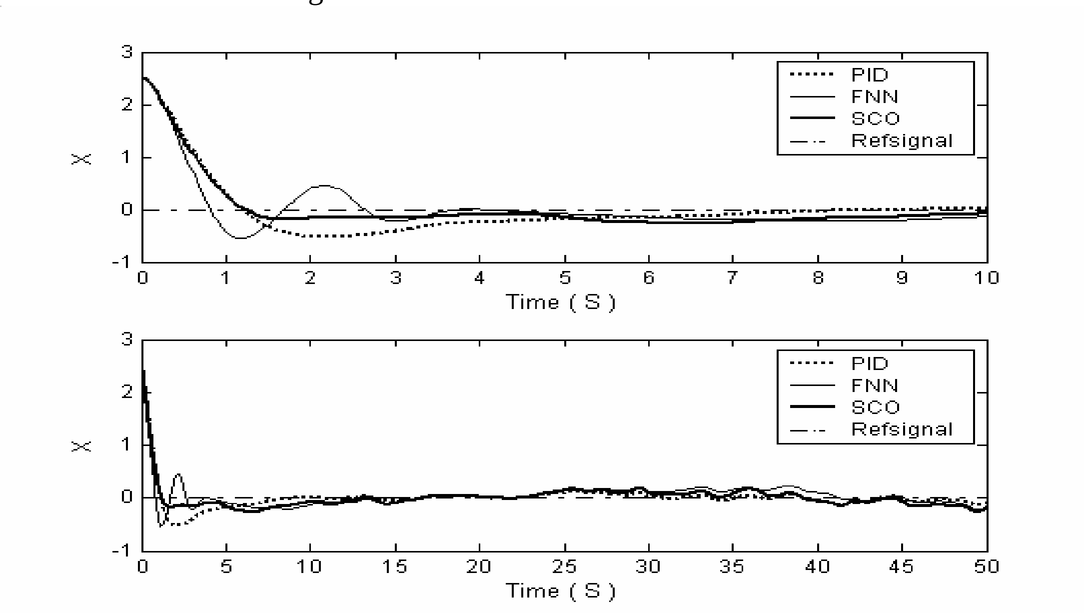


Figure 7.5-10. Example 5 oscillator motion. TS control situation

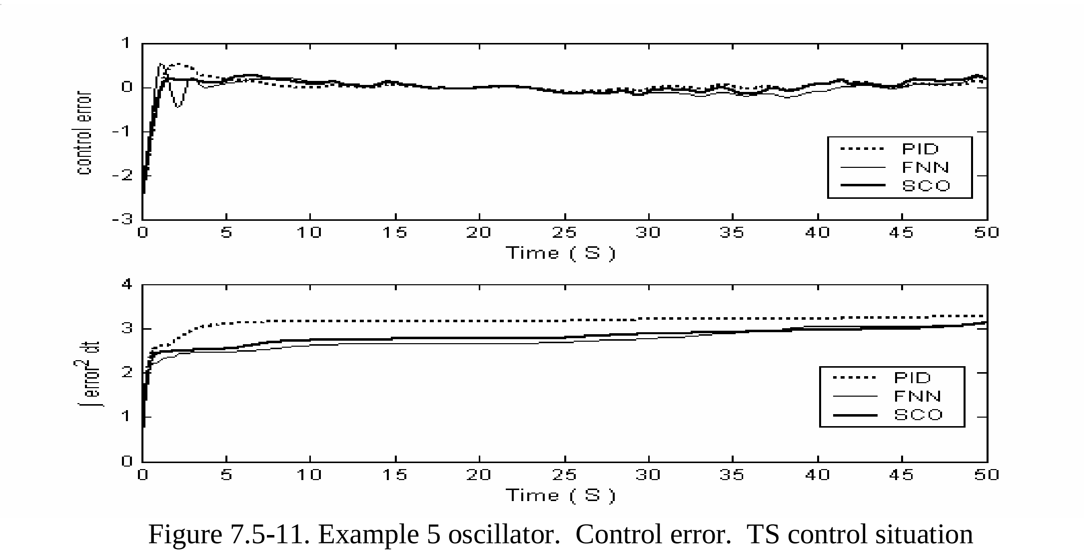


Figure 7.5-11. Example 5 oscillator. Control error. TS control situation

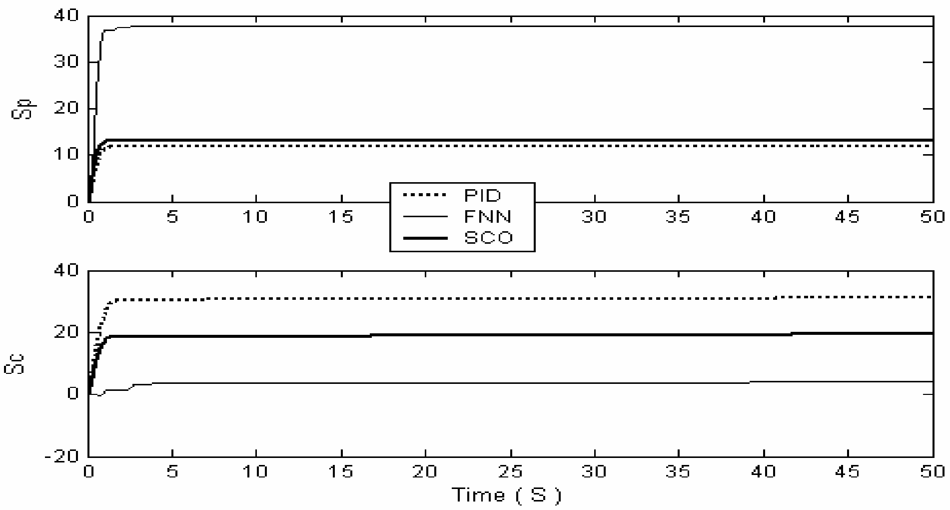


Figure 7.5-12. Example 5. Entropy productions of plant and Controller. TS control situation

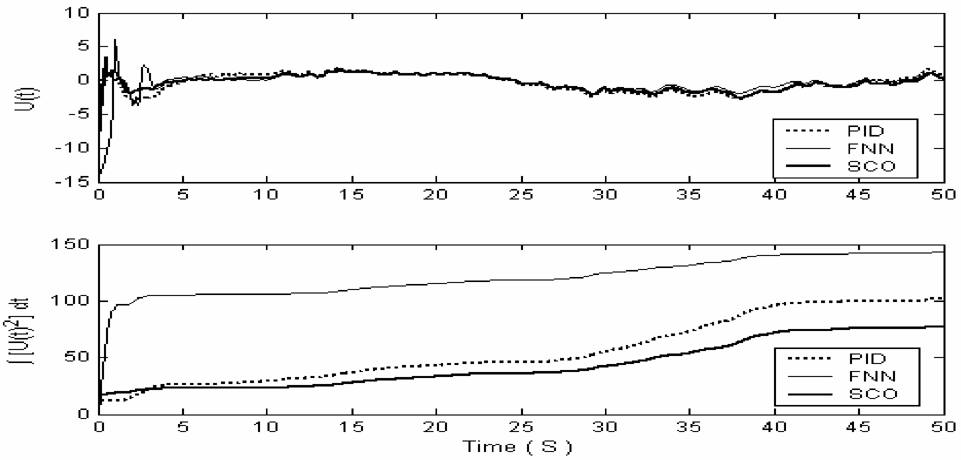


Figure 7.5-13. Example 5. Control force. TS control situation

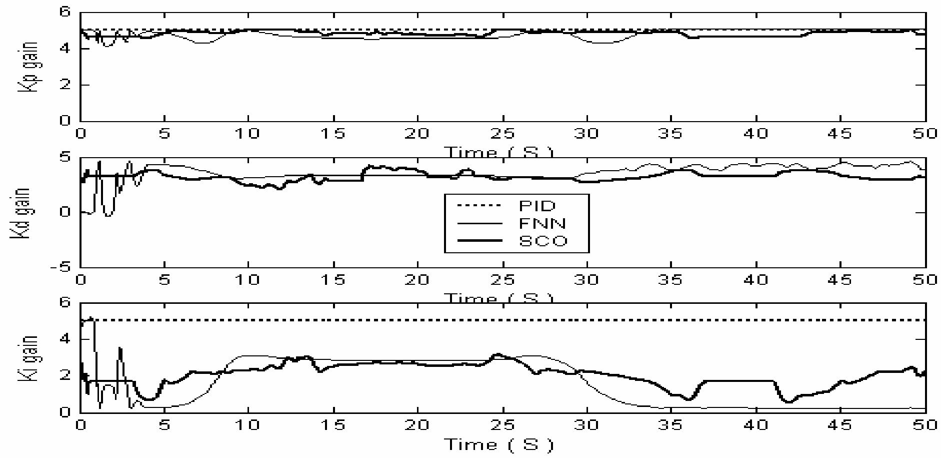


Figure 7.5-14. Example 5 oscillator. Control laws. TS control situation

Conclusions

- FC_SCO has smaller transit time and control error than classical PID and FC_FNN.
- From control quality point of view including a minimum of control error, minimum of entropy production in a plant and minimum of control force, Fuzzy PID-controller designed by SC Optimizer with *20 rules* realizes more effective (optimal) control in comparison to FC-FNN with *125 rules* and traditional PID-controller.

Robustness investigation of KB FC designed for the given TS control situation

Let us take FC_SCO, FC_FNN developed for the TS control situation and use them in new control situation, where

- new initial conditions [1] [0];
- new reference signal $x_{ref} = 0.1$;
- new type of noise: Gaussian (max amplitude =4);
- new model parameters: $\beta = 3\,; \alpha = 1\,; k_1 = 0.5; k = 1$

are considered.

In Fig. 7.5-15, 7.5-16, 7.5-17 and 7.5-18 results of comparison of CO stochastic motion under three types of control in the new control situation are shown.

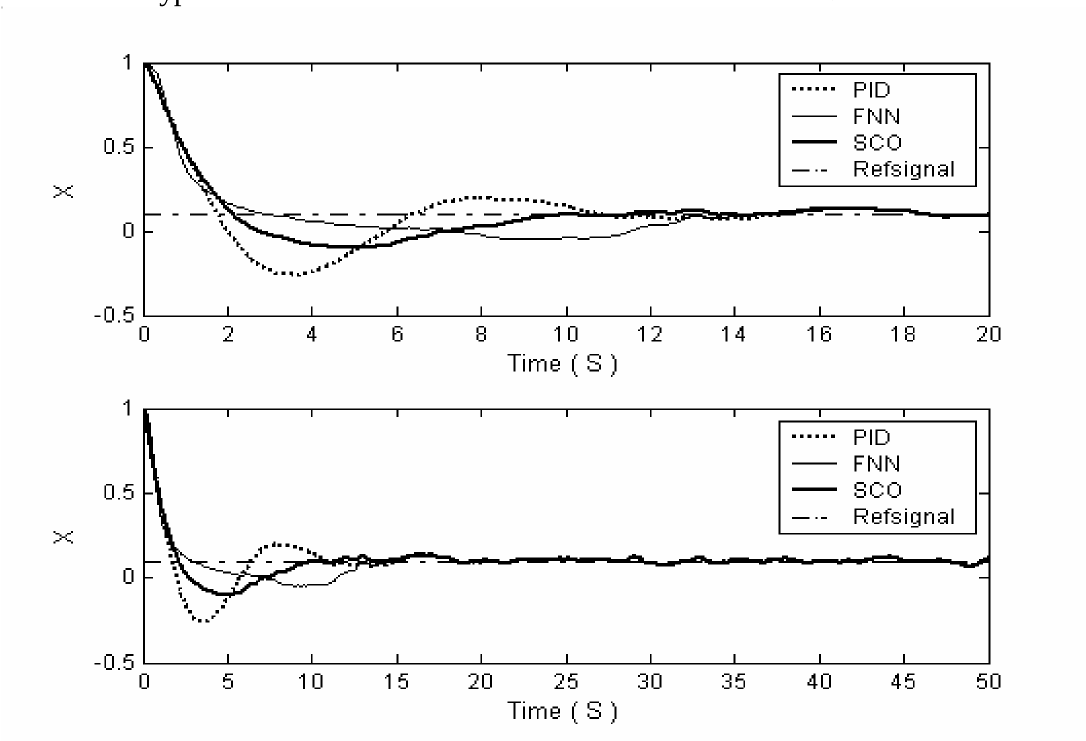


Figure 7.5-15. Example 5. Stochastic motion under 3 types of control. New control situation

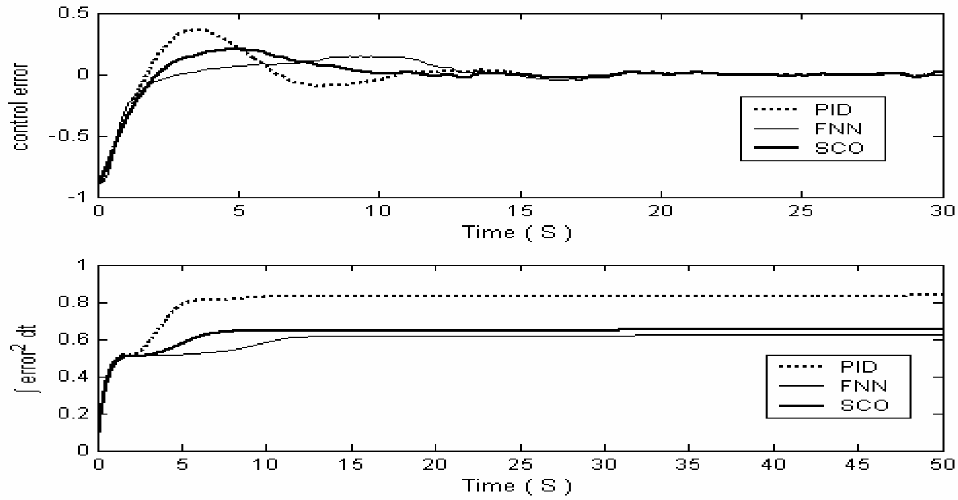


Figure 7.5-16. Example 5 oscillator. Control error. New control situation

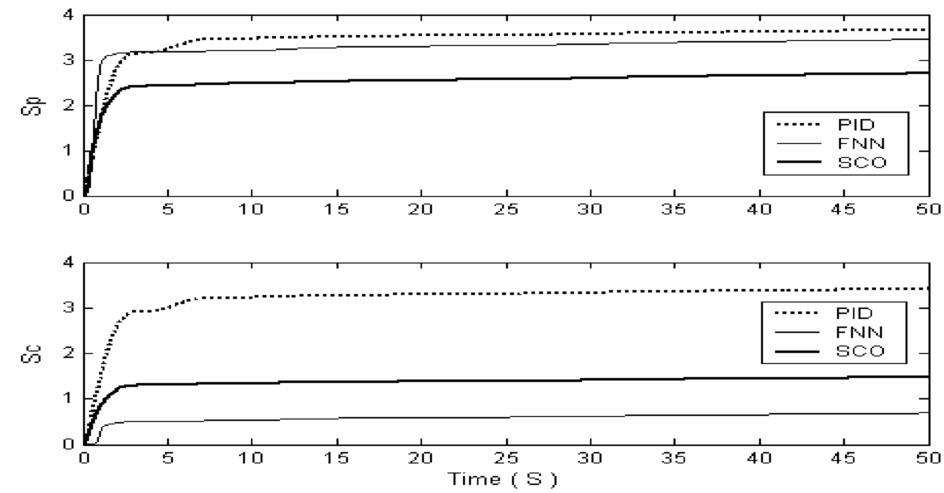


Figure 7.5-17. Example 5. Entropy productions of plant and Controller. New control situation

