[AQ: Please select a Creative Commons license type via the SAGE Open Access Portal. Please annotate the following proof with a comment stating the license type you have selected in the SAGE Open Access Portal. Please ensure that, where applicable, the responsible bill payer has paid the Article Processing Charge (APC) via the SAGE Open Access Portal for timely publication of your article.]

[GQ1: Please confirm that all author information, including names, affiliations, sequence, and contact details, is correct.]

[GQ2: Please confirm that the Funding and Conflict of Interest statements are accurate.]

[GQ3: Please note that we cannot add/amend ORCID iDs for any article at the proof stage. Following ORCID’s guidelines, the publisher can include only ORCID iDs that the authors have specifically validated for each manuscript prior to official acceptance for publication.]

Research Article

DOI: 10.1177/16878132231223387

Advances in Mechanical Engineering

Yang et al.

1Department of Automation, Tsinghua University, Beijing, China

2Beijing Aerospace Automatic Control Institute, Beijing, China

**Corresponding author:**

Qing Li, Department of Automation, Tsinghua University, Room 605, Main-Building, Beijing 100084, China.

Email: liqing@tsinghua.edu.cn

A controller design method based on the integration of structure search and parameter optimization

Kan Yang1**[INSERT ORCID iD LOGO]**, Zhaolei Wang2, Pengpeng Guo1 and Qing Li1**[GQ: 1]**

Abstract

For the controller structure design problem, a controller design method based on the integration of structure search and parameter optimization is proposed with the idea of ENAS, which automates the controller structure design work, and searches for the most suitable controller structure and completes the parameter optimization in a relatively short time. The simulation study with the galvano scanner as the controlled plant shows that the number of iterations and the consumption time of the method are only 15% of the traditional method in achieving the same performance compared with the traditional design method.

Keywords

Structure search, integrated design, controller, ENAS, parameter optimization

Date received: 20 June 2023; accepted: 11 December 2023

Handling Editor: Chenhui Liang

Introduction

After years of theoretical research and engineering practice, a variety of structures and forms of control theory and its controller design methods have been created and developed. The construction of a new control system structure has been an important direction in the development of control theory and control engineering.

In actual engineering, the structure of the constructed control system will be more complex, as shown in Figure [1](#F1), which is the composition and principle diagram of a single-channel longitudinal electrotransmission manipulation system. The electrotransmission steering system is actually composed on the basis of control stabilization, eliminating the irreversible power-assisted mechanical steering channel and retaining only the channel of electrical command signal output from the driving rod via the rod force sensor. The function and control principle of the system is the same whether analog or digital. For each given aircraft there may be different system structure and functional requirements, that is, there are different control laws, but in the basic structure of the controller will be more or less the same.

[FIGURE 1 ABOUT HERE]

Figure 1.

Composition and principle of single-channel longitudinal telegraphic manipulation system.[1](#1)

Therefore, for a control system used in engineering, its topology has a meaning and there exists a clear physical meaning and logical relationship. The development of traditional control system topology is also based on the continuous deepening of the mechanism of the controlled plant and control theory.

Most of the current control system optimization algorithms are for the optimization of control system parameters, control system structure optimization, or mainly rely on human intelligence and judgment.

Zoph and Le proposed a Neural Architecture Search (NAS) method to find a suitable neural network structure in 2017.[2](#2) They used an RNN generator sampling to obtain a new neural network structure. The data is trained under this neural network structure and the accuracy on the corresponding validation set is obtained. This accuracy is used to characterize the performance of the neural network obtained from this search, which is then used as a reward function to train the RNN generator. The conceptual structure is shown in Figure [2](#F2).

[FIGURE 2 ABOUT HERE]

Figure 2.

NAS basic framework.[2](#2)

Since NAS methods consume a lot of computational resources, Pham et al. proposed an Efficient Neural Architecture Search (ENAS) method to quickly design the structure and weights of neural networks based on NAS.[3](#3) It uses an LSTM (Long Short-Term Memory) as a generator to express the network structure as a directed acyclic graph. The directed acyclic graph has edges and nodes, where each node represents an operation and the edges represent the information flow.ENAS trains a generator to select an optimal subgraph from the directed acyclic graph (which determines which edges are activated in the directed graph), while the selected subgraph is trained using the classical Cross Entropy Loss.

