Amplitude-Frequency Characteristic of a Neural Control Based DC Drive

S.A.Sergeev, V.B.Klepikov, K.V.Mahotilo[1]

Abstract

The paper interprets characteristics of a neural-control-based DC servodrive in terms of the classical theory of automatic control. It also touches on the problem of choosing training patterns to synthesize a nonlinear PID-controller with a desired amplitude-frequency characteristic and analyses the efficiency of using for this purpose input signals in form of a step function and a harmonic one. Synthesis of the neurocontroller has been performed within the framework of a three-layer perceptron. To train it, a genetic algorithm has been developed.

Introduction. Classical and evolutionary paradigms

The 20-th century has proved the classical (Newtonian) paradigm to be a most fundamental one in technical sciences. Created as a powerful means of analysing physical phenomena, it acted as a magnifying glass which lets the mankind approach the essence of the surrounding world. Having developed differential and integral calculus, scientists got a universal tool to mathematically describe, in the most general form, laws of parameter changes in a process under investigation.

Also, the 20-th century has witnessed the appearance of a new, so called evolutionary paradigm. Intensive biological research succeeded in lifting the veil of mystery over heredity mechanisms, principles of functioning higher organisms central nervous system and cerebrum, stimulated attempts to simulate an evolution of artificial technical object populations and promoted developing new devices on the base of artificial neural networks (ANNs).

The new are always brought in through fighting with the old, the latter being swept aside. This statement is quite easy to prove analysing the current situation in the automatic control. ANNs keep on further conquering problems traditionally solved by the classical, Newton-paradigm-based theory. Evolutionary methods utterly suffice for ability to solve these problems in their own manner via non-typical sight on the problems. Nevertheless, we should admit that the classical theory has worked out numerous valuable abstractions and techniques (for instance, the conception of dynamic links, frequency methods of analysis and synthesis, etc.) which remain of the same value in the framework of the evolutionary paradigm.

[1] Kharkov State Polytechnic University, 21 Frunze St., 310002, Kharkov, Ukraine
E-mail: sergeev@lotus.kpi.kharkov.ua

What we have tried to do is to apply classical control theory methods (studying frequency characteristics) to analysing a neurocontroller synthesized through new evolutionary paradigm methods.

Amplitude frequency characteristic

Amplitude-frequency characteristic (AFC) is suitable for its containing requirements as to transient processes, energy limitations and criteria of optimal control system operation as a whole.

Two following circumstances which affect AFC shape are usually taken into account.

1.According to Kesler, a system tuned to a magnitude optimum (providing for the compromise between the over-control and the transient process duration) must be a low-pass filter.

2.The higher the input signal frequency as compared to the natural dynamical object frequency, the larger the resources required to work it out. The resources being always limited, it is desired that the controller filter out signals which break the normal system operation.

Spectrum of problems

The problem of training a neural network in the continuous (not a discrete) input signal spectrum differs from the classical problem of recognizing handwritten figures due to unclear requirements as to the training pattern set. Unlike the latter case when the network must be capable of telling the known samples, in the first case it must learn to interpolate the output signal in the frequency band where training has not occurred. So such questions as how huge the training pattern set must be and what the sample distribution along operating frequencies and amplitudes are like remain existing and cannot be answered *a priori*.

Our research aims to test whether a neurocontroller, which actually comprises no timing circuits (RL-RC), is capable of implementing a given AFC and also to determine to what extent additionally training the neurocontroller may influence the network architecture.

Neurocontroller training scheme

We have chosen to investigate quite a known scheme of a drive with a neurocontroller in the direct control channel (see fig.1). The dynamical object (in our case, it is a DC motor) and the neurocontroller are variable x_1 feedbacked. The neurocontroller training is performed via a genetic algorithm (GA) through the mismatch between the output of the object and that of the reference model with a given AFC.

Learning was accomplished at fixed input signal frequencies of 2, 4, 6, 8, and 10 Hz and also with the step function.

So that the neurocontroller could be able to work not only at a fixed input signal amplitude but within an amplitude band, the pattern amplitudes changed relative to the frequencies. For example, if the step amplitude was 0.3, the amplitude of 10 Hz pattern was increased up to 2.

