Optimization Framework and Application Research of Automatic Test System for Armored Vehicles

Peng Liu

1 Academy of Special Operations, Guilin, 541000 China

*Corresponding author’s e-mail: 2643240412@qq.com

Abstract. The optimization of the automatic test system for armored vehicles is a system engineering. Based on the analysis of the current research status of automatic test systems, automatic test system optimization framework is introduced. According to the relevance with the architecture and software function, the software optimization is divided into three categories, focusing on the first two categories of nine optimization content. Aiming at the accuracy optimization of software prediction in the Class II software optimization, a state prediction algorithm based on GA-ANFIS is proposed, which can be used to predict the SOH of lead-acid batteries in armored vehicles. The experimental results verify the effectiveness of the GA-ANFIS algorithm.

1. Introduction

With the development of science technology, weapons and equipment are changing with each passing day. Key technologies continue to make new breakthroughs, and the renewal cycle is obviously accelerated. Whether the equipment can truly form combat effectiveness depends not only on the performance of the equipment, but also on the technical support capabilities of the equipment. In order to meet the needs of armored equipment support, the testing technology of weapons and equipment has been rapidly developed and widely used. In the field of military equipment support, the testing technology has experienced more than 50 years of development, and has achieved fruitful results and matured.

In the past, the technical state test of armored vehicles was usually carried out according to manual inspection or using special test equipment. The lack of a complete and high-performance universal automatic test system resulted in long test time, incomplete test results, and failure to quickly and accurately diagnose faults. With positioning, it is difficult to make a comprehensive and accurate evaluation of the overall technical state of the equipment. After extensive research and analysis, the research team started the research work on the automatic test system of armored vehicles, designed the hardware platform of the automatic test system of armored vehicles and several software subsystems such as gun control, fire control and communication. However, armored vehicle automatic test systems have encountered various complex occasions and application environments in use. Due to the different component manufacturers, the working status of key components of armored vehicles cannot be scientifically and accurately predicted and estimated. Many problems have been exposed in the use process, such as the low openness and universality, the inconvenience of software operation, the low efficiency of software operation and the inaccuracy of state prediction.
2. Related works
In the field of military automatic testing, the level of testing greatly affects the function and performance level of weapons and equipment. In the early part of this century, the United States proposed the concept of the Next Generation Automatic Test System (NxTest), which generated a lot of research work [1-5] based on NxTest. NxTest is designed to reduce the full cycle cost of automated test systems, providing flexibility and interoperability, effectively reducing system development, system deployment and system upgrade time.

In different periods of the development of China's military automatic testing system, the detailed technical status testing rules for diverse equipment have been formulated. Many automatic test systems for armored vehicles, aviation, ships and other fields has been formed. In the past five years, many new research results have been achieved. In 2014, Li expounded the development status of avionics systems, and analyzed the research, improvement and development trend of automatic test system [6]. In 2015, Chen discussed the issues related to the standardization of military test system based on the comparative analysis of international test standards, and provided reference for the standardization of military automatic test system [7]. In 2015, Zhang studied the automated test technology of military power modules and realized the reliability test of power modules [8]. In 2017, Mei used the Klocwork test framework for automated design under the Kirin Linux operating system, reducing the cost of the entire test process [9]. In 2018, Ma analyzed the meaning and current status of military software automated testing, and discussed in depth the system structure of military software automated test architecture [10].

3. Armored Vehicle Automatic Test System Optimization Framework

3.1 Optimization Framework

The armored vehicle automatic test system consists of four parts, which are the main control platform, the measurement and control platform, the vehicle body attitude generation platform and the artillery action excitation platform. The system platform and functions can be referenced in reference [11]. The theoretical framework of the automatic test system optimization of armored vehicles is shown in Figure 1. It consists of three levels of optimization. The software optimization on the left side is strongly related to the architecture, which is called the Class I software optimization. The optimization of the software level depends partly on the selected software. The hardware and computer architecture used are highly correlated with computer hardware. The lower right part is software optimization with weakly related to the architecture, called the Class II software optimization. It can effectively support the function optimization of each subsystem. The upper right part is the function optimization of each sub-system, which is called the Class III software optimization, mainly involving the optimization of the algorithm level of each sub-system.