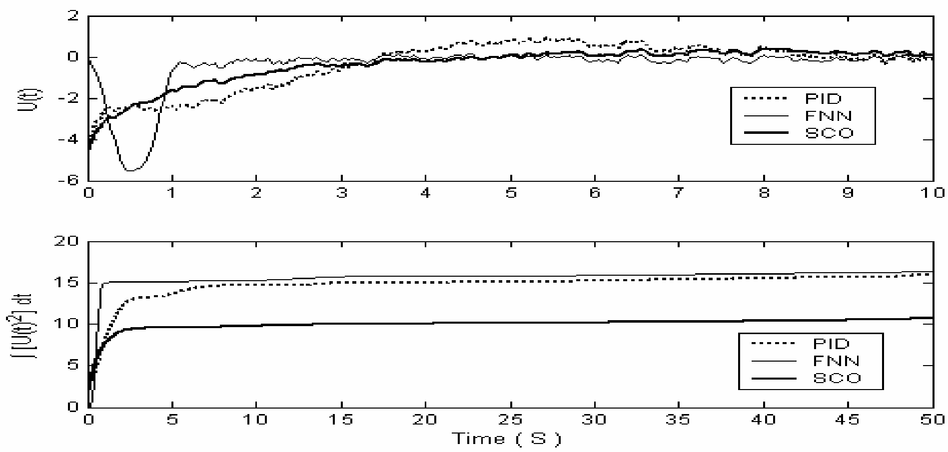


Figure 7.5-18. Example 5. Control force. New control situation

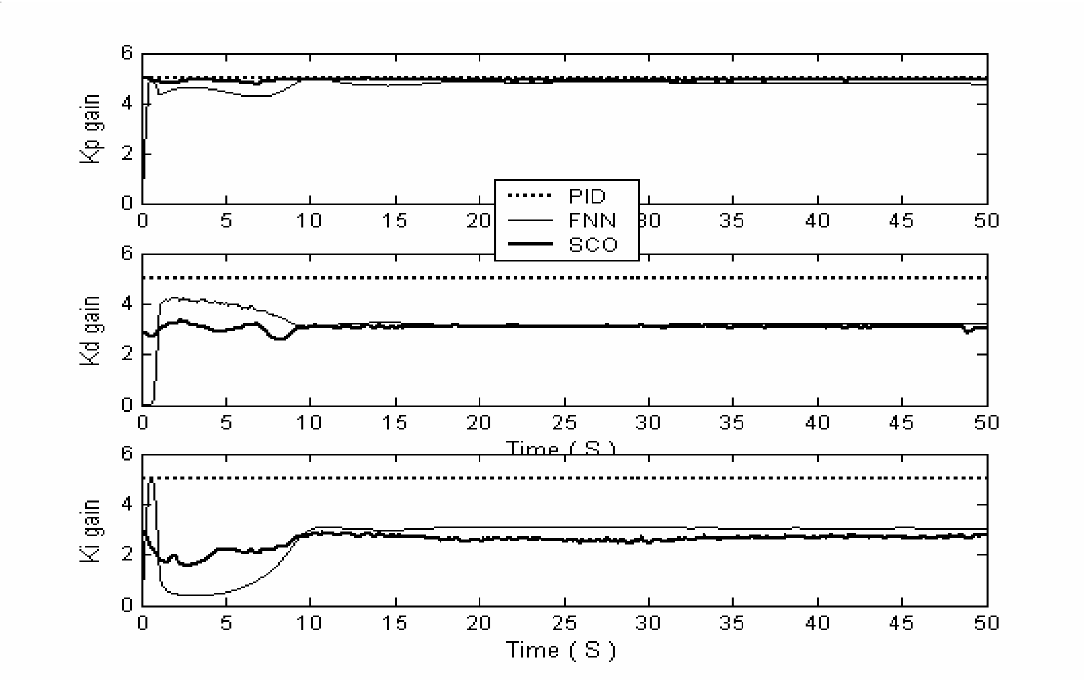


Figure 7.5-19. Example 5. Control laws. New control situation

Conclusions

- FC_SCO and FC_FNN are robust.
- Integral control error values of FC_SCO and FC_FNN are compatible.
- CO entropy production under FC_SCO control is smaller than under FC_FNN control and PID control.
- Control force and its integral value of FC_SCO is smaller than FC_FNN and PID.
- KB FC_SCO has smaller number of rules (*20 rules*) than KB FC-FNN (*125 rules*).

7.6 Example 6: *Nonlinear oscillator with sizable nonlinear dissipative components*

Equations of motion and entropy production rate:

$$\ddot{x} + 2\beta|\dot{x}|^5 \, sign(\dot{x}) + kx + \gamma x^3 = \xi(t) + u(t); \quad \frac{dS_x}{dt} = 2\beta sign(\dot{x})|\dot{x}|^5 \, \dot{x}.$$

Here $\xi(t)$ is a given stochastic excitations with an appropriate probability density function, and $u(t)$ is a control force. Consider the following model parameters and initial conditions. Model parameters: $\beta = 0.3; \gamma = 0.5; k = 4$. Initial conditions: $[x_0][\dot{x}_0]$ = [1.5] [0.1]. In Fig. 7.6-1 and 7.6-2 CO free motion (dynamic and thermodynamic behavior) with the given above parameters are shown.

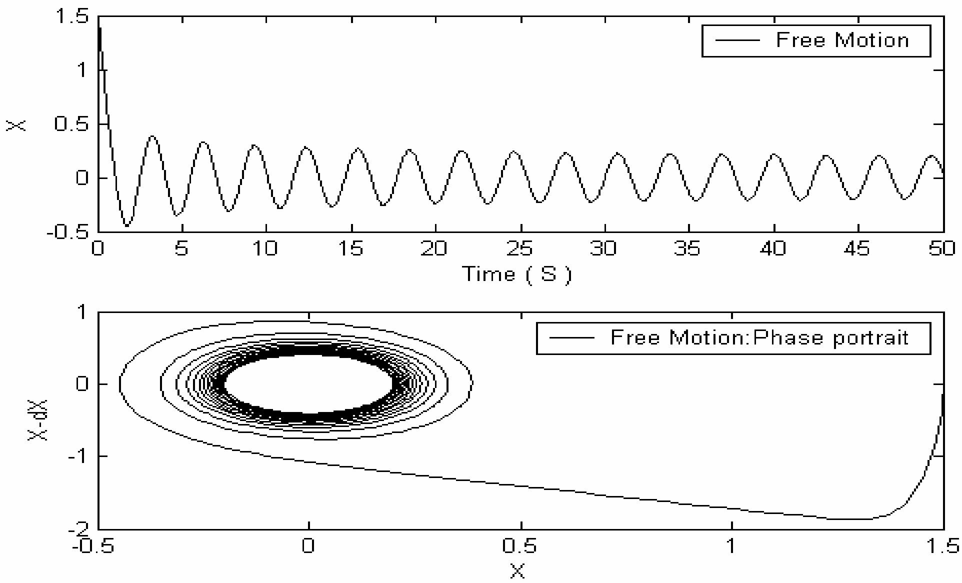


Figure 7.6-1. Example 6 oscillator. Free motion

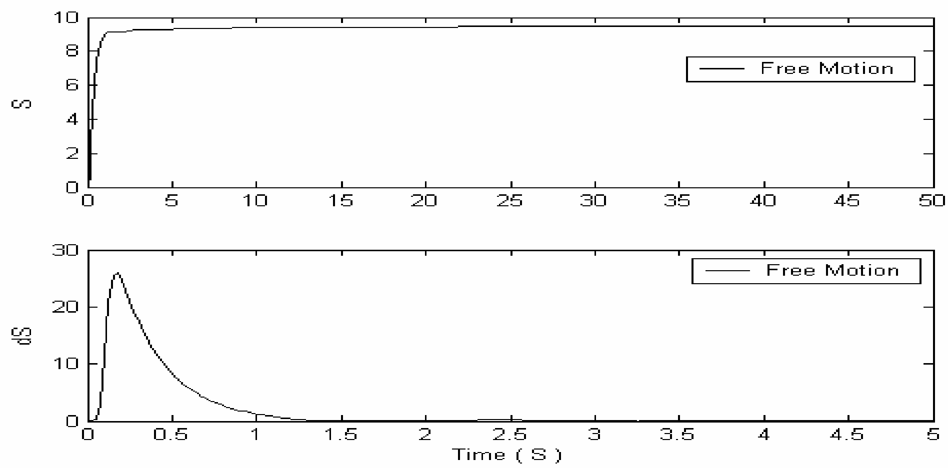


Figure 7.6-2: Example 6 oscillator. Free motion. Thermodynamic behavior

Simulation results show that the given CO represents an *asymptotically stable dynamic system*. Consider behaviour of this control object under two different types of stochastic excitations (Gaussian and Rayleigh noises) shown in Fig.7.6-3.

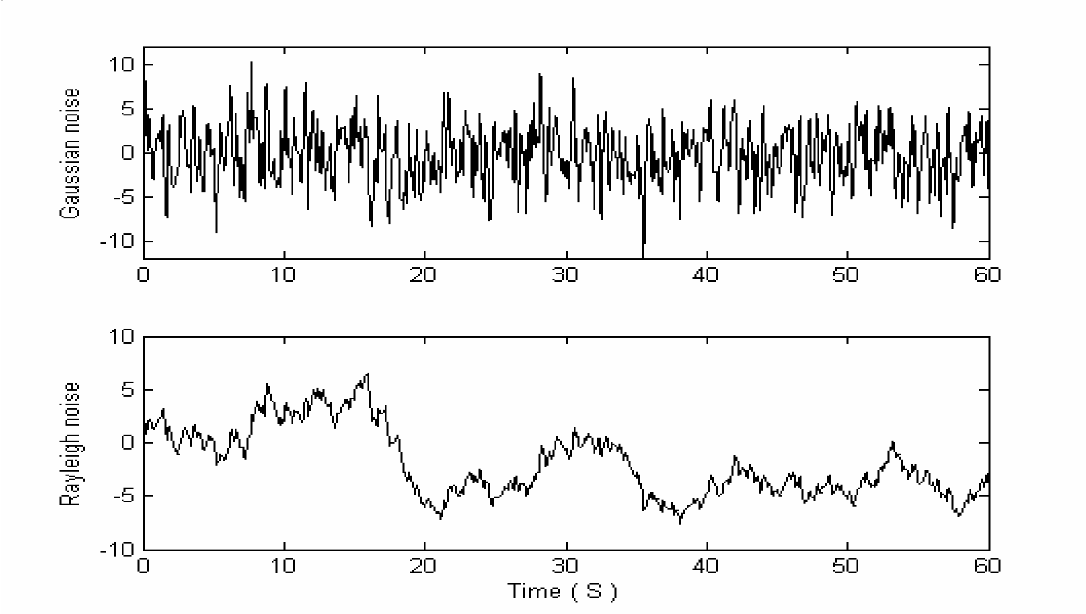


Figure 7.6-3. Stochastic noises: Gaussian (top) and Rayleigh (below)

In Figures 7.6-4, 7.6-5 and 7.6-6 dynamic and thermodynamic CO motion under stochastic noises is shown.

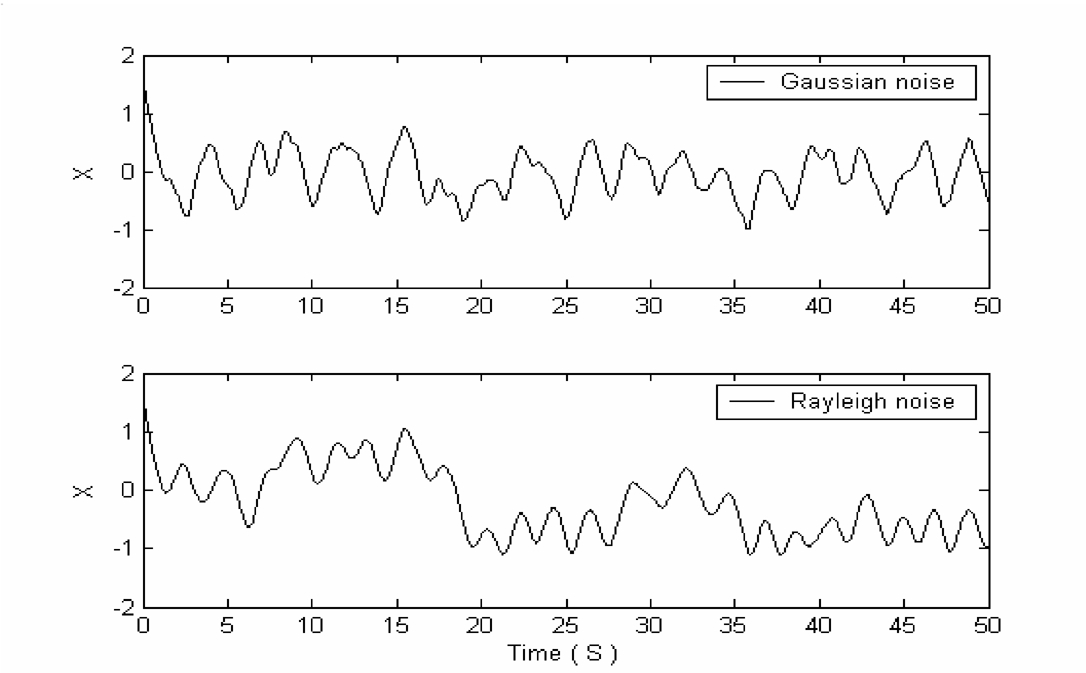


Figure 7.6-4. Example 6 oscillator. Stochastic motion

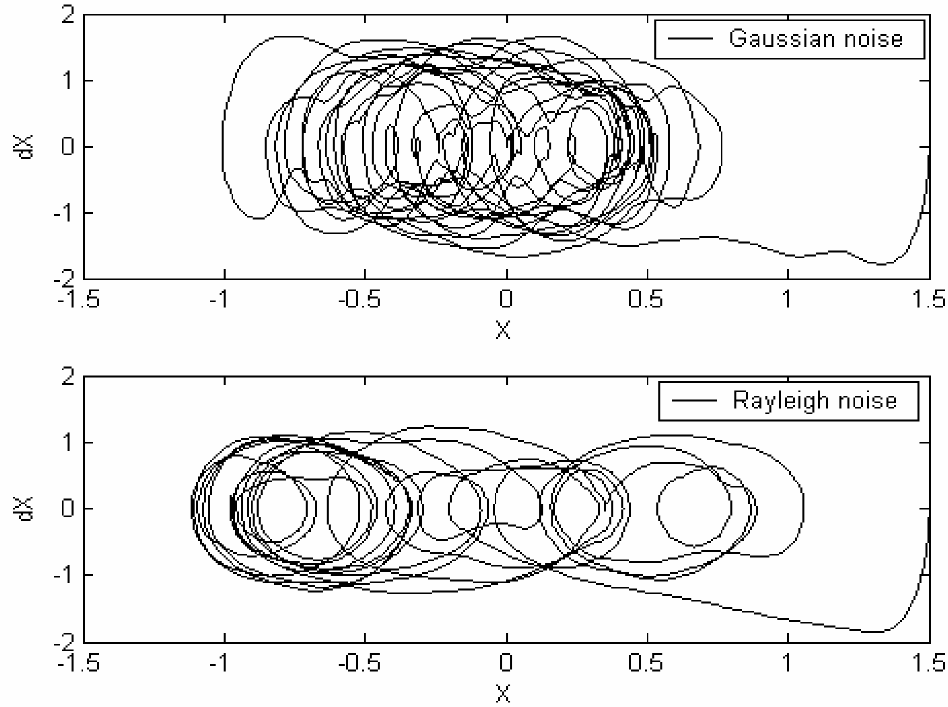


Figure 7.6-5. Example 6 oscillator. Stochastic motion. Phase portraits

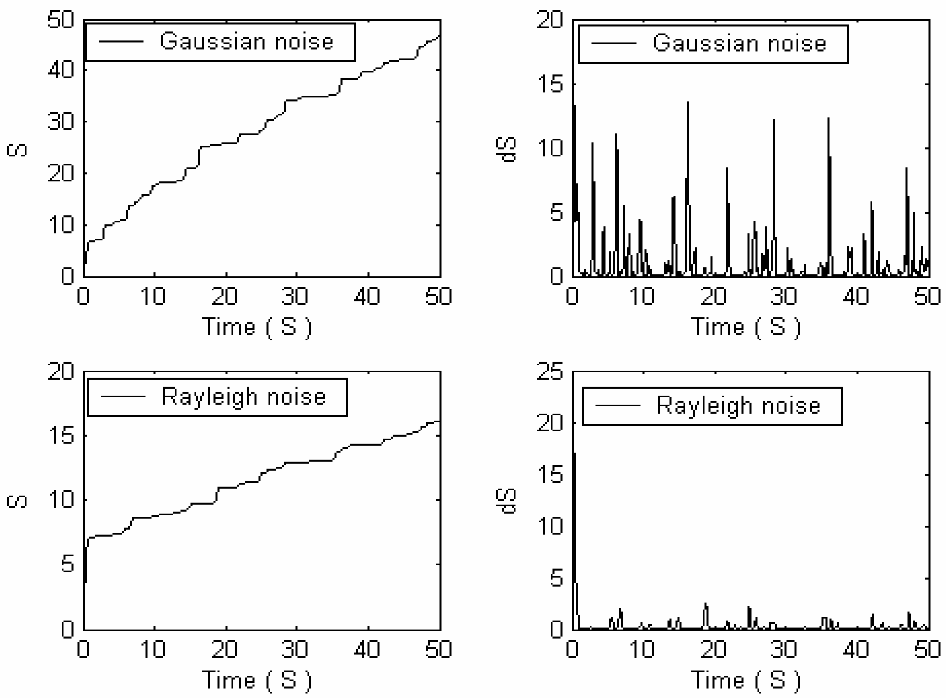


Figure 7.6-6. Example 6 oscillator. Stochastic motion. Thermodynamic behavior

Simulation results show that CO dynamic behavior as under Rayleigh and under Gaussian excitations are very chaotic.

Consider the following *control task* for this example: in the presence of Rayleigh (Gaussian) noise maintain motion of CO at the given reference signal $x_{ref} = 0$.
Let us design intelligent control system for the given above control problems by using our KB FC design tools and compare results with traditional PID Controller.

GA-based TS design

At this step we design TS of optimal control based on the given control quality criterion as a GA fitness function. For this case we will use the following GA parameters:

- GA parameters: search space for K-gains parameters: (0,10);
- GA FF : minimum of " control error and control error rate".