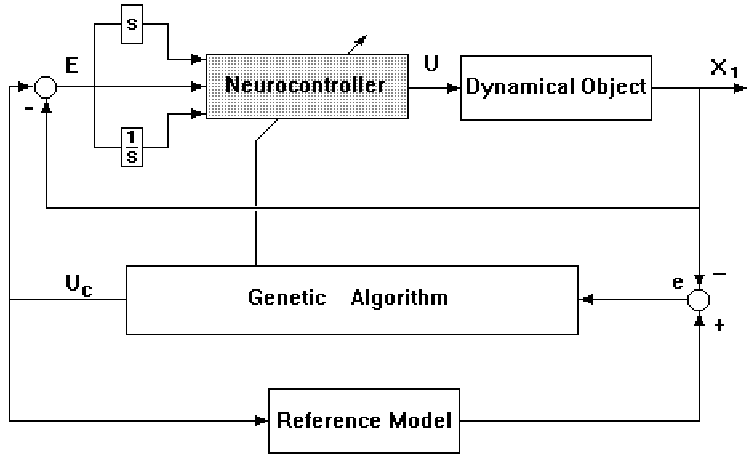

Fig.1. The scheme of training the neurocontroller

The cumulative training error was calculated by summing up the mean-square differences between the object output x_1 and the reference model output over the used patterns. We should note that even though we seem not to be touching on analysing phase-frequency characteristics, the problem of minimizing the training error implies minimization of the phase shift between the object and the reference model outputs.

Dynamical object and reference model

Being a bit experienced in numerical experimentation on ANN-based drive, we have chosen the same motor model as in [1], namely

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = \left(-2T\varsigma x_2 - x_1 + kU\right)/\sqrt{T} \end{cases} \tag{1}$$

where x_1 and x_2 are, respectively, the rotor angular speed and the armature current. The parameter values $k = 1$, $T = 0.5$ and $\varsigma = 0.1$ have been chosen so that oscillating properties of the dynamical object would be distinct.

Figure 2 shows the dynamical object AFC. Its natural resonance frequency is about 2 Hz (curve 1). Just after this point the oscillation amplitude sharply decreases to reach 0.12 at frequency of 6 Hz.

Now let's set more strict requirements as to the drive. The discontinuity point at the reference model AFC is chosen to be quite far from the resonance frequency, namely, at about 6 Hz (curve 2). Thus, the neurocontroller task is reduced to attenuating the object oscillations in zone I and amplifying them in zone II.

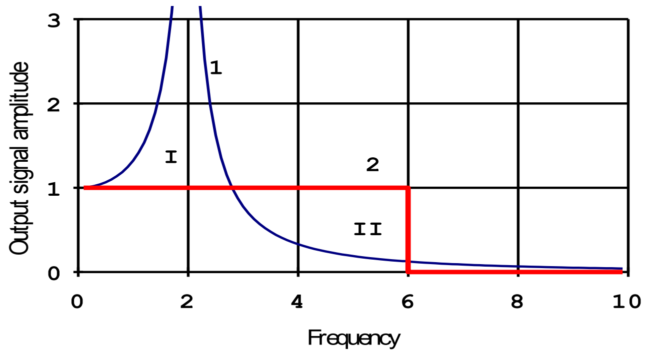

Fig.2 AFC of the dynamical object (curve 1) and that of the reference model (curve 2)

Network architecture

Having at hand the results of synthesizing a neurocontroller only through step-function input signals of different amplitude [1], we checked up on the neurocontroller dynamical properties within frequency band [0, 10 Hz]. The results of the simulation are shown in fig.3. As one can see, despite the neurocontroller treating the step quite well, its AFC is far from being satisfactory. Our tries to additionally train it through harmonic signals without changing the network architecture failed. The 4 hidden-layer neurones may have appeared to be too small number for such a complicated task.