Among the three categories of software optimization, the Class I software optimization is highly correlated with hardware. Excellent programmers familiar with the architecture and have a deep understanding of computer compilation technology are needed, which can support the other two
categories. The effect of software optimization. Through the Class I software optimization, the optimization of the architecture level can be achieved according to compilation elements such as the program logic, hotspot function, hotspot loop, program structure, etc.

The Class II and Class III software optimization are weakly related to the architecture. The Class III software optimization focuses on the function development of various sub-systems. There are some differences in the software design and main ideas of different sub-systems. Therefore, it is necessary to consider the specific implementation details of each sub-system, and improve the overall capability of the automatic test system by promoting the functions of the sub-system. Relatively speaking, the Class II software optimization is the universal optimization in the automatic test system, which improves the overall capability of the automatic test system from five aspects, including ease of use, versatility, openness, timeliness and accuracy.

3.2 The Class I software optimization

3.2.1 Computing Resource Optimization
With the increasing popularity of homogeneous multi-core servers, the measurement and control platform in the automatic test system also uses a general-purpose multi-core architecture server. The advantages of multi-core architecture can be fully exploited in task-level parallelism, thread-level parallelism, instruction-level parallelism, and resource scheduling. The division of computing resources on multiple CPU cores can be achieved using MPI, OpenMP, SIMD and other parallel programming models, improving the efficiency of each sub-system software implementation.

3.2.2 Optimization of communication resources
Since the automatic test system uses multiple platforms and systems, different software runs on different nodes and processes, which involves inter-node communication, inter-process communication, wire-speed access of network traffic, and wire-speed forwarding of network traffic, ensuring efficient flow of control flow and data flow of platforms and systems in automated test system.

3.2.3 Storage Resource Optimization
Storage resource optimization is an important part of architecture-related software optimization. In the implementation of the software functions of each subsystem, programmers who have more in-depth understanding of compiler optimization techniques can optimize the core algorithms and hotspot functions of the automatic test system, from improving the use of registers, managing caches to designing high-speed file systems, improving the performance of the software system to a certain extent.

3.2.4 Data Level Optimization
Various data acquisition cards, graphic image cards, and sensing components in the automatic test system generate a large amount of data. The main stages of big data include data acquisition, data storage, data processing, data analysis, data evaluation and assistant decision-making. As a whole, bottlenecks in any one link can cause system performance degradation. The processing delay and throughput of each link are analyzed, and each stage is optimized to form the theoretical composition of data level optimization.

3.3 The Class II software optimization

3.3.1 Usability of software operations
While realizing the basic functions of the automatic test system, the software should be developed in terms of usability. The user interface should conform to the current standards and specifications, and should have different views for users to choose flexibly. In the meanwhile, the user should be designed
as the center as much as possible, so that the user can use it comfortably, which will not create obstacles and difficulties for users. User-oriented, user-centered is an important optimization method from the prototype of the laboratory to popularization and promotion. It strives to make the operation convenient and easy to use. The layout of the user's graphical interface is simple and reasonable, and the user experience comfort during software use.

3.3.2 Universality of the software system
When the number of different types of armored vehicles is small, test systems can be customized for different armored vehicles. However, the automatic test system software size and runtime space-time overhead will increase linearly with the number of equipment. Therefore, the design of the software function of the automatic test system must face the development from the specificity to the versatility. By analyzing the common characteristics of different armored vehicles, the universal design of the software function is developed. The detection index and the detection function are tailored to adapt to different armored vehicles.

Versatility is the general direction to be considered for software system optimization. At the beginning of the design of the automatic test system for armored vehicles, software development was carried out based on a certain type of armored vehicle. With the increasing number types of armored vehicles, many types of test equipment have been introduced. The test process is more complicated, and the adaptation and interface are also diversified. Then, unified interface, standard unification, and process unification are the requirements that must be followed to develop the universal optimization of software systems.

3.3.3 Openness of the software interface
The openness of the software interface is a development trend of software design, and is also applicable to the development of software systems in the automatic test system of armored vehicles. However, the current automatic test system software design is not open. After the user has new requirements, they need to be fed back to the developer. The developer will modify the software system in the automatic test system according to the new requirements. After the test passes, the software product is delivered to the user. The open software design implements interfacing and componentization of the program design. Users can add some functions to the detection platform according to their own characteristics, and carry out secondary development. This opens the software development interface to users or external programmers. The process can open up another way to add new software features, significantly improve the efficiency of software development.