We have the following TS control situation:

- Model Parameters: $\beta = 0.3; \gamma = 0.5; k = 4$.
- Initial conditions: [1.5] [0.1]; Reference signals: $x_{ref} = 0$
- Rayleigh noise (max amplitude = 6);

In Fig.7.6-7 comparison of motion under GA and PID control is shown.

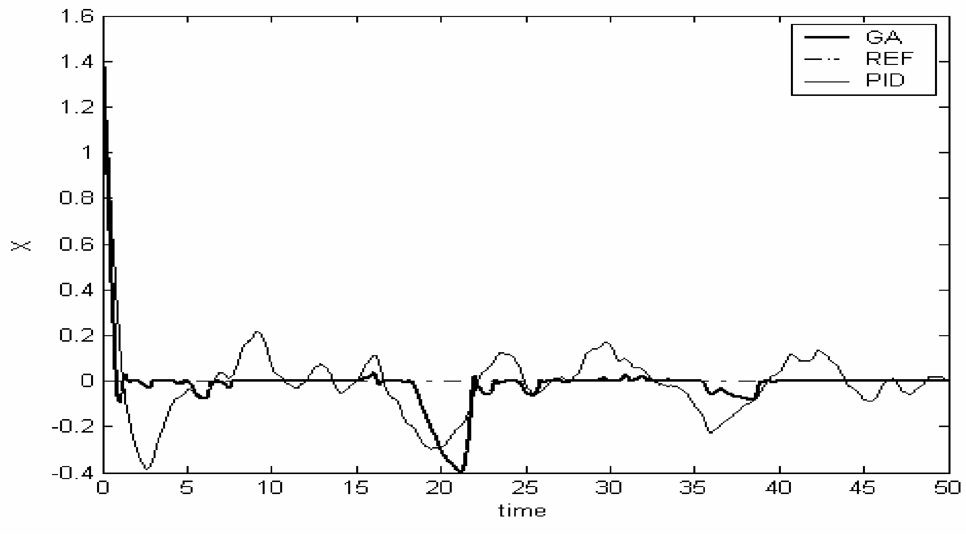


Figure 7.6-7. Example 6 oscillator. GA-PID control

FNN-based KB FC design process (step 1 technology)

For the given control tasks we will design FC-PID controller with three input variables to FC as $\left\{e, \dot{e}, \int edt\right\}$ and three output variables of FC as $\left\{k_p, k_d, k_i\right\}$.

AFM based KB design process is described as follows:

- Manual design of numbers of membership functions for each input variables: 5;

- Complete number of fuzzy rules: 5x5x5= 125 rules;
- Number of activated rules in KB**: 125 rules.**

In Fig. 7.6-8 AFM representation of membership functions for input FC variables is shown.

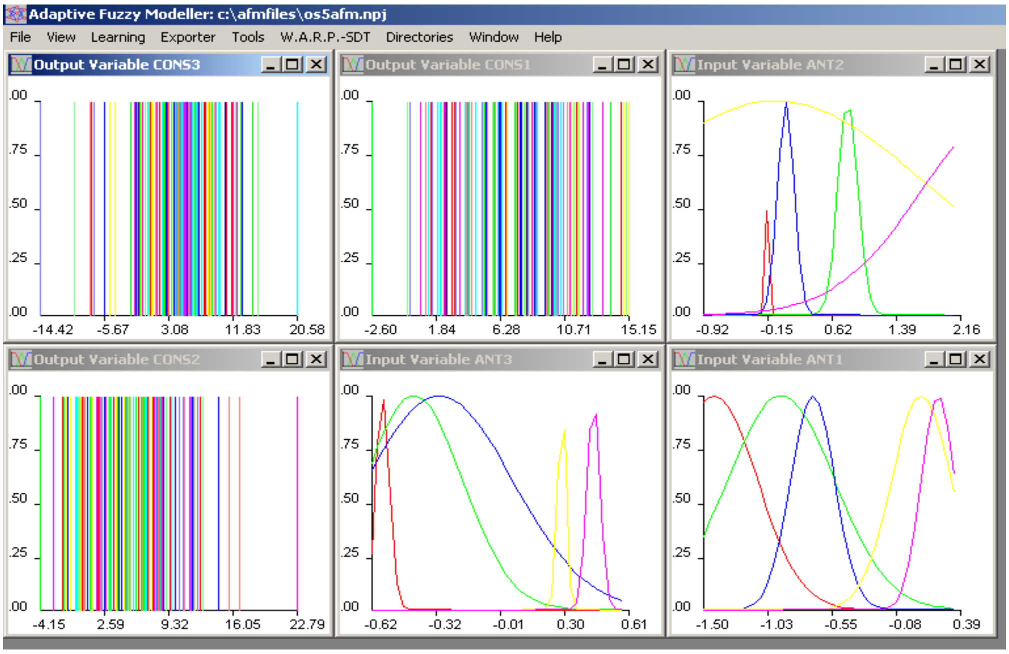


Figure 7.6-8. Example 6. AFM based membership functions representation

SC Optimizer-based KB FC design process (step 2 technology)

The process of KB FC design is described as follows:

- *Creation of linguistic values* by GA1 : number of membership functions for each input variables: 6,9,8 ;
- Complete number of fuzzy rules: 6x9x8 = 432 rules;
- *Rule selection* : SUM criterion with automated choice of threshold level;
- *KB optimization by* GA2: optimized KB contains **15 rules.**

In Fig. 7.6-9, SC Optimizer representation of membership functions and their shapes for third input FC variables is shown.

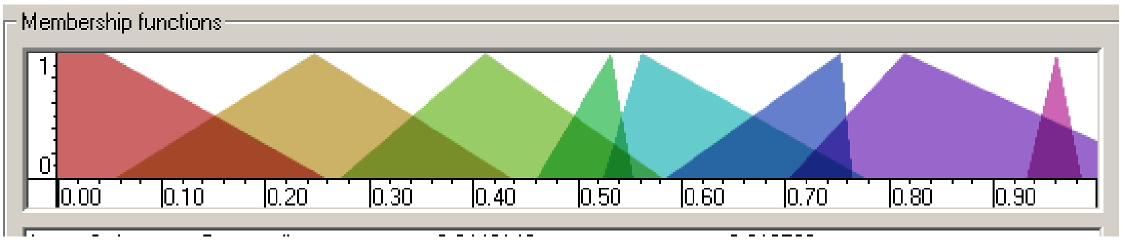


Figure 7.6-9. Example 6. SC Optimizer based membership functions representation

Remark. In AFM based representation number and MF shapes are selected manually by a user. In SC optimizer for each input variables optimal MF shapes and MF number are defined by GA.

Compare control quality of FC_SCO obtained by SC Optimizer with *15 rules*, FC_FNN with *125 rules* obtained by traditional SC approach based on FNN-tuning and traditional PID controller with constant gains K = (10 10 10).
In Figures 7.6-10, 7.6-11, and 7.6-12 results of comparison of CO stochastic motion under three types of control are shown.
Control laws for TS control situation are shown in Fig. 7.6-13.

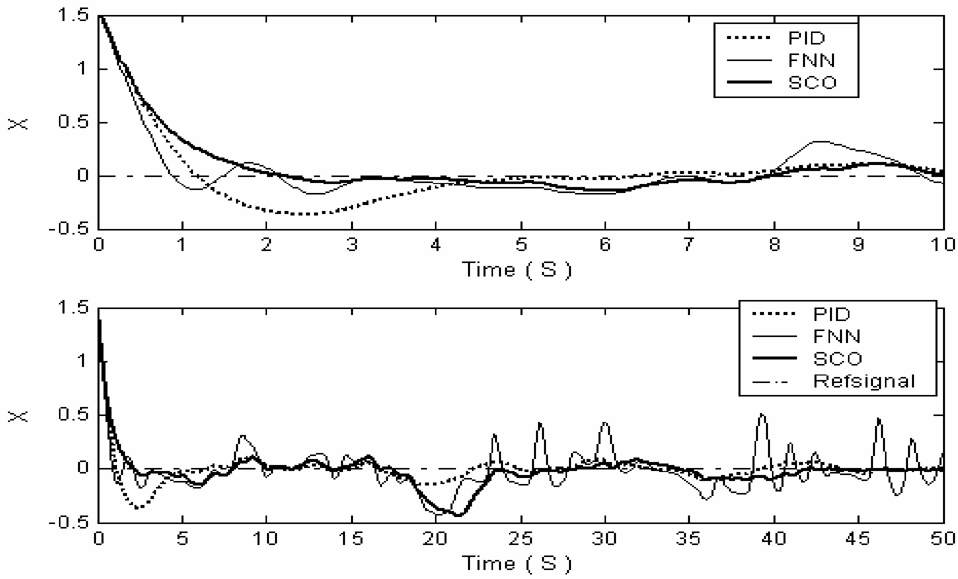


Figure 7.6-10. Example 6 oscillator motion. TS control situation

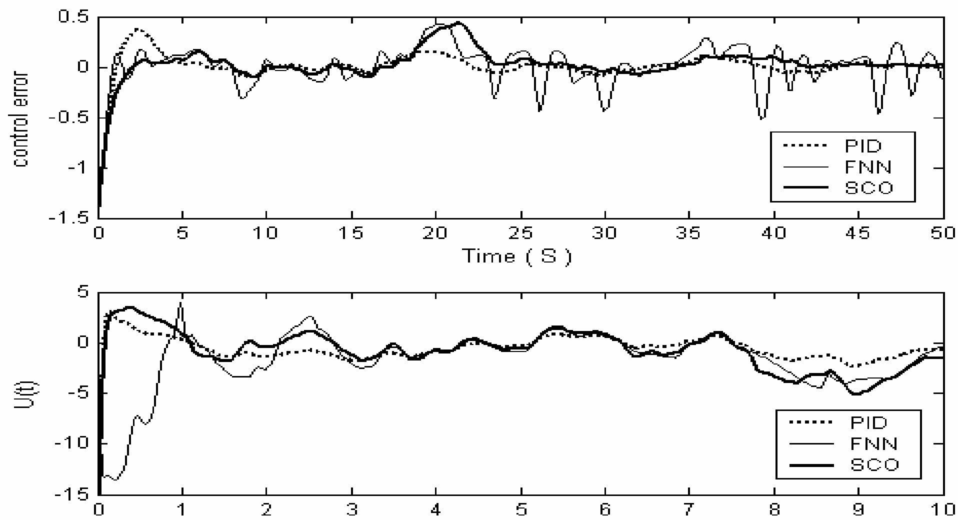


Figure 7.6-11. Example 6 oscillator. Control error and control force. TS control situation

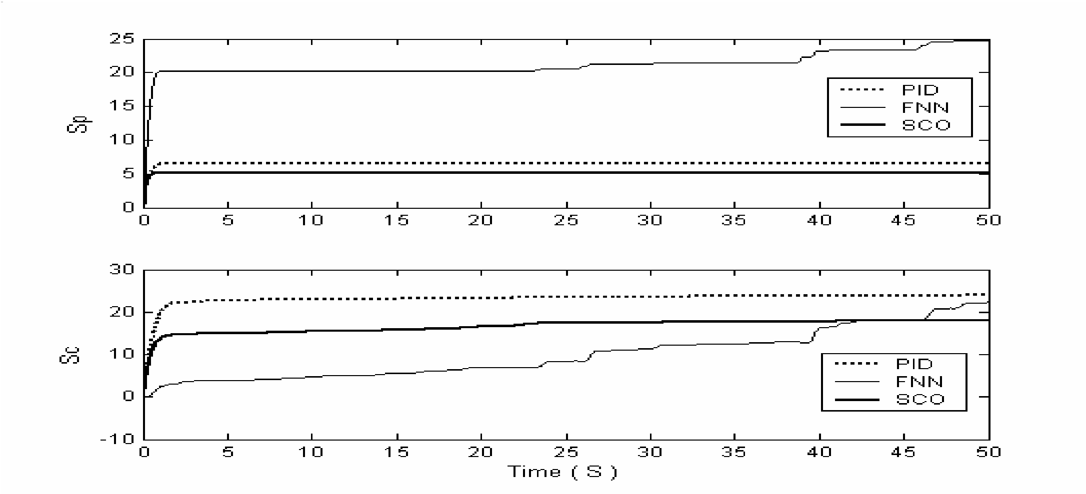


Figure 7.6-12. Example 6. Entropy productions of plant and Controller. TS control situation

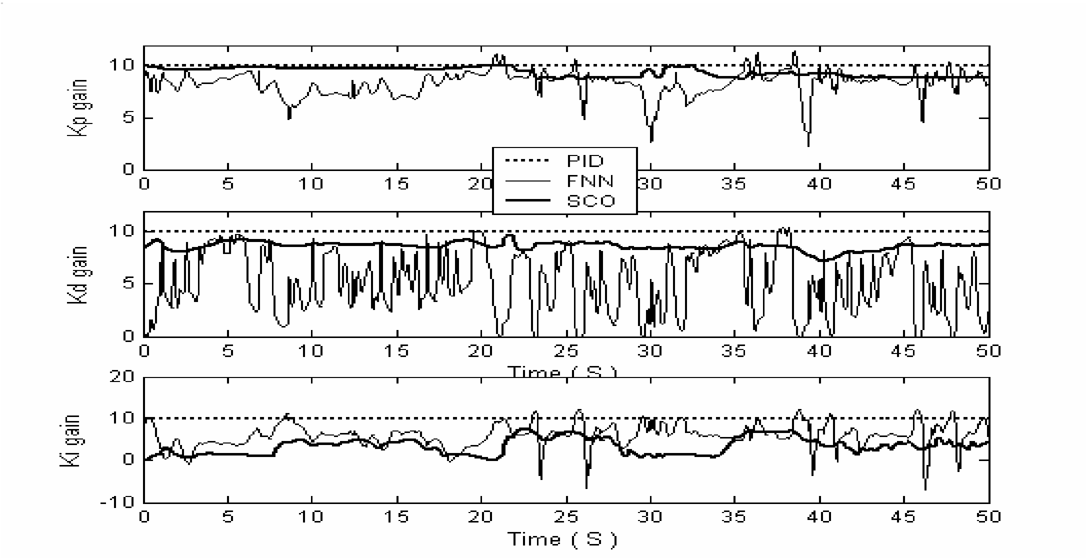


Figure 7.6-13. Example 6 oscillator. Control laws. TS control situation

Conclusions

- FC_SCO has smaller transit time and control error than classical PID and FC_FNN.
- At time 10-50 sec FC_SCO and PID performance in the given TS control situation are compatible.
- FC_FNN has worst performance (comparatively big control error).
- From control quality point of view including minimum of control error, minimum of entropy production in a plant and controller, and a minimum of control force, Fuzzy PID-controller designed by SC Optimizer with *15 rules* realizes more effective (optimal) control than FC-FNN with *125 rules.*

Robustness investigation of KB FC designed for the given TS control situation

Let us take FC_SCO, FC_FNN developed for the TS control situation and use them in new control situation, where

- new initial conditions [1] [0];
- new noise Rayleigh (max amplitude = 3);
- new model parameters: $\beta = 0.4\,;\gamma = 0.3\,;k = 5$

are considered.

In Fig.7.6-15, 7.6-16, 7.6-17 and 7.6-18 results of comparison of CO stochastic motion under three types of control in the new control situation are shown.