Some specific applications of structure search have already emerged. Luo used NAS to compress large scale pre-trained language model and text-to-speech model by proposing or utilizing block-wise training, progressive pruning and performance approximation.[4](#4) Zhou combined Huffman coding with NAS algorithm and applied it to image segmentation task.[5](#5) Luo proposed an efficient NAS algorithm by combining Dynamical Isometry and achieve an high accuracy rate on the ImageNet classification task.[6](#6) To overcome the ENAS model generalization error problem, Ahmed et al. examined the effectiveness of a range of techniques including reducing model complexity, use of data augmentation, and use of unbalanced training sets, and achieved remarkable results on the ultrasound image classification for breast lesions.[7](#7)

Although the ENAS method was proposed for designing the structure and weights of neural networks,[3](#3) the idea of ENAS can be extended to design the structure and edge weights of any directed acyclic graph. Since the control system (CS) can be transformed into the form of a directed acyclic graph, this paper draws on the idea of ENAS to design the structure and parameters of the control system (CS) in an integrated manner, as shown in Figure [3](#F3).

[FIGURE 3 ABOUT HERE]

Figure 3.

The basic framework of the methodology in this paper.

In this paper, a controller design method based on the integration of structure search and parameter optimization is proposed to automate the controller structure design. The method introduces the idea of ENAS, which can search for the most suitable controller structure and complete parameter optimization in a relatively short time. In this paper, the galvano scanner in Maeda and Iwasaki’s paper[8](#8) is selected as the research plant to verify the effectiveness of the method and compare it with the traditional full search-based method.

Controller structure automatic generation technology

*Conversion method of control law and control system structure diagram*

The central scientific and technical problem of this section is to transform the control system structure into a directed acyclic graph structure that the LSTM can handle.

A complex system structure diagram, the connection between its boxes is bound to be intricate, but the only three basic ways to connect the boxes are series, parallel and feedback connections. Therefore, the general method of simplifying the structure diagram is to move the lead or comparison points, exchange the comparison points, and perform box operations to merge the boxes connected in series, parallel and feedback. The principle of maintaining equivalent variable relationships before and after the transformation should be followed in the simplification process. Eventually, the structure of all closed-loop control systems can be expressed in the form of Figure [4](#F4).

[FIGURE 4 ABOUT HERE]

Figure 4.

Classic FB control system block diagram.[9](#9)

where is the controller transfer function. The transfer function is set to the N-layer structure controller as shown in Figure [5](#F5) (any controller can be written in the form of N transfer functions multiplied by each other).

[FIGURE 5 ABOUT HERE]

Figure 5.

N-layer cascade structure controller block diagram.[10](#10)

Since the controller is a cascade structure, the transfer function is

Here denotes a transfer function picked from the ith layer, that is, the transfer function of the connected part of each layer in Figure [5](#F5). Each can be set to the following form.

The transfer function can be decomposed into the form of the product of N transfer functions according to the above method.

*Structural Modeling of control systems based on directed acyclic graphs*

According to the Section [2.1](#H2\.1), an arbitrary controller can be written in the form of N transfer functions multiplied by each other. Further, we can convert the cascaded structure controller shown in Figure [5](#F5) into a directed acyclic graph, as shown in Figure [6](#F6), for comparison with the graph defined by Pham et al.,[3](#3) following the following steps:

**Step 1** Define each box of the controller structure diagram as a node.

**Step 2** Which edge is activated: each edge in the controller structure graph is activated.

**Step 3** What is the operation of a node: The operation of a node is a certain controller that is selected.

[FIGURE 6 ABOUT HERE]

Figure 6.

Controller structure diagram.

Based on the above steps, the controller can be completely transformed into a directed acyclic graph. Then, the structure of this directed acyclic graph can be searched/optimized by the ENAS method.

Note that, as shown in Figure [6](#F6), we only convert the controller to a directed acyclic graph, while for the whole control system, it remains a directed cyclic graph structure.

*Method for encoding topological relations of directed acyclic graphs*

The encoding method for the topological relations of a directed acyclic graph as shown in Figure [6](#F6) is to generate an N-dimensional vector

Where and is the number of candidate structures for the i-th layer controller.

Thus, the number of all possible controller structures in the search space represented by this graph is .

For example, the whole control system is 4-layer, if the diagram code is (1 3 2 2), it means that the first structure is selected for the first layer, the third structure is selected for the second layer, the second structure is selected for the third layer, and the second structure is selected for the fourth layer, as shown in Figure [7](#F7).

[FIGURE 7 ABOUT HERE]

Figure 7.

Example of diagram coding.

*Controller design rules embedding method*

Section [2.1](#H2\.1) mentions the decomposition of the controller transfer function into the form of a product of N transfer functions. Further, each part can be considered as a filter, and the design of the controller is actually the design of N filters. When designing a circuit for a filter, it is difficult to directly implement a circuit with a transfer function of order 3 or higher. When it is necessary to design a filter greater than or equal to 3rd order, it generally takes the form of decomposing the higher-order transfer function into the product of several lower-order transfer functions. For example, to design a 5th-order filter, two 2nd-order filters and a 1st-order filter cascade can be obtained. Common filters are: low-pass filter, high-pass filter, band-pass filter, band-stop filter (notch filter), all-pass filter, etc., as shown in Table [1](#T1).