3.3.4 Effectiveness of software execution
The future war is quicker and faster, and the rhythm is faster. It puts new requirements on equipment support, and should focus on the development direction of future equipment support optimization with effectiveness. As the core means of equipment support, the automatic test system must have the fast guarantee capability consistent with high-speed, fast-paced combat operations. Therefore, the effectiveness of software system implementation is an important research direction in this field. Effectiveness includes computational efficiency optimization, storage efficiency optimization, network optimization, and database optimization. These optimization techniques improve the effectiveness of software execution from different aspects. It should be pointed out that this part of the optimization is different from the Class I software optimization. For example, the spatial locality in memory access can be regarded as the Class I software optimization. However, and the large cache space is divided into multiple bank round-robin accesses to solve the bottleneck caused by storage to a certain extent, which can be regarded as the Class II software optimization.

3.3.5 Accuracy of software prediction
The state prediction of the armored vehicle components is an important maintenance technology, which can effectively avoid the "under-maintenance" and "over-maintenance" caused by regular
maintenance. It is easy to grasp the trend of the running state of the equipment. Although various technical methods can be used such as mathematical statistics, support vector machines and neural networks, most of the studies are difficult to achieve the desired results. Therefore, it is necessary to improve the predictive ability of the model. The research on the accuracy of state prediction is a difficult and important research direction in this field.

Accuracy of software prediction includes component prediction, state prediction, and accuracy improvement. An armored vehicle is an organic whole consisting of numerous components. The input data of component prediction, state prediction is derived from the detection result of automatic test system. The detection data is a set of discrete time series data. The machine learning model such as support vector machine and neural network is established to predict the state of next time point.

4. Application research on prediction accuracy optimization

4.1 Battery SOH prediction
In the process of armored vehicles operation and use, it is necessary to timely and accurately monitor the SOH of battery in order to ensure good battery performance. The SOH prediction can identify battery performance decrease and detect the end of its life cycle. It is generally believed that when the SOH is less than a certain ratio, the battery cannot be used anymore and needs to be repaired or replaced. SOH is a theoretical concept that cannot be directly measured. It is more complicated than the prediction of SOH and needs to be inferred from other factors related to battery health. The degree of degradation of battery power depends on many factors such as usage, operating mode and working environment. SOH estimation accuracy depends not only on the reliability of the battery model, but also on determining the accuracy of parameter selection in the model. At the same time, military batteries and civilian batteries have certain differences in terms of business needs and usage scenarios.

4.2 State prediction algorithm based on GA-ANFIS
Combining genetic algorithm with adaptive neuro-fuzzy inference system, a state prediction algorithm is designed based on GA-ANFIS (Figure 2).

The core of the algorithm is the process of training ANFIS using GA. The process first obtains the membership-related parameters and serializes the parameters. Then, the configuration parameters of the genetic algorithm are initialized, and the initial population of the genetic algorithm is generated according to some of the above parameters. For the initial population, the FIS fuzzy inference is calculated and the cost is calculated. The input variable is the input matrix of the training data, and the output variable is the battery SOH estimated by FIS. The overhead is represented by two error statistics terms MSE and RMSE.

In each generation computation of genetic algorithm, the algorithm normalizes the cost to calculate the fitness of each population. Each iteration generates two descendants through a crossover operator and evaluates the cost of the descendants. The mutation operator is executed after the crossover operator. The father node is randomly selected, and the offspring are generated by the mutation operator operation. When the progeny generated by the crossover and mutation meet the specified number, the population is merged along with the previous generation node, and the cost is sorted. The algorithm records the optimal FIS parameters and generates an optimal FIS, which will ultimately be used for the prediction of the battery SOH.
4.3 Experiment and analysis

The test object is a lead-acid battery of a certain kind of armored vehicle. Considering the universality of input variable selection, the two parameters of discharge depth and output energy are selected as input variables. Since the working environment of the lead-acid battery of the armored vehicle differs greatly from the use environment of the industrial battery. Two input variables of temperature and altitude are added. To test the accuracy of the optimization, four sets of experiments were performed, namely a two-input ANFIS experiment (ANFIS2), a four-input ANFIS experiment (ANFIS4), a two-input GA-ANFIS experiment (GA-ANFIS2), and a four-input GA-ANFIS experiment (ANFIS4). Due to space reasons, only the experimental results of the four-input GA-ANFIS are shown (Figures 3, 4, and 5).