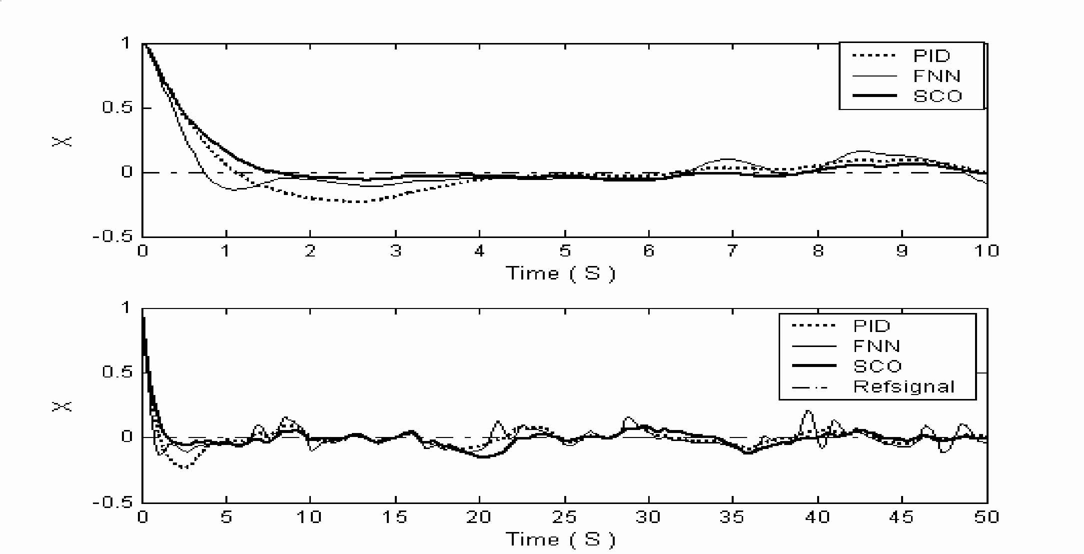


Figure 7.6-14. Example 6 oscillator motion. New control situation

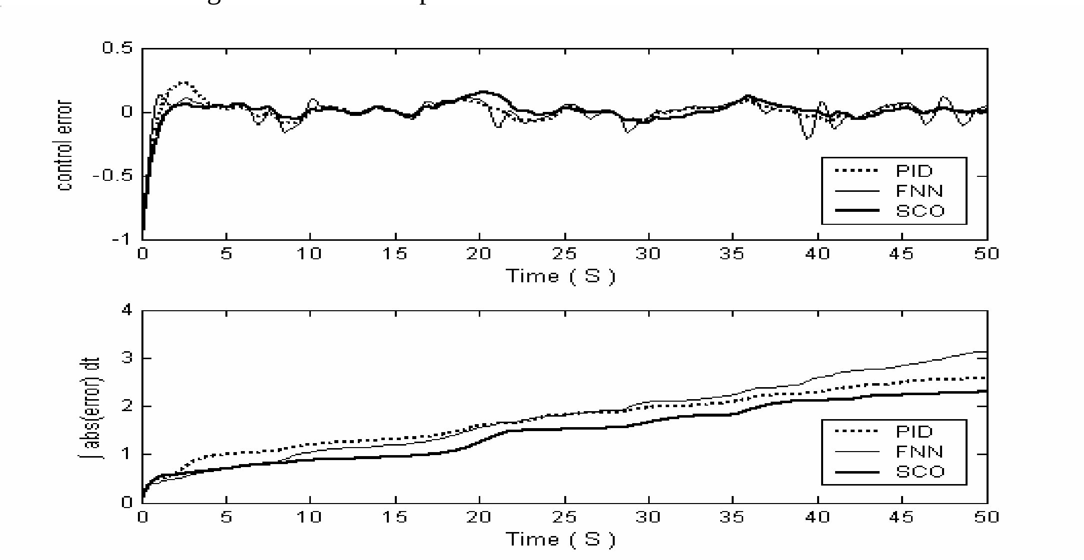


Figure 7.6-15. Example 6 oscillator. Control error. New control situation

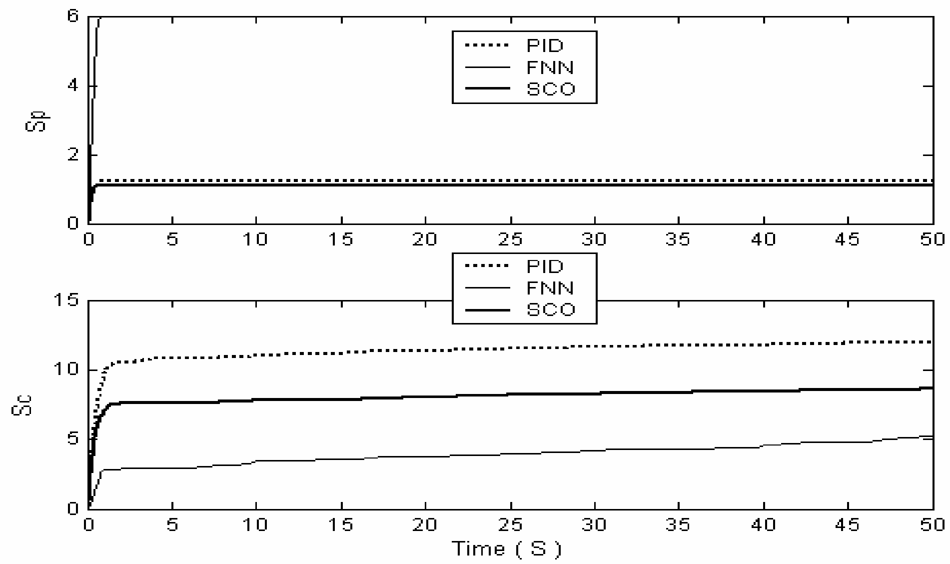


Figure 7.6-16. Example 6. Entropy productions of plant and Controller. New control situation

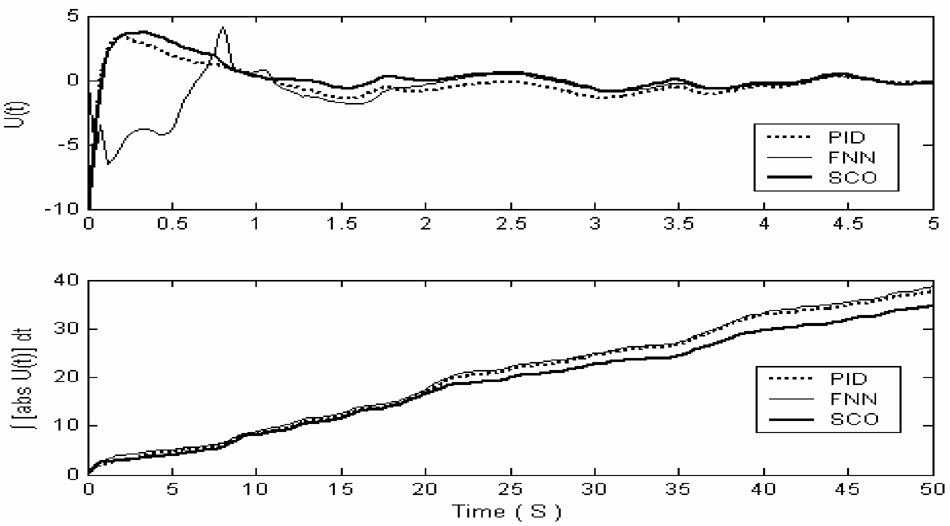


Figure 7.6-17. Example 6. Control force. New control situation

Conclusions

- FC_SCO is robust.
- FC_SCO has smaller transit time and control error than classical PID and FC_FNN.
- At time 10-50 sec, FC_SCO and PID performance in the given TS control situation are compatible.
- FC_FNN has worst performance (comparatively big control error).

- From control quality point of view (minimum of control error, minimum of entropy production in a plant and in controller, and minimum of control force) Fuzzy PID-controller designed by SC Optimizer with **15** rules is more effective (optimal) than FC-FNN with **125** rules and traditional PID-controller.

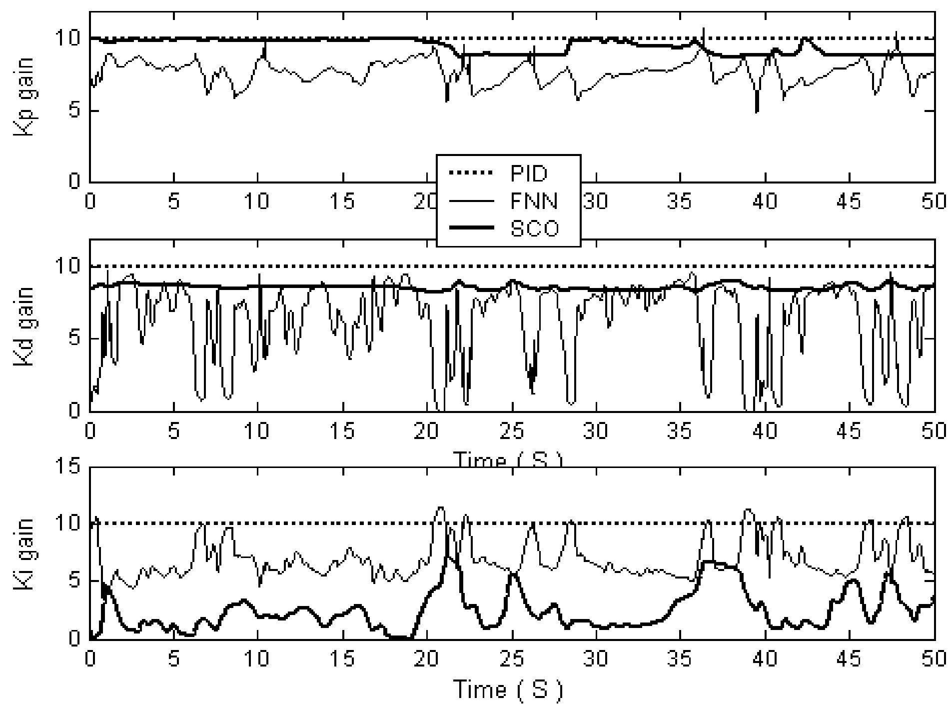


Figure 7.6-18. Example 6. Control laws. New control situation

7.7 Example 7: *Nonlinear Nose-Hoover oscillator*

Equations of motion and entropy production rate are as follows:

$$\begin{cases} \ddot{x} + \xi\dot{x} + x = \eta(t) + u(t) \\ \dot{\xi} = \dot{x}^2 - T \end{cases}, \quad \frac{dS_x}{dt} = \xi\dot{x}\dot{x}.$$

Here $\eta(t)$ is a given stochastic excitations with an appropriate probability density function, and $u(t)$ is a control force.

Model parameters: $T = 1$. Initial conditions: $[x_0\ \dot{x}_0\ \xi_0][\dot{x}_0]$= [0.001 0.01 –0.1], [0.01].

In Fig. 7.7-1, 7.7-2 and 7.7-3 free motion (dynamic and thermodynamic behavior) of CO is shown.

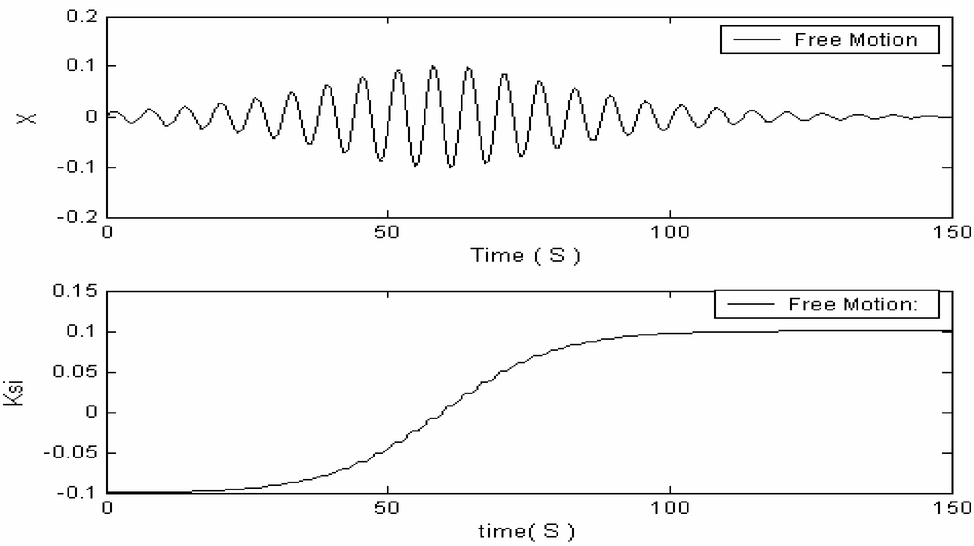


Figure 7.7-1. Nose-Hoover oscillator. Free motion

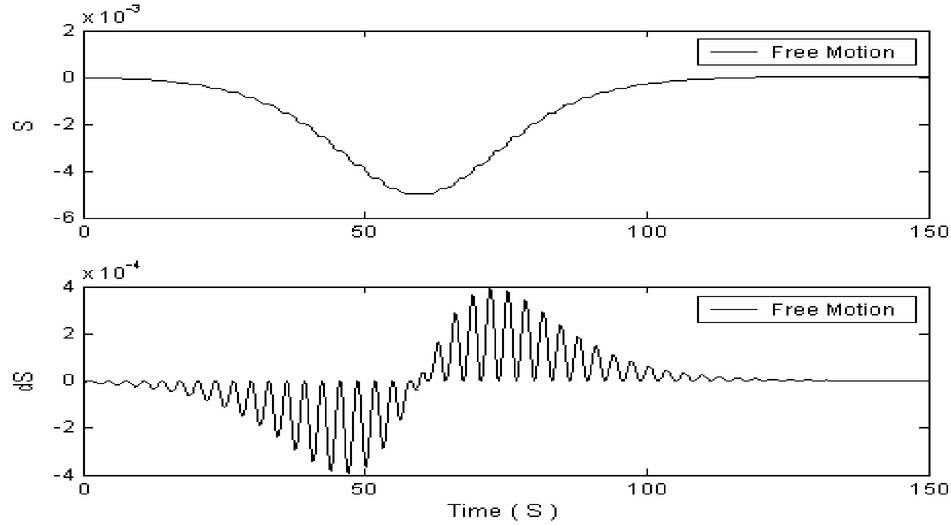


Figure 7.7-2: Nose-Hoover oscillator. Free motion. Thermodynamic behavior

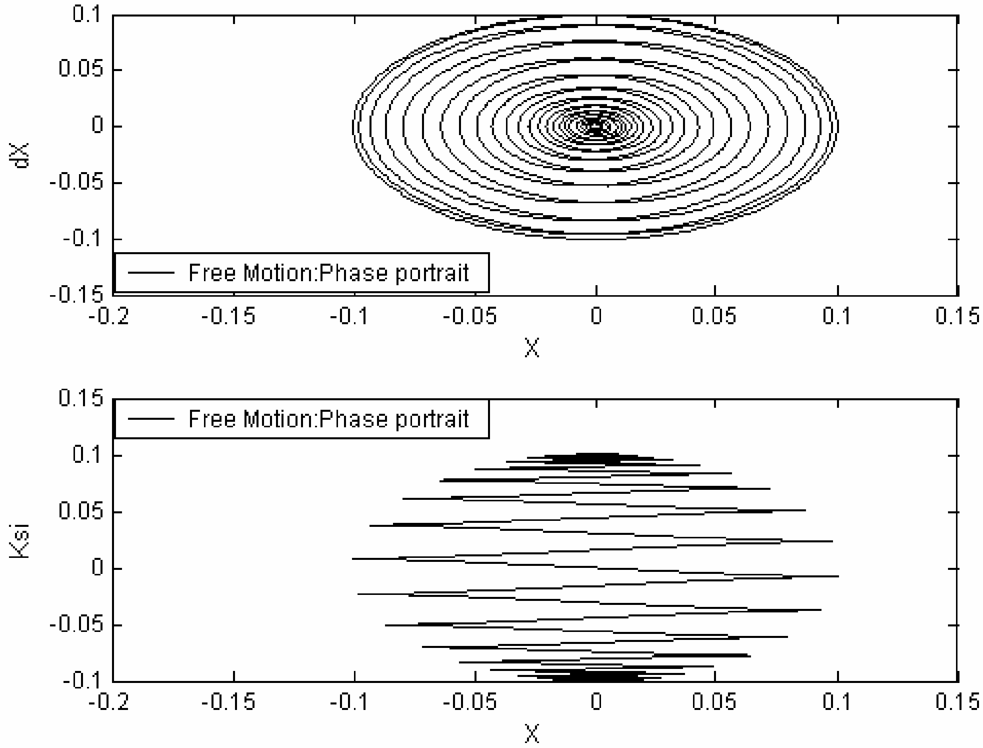


Figure 7.7-3. Nose-Hoover oscillator. Free motion. Phase portraits

Free motion simulation results show that $\exists \Delta t : S_x(t) < 0, \dot{S}_x(t) < 0$. So, CO motion is *locally unstable* in Lyapunov sense.

Consider CO behaviour under two different types of stochastic excitations (Gaussian and Rayleigh noises) shown in Fig. 7.7-4.

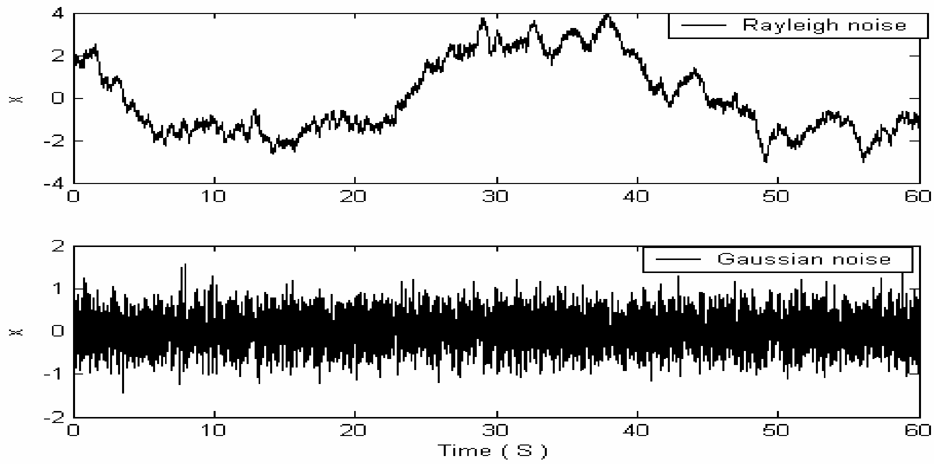


Figure 7.7-4. Stochastic noises: Gaussian (top) and Rayleigh (below)

In Figures 7.7-5, 7.7-6 and 7.7-6 dynamic and thermodynamic motion under stochastic noises is shown.