Table 1.

Transfer function of typical filters.

|  |  |
| --- | --- |
| Type |  |
| 1st-order LPF |  |
| 1st-order HPF |  |
| 1st-order APF |  |
| 2nd-order LPF |  |
| 2nd-order HPF |  |
| 2nd-order BPF |  |
| 2nd-order BSF(NF) |  |
| 2nd-order APF |  |

For the actual plant to be controlled, determining the number and type of filters, etc. requires the use of rules and a priori knowledge from the controller design domain. At this point, we can embed these rules and a priori knowledge by adjusting the parameter range. For example: by analyzing the bode plot of the controlled plant, it is found that there are two wave peaks to be handled, we can set the number of filters as 2 and the frequency selection range of the filters as the periphery of the two wave peak positions respectively; according to the a priori knowledge, the improvement of the controller performance may require band-stop filters or all-pass filters, then the filters can be designed as

The change from BSF to APF can be achieved by first fixing so that varies between and .

Automatic controller parameter optimization technique based on efficient heuristic search algorithm

*General topology study of control systems*

The structure of the controller varies significantly depending on the control theory on which the controller is based, and it is often very complex in engineering because of the need to solve various characteristic problems and constraint problems of the controlled plant. Based on this, we adopted the expression of a cascade controller. After analysis, it is believed that the classical multi-loop control systems and control laws can be basically transformed into this form.

In addition, for the actual control system, there are multiple high-frequency resonant mode disturbances in addition to the nonlinear characteristics. Therefore, when designing a controller, a basic controller is generally designed for completing the stability control, and then multiple filters are designed for each resonant mode to improve the overall performance of the controller. As shown in Figure [8](#F8). The basic controller can use PID controller, robust controller, LQR controller, sliding mode controller, etc.

[FIGURE 8 ABOUT HERE]

Figure 8.

Common topology for controller.

For example, the basic controller uses a PID controller and there are two resonant modes that need to be filtered, when C(s) can be designed as a cascade structured FB controller using a PID controller for the rigid mode and two second order filters for the first and second resonant modes. the mathematical expression of C(s) in the continuous time domain is defined by the following equation.

MathMLToLaTeX

*Filter structure design method*

Taking into account the study of controller topology in Section [3.1](#H3\.1) and the control law design domain rule embedding method in Section [2.4](#H2\.4), this paper adopts the structure of multiple filters cascaded with three possible forms for each filter, as shown in Figure [9](#F9).

[FIGURE 9 ABOUT HERE]

Figure 9.

The filter structure in this paper.

where the transfer functions of the three filters are:

(1) 0-order filter:



(2) 1st-order filter:

(3) 2nd-order filter:

It can be seen from the transfer functions of the three filters: the 0-order filter is equivalent to an all-pass filter with no change in phase; the 1st-order filter varies between an all-pass filter and a high-pass filter as varies; the 2nd-order filter varies between an all-pass filter, a band-stop filter and a band-pass filter as and vary.

*Controller parameter optimization method*

After obtaining the structure and initial parameters of the controller, it is necessary to substitute this controller and parameters into the closed loop of the whole system, and to adjust and optimize the parameters of the controller.

In the control system shown in Figure [4](#F4), is defined as a cascade structure controller with a basic controller and N filters connected in series:

The design problem of is to obtain all controller parameters that can maximize the control bandwidth while satisfying the specified gain margin and phase margin . The controller parameter is divided into two parts: the basic controller parameters and the other parameters .[9](#9)

The controller design method based on hybrid optimization uses different optimization methods to obtain the parameters of the cascade structure controller for the above two parts of the parameters, respectively. For the basic controller parameter , the theory and method related to the basic controller are used to tune the controller parameters; while for the other parameters , the GA-based optimization method is used to tune the parameters. The detailed design process is as follows:

**Step 1** Randomly generate the initial population (population size ) as the first generation (=1).

**Step 2** Obtain the parameters of candidates applicable to by this basic controller related theory and method.

**Step 3** According to the defined fitness function (which reflects the control bandwidth and stability margin), the fitness scores of all individuals are evaluated in the GA using and to obtain the elite individuals.

**Step 4** If the generated is less than the specified number , then and go to step 5; otherwise, go to step 6.

**Step 5** Perform genetic operations, such as selection, crossover and mutation, and generate a new population for the next generation. repeat steps 2∼5 when .

**Step 6** Use the elite and to obtain the desired , which extends the control bandwidth while satisfying a specific stability margin.

*Selection of the fitness function and determination of important parameters*

The most important requirement of the control system is fast and accurate, which requires the bandwidth to be as high as possible, but too much bandwidth can lead to system stability degradation, so the fitness (reward) function is selected considering both high bandwidth and sufficient stability margin, and we have determined the following forms of fitness functions:

(1) Summation type

where is the bandwidth (in rad/s), is the gain margin, and is the phase margin (in degrees).

