A Method of Generating Engineering Test Sequence for Urban Rail Transit CBTC System Based on Formalization

Xinyue Si*, Wenzhen Kuang2, Qiang Li3

1 School of Automation & Electrical Engineering, Lanzhou Jiaotong University, Lan Zhou, Gan Su, 730070, China
2 Gansu Research Center of Automation Engineering Technology for Industry & Transportation, Lan Zhou, Gan Su, 730070, China
3 School of Automation & Electrical Engineering, Lanzhou Jiaotong University, Lan Zhou, Gan Su, 730070, China

*Corresponding author’s e-mail: 448197369@qq.com

Abstract. In recent years, with the rapid development of urban rail transit, Communication Based Train Control system is very important to improve the operation efficiency and ensure the traffic safety. Before urban rail transit lines are put into operation, they need to be tested strictly. In view of the lack of unified test standards, specifications and low test efficiency, a formal test sequence generation method for CBTC system is proposed. Firstly, the test line is divided into several test segments; secondly, analyze the constraint conditions of test cases and get the expression of constraint relationship of test cases; thirdly, analyze the characteristic quantity that can represent the line characteristics, and get the line characteristic expression; and then match test cases with test segments using matching algorithm; finally, concatenate test cases into test sequences using the concatenation algorithm. The results show that the test sequences generated by this method meets the test requirements of CBTC system. The test sequences cover all test cases. And the method can effectively improve the test efficiency.

1. Introduction

CBTC system can transmit the train-ground information of continuous, two-way and large capacity. It can control the train with accurate positioning technology and real-time and efficient communication technology. Instead of the traditional method that CBTC system use the ground track circuit to determine the position of the train, and to exchange the information. CBTC system realizes moving block [1]. At present, China's urban rail transit new and modified lines all use the CBTC system. The CBTC system plays a key role in ensuring the safe and efficient operation of trains, and it is also a necessary condition for interconnection between lines. Therefore, the system must be fully tested before it is put into use to ensure the safe operation of urban rail transit.

The testing of the CBTC system is divided into white box testing at the R & D stage and black box testing at the engineering acceptance stage [2]. Testing in the R & D stage mainly tests the program code to check the logic, algorithms, and program input and output; Engineering tests perform integrity tests on system functions, data, interfaces, etc., including laboratory tests and field tests. Test cases are the basis for testing. Engineering tests are the final step to ensure the safe operation of the system before it goes online. This paper mainly studies the generation method of CBTC system engineering test sequence. Through the investigation of many domestic mainstream signal equipment
manufacturers, the CBTC system engineering test is still tested by test cases, and the test efficiency is low. Therefore, it is necessary to study the generation method of CBTC system engineering test sequence.

The formal language can be well recognized by the computer. The research on the generation method of the test sequence is based on the formal realization of the test case. In the following, the test is used instead of the engineering test [3].

2. Test requirements and objectives
Because of its fast transmission speed, large transmission data capacity, and high real-time characteristics, the CBTC system has become the development trend of urban rail transit in China and the world. The reliability and functional integrity of the CBTC system directly affect the safety and security of urban rail transit operations. To ensure the safety of passengers' lives and property, in order to ensure that the CBTC system can meet the system safety and functional requirements, it must be tested for completeness, and scientific test methods can greatly improve the efficiency of testing and the coverage of requirements.

To clarify the test objectives of the CBTC system, you must first determine the test content. The test work of CBTC system engineering test mainly includes unit test, integration test and field test. No matter which kind of test needs to test the specific functions of the system according to the test cases under a specific environment, this article writes 305 test cases based on the relevant standards of Chongqing Metro Line 10, and covers all the requirements in the latest functional requirements specification of the CBTC system. Perform a completeness test on the CBTC system, the test goal is that the system passes all test cases, and the function of the system meets all system requirements.

3. Research methods
Because the test environment of the CBTC system engineering acceptance test is complex and cannot be fully identified and modelled, this article proposes a formal language description method to convert the test line, test train and test case into a formal language that the machine can recognize [4]. Use test case assisted generation tools to connect test cases into executable test sequences. The following figure is the basic flow chart of the automatic test sequence generation method based on formal language description:

![Figure 1. Basic process of test sequence generation.](image)

3.1. Divide the test segment
CBTC system engineering acceptance test cases have certain requirements for the execution position. If you want to connect the test cases into a test sequence, you should first find the best execution position for each test case, and divide the test circuit into test sections [5]. Determine the execution position of the test case and pave the way for generating the test sequence. In this paper, the axle counter is selected as the dividing point of the test section. By analysing the layout method of the domestic CBTC system axle counter and the signal plane layout of Chongqing Metro Line 10, the principles to be followed when dividing the test section are summarized:

(1) Station area: Combine the axle counting segment and the protection segment in the station into one test segment;
(2) Section area: Incorporate the axle counting segment less than 200m into the next axle counting segment
(3) Turn-back area: Turn-back turnout section and turn-back parking section are combined into one test section.