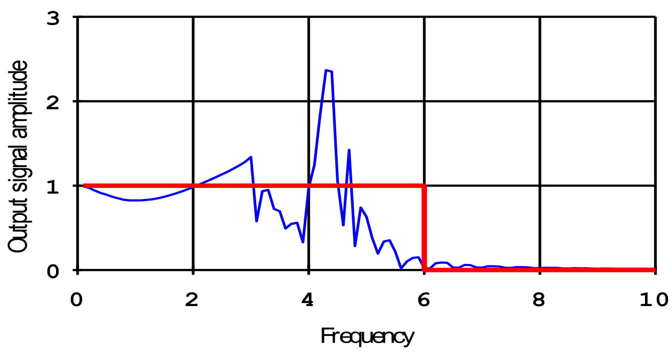

Fig.3. AFC of the dynamical object with the step-function-signal-trained neurocontroller

It made us enlarge the hidden-layer size up to 10 neurones with sigmoidal activation functions. Now the general network configuration is following: a three-layer perceptron 3-10-1. The network input receives an error signal, its derivative and integral. The network output serves as control signal U. Synaptic weights initialization is performed in the interval of [-10,10].

GA as a training procedure

GAs are traditionally regarded as something opposing to earlier-developed network-training techniques, for example, BackPropagation (BP). We would like to concentrate on the other side of this competition, namely, on the points where they coincide. The both algorithms do correspond to the scheme given in fig.4. So in fact, in the design variable space there exists an apparent (GAs) or a hidden

(BP) current population of probable solutions. A mathematical model performs mapping of these points on to the criteria space. Then there goes the procedure of selecting best solutions (parents) followed by the procedure of their mating.

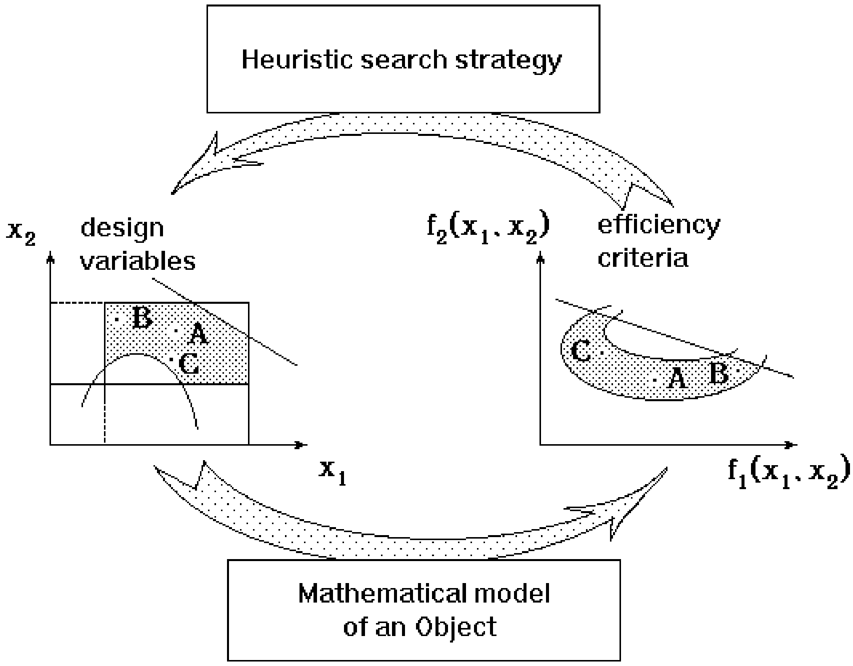

Fig.4. Cyclic structure of numerical optimization methods

Strictly speaking, such an interpretation exists for any numerical optimization method. It should be cleared up what we mean by the term of mating. In general, we use this term to name the process of generating any new point coordinates in the variable space (an offspring) through the information about the task function landscape which a current population possesses.

As far as gradient methods are concerned, a current population consists of the auxiliary points placed in the neighbourhood of the searching one which are used to calculate particular derivatives. Though it may seem that the searching point is the only one to perform the search, in fact, every new point is a collective offspring of the whole population created at a current algorithm step.

It is heuristics available for mating that differentiate GAs. GAs borrow these mechanisms, called genetic operators (mutation, crossover and inversion), from Nature. Since the mechanisms of passing genetic information on to offspring have been subject to natural selection themselves for millions of years of organic evolution, they deserve to be treated as the complete and perfect set of heuristics which have proved their efficiency in competition with spontaneously-appeared alternative mechanisms of heredity.