(2) Additional bonus type

where 5° and 30° are referenced from the literature [9]. and are additional bonus values for gain margin and phase margin, respectively, which can be adjusted.

(3) Feasible domain type

The direct summation-type fitness function focuses too much on the bandwidth and ignores the gain margin and phase margin, and from the simulation results, the system is in the edge of stability in most cases. The feasible domain adaptability function varies too much in the feasible domain edge function value, which is easy to fall into the local optimum and not easy to find a better solution. By adjusting the extra reward value, the local optimal solution can be gradually transitioned to a better solution through genetic iteration. The extra reward value cannot be too large, which will lead to a local optimum, or too small, which will make the fitness function focus too much on bandwidth and ignore the system stability. Therefore the extra reward equation is a form of fitness function between the direct summation equation and the feasible domain equation. The actual reward value needs to be finally determined by repeated experiments.

The larger the number of populations in the genetic algorithm, the better the population diversity and the higher the probability of convergence to the global optimal solution, but the more computational resources are consumed. Taking into account the convergence of the genetic algorithm and the time overhead of the computational process, the number of populations in the genetic algorithm is set to 50, the number of elites is set to 7, the number of crossovers is set to 40, the number of variants is set to 3, the number of iterations is set to 2000, and other default settings are used.

Lightweight and efficient search method for overall optimization of controller structure and parameters

*Design of LSTM networks*

Following the ENAS approach proposed by Pham, the LSTM samples the decisions in an autoregressive manner by means of a Softmax classifier: the decisions from the previous step are embedded as inputs to the next step. In the first step, the controller network receives the empty embedding as an input. For the control system, according to the structure of the filters in Section [3.2](#H3\.2), what needs to be determined is the number of filters and the order of each filter. The number of filters generally depends on the needs of the actual problem.

The filters are divided into 0-order, 1st-order , and 2nd-order, so the number of output categories of LSTM is 3. Assuming that the number of filters is N, as in Pham et al.’s original paper,[3](#3) the input size of LSTM is 1, and the length of the input stream is N, which denotes N filters, respectively. As shown in Figure [10(a)](#F10) represents the sampled output results of the LSTM recurrent cell, and (b) represents the filter structure corresponding to the output of the LSTM recurrent cell. Note that the input of the initial recurrent cell is 0 and the output is 3-category probability. After sampling out the first filter structure as a 1st order filter, this is used as the input to transmit to the recurrent cell, and then the 3rd-order filter is sampled from the output of the 3-category probability, and so on.

[FIGURE 10 ABOUT HERE]

Figure 10.

An example of an LSTM recurrent cell: (a) the sampled output results of the LSTM recurrent cell and (b) the filter structure corresponding to the output of the LSTM recurrent cell.

Pham et al. used an LSTM with 100 hidden units in the original paper.[3](#3) Comparing the size of the problem in this paper with Pham et al.’s paper, we use an LSTM with 10 hidden units.

*Overall Training Scheme for controller structure search and parameter optimization*

Throughout the training process, there are two sets of learnable parameters: the parameters of the LSTM are denoted by , and the shared parameters of the filter are denoted by . The training process consists of two interleaved phases. The first stage trains the shared parameters of the filter, , and optimizes the parameters from the whole by genetic algorithm. In the actual experiment, the number of training steps per round of is set to 20 (taking into account the convergence of the genetic algorithm and the time overhead of the computational process, the number of iterations per round of training of the genetic algorithm is set to 20). The second stage trains the parameters of the LSTM . Unlike the original ENAS paper, where the object of study is the data set and multiple sets of data need to be sampled for updating, the object of study in this part is a single controlled system, which is set to perform one parameter update in the actual experiment. These two phases are alternated during the training process, and the total number of rounds is generally set between 100 and 200. More details are as follows.

*Train the filter with shared parameters .*

In this step, fix Generator’s policy and perform genetic iterations on to maximize the fitness function . The filter structure is sampled from . We can use any individual model sampled from to update . As mentioned before, we train throughout the process of finding a better value of the fitness function.

*Train the generator parameters .*

In this step, fix and update the policy parameters with the goal of maximizing the desired fitness . We use the SGDM or Adam optimizer, for which the gradient is computed using the classical REINFORCE algorithm in reinforcement learning, and a sliding average baseline to reduce the variance. The fitness is computed by imposing the controller on the target plant to encourage the selection of a well-performing controller structure. The fitness function is defined in Section [3.4](#H3\.4). The sliding average baseline is given by:

Where is the sliding ratio, taken as 0.9; is the coefficient of the fitness function, taken as 0.001, because the fitness is generally greater than 103; and the initial value of is taken as 0.