Figure 3. Comparison of test data ANFIS output to target value (four inputs).

Figure 4. Error indicator of test data output results (four inputs).
Figure 5. Statistical characteristics of the first-order error of the test data (four inputs).

Figure 6. Comparison of the average error of the four groups of experiments.

The average error comparison of the four sets of experiments is shown in Figure 6. As can be seen from the figure, combining genetic algorithm with adaptive neuro-fuzzy inference system, the effect of fuzzy reasoning of the ANFIS algorithm can be significantly improved. Among them, the error of test data of GA-ANFIS2 is reduced by 24.6% compared to ANFIS2. The error of test data of GA-ANFIS4 is reduced by 47.6% compared to ANFIS4. Using the four input variables of discharge depth, output energy, temperature and altitude as the battery SOH, the model prediction effect of the two input variables of discharge depth and output energy can be significantly improved. Among them, the error of test data of ANFIS4 is reduced by 29.7% compared with ANFIS2. The error of test data of GA-ANFIS4 is reduced by 51.2% compared with GA-ANFIS2. Through comparison and analysis, the validity of accuracy optimization is verified, and the algorithm can be further extended to general. As a general state prediction technology, it can be used to predict the state of armored vehicle components and optimize the accuracy of system software prediction.

5. Conclusion
The paper focuses on the optimization research of armored vehicle automatic test systems. According to the relevance with architecture and software function, the optimized content is divided into three categories. Four contents of the Class I software optimizations are described, including computational resource optimization, communication resource optimization, storage resource optimization, and data-level optimization. Then, five contents of the Class II software optimizations are described, including usability of software operations, universality of software systems, openness of software interfaces, effectiveness of software execution and accuracy of software predictions. The accuracy of software prediction in Class II software optimizations is applied to the SOH prediction of armored vehicle battery, and a state prediction method based on GA-ANFIS is proposed. Four groups of experimental results show that GA-ANFIS4 has a good prediction effect and can significantly improve the accuracy of battery SOH prediction.

References
[1] Y. Z. Hu, Y. K. Wen. Yongkang. Technical Architecture of New Generation Technology State Test Line [J]. Scientist, 2017, 5(24): 80-83.
[2] Hrobak M. Critical mm-wave Components for Synthetic Automatic Test Systems[M]. Springer, 2015.

[3] X. J. Chen, Q. Zhang, Y. X. Zhang et al. Discussion on parallel testing methods of airborne products [J]. Proceedings of 2015 Aviation Experimental Testing Technology Academic Exchange Conference.

[4] Romania J, Ross W, Butcher S. Army and navy management of automatic test systems for weapon system support: Comparing us army and us navy ats management practices[C]//AUTOTESTCON, 2017 IEEE. IEEE, 2017: 1-9.

[5] G. F. Peng. Research on Military Technology State Test Line Based on NxTest[J]. Journal of Aerospace Computing Technology, 2008, 38(2): 93-95.

[6] Y. Li. Development of Military Aviation Electronic Technology State Test Line [j]. Silicon Valley, 2014, 7(16): 165-165.

[7] J.S. Chen, X. Y. Zheng, Y. Chen, et al. Discussion on the Relevant Standards of Military Test Systems [J]. Automation and Instrumentation, 2015(6): 179-180, 184

[8] B. Zhang. Design and Implementation of Military Power Module Automatic Test System [D]. University of Chinese Academy of Sciences, 2015.

[9] L. Mei, X. B. Liu. Application of Klocwork in Military Software Testing[J]. Aerospace Standardization, 2017(3): 38-42, 47.

[10] H. X. Ma, Y. Zhang. Discussion on Automated Testing and Quality Supervision of Military Software [J]. Reliability and Environmental Testing of Electronic Products. 2018(12): 74-77.

[11] X. Y. Tian, Architecture of a Kind of General-purpose Automatic Test System for Armored Equipments[J]. Computer Measurement and Control, 2011.7.