3.2. Analyze test case constraints
First, extract the execution constraints of the test cases, analyse all the test cases, and analyse the execution constraints related to the test position and series conditions [6]. Seven types of test case execution constraints are summarized, as follows: Route constraint, Semaphore condition, Execution area constraints, Execution times constraints, Train condition constraints, Station type constraints, Equipment failure constraints.

(1) Route constraint (Route_Constraint): Balise constraints (Ba_Constraint), Semaphore constraints (Sp_Constraint), axle counter constraint (Ax_Constraint), switch segment constraint (Sw_Constraint), approach section constraint (Ap_Constraint); track conversion segment constraint (Tc_Constraint); Route triggering section constraint (Tr_Constraint); ZC point constraint (Zc_Constraint) and parking point constraint (Po_Constraint)
(2) Semaphore condition (Semaphore__Constraint): The status of the Semaphore is: signal red light (Sp_R_Constraint), signal green light (Sp_G_Constraint)
(3) Execution area constraints (Area__Constraint): Define the constraints of the execution area: Station(Station), Section(Section), Depot(Depot), Operation(Operation), All_Line(All_Line).
(4) Execution times constraints (TestcaseNum__Constraint): the entire line is only executed once (All_Line); each concentrated area is executed once (Every_Zc_area); each The areas that meet the conditions are executed once (Every_Target).
(5) Train condition constraints(TrainState_Constraint): Before test case execution, definition the train state constraint is: (Pre_TrainState_Constraint); After test case execution, definition the train state constraint is: (Pro_TrainState_Constraint). The following table is train state constraints:

| Constraint                  | State                | Definitions       |
|-----------------------------|----------------------|-------------------|
| Train control level and     | ATP excision mode    | EUM               |
| driving mode                | Manual mode at       | ILC-RM            |
|                             | interlock level      |                   |
|                             | Point-level coding   | ITC-CM            |
|                             | mode                 |                   |
|                             | Auto mode in point   | ITC-AM            |
|                             | level                 |                   |
|                             | CBTC level coding    | CBTC-CM           |
|                             | mode                 |                   |
|                             | Automatic mode       | CBTC-AM           |
|                             | under CBTC level     |                   |
| Train speed                 | motion               | Moving            |
|                             | stop                 | Stopped           |
| Door control method         | Automatic door       | AA                |
|                             | opening and closing  |                   |
|                             | Manual door opening  | MA                |
|                             | , automatic door     |                   |
|                             | closing              |                   |
|                             | Automatic door       | AM                |
|                             | opening, manual door |                   |
|                             | closing              |                   |
| Train positioning           | Targeted             | Already_Location |
|                             | Lost location        | Unlocated         |

(6) Station type constraints (Station_Type_Constraint): The constraints that define the station type are: Platforms on single side (Single_Station) and Platforms on both sides (Double_Station).
(7) Equipment failure constraints (Device_Fault_Constraint): By analysing the test case library, the device faults obtained are as follows: Signal failure; Turnout failure; Balise failure; Lost train integrity; VOBC failure; ATO failure; ATS failure; CI failure; Handover ZC fault; Take over ZC failure; Door fault; Screen door failure; Communication failure between train and ZC; Communication failure
between train and CI; ZC and CI communication failure; ZC and ZC communication failure; Communication failure between ZC and DSU.

3.3. Analysis of test section characteristics
According to the execution constraints of the test cases that have been obtained, the relevant information that can characterize the line characteristics of the test section is analyzed, and 11 kinds of feature quantities that can characterize the line characteristics are obtained. Include: Balise, Semaphore, Axle counter, Switch segment, Approach segment, Conversion track segment, Route trigger segment, ZC point, Parking spot, Execution area, Platform type, Current ZC, Current CI, Current route.

3.4. Match test segment and test case
The test circuit has been divided into test sections, so how to reasonably allocate all of them to the test section according to the execution constraints of the test case is the focus of this section. Matching the test case with the test section is to use the execution constraint of the test case to compare with the feature amount of the test section. If the feature amount of the test section meets the execution constraint of the test case, the test case is placed in this test section, and so on, until all test cases are matched [7].

When comparing the execution constraints of the test case with the feature quantities of the test section, it is only necessary to be able to determine the execution constraints of the test location. The execution constraints of such test cases are line constraints, execution area constraints, execution times constraints and platform constraints. Such constraints are called test site constraints. Analyzing the test site constraints of each test case can get a set of constraint relationship expressions. Similarly, analyzing each test segment can get a corresponding set of test segments Line characteristic expression, match this constraint relation expression with the line characteristic expression of each test section, and match it into the corresponding test section if successful [8]. So, how to match all test cases correctly and reasonably to all test sections that meet the conditions is difficult and important.