In order to model the problem as a reinforcement learning problem, it can be assumed that the generation of the controller structure is the action of an agent whose action space is the same as the search space of the controller structure. The “reward” of the agent is based on the estimation of the control performance of the controller structure. The policy samples various “actions” to sequentially generate the controller structure, and the “state” of the environment consists of the set of actions sampled so far, and is rewarded only after the final action. As shown in Table [2](#T2).

Table 2.

Reinforcement learning modeling.

|  |  |
| --- | --- |
| State | The set of actions sampled so far |
| Action | The next part of the structure obtained by sampling |
| Action space | The search space of the controller structure |
| Reward | Estimation of the control performance of the controller structure |

*Determining the optimal architecture and parameters.*

After the overall training for controller structure search and parameter optimization, we first select the controller structure with the highest probability from the trained strategy . Then we take only the model parameter with the highest adaptation from the shared parameters.

The whole process is shown in Figure [11](#F11).

[FIGURE 11 ABOUT HERE]

Figure 11.

Training framework.

*Selection of the optimizer and determination of important parameters*

Currently, the mainstream optimizers are stochastic gradient descent (SGD),[11](#11) momentum stochastic gradient descent (SGDM),[12](#12) adaptive gradient descent (Adagrad),[13](#13) root mean square propagation optimization (RMSProp),[14](#14) and adaptive moment estimation optimization (Adam).[15](#15)

The five major optimizers are actually divided into two categories, SGD, SGDM, and Adagrad, RMSProp, Adam. the more used ones are SGDM and Adam. as shown above, SGDM is more used inside computer vision (CV), while Adam basically sweeps natural language processing (NLP), reinforcement learning (RL), generative adversarial networks (GAN), and Speech synthesis and other fields. For example, in the field of NLP, the classical models Transformer and BERT all use Adam, and its variant AdamW.

In this paper, the controller structure is generated in LSTM, which is different from all of the above application areas and requires several experiments to determine the specific problem. The LSTM can be trained with input weights, recurrent weights and bias parameters. The input weights are initialized using the Glorot initializer (also known as the Xavier initializer). The Glorot initializer samples independently from a uniform distribution with mean zero and variance 2/(inputsize+numout) (inputsize for this part of the LSTM is 1), where numout▓=▓4\* numhiddenunits.  For the initialization of the cyclic weights the orthogonal matrix Q given by the QR decomposition of the random matrix Z sampled from the unit normal distribution is used. For the bias, we use 1 to initialize the oblivious gate bias and 0 for the rest of the bias.

Simulation example

*Target plant*

This example draws on the laboratory galvano scanner of the laser processing machine in Makoto Iwasaki’s paper as the target plant under control. The galvano scanner consists of a rotating motor, a mirror, and an optical encoder, and fast response and high-precision control of the motor angle is required to achieve high productivity in high-density interconnect (HDI) printed circuit boards.

The mathematical model used in this example is structurally typical in that it can simulate both delay characteristics and high frequency disturbances. It has a primary resonant mode of 2960▓Hz and a secondary resonant mode of 6100▓Hz. The transfer function of the system is as follows:

The parameters are shown in Table [3](#T3).

Table 3.

Parameters of the target plant model.

|  |  |
| --- | --- |
|  | 1.9▓×▓10−5 |
|  | 1▓×▓106 |
|  | 0.41 |
|  | −1.65 |
|  | 2▓×▓2960 |
|  | 2▓×▓6100 |
|  | 0.004 |
|  | 0.015 |

Its Bode plot is shown in Figure [12](#F12).

[FIGURE 12 ABOUT HERE]

Figure 12.

Bode plot of the target plant.

*Filter structure design*

Taking into account the study of the controller topology in Section [3.2](#H3\.2), this example uses a structure with two filter cascades, each with three possible forms, for a total of nine possible structures, as shown in Figure [13](#F13). The basic controller uses a PID controller with a low-pass filter with a transfer function of the form:

[FIGURE 13 ABOUT HERE]

Figure 13.

The filter structure used in this example.

Therefore, the basic controller parameters for this example are , including {{EQU00117}}、{{EQU00118}} and {{EQU00119}}, and the other parameters {{EQU00120}} including {{EQU00121}}、{{EQU00122}}、{{EQU00123}}、{{EQU00124}}、{{EQU00125}}、{{EQU00126}}、{{EQU00127}}. The range of parameters {{EQU00128}} for the other parameters is shown in Table [4](#T4).

Table 4.