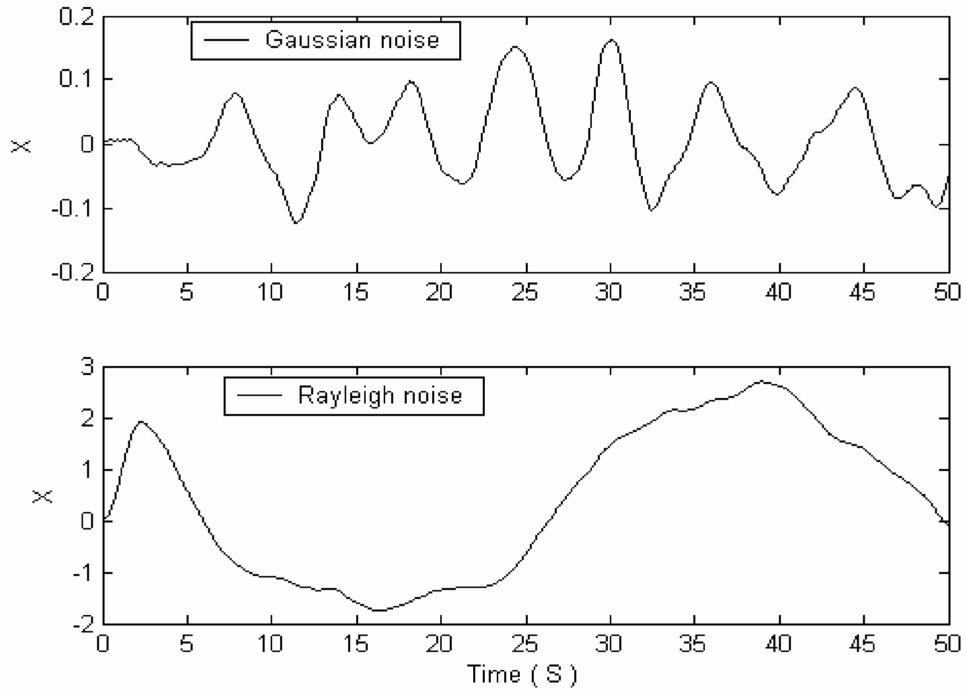


Figure 7.7-5. Example 7 oscillator. Stochastic motion

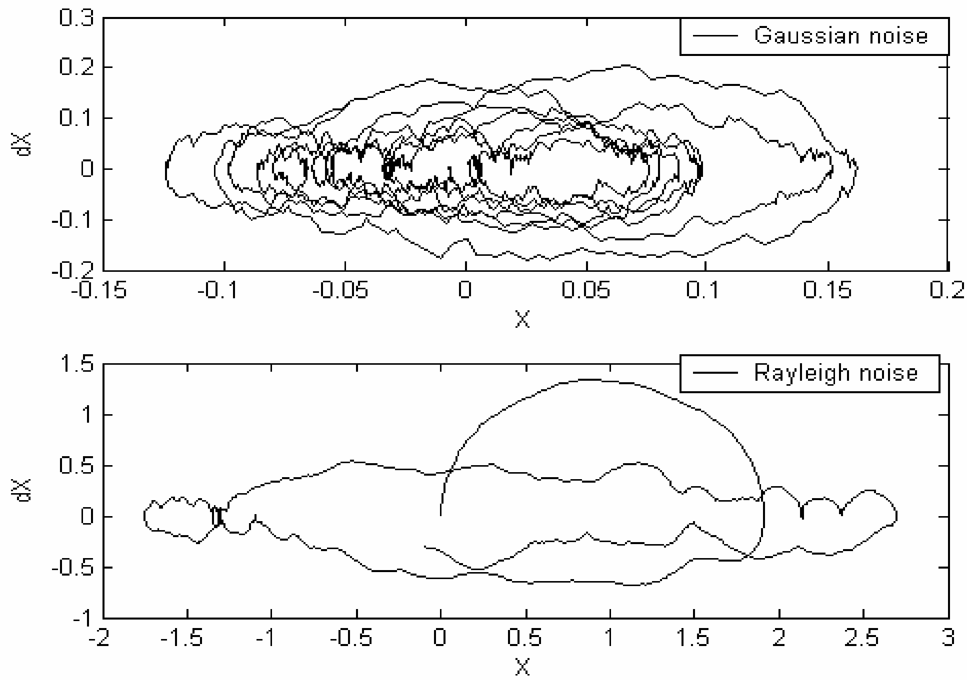


Figure 7.7-6. Example 7 oscillator. Stochastic motion. Phase portraits

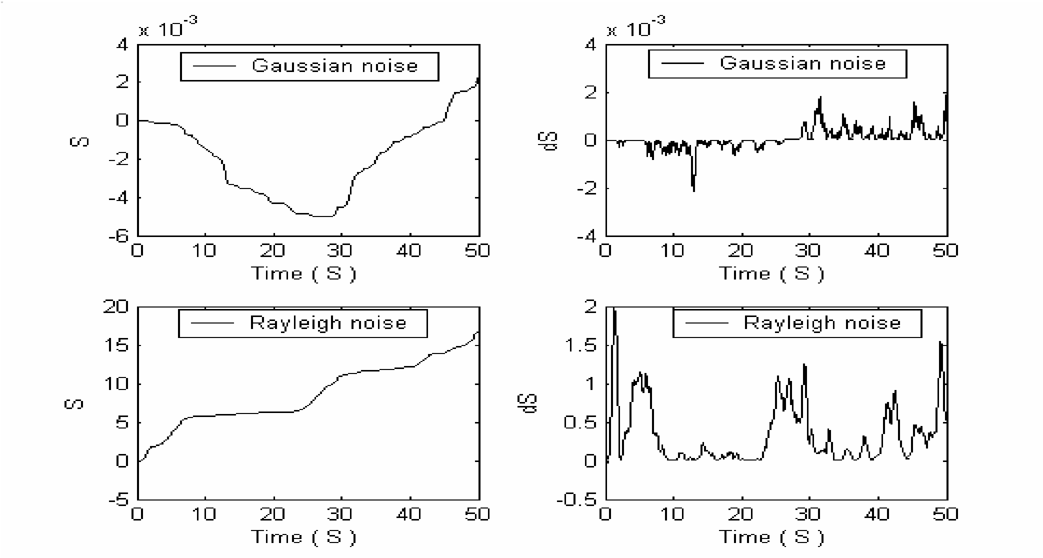


Figure 7.7-7. Example 7 oscillator. Stochastic motion. Thermodynamic behavior

Simulation results show that CO dynamic behavior as under Rayleigh and under Gausssian excitations are very chaotic.

Consider the following *control task* for this example: in the presence of Rayleigh noise maintain motion of CO at the given reference signal $x_{ref} = 0.1$.
Let us design intelligent control system for the given above control problems by using our KB FC design tools and compare results with traditional PID Controller.

GA-based TS design

At this step we design TS of optimal control based on the given control quality criterion as a GA fitness function. For this case we will use the following GA parameters:

- GA parameters: search space for K-gains parameters: (0,30);
- GA FF : minimum of "control error and entropy production rate of CO".

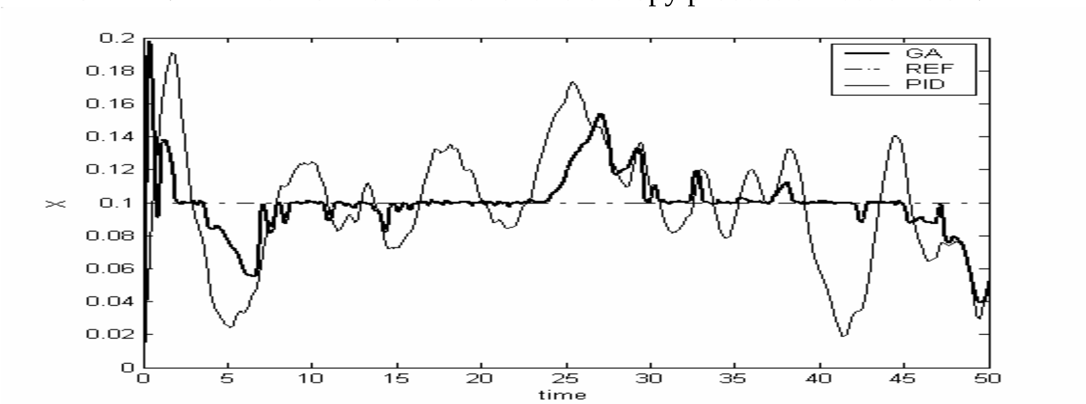


Figure 7.7-8. Example 7 oscillator. GA-PID control

We have the following TS control situation:

- Model Parameter: $T = 1$;
- Initial conditions: [0.001 0.01 –0.1], [0.01]. ; Reference signals: $x_{ref} = 0.1$
- Rayleigh noise (max amplitude = 4);

In Fig.7.7-8 comparison of CO motion under GA and PID control is shown.

FNN-based KB FC design process (step 1 technology)

For the given control tasks we will design FC-PID controller with three input variables to FC as $\{e, \dot{e}, \int edt\}$ and three output variables of FC as $\{k_p, k_d, k_i\}$.

AFM based KB design process is described as follows:

- Manual design of numbers of membership functions for each input variables: (5, 3, 5);
- Complete number of fuzzy rules: 5x3x5 = 75 rules;
- Number of activated rules in KB**: 75 rules.**

In Fig. 7.7-9 AFM representation of membership functions for input FC variables is shown.

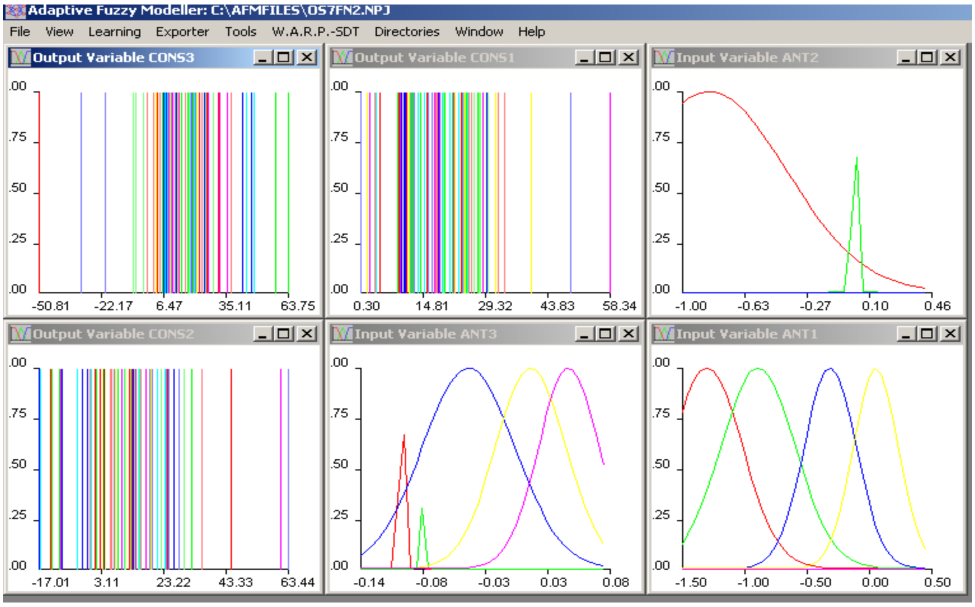


Figure 7.7-9. Example 7. AFM based membership functions representation

SC Optimizer-based KB FC design process (step 2 technology)

SC Optimizer based KB design process is characterized as follows:

- *Creation of linguistic values* by GA1: optimal number of membership functions for each input variables: 7,9,6;
- Complete number of fuzzy rules:7x9x6 = 378 rules;

- *Rules selection by:* with SUM criterion and limited number of rules = 35;
- *KB optimization by* GA2: number of rules = 35;
- *KB refinement by* GA3: with minimum approximation error criterion;
- Optimized KB contains **35 rules.**

In Fig. 7.7-10 example of SC Optimizer representation of membership functions and their shapes for input FC variables is shown.

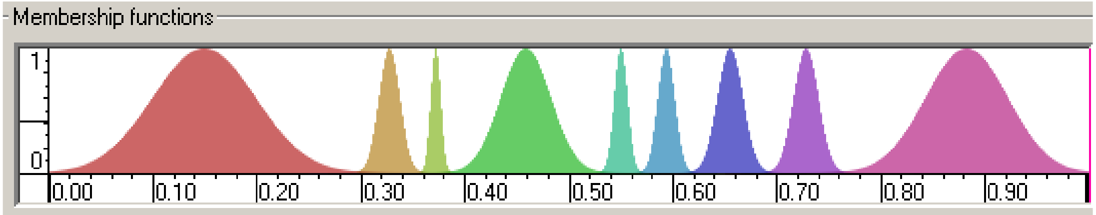


Figure 7.7-10. Example 7. SC Optimizer based membership functions representation

Remark. In AFM based representation number and MF shapes are selected manually by a user. In SC optimizer for each input variables optimal MF shapes and MF number are defined by GA.

Compare control quality of FC_SCO obtained by SC Optimizer with 35 rules, FC_FNN obtained by traditional SC approach based on FNN-tuning (with AFM tools) with 75 rules and traditional PID controller with constant gains K = (15 15 15).

In Figures 7.7-11, 7.7-12, 7.7-13, 7.7-14, 7.7-15 and 7.7-16 results of comparison of CO stochastic motion under three types of control are shown.