Through the above test case and test section matching method, combined with the key variable of test case execution constraints, a test case and test section matching algorithm is designed. The algorithm flow chart is as follows:

![Figure 2. Flow chart of concatenation algorithm.](image)
The number of test case executions serves as a key variable, which determines the number of test sections matched by the test case. Through this algorithm, all corresponding execution sections of each test case can be found.

3.5. CBTC system test sequence generation

After the matching of test cases and test sections is completed, the acceptance test cases of CBTC system engineering will be all assigned to the corresponding test sections, and then these test cases will be connected in series by a series of test case series rules [9]. The following figure shows the formation of a test sequence. Multiple test cases can be executed simultaneously in a test section. TC is test case.

According to the above test case concatenation rules, a test case concatenation algorithm is designed. The algorithm flow chart is as follows:

Traverse the acceptance test cases of CBTC system engineering, search for mode conversion test cases, point-level test cases, interlocking level test cases, and CBTC level test cases in series, and finally generate several executable test sequences.

4. Software implementation

In the third section, the generation method of the CBTC system engineering acceptance test sequence is studied. According to the above theoretical method, the test sequence generation auxiliary tool is designed to verify the feasibility of this method.
The composition framework of the CBTC system engineering acceptance test sequence auxiliary generation tool is shown in the figure. Designed to store the execution constraints of the test cases and the feature quantities of the test sections in the data storage layer, establish and call the database through SQL Server software, implement database query and test sequence generation functions at the application layer, and display the loaded test case data at the presentation layer, Load test section data, export test sequence and other functional options [10]. In short, the function realized by the test sequence auxiliary generation tool is mainly to generate test sequences using test case data and test segment data.

The interface of the auxiliary tool for generating test sequences is shown in the figure. 305 CBTC system engineering acceptance test cases written according to Chongqing Metro Line 10 are entered into this tool. The final generated test sequence is 65, which can be executed after manual testing.

5. Conclusion
There are many deficiencies in the use of test cases for the acceptance test of CBTC system engineering. With reference to the relevant line data and test standards and requirements of Chongqing Metro Line 10, an engineering acceptance test case database covering 100% of the latest CBTC system interconnection requirements have been written, combining forms The description describes the method of automatically generating CBTC system engineering acceptance test sequence. This method was actually applied to the CBTC system simulation test platform for testing, which verified the feasibility. The automatic test sequence generation method based on formalization is universal, and the database can be changed according to different lines to generate test sequences for different test lines. This test sequence generation method is beneficial to the development and realization of urban rail transit interconnection.
Acknowledgments
This publication is based upon work supported by the National Natural Science Foundation Regional Science Foundation Project under Project approval number: N2019G017; GSITA201910. The authors would like to thank anonymous referees whose comments and suggestions have improved the content and presentation of this work.

References
[1] Hessel A, Pettersson, P.A Test Case Generation Algorithm for Real-Time Systems[C]. Quality Software,4th International Conference.2004:268-273.
[2] L C, Zheng W, Li K C, Tang T. Automatic Generation and Verification of Test Sequence Based on Path Optimization Algorithm. Journal of Railway. (2013), 35(6).53-58.
[3] DE SMITH M J. Determination of Gradient and Curvature Constrained Optimal Paths. Computer-aided Civil and Infrastructure Engineering. 2006, 21(1); 24-38.
[4] Chen X, Jiang P, Zhang Y F. Method for Automatically Generating Test Cases Oriented to Safety-Related Scenarios in Train Control System. Journal of Software. (2015), 26(2):269-278.
[5] Hu W W. CBTC Vehicle-Mounted System Test Case Design and Optimization Method Research. Beijing Jiaotong University, (2010).
[6] Song S, Liu B H, Zhou K, Chen Y D. Method for Generating Regional Controller Test Sequence Based on Time Automaton Model. Railway Standard Design. (2018), 62(8).168-173.
[7] Song S. Research on Model-based Automatic Generation Method of CBTC Regional Controller Test Sequence. Lanzhou Jiaotong University, (2018).
[8] Wang F. CTCS-3 Level Train Control System Function Test Method Based on Simulation. Railway Communication Signal, (2010),46(8):13.
[9] Liu L. Research on CTCS-3 Level Train Control System Field Test and Auxiliary Tools. Beijing Jiaotong University, (2010).
[10] Wang S. Research on Expert System of CTCS-3 Train Control System Test Sequence Generation. Beijing Jiaotong University, (2012).