Parameter range of {{EQU00129}}.

|  |  |  |
| --- | --- | --- |
| Parameters | Min. | Max. |
| {{EQU00130}} | 1▓×▓10−4 | 1▓×▓10−4 |
| {{EQU00131}} | 2{{EQU00132}}▓×▓1000 | 2{{EQU00133}}▓×▓2960 |
| {{EQU00134}} | −1 | 1 |
| {{EQU00135}} | 0 | 1 |
| {{EQU00136}} | 2{{EQU00137}}▓×▓5965 | 2{{EQU00138}}▓×▓12,000 |
| {{EQU00139}} | −1 | 1 |
| {{EQU00140}} | 0 | 1 |

From Tables [3](#T3) and [4](#T4), it can be seen that the first filter is mainly for the first resonant mode and the second filter is mainly for the second resonant mode. From the transfer functions of the three filters, it can be seen that: the zero-order filter is equivalent to an all-pass filter with no phase change; the first-order filter varies between an all-pass filter and a high-pass filter as {{EQU00141}} changes; the second-order filter varies between an all-pass filter, a band-stop filter (trap filter) and a band-pass filter as {{EQU00142}} and {{EQU00143}} change. In addition, since the parameter {{EQU00144}} is independent of the filter structure, {{EQU00145}} is fixed to 1▓×▓10−4 for the convenience of comparing and analyzing the simulation results of full-search and structure search.

*Hybrid optimization-based controller design method and selection of fitness function*

From the analysis in Section [5.2](#H5\.2), it can be seen that the parameters to be optimized are {{EQU00146}} and {{EQU00147}}. Among them, {{EQU00148}} uses MATLAB’s own function “pidtune” for parameter tuning, while the parameter tuning of {{EQU00149}} is carried out using a genetic algorithm GA. The detailed design process is as follows:

**Step 1** Randomly generate the initial population {{EQU00150}} (population number {{EQU00151}}) as the first generation ({{EQU00152}}).

**Step 2** Obtain the PID parameters {{EQU00153}} applicable to the candidate of population {{EQU00154}} of {{EQU00155}} by the pidtune function, which can automatically adjust the controlled objects under the specified phase margin according to the user requirements (signal tracking, interference suppression).

**Step 3** According to the defined fitness function, the fitness scores {{EQU00156}} of all individuals are evaluated with {{EQU00157}} and {{EQU00158}} to obtain the elite parameters {{EQU00159}}for elite individuals.

**Step 4** If the generated {{EQU00160}} is less than the specified number {{EQU00161}}, then {{EQU00162}} and go to step 5; otherwise, go to step 6.

**Step 5** Perform genetic operations, such as selection, crossover and mutation, and generate a new population {{EQU00163}} for the next generation. then {{EQU00164}}. Repeat steps 2∼5.

**Step 6** Use elites {{EQU00165}} and {{EQU00166}} to obtain the desired {{EQU00167}}. This extends the control bandwidth while meeting specific stability margins.

Based on iterative experiments, the final fitness function is determined as:

{{EQU00168}}

{{EQU00169}}

{{EQU00170}}

Refer to Section [3.4](#H3\.4) for other settings of the genetic algorithm.

*Design of LSTM networks and selection of optimizers*

For the controlled object of this example, the number of filters is determined to be 2 according to the analysis in Section [5.2](#H5\.2), and what needs to be determined is the order of the two filters.

The number of output categories of the LSTM is 3, representing three filters. the input size of the LSTM is 1, and the length of the input stream is 2, representing two filters respectively. The number of hidden cells is 10. The network architecture is shown in Figure [14](#F14).

[FIGURE 14 ABOUT HERE]

Figure 14.

The architecture of this LSTM.

Figure [15](#F15) shows an example where (a) represents the filter structure corresponding to the output of this LSTM recurrent cell, and (b) indicates that the sampled output results of this LSTM recurrent cell. Unlike Section [4.1](#H4\.1) only two filter structures need to be determined here.

[FIGURE 15 ABOUT HERE]

Figure 15.

An example of this LSTM recurrent cell: (a) the sampled output results of this LSTM recurrent cell and (b) the filter structure corresponding to the output of this LSTM recurrent cell.

For the optimizer, the learning rate is experimentally found to be the key parameter. Too high a learning rate will lead to too fast convergence, and the Generator may not search sufficiently for various structures and may even converge to non-optimal structures; too low a learning rate will lead to insignificant convergence. To achieve a lightweight optimizer, the SGDM is modified as follows, drawing on Adam’s idea:

{{EQU00171}}

The decay of the learning rate is linked to the number of iterations, which not only realizes the adaptive update of the learning rate, but also makes the decay process of the learning rate smoother. After several experiments, the initial learning rate **initialLearnRate** is taken as 0.003 and the decay rate **decay** is taken as 0.01.