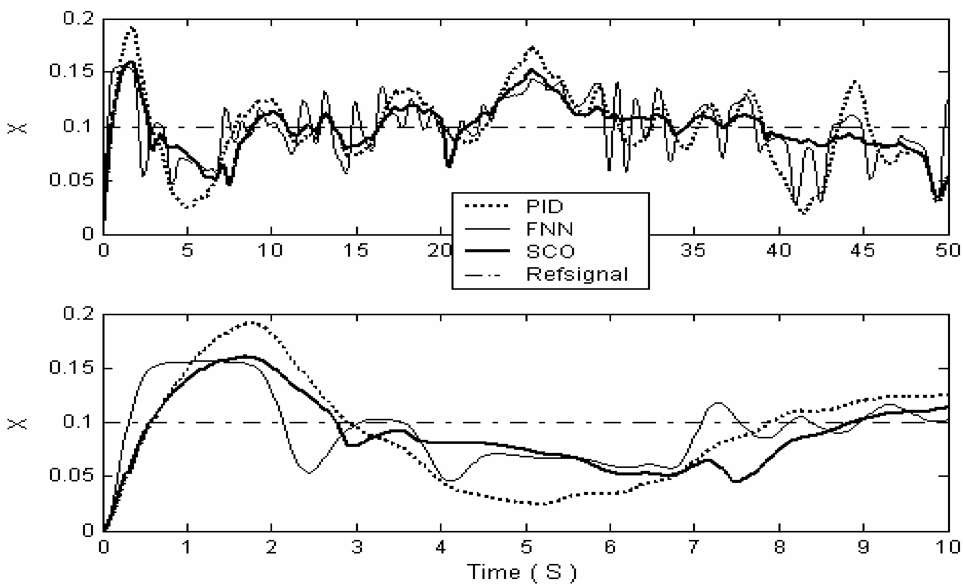


Figure 7.7-11. Example 7 oscillator motion. TS control situation

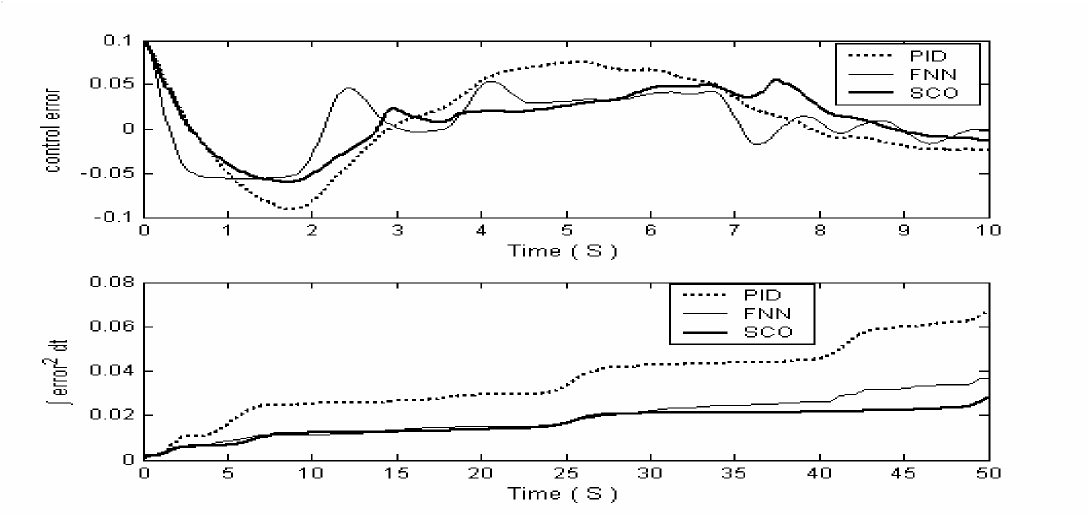


Figure 7.7-12. Example 7 oscillator. Control error. TS control situation

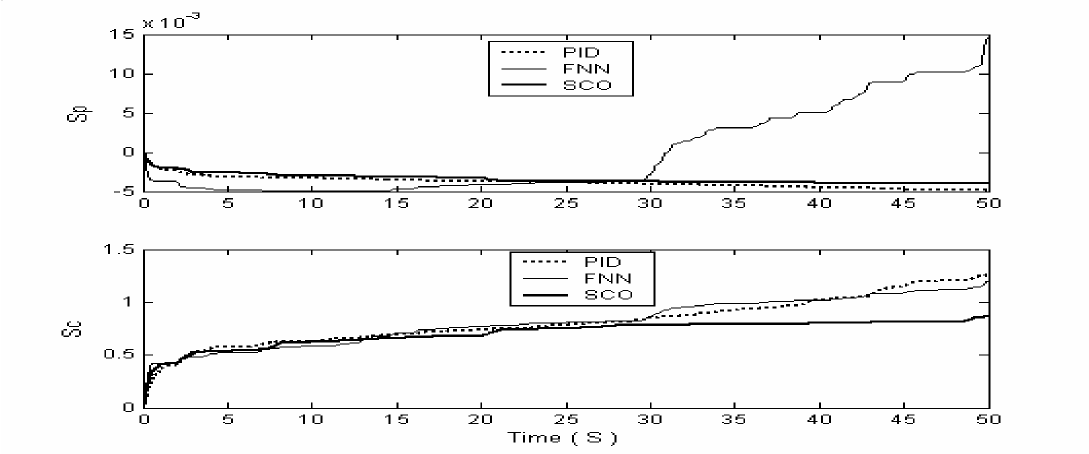


Figure 7.7-13. Entropy productions of plant and controller. TS control situation

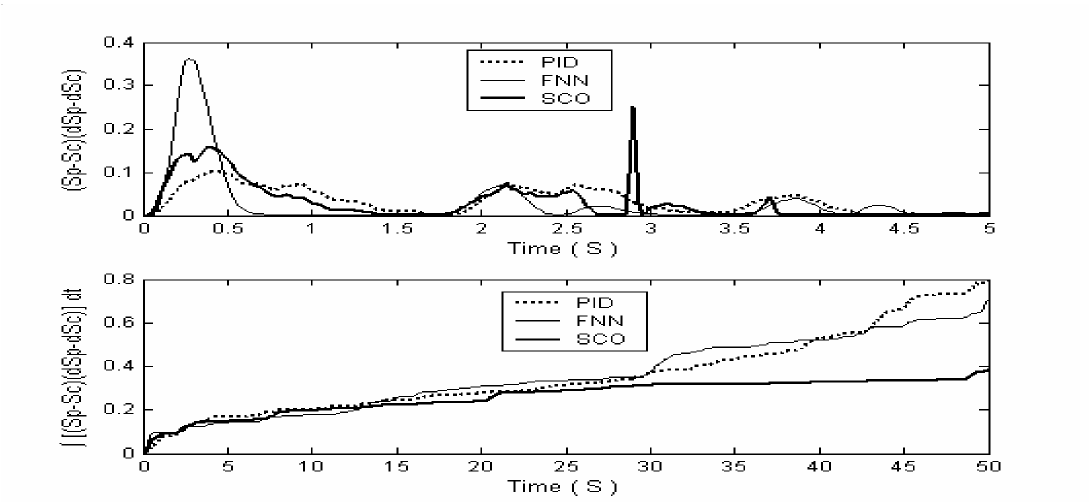


Figure 7.7-14. Example 7. Generalized entropy production. TS control situation

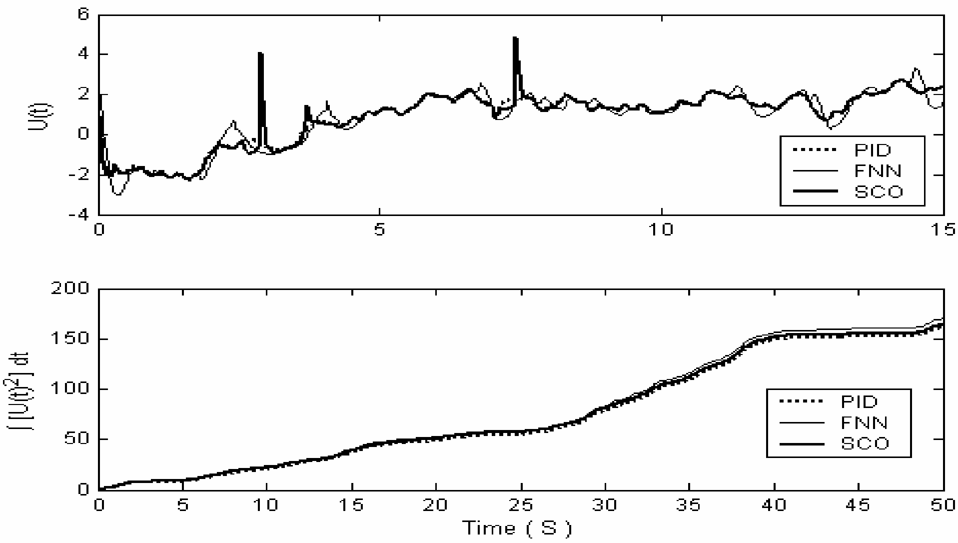


Figure 7.7-15. Example 7 oscillator. Control force. TS control situation

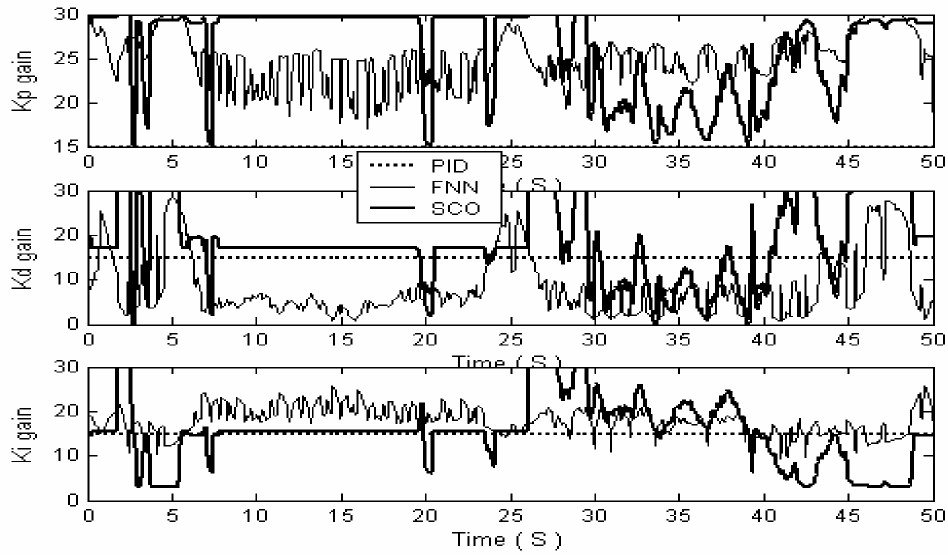


Figure 7.7-16. Example 7 oscillator. Control laws. TS control situation

Conclusions

- FC_SCO, FC_FNN and PID performance in the given TS control situation are compatible.
- From control quality point of view including minimum of control error and minimum of generalized entropy, Fuzzy PID-controller designed by SC Optimizer with *35 rules* realizes more effective (optimal) control than FC-FNN with *75 rules* and traditional PID-controller.

Robustness investigation of KB FC designed for the given TS control situation

Let us take FC_SCO, FC_FNN developed for the TS control situation and use them in new control situation, where

- new initial conditions: [1.5 0.01 -5.5] [0.01];
- new reference signal = 0; and
- new type of noise: Gaussian (max A = 4).

are considered. In Fig. 7.7-17, 7.7-18 and 7.7-19 results of comparison of CO stochastic motion under 3 types of control in the new control situation are shown. Classical PID K-gains are: K = (25 15 15).

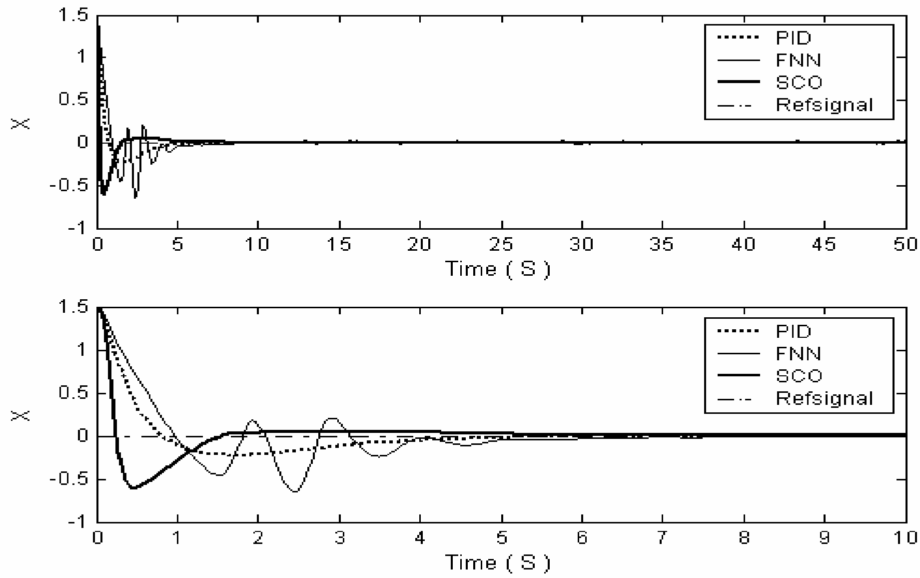


Figure 7.7-17. Nose-Hoover oscillator motion. New control situation

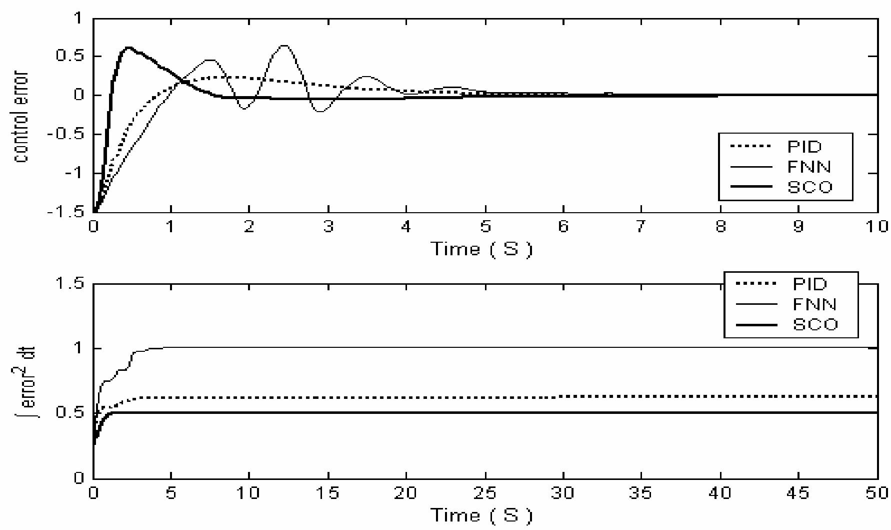


Figure 7.7-18. Nose-Hoover oscillator. Control error. New control situation

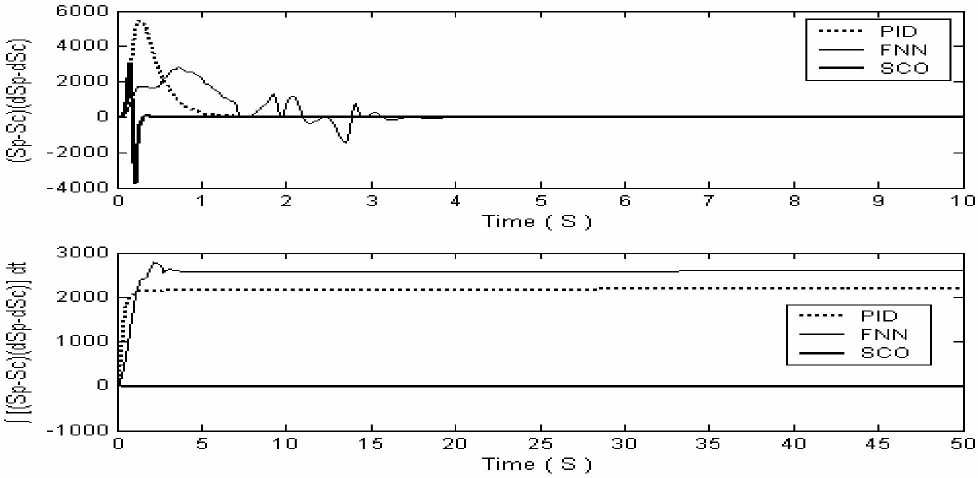


Figure 7.7-19. Example 7. Generalized entropy production. New control situation

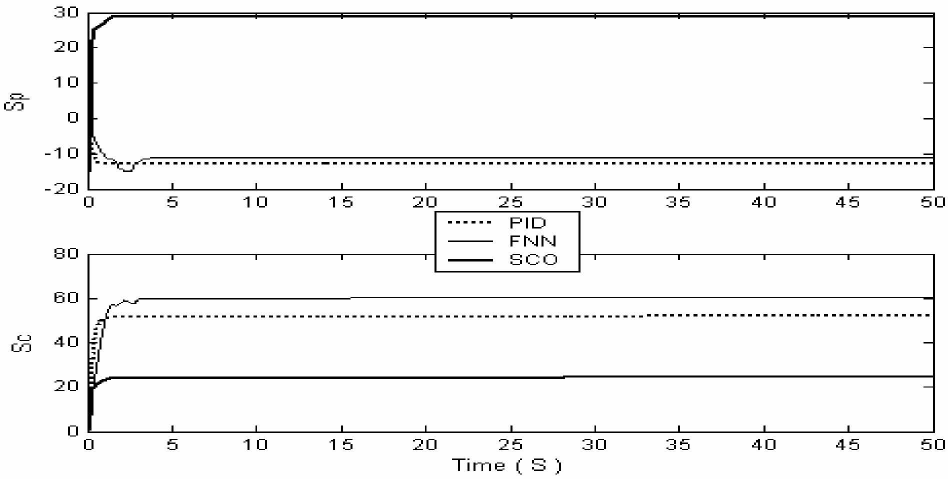


Figure 7.7-20. Example 7. Entropy productions of plant and Controller. New control situation

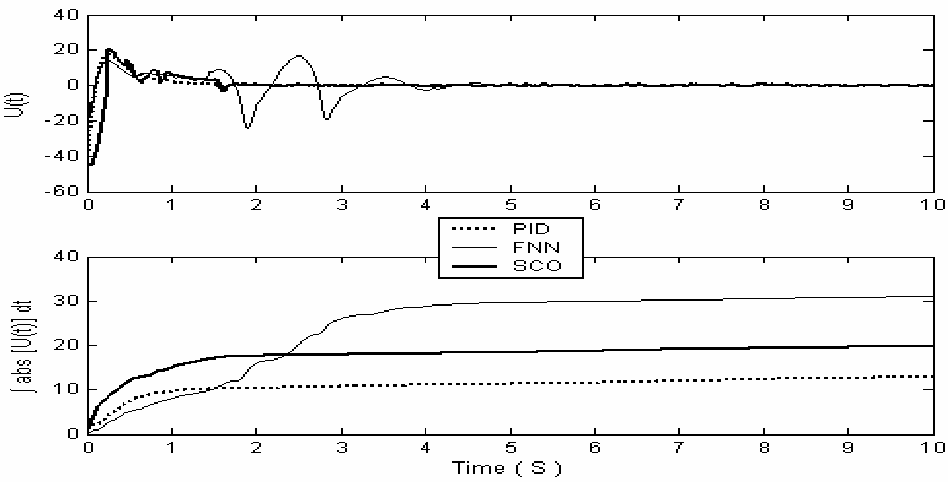


Figure 7.7-21. Example 7 oscillator. Control force. New control situation

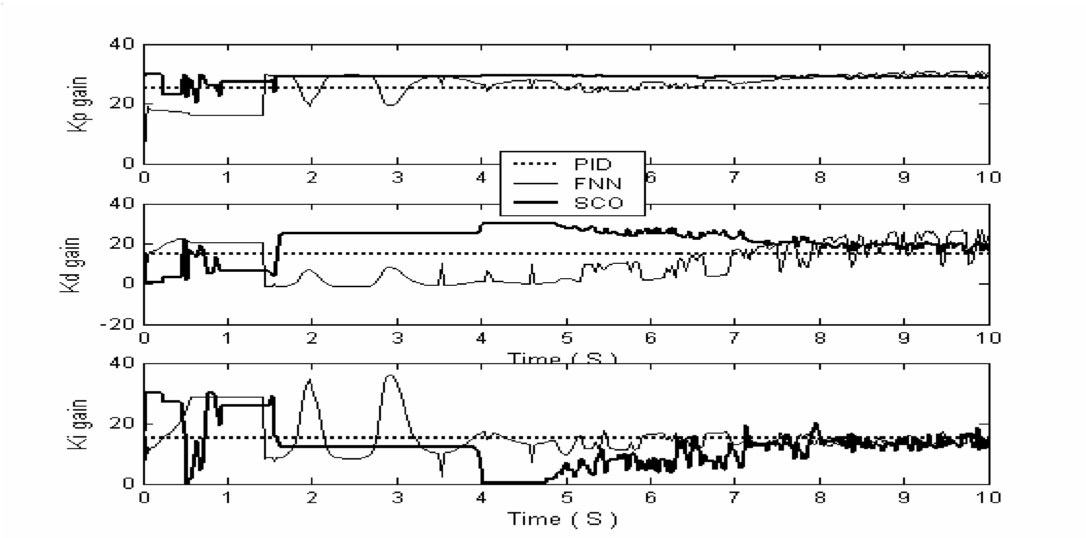


Figure 7.7-22. Example 7 oscillator. Control laws. New control situation

Conclusions

- FC_SCO and FC_FNN performance in the given new control situation are robust.
- FC_FNN has worst performance (comparatively big control error).
- FC_SCO and PID performance in the given control situation are compatible.
- From control quality point of view including a minimum of control error and minimum of generalized entropy production, Fuzzy PID-controller designed by SC Optimizer with **35** rules is more effective (optimal) than FC-FNN with **75** rules.