We want to point out two peculiarities of the GA version used here which make it different from the canonical one. First, we use a diploid representation of individuals (those interested may address to [2]) which we consider capable of significantly decreasing the population exhaustion rate. Second, we treat a parametric ANN synthesis not as a monocriterion problem but as a problem of vector criteria optimization with the criteria number equal to the size of the training pattern set. To solve such a problem, we find hierarchic approach especially suitable.

The main purpose we try to accomplish is to reduce the number of addressing to the procedure which calculates the training error over a current pattern. Replacing the criteria weighted sum by the hierarchic approach lets us do it. The introduced innovations only touch on the stage of choosing the parental group in the population.

So, first, we generate a population of N individuals and evaluate the training error over the first pattern (say, at frequency of 10 Hz for a start). Then we rank the population over the first criterion and expel from further consideration $m=kN$ individuals ($k<1$ is a coefficient dependent on the total number of criteria) which turn out to be the worst.

In the second stage we estimate the fitness of the rest N-m individuals over the second criterion (for example, under the step-function signal) and repeat ranking the formed subpopulation followed by rejecting the worst. The individuals which have survived after these two stages are considered identical over the first two criteria.

We carry on like that. We shoud note that the higher the number of criterion, the less addressing to the analysis unit is required. The subpopulation which has survived after ranking over the last criterion and rejecting the worst is considered as the parental group.

Generating an offspring is followed by checking up on its ability to squeese into the population ranked over the first criterion not closer than m individuals to the tail. If it fails, it is expelled from the further estimation, that is its second and other criteria are ignored. If it manages to shift more than m individuals towards the tail, its second criterion is calculated and the entire procedure of checking up is repeated.

Both the number of the individuals rejected after every ranking and the pattern hierarchy which determines the turn of producing the patterns are adaptive strategy parameters. The pattern which gives the largest error takes the first place in the criteria hierarchy in the next generation.

Calculation results

Fig.5 shows the results of the neurocontroller parameter optimization achieved by the 3,000-th generation of a population of 120 individuals. In each generation, 12 new offspring appeared, so during the search 6 training patterns were processed about 20,000 times. The number of calculations is impressive but we should bear in mind that any probable solution chromosome consisted of 510 bits which coded 51 network parameters.

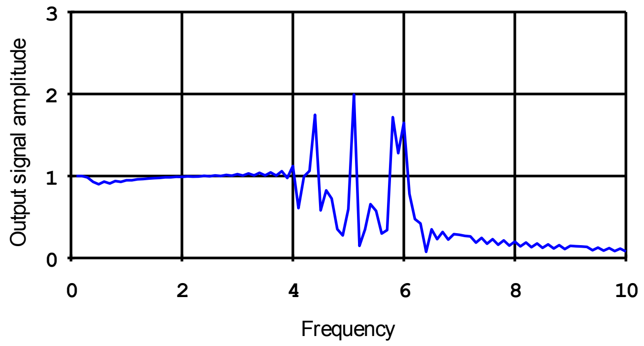

Fig.5 AFCs of the dynamical object with
the final neurocontroller version

Also, fig.5 witnesses that our initial idea about training patterns of different amplitude has carried out the task assigned to it. The neurocontroller does provide for the given AFC in quite a large band of input signal amplitudes.

The synthesized neurocontroller should not be regarded as an optimal PID-controller in the meaning which is implied by the classical theory of automatic control. It is an utterly different (in its essence) device, and naming it a PID-controller, we just point out its relative similarity to a certain type of apparatus capable of performing like controlling functions.

Conclusion

Thus, the obtained results demonstrate the neurocontroller capability of being trained to a certain behaviour within a given frequency band. Numerical calculations prove its ability both to suppress the dynamical object resonance oscillations and to extend the operating frequency band.

The developed method of choosing the parental group in the GA, which is based on hierarchic approach as to vector evaluation of individual fitness, permits to substantially decrease the number of task-function calculations and speed up the GA convergence.

Literature

1. Клепиков В.Б. и др. Искусственные нейронные сети: Новая парадигма в управлении // В: Проблемы автоматизированного электропривода. Теория, практика.- Харьков, Основа, 1995.- с.с.111-115.

2. Klepikov V.B. et al. Modification of Holland's reproductive plan for diploid populations // In: Artificial Neural Nets and Genetic Algorithms (Eds. D.Pearson et al)., Springer Verlag, 1995.- pp.337-339