*Results*

Set the number of LSTM network update iterations to stop when a certain structure reaches 100 first. The results are as follows:

As can be seen in Figure [16](#F16), the filter structure converges to the 2nd order-2nd order form. the output of the LSTM is shown in Table [5](#T5). Figure [17](#F17) shows the comparison of the Bode plot before and after filtering, where SG1 and SG2 denote the sensitivity gain at the two resonant frequencies, respectively, and it can be seen that the first resonance frequency has a significant decrease in sensitivity, while the second resonance frequency has a small decrease in sensitivity.

[FIGURES 16 AND 17 ABOUT HERE]

Figure 16.

Simulation results: (a) number of iterations per structure, (b) loss function with baseline, (c) output of the first time step of the LSTM, and (d) output of the second time step of the LSTM.

Table 5.

Output of LSTM.

|  |  |  |
| --- | --- | --- |
|  | The first filter | The second filter (when the input is of 2nd-order) |
| 0-order | 0.0206 | 0.0176 |
| 1st-order | 0.0189 | 0.0119 |
| 2nd-order | 0.9606 | 0.9705 |

Figure 17.

Comparison of Bode plot before and after filtering.

For comparison, the conventional exhaustive method was used for the experiments, and the results are shown in Table [6](#T6). It can be seen that: (1) The optimal structure form is PID-2nd-2nd; (2) The structure of the first filter is critical to the system bandwidth; (3) The controller structure search and parameter optimization method used in this example can effectively converge to the optimal structure, and the control performance is close to the optimal control performance of the full-search method; (4) Compared to the full-search method, the number of iterations and the time consumed by the method used in this example is only 15%.

Table 6.

Comparison of simulation results of two methods.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Method | Structure | BW（rad/s） | GM | PM（deg） | Fitness | SG1  （dB） | SG2  （dB） | Iteration number |
| Full-search | PID-0-0 | 887.8 | 0.06 | 71.7 | 4959.56 | 0 | 0 | 18,000 |
| PID-0-1st | 1464.0 | 7.6 | 59.3 | 9530.9 | −5.6 | −2.9 |
| PID-0-2nd | 1705.3 | 7.3 | 59.4 | 9772.0 | 0.4 | 2.1 |
| PID-1st-0 | 1762.6 | 5.5 | 64.6 | 9832.7 | −0.8 | −0.2 |
| PID-1st-1st | 2091.5 | 7.8 | 59.4 | 10,158.7 | −11.1 | −6.5 |
| PID-1st-2nd | 1721.8 | 5.2 | 60.0 | 9787.0 | −3.0 | −6.2 |
| PID-2nd-0 | 6525.5 | 5.7 | 60.0 | 14,591.2 | −1.4 | −0 |
| PID-2nd-1st | 6577.1 | 5.7 | 59.4 | 14,642.2 | −1.7 | −0 |
| PID-2nd-2nd | 6845.5 | 5.4 | 60.0 | 14,910.9 | −7.5 | −0.7 |
| Structure search | PID-2nd-2nd | 6691.3 | 5.5 | 60.0 | 14,756.8 | −5.2 | -1.0 | 2780 |

Table [7](#T7) shows the comparison of simulation results using different fitness functions, from which it can be seen that: (1) The **Summation type** leads to too much focus on bandwidth and results in the system not being able to achieve sufficient stability margins; (2) In the **Additional bonus type**, if the reward value is too small it will lead to a similar situation as in the **Summation type**, and if the reward value is too large it will make the system easy to fall into a local optimum and not be able to achieve sufficient bandwidths; (3) The **Feasible domain type** can balance bandwidths and stability margins, but the bandwidths are lower compared to the optimal results.

Table 7.

Comparison of different fitness functions.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Fitness Function | Structure | BW（rad/s） | GM | PM（deg） | SG1  （dB） | SG2  （dB） |
| Summation type | PID-2nd-2nd | 12,707 | 1.2 | 24.6 | −1.5 | −9.7 |
| Additional bonus type ({{EQU00172}}={{EQU00173}}=1000) | PID-2nd-2nd | 11,188 | 2.0 | 60.0 | −11.5 | 0.7 |
| Additional bonus type ({{EQU00174}}={{EQU00175}}=3000) | PID-2nd-2nd | 10,297 | 2.0 | 60.0 | 6.0 | 3.3 |
| Additional bonus type ({{EQU00176}}={{EQU00177}}=4000) | PID-2nd-2nd | 6691.3 | 5.5 | 60.0 | −5.2 | −1.0 |
| Additional bonus type ({{EQU00178}}={{EQU00179}}=5000) | PID-1st-1st | 1503.1 | 5.7 | 60.5 | −6.7 | −3.7 |
| Additional bonus type ({{EQU00180}}={{EQU00181}}=10,000) | PID-1st-1st | 1506.1 | 5.9 | 59.4 | −7.8 | −5.1 |
| Feasible domain type | PID-2nd-2nd | 5835.6 | 5.7 | 60.0 | −13.4 | 2.1 |

In addition, for the structure search method, Maeda et al. used GA to implement it, where he viewed all possible structures as individuals in a population and selects the optimal structure through repeated iterations.[10](#10) Compared with the LSTM-based structure search method proposed in this paper: (1) The scalability of structure search based on GA is not good, which is not efficient in dealing with multiple controllers in cascade, whereas the method in this paper can handle it easily; (2) The structure search based on GA can not reflect the correlation between substructures, whereas the LSTM can model the correlation between each time step, and it is more interpretable.