General conclusions from benchmarks simulation results

- Classical control system doesn't work well, if we have globally unstable or essentially non-linear CO in the presence of Rayleigh noises (with non-symmetric probability distribution densities) (see benchmarks 1 and 4).
- FC based on traditional SC approach with given by a user FNN structure and FNN tuning by error back propagation algorithm are not robust in cases of globally unstable or essentially non-linear CO in the presence of Rayleigh noises (with non-symmetric probability distribution densities) and Gaussian noises (see benchmarks 1 and 4).
- In the case of stable CO without deep nonlinearity, FC_SCO are compatible with FC_FNN and PID, but FC_SCO have more fine control quality (control error, force and thermodynamic characteristics), i.e. FC_SCO performance is more optimal.
- Finally, the developed design methods and tools demonstrate their efficiency and robustness for design intelligent control systems (ISC) of essentially nonlinear, stable and globally unstable dynamic objects performance in wide range of plant's parameters, reference signals, and external disturbances.

Conclusions

In this manuscript we have demonstrated the possibilities of advanced soft computing approach for design of robust and smart fuzzy controllers. We have shown the efficiency of this approach by using different benchmarks in the form of essentially nonlinear dynamic systems motion under different kinds of stochastic excitation.

The peculiarity of our non traditional SC-based approach to a smart control systems design consists of the following:

1) For optimal robust control design, we use primarily our stochastic simulation system based on mathematical model of control object, stochastic noise modeler and GA optimization subsystem. For stochastic noises generation we apply methods of nonlinear forming filters by using Fokker-Plank-Kolmogorov equations.

2) If it is difficult to define mathematical model of control object, experimental data based on physical model of control object can be used. In this case we consider given experimental data as a teaching signal for SC Optimizer.

3) We use GA-based optimization of control laws of PID-controller. As optimization criteria, we use a principle of minimum of generalized entropy production and a minimum of control error. Thus, GA fitness function (evaluating a control quality) represented as a vector function consisting of a generalized entropy production and a control error. We calculate entropy production and its rate through the parameters of the developed model.

4) The integration of the equations of mechanical model motion and the equations of entropy production rate enables us to find optimal and robust control laws.

5) We use SC Optimizer-based approximation of control laws obtained by GA and extract optimal Knowledge Base of a Fuzzy Controller.

6) Obtained Knowledge Base is used then for a robust Fuzzy Control of a given control object.

SC Optimizer has the following basic peculiarities:

1) It uses the chain of GAs to solve optimization problems connected with the optimal choice of number of membership functions (MFs) for input variables values description, their shapes and parameters and with optimal choice of fuzzy rules;

2) To design GA fitness functions we use an information-thermodynamic approach based on the analysis of dynamic behavior of control object and FC

3) SC Optimizer works as a universal approximator, which extracts information from simulated (or measured) data about the modeled system. SC Optimizer guarantees the robustness of FC, i.e. successful control performance in wide range of plant's parameters, reference signals, and external disturbances.

The developed design methods and tools are considered as effective tools for design robust intelligent control systems (ICS) of essentially nonlinear, stable and globally unstable dynamic objects. The developed design methods and tools are protected by patents given in references. In beginning of our paper we talked about difficulties of control methods in complex dynamic systems. Complex dynamic systems are characterized by uncertain model, a high degree of nonlinearity, instability, distributed sensors and actuators, high level of noise, abrupt changes in dynamics and so on. As a result, the reliability of control systems is decreased. The degree to which a control system deals successfully with above difficulties depends on the level of *intelligence* of control system. We showed that our tools allows to design robust ICS which can perform successfully a given (local) control task.

Developed tools for design of ICS serve our ultimate goal: design of ***integrated intelligent control*** systems which can realize a hybrid intelligent control of autonomous machine. Integrated ICS may be considered as interrelated corporation of local ICS (with internal local control tasks) with a top layer of intelligent automatic control. The top layer would organize local control processes performed in lower layers.

References

[1] **Zadeh L.A.** (1975) The concept of a linguistic variable and its application to approximate reasoning, *Information Sciences*, 8, 199-249; 9, 43-80.

[2] **Kosko B.** (1992) Fuzzy Systems as Universal Approximators, *Proc. Intern.Conf. on Fuzzy Systems*, San Diego, CA, March 1992, 1143-1162; Fuzzy Systems as Universal Approximators, *IEEE Trans. on Computers*, Vol. **43**, No 11, 1329-1333, 1994; Additive fuzzt systems: from function approximation to learning, In: Chen C.H. (ed.), *Fuzzy Logic and Neural Network Handbook*, McGraw-Hill, N.Y., 9-1-9-22, 1996.

[3] **Wang L.-X.** (1992) Fuzzy Systems are Universal Approximators, *Proc. of Second IEEE Intern. Conf. on Fuzzy Systems*, San Diego, CA, March 1992, 1163-1170; Stable adaptive fuzzy control of nonlinear systems, *IEEE Trans. Fuzzy Systems*, Vol. **1**, 146-155, 1993

[4] **Buckley J.J (**1993) Universal Fuzzy Controllers, *Automatica*, Vol. **28**, 1245-1248, 1992; Sugeno type Controllers are Universal Controllers, *Fuzzy Sets and Systems*, Vol.**53**, 299-303, 1993.

[5] **Litvintseva L.V. and Ulyanov S.V.** (2000) Artificial Intelligence applied to design of intelligent systems (a Soft Computing Approach), *Lectures Notes***,** Vol.38, Note del Polo (Ricerca), Universita degli Studi di Milano, 2000, 134p.

[6] **S.A.Panfilov, L.V. Litvintseva, I.S Ulyanov, K. Takahashi, S.V. Ulyanov** (2003) Soft computing optimizer for intelligent control systems design: the structure and applications**,** Proc. Intern. Conference on Computer, Communication and Control Technologies (CCCT'03), July 31 - August 1-2, Orlando, Florida, USA, 2003.

[7] **S.V. Ulyanov, K. Takahashi, S.A. Panfilov, L.V. Litvintseva, I.S. Ulyanov, I. Kurawaki, T. Hagiwara and A.V. Yazenin** (2003) Design of robust intelligent control systems based on soft and quantum computing: From 1965 with our teacher B.N. Petrov, In: Proc. Conference on Control System Design (*Dedicated to the memory of B.N. Petrov*), Moscow, 11 March, 2003, *Sci. Publ.*, 2003, pp. 34-35.

[8] **US** *Patent* N 6,411,944 B1, 1997, **Self-organizing control system** (*Inventor*: S.V.Ulyanov)

[9] **US** *Patent* N 6,415,272 B1, 1998, **System for intelligent control based on soft computing** (*Inventor*: S.V. Ulyanov)

[10] **Dubois D. and Prade H.** (1980) Fuzzy Sets and Systems and Applications, Academic Press, New York.; (1987) Fuzzy numbers: an overview, In: Bezdek JC (Ed) *Analysis of Fuzzy Information*, Vol 1. CRC Press, 3-39.

[11] **Klir G.J.** and **Folger T.A.** (1996) *Fuzzy sets, uncertainty and information*, Prentice-Hall, Englewood Cliffs, New Jersey.

[12] **Tanaka K.** (1991) *An Introduction to Fuzzy Logic for practical Applications*, Springer Verlag.

[13] *Fuzzy systems. Modeling and Control*. (Eds. H.T. **Nguen and M.Sugeno**) (1998) Kluwer Academic Publ.

[14] **Wang Li-Xin** (1994) *Adaptive fuzzy systems and control*. PTR Prentice Hall, Englewood Cliffs, NJ.

[14a] **Batyrshin I. and Kaynak O.** (1999) Parametric Classes of Generalized Conjunction and Disjunction Operations for Fuzzy Modeling, *IEEE Transactions on Fuzzy Systems*, Vol. 7, N 5, 586-597

[15] **Jang J-S.R. , Sun C-T., Mizutani E**. (1996) *Neuro-Fuzzy Modeling and Soft Computing (A computational approach to learning and machine intelligence*), Matlab Curriculum series.
[16] **Holland J.H.** (1975) *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor.
[17] **Goldberg D.E.** (1989) *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, Reading, MA.
[18] **S.A. Panfilov**, **L.V. Litvintseva, S.V. Ulyanov, K.Takahashi and A.V.Yazenin** (2002) The stochastic simulation system of robust fuzzy control of essentially non-linear dynamic systems based on soft computing, **Proc. ICAFS'2002**, Milan, Italy, September 17-18, 2002, pp. 69-80.
[19] **T.Hagiwara, S.A.Panfilov, S.V.Ulyanov and I.Kurawaki** (2000) Intelligent robust control of suspension system based on soft computing. **Proc. ICAFS'2000**, June 27-29, Siegen, Germany, 2000, pp.180-187.
[20] **Van Rooij J.F, Jain L.C. and Johnson R.P.** (1996) *Neural Network Training Using Genetic Algorithms*, series "Machine Perception, Artificial intelligence", Vol.26, World Sceintific.
[21] **Kasabov N.K.** (1996) *Foundations of Neural Networks, Fuzzy Systems, and Knowledge Engineering*, MIT Press, Cambridge, Massachusetts
[22] **Anil K.J,** Jianchang M., Mohiuddin K.M. (1996) Artificial Neural Networks: A Tutorial, *Computer*, March-96.
[23] **Wasserman P.** (1993) *Advanced Methods in Neural Computing*. Van Nostrand Reinhold, New York.
[24] **Patterson D.W**. (1996), Artificial neural networks, theory and applications, Prentice Hall, Singapore.
[25] **B.N. Petrov, G.M. Ulanov, I.I. Goldenblat and S.V. Ulyanov** (1977) *Models of control processes: Information-thermodynamics approach*, Moscow, *Sci. Publ*, 1978; *Information-semantic problems in control and organization systems*, *ibid*, 1977.
[26] **S.V. Ulyanov, K. Yamafuji, V.S. Ulyanov, S.A. Panfilov,** *et al*. (1999) Computational intelligence for robust control algorithms of complex dynamic systems with minimum entropy production. Part1: Simulation of entropy-like dynamic behavior and Lyapunov stability, *J. of Advanced Computational Intelligence*, 1999, Vol.3, No.2, pp. 82-98.
[27] **S.V. Ulyanov, M. Feng, K. Yamafuji, I. Kurawaki** *et all* (1998) Stochastic analysis of time-variant nonlinear systems. Pts 1, 2, *Intern. J. Probabilistic Engineering Mechanics*, 1998, Vol. 13, No 3, pp. 183-226.

[28] **L.V. Litvintseva, K. Takahashi, T. Hagiwara, I.S. Ulyanov, Sh. Fujii and A.V. Yazenin** (2003**)** Robust Smart Control of Essentially Non-linear Dynamic Systems Based on Soft Computing Optimizer: Stochastic and Fuzzy Simulation, Proc. ICSCCW-2003, 2nd Intern.Conf. on Soft Computing with Words in System Analysis, Decision and Control, September 9-11,Antalya,Turkey,2003, pp.91-104.
[29] **Panfilov S.A., Litvintseva L.V.,Ulyanov I.S., Takahashi K.** (2003) Soft Computing Optimizer: Structure and SW for simulation of Universal Fuzzy Approximatots, Proc. ICSCCW-2003, 2nd Intern.Conf. on Soft Computing with Words in System Analysis, Decision and Control, Sept. 9-11,Antalya,Turkey,2003, pp.112-123.
[30] http://eu.st.com/stonline/index.shtml

Appendix

A1. Stochastic simulation theoretical backgrounds. Noise Modeler

Before the introduction of forming filters structures used for random noises generation in our stochastic simulation system [27] let us refresh main concepts in random processes theory.

Consider stochastic (or random) process $x(t)$ where random values "$x = X$" constitute a set of real numbers. In Fig.A1 an example of some stochastic process $x(t)$ at an observation period of time and its stochastic characteristics are shown.

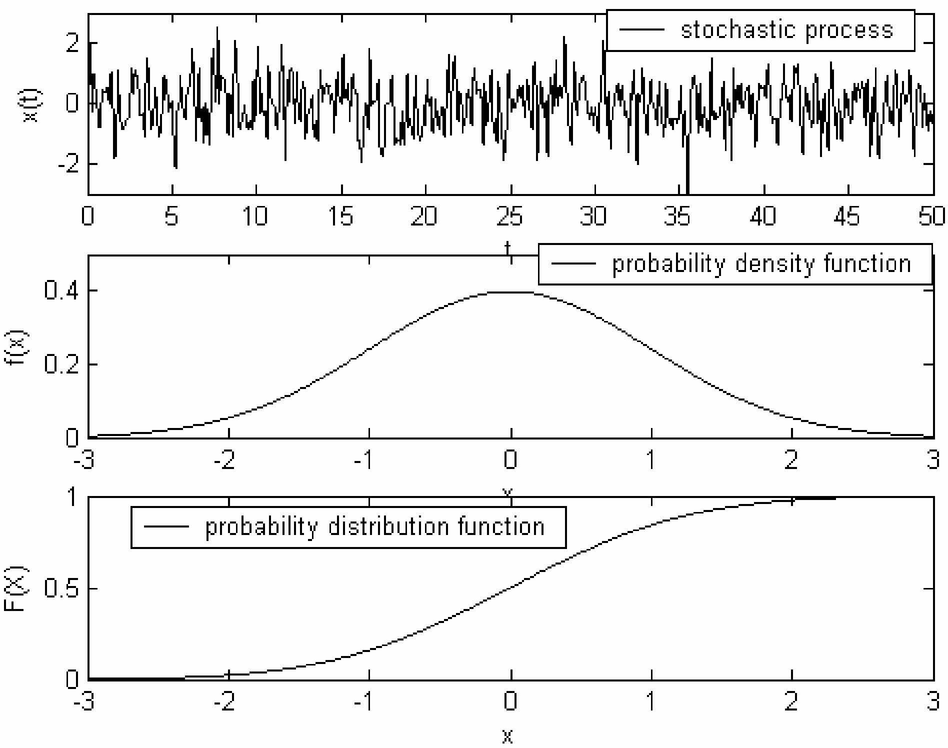


Figure A1. Example of random process (Gaussian) and its stochastic characteristics

Remind the following basic definitions.

Definition 1. The distribution of any real numerical random variable x is uniquely described by its *probability distribution function*

$$F(X) \equiv P[x \le X],$$

where X is a real number.

Random variable x is a continuous random variable if and only if its probability distribution function is continuous.

For description of $a < x \le b$ we have:

$$P[a < x \le b] = F(b) - F(a) .$$

Definition 2. *The probability density function* of random variable *x* is defined as

$$f(X) = \lim_{\Delta x \to 0} \frac{P[X < x \le X + \Delta x]}{\Delta x} \equiv \frac{dF}{dX} .$$

.

$$F(X) = \int_{-\infty}^{X} f(x)dx \ , \quad P[a < x \le b] = F(b) - F(a) = \int_{a}^{b} f(x)dx \text{, and } \int_{-\infty}^{\infty} f(x)dx = 1$$

Remark. The values $x = X$ of a random variable *x* will be denoted simply by *x* whenever this notation does not lead to ambiguities.

Definition 3. *Mean value* (or mathematical expectation) E of random variable x is defined as

$$E\{x\} = \overline{x} = m_x = \int_{-\infty}^{\infty} x \cdot f(x)dx ,$$

a *variance* $\mathrm{var}\{x\}$ of random variable x is defined as

$$\mathrm{var}\{x\} = \sigma^2 = E\{(x - m_x)^2\} ,$$

and *mean value* E of a function $y(x)$ is defined as

$$E\{y(x)\} = \int_{-\infty}^{\infty} y(x) f(x)dx .$$

Definition 4. *Random process* $x(t)$ is described uniquely by its *probability distribution function*

$$F(t, X) \equiv P[x(t) \le X] = \int_{-\infty}^{X} f(t, x)dx ,$$

where $f(t, X) = \lim_{\Delta x \to 0} \frac{P[X < x(t) \le X + \Delta x]}{\Delta x}$.