Conclusion and outlook

*Conclusion*

In this paper, a controller design method based on the integration of structure search and parameter optimization is proposed, which can find the most suitable controller structure and complete parameter tuning in a shorter time compared with the traditional full-search based method, and meet a specific stability margin while extending the control bandwidth, which can significantly improve the efficiency of controller designers. The validity of the method was verified on a galvano scanner.

*Outlook*

The next step can be further investigated in the following aspects:

(1) Embedding more rules of the controller structure design domain and adopting a more complex way of describing the controller structure to make it closer to the actual application scenarios.

(2) Researching on the encoding of directed cyclic graphs, which can be applied to controllers that introduce feedback structures, allows a wider range of structure search.

(3) Selecting more complex controlled objects, such as variant vehicles.

(4) Parameter optimization can use other heuristic algorithms.

(5) Accomplish more complex control objectives.

Declaration of conflicting interests**[GQ: 2]**

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This research is supported by the science and technology innovation 2030 - “new generation artificial intelligence” major project (2018AAA0101605), the National Natural Science Foundation of China (No.61771281, No.61174168), the special project for industrial transformation and upgrading of MIIT 2018 (ZB182505), and independent research program of Tsinghua University (2018Z05JZY015).

ORCID iD**[GQ: 3]**

Kan Yang https://orcid.org/0009-0003-8089-0575

References

1. Cai M. *Flight Control System*. Beijing: National Defense Industry Press, 2007. pp.196.

2. Zoph B and Le QV. Neural architecture search with reinforcement learning. In: *International conference on learning representations*, 2017.

3. Pham H, Guan MY, Zoph B, et al. Efficient neural architecture search via parameter sharing. In: *International conference on machine learning*, 2018.

4. Luo R. *Efficient neural architecture search: algorithms and applications*. Hefei, Anhui: University of Science and Technology of China, 2021. DOI: 10.27517/d.cnki.gzkju.2021.001094.

5. Zhou T. *Study and application of search space optimization for structure search with convolutional neural network*. Beijing: Minzu University of China, 2021. DOI: 10.27667/d.cnki.gzymu.2020.000294.

6. Luo J. *Research and application of efficient neural network structure search algorithm based on dynamical isometry theory*.  Chengdu: University of Electronic Science and Technology of China, 2023.

7. Ahmed M, AlZoubi A and Du H. Improving generalization of ENAS-based CNN models for breast lesion classification from ultrasound images. In: Papież BW, Yaqub M, Jiao J, et al. (eds) *Medical image understanding and analysis. MIUA 2021. Lecture Notes in Computer Science*. Cham: Springer, 2021, Vol. 12722.**[AQ: Please provide page range for Reference ‘7’.]**

8. Maeda Y and Iwasaki M. Improvement of adaptive property by adaptive deadbeat feedforward compensation without convex optimization. *IEEE Trans Ind Electron* 2015; 62: 466–474.

9. Maeda Y, Kuroda E, Uchizono T, et al. Hybrid optimization method for high-performance cascade structure feedback controller design. In: *IECON 2018 - 44th annual conference of the IEEE Industrial Electronics Society*, 2018, pp.4588–4593. New York, NY: IEEE.

10. Maeda Y, Kunitate S, Kuroda E, et al. Autonomous cascade structure feedback controller design with genetic algorithm-based structure optimization. *IFAC-PapersOnLine* 2020; 53: 8419–8425.

11. Robbins H and Monro S. A stochastic approximation method. *Ann Math Stat* 1951; 22: 400–407.

12. Qian N. On the momentum term in gradient descent learning algorithms. *Neural Netw* 1999; 12: 145–151.

13. Duchi J, Hazan E and Singer Y. Adaptive subgradient methods for online learning and stochastic optimization. *J Mach Learn Res* 2011; 12: 2121–2159.

14. Tieleman T and Hinton G. Lecture 6.5-rmsprop: divide the gradient by a running average of its recent magnitude. *Coursera* 2012; 4: 26–30.

15. Kingma D and Ba J. Adam: A method for stochastic optimization. In: *Proceedings of the 3rd international conference on learning representations*, San Diego, CA, Workshop Track, 2015, pp.1–13.