In general, random process $x(t)$ can be described as

$$x(t) = \overline{x}(t) + \xi(t) ,$$

where $\overline{x}(t)$ is a mean value of a random process *x* and $\xi(t)$ is some fluctuation component with $E(\xi(t)) = \overline{\xi} = 0$.

Definition 5. *Auto-correlation function* $R_x(t, \tau)$ of random process $x(t)$ is

$$R(t, \tau) = E[X(t - \tau)X(t)] = \int_{0}^{\tau} x(t - \tau)x(\tau) f(x(\tau))dt ,$$

where $\tau = t_2 - t_1$ and t_1, t_2 are some time moments, and

a *correlation time* of a random process $x(t)$ is defined as

$$t_c = \frac{2}{\sigma^2} \int_0^\infty R^2(\tau) d\tau .$$

Semantic meaning of auto-correlation function is that it describes a mean value of joint distribution $P[x(t_1) \le X_1, x(t_2) \le X_2]$, i.e. having value $x(t_1)$ and knowing $P[x(t_1) \le X_1, x(t_2) \le X_2]$ we can predict value of x at time t_2.

Definition 6. *Spectral density* of a random process $x(t)$ is defined as a function S_x so that

$$E\{(x - m_x)^2\} = \int_{-\infty}^{\infty} S_x(\omega) d\omega .$$

Stationary stochastic processes

We will consider *stationary stochastic processes*. Random process is called stationary one if all his statistical characteristics are invariant relative to temporal shift on any value τ, i.e.

$$F(x(t+\tau)) = F(x(t)) .$$

In this case, the joint probability density function $f(x(t_1), x(t_2))$ does not depend on time points and depends only on time interval $\tau = t_2 - t_1$.

Important properties of stationary random processes

Stationary random processes have the following important properties:

1) $\bar{x}, \sigma$ don't depend from a time, i. e.

$$\bar{x}(t) = \bar{x}(0) = m_x ; \sigma(t) = \sigma(0) = \sigma .$$

2) Correlation function $R(t, \tau)$ depends only from τ.

3) For stationary random processes there are following *relationships between auto-correlation and spectral density functions*:

$$R(\tau) = \int_{-\infty}^{\infty} S_x(\omega) e^{i\omega\tau} d\omega \quad \text{and} \quad S_x(\omega) = \frac{1}{2\pi} \int_{-\infty}^{\infty} R(\tau) e^{-i\omega\tau} d\tau .$$

Markov processes

Important class of random processes constitutes a so called *Markov processes*. Describe some random process $x(t)$ at $t_0, t_1, ..., t_m$ time moments as follows:

$$x(t_1) = F[x_0(t_0); \xi(\theta); t_0 \le \theta \le t_1], \ \dots,$$

$$x_m = x(t_m) = F[x_{m-1}, x_{m-2}, ..., x_1; \xi(\theta); t_{m-1} \le \theta \le t_m].$$

By definition, for Markov processes we have:

$$x(t_m) = F[x_{m-1}; \xi(\theta); t_{m-1} \le \theta \le t_m],$$

i.e. the value of x_m depends only from value x_{m-1} and doesn't depend from other values (a history of the process). As a consequence, the following property is also true:

$$f(x_m \,|\, x_{m-1}, x_{m-2}, ..., x_1) = f(x_m \,|\, x_{m-1}).$$

Stochastic simulation of random processes with required statistical characteristics based on forming filters: FPK equations approach

Typical random noises

We will consider two main classes of random noises (as random processes): a so called *Gaussian noises* and *non-Gaussian noises.* They are described with following probability density functions.

Gaussian noises

1. *Gaussian noises* are described by a normal (Gaussian) random distribution with the following symmetric probability density function:

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} \mathrm{e}^{-\frac{1}{2}\left(\frac{x-m_x}{\sigma}\right)^2}, \; -\infty \le x < \infty .$$

Spectral density and auto-correlation functions for Gaussian noises are:

$$S_x(\omega) = S_0 e^{-\omega^2/2\Delta^2} \;;\; R(\tau) = \sigma^2 e^{-\tau^2\Delta^2/2} \text{ and } \sigma^2 = S_0 \Delta\sqrt{2\pi} .$$

Example of Gaussian random process is shown in Fig.A-1.

1.1 *White noises* represent a special subclass of Gaussian noises.
Random process $x(t)$ with constant $S_x(\omega) = S_0 (-\infty \le \omega \le \infty)$ is called "white noise". White noise is described by the auto-correlation function

$$R(\tau) = S_0 \delta(\tau) \; \left(\int_0^\infty \delta(\tau) d\tau = 1 \right),$$

where $\delta(\tau)$ is so called Dirac function.

Continuous white noise has a correlation time = 0 and $E(x - m_x)^2 = \int_{-\infty}^{\infty} S(\omega) d\omega \to \infty$. It means that physically it is non possible to realize this noise. But white noise is a useful theoretical approximation when the noise disturbance has a correlation time that is very small relative to the natural bandwidth of a system. We will use white noise in our forming filters for noise generation.

Remark. In Simulink, white noise is generated by using special block with the following noise parameters: *noise power* is the height (S_0) of the spectral density function of the white noise (the default value is 0.1); *sample time* is the correlation time of the noise (the default value is 0.1) and s*eed* is the starting seed for the random number generator (the default value is 23341).
Example of white noise with noise power =1 and sample time = 0.01 is shown in Fig.A-2.

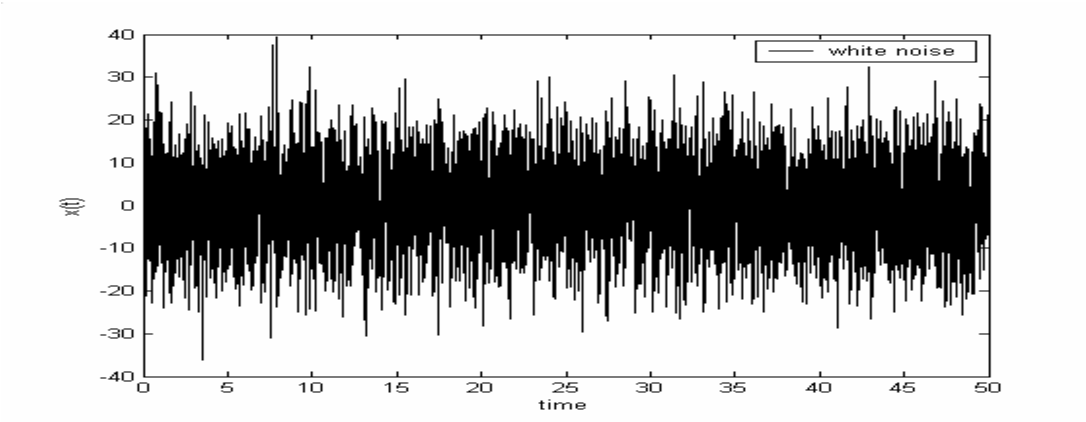


Figure A-2. Example of white noise

Non-Gaussian noises

2. *Non-Gaussian noises* are described by the following random distributions.
2.1 *Uniformly distribution* with the following probability density function:

$$f(x) = \frac{1}{2\Delta}, -\Delta \le x \le \Delta .$$

2.2. *Rayleigh distribution* with probability density function as follows (unsymmetrical probability density):

$$f(x) = \gamma^2 x e^{-\gamma x}, \gamma > 0, 0 \le x < \infty .$$

Auto-correlation and spectral density functions for Rayleigh noises are:

$$R(\tau) = \sigma^2 e^{-\alpha|\tau|}, \qquad S_x(\omega) = \frac{\sigma^2}{\pi} \frac{\alpha}{\alpha^2 + \omega^2} .$$

Example of Rayleigh random process is shown in Fig.A-3.

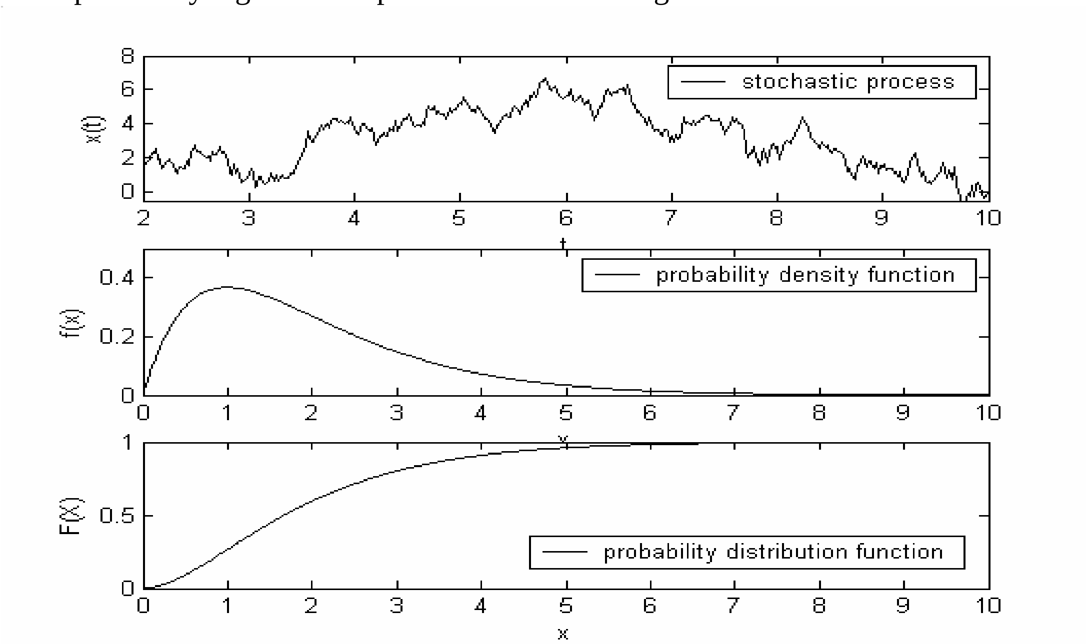


Figure A-3. Example of Rayleigh random process and its stochastic characteristics

Fokker-Planck-Kolmogorov (FPK) *stochastic differential equation*

In general, random processes obey to *Fokker-Planck-Kolmogorov* (FPK) *stochastic differential equation*:

$$\frac{\partial f(x)}{\partial t} = -\frac{\partial}{\partial x}\left[A(x) f(x)\right] + \frac{1}{2}\frac{\partial^2}{\partial x^2}\left[B(x) f(x)\right],$$

where $A(x)$ and $B(x)$ are known as drift and the diffusion coefficients, respectively. Below we consider special cases of this equation.

Stochastic noises simulation based on Forming Filter. Design of Forming Filter Structures based on FPK equation

Consider a stationary stochastic process $x(t)$ defined on the interval $\left[x_l, x_r\right]$, which can be either bounded or unbounded. With the knowledge of the probability density $f(x)$ and the spectral density $S_x(\omega)$ (or auto-correlation function) of $x(t)$, we wish to establish a procedure to model the random process $x(t)$.

Describe the methodology of forming filter structure according to given type of auto-correlation functions

$$R(\tau) = \sigma^2 \mathrm{e}^{-\alpha|\tau|} \tag{3-1}$$

and different probability density functions.

If $x(t)$ is also a diffusive Markov process, then it is governed by the following *Ito stochastic differential equation*:

$$dx = -\alpha x dt + D(x) dB(t) \tag{3-2}$$

where α is the same parameter as in Eq. (3-1), $B(t)$ is a unit Wiener process, and the coefficients $-\alpha x$ and $D(x)$ are drift and the diffusion coefficients, respectively [].

The Ito type stochastic differential equation (3-2) may be converted to the simplified version called as Stratonovich type differential equation as follows:

$$\dot{x} = -\alpha x + D(x)\frac{\sigma}{\sqrt{2\pi}}\xi(t), \tag{3-2a}$$

where $\xi(t)$ is a *white noise* with a *unit spectral density*.

Function $D(x)$ in Eq.(3-2 and 3-2a) is calculated as [27]:

$$D^2(x) = -\frac{2\alpha}{f(x)}\int_{x_l}^{x_r} u f(u) du\,. \tag{3-3}$$

Solutions of Eq. (3-2 and 3-2a) give us the random process $x(t)$ which we want to generate.

Consider typical random noises forming filter design task. Given the probability density $f(x)$ and auto-correlation function $R(\tau)$ of stochastic process $x(t)$, obtain dynamic behavior of random process $x(t)$.

In Fig.A-4 the scheme of $x(t)$ generation based on Eq.(3-2a) is shown.

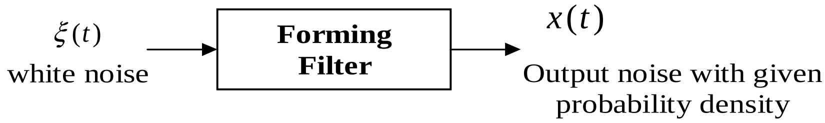


Figure A-4. Forming Filter for random noise generation

Forming filters examples

Example 1. Assume that $x(t)$ is uniformly distributed random process with

$$f(x) = \frac{1}{2\Delta}, -\Delta \le x \le \Delta . \tag{3-4}$$

Substituting Eq.(3-4) into Eq.(3-3) we have

$$D^2(x) = \alpha(\Delta^2 - x^2) . \tag{3-5}$$

In this case, the desired Ito equation is given by

$$dx = -\alpha x dt + \sqrt{\alpha(\Delta^2 - X^2)} dB(t) . \tag{3-6}$$

Corresponding equation for $x(t)$ in the Stratonovich form is

$$\dot{x} + \alpha x = \frac{\sigma}{\sqrt{2\pi}} \sqrt{\alpha(\Delta^2 - x^2)}\ \xi(t) . \tag{3-6a}$$

A solution of Eq.(3-6 and 3-6a) gives us uniformly distributed random process $x(t)$.

Example 2. Let $x(t)$ be governed by a Rayleigh distribution

$$f(x) = \gamma^2 x \mathrm{e}^{-\gamma x}, \quad \gamma > 0, 0 \le x < \infty .$$

Its centralized version $y(t) = x(t) - 2/\gamma$ has a probability density

$$f(y) = \gamma(\gamma y + 2) \mathrm{e}^{-\gamma y + 2}, \quad -2/\gamma \le y < \infty .$$

From Eq.(3-3), we define

$$D^2(y) = \frac{2\alpha}{\gamma}(y + \frac{2}{\gamma}) . \tag{3-7}$$

The Ito equation for $y(t)$ in this case is

$$dy = -\alpha y dt + \left[\frac{2\alpha}{\gamma}(y + \frac{2}{\gamma}) \right]^{1/2} dB(t) \tag{3-8}$$

and the corresponding equation for $y(t)$ in the Stratonovich form is

$$\dot{y} + \alpha y = \frac{\sigma}{\sqrt{2\pi}} \sqrt{\frac{2\alpha}{\gamma}(y + \frac{2}{\gamma})}\ \xi(t) \tag{3-8a}$$

A solution of Eq.(3-8 and 3-8a) gives us random process $y(t)$ governed by a Rayleigh distribution.

Example 3. Let $y(t)$ be governed by a Gaussian distribution. For Gaussian distributions coefficient $D^2(y) = 2\pi\sigma^2$. So, the corresponding equation for $y(t)$ in the Stratonovich form is

$$\dot{y} + \alpha y = \sigma^2 \xi(t).$$

Finally, we can describe typical structures of forming filters as shown in Table A-1.

Table A-1: The Structures of Forming Filters for Typical Probability Density Functions

Auto-correlation function	**Probability density function**	**Forming filter structure**
$R(\tau) = \sigma^2 e^{-\tau^2\Delta^2/2}$	*Gaussian* $f(y) = \frac{1}{\sqrt{2\pi}\sigma} \mathrm{e}^{-\frac{1}{2}\left(\frac{y-m_y}{\sigma}\right)^2}$	$\dot{y} + \alpha y = \sigma^2 \xi(t)$
$R(\tau) = \sigma^2 \, \mathrm{e}^{-\alpha\lvert\tau\rvert}$	*Uniform* $f(y) = \frac{1}{2\Delta}, -\Delta \le y \le \Delta$	$\dot{y} + \frac{\alpha}{2} y = \frac{\sigma}{\sqrt{2\pi}} \sqrt{\alpha(\Delta^2 - y^2)} \, \xi(t)$
$R(\tau) = \sigma^2 \, \mathrm{e}^{-\alpha\lvert\tau\rvert}$	*Rayleigh* $f(y) = \gamma(\gamma y + 2) e^{-\gamma y + 2}$ $-\frac{2}{\gamma} \le y \le \infty$	$\dot{y} + \alpha y = \frac{\sigma}{\sqrt{2\pi}} \sqrt{\frac{2\alpha}{\gamma}(y + \frac{2}{\gamma})} \, \xi(t)$