

Complex Logic Control Algorithm and Simulation Application Based on Matrix Modeling

Yu Li^{1,a}, Gang An^{2,b} and Xin Li^{3,c}

¹ System Integration Center, FAI AVIC, Xi'an, CN

² School of management, Xi'an Jiaotong university, Xi'an, CN

³ Landing Gear System Research Institute, FAI AVIC, Xi'an, CN

Abstract. The complex logic control of discrete signals is a common problem in engineering applications. The traditional program design using multi branch conditional transfer algorithm, program complexity increases with the complexity of the problems, therefore, the ability to solve logic control severely limits the scale and complexity of problems. The complex logic control algorithm based on matrix modeling is based on the analysis and modeling of logical relations, and the implementation of the two-dimensional matrix makes the complexity of the program irrelevant to the problem. The algorithm can well solve the display and control simulation problem of the multi-function display in aircraft cockpit display and control equipment. It has good expansibility and popularization.

1. Introduction

The multi-function display (MFD) is an important component of human-computer interaction system in modern cockpit. In flight and execution tasks, it provides real-time video and graphical information required by pilots. MFD is a controllable display device, providing a rich working mode and a variety of control modes. The pilot can toggle the work mode by function keys as needed. Because the function keys have different control function definitions in different working modes, there is a complex logic control relationship between the conversion of working states and the control of peripheral keys. In the simulation of MFD display control, matrix modeling is used to solve the complex logic control relation transformation problem.

The matrix method is different from the traditional multi branch conditional transfer control algorithm. It uses the two-dimensional array to represent the logic relation[1]. When the program is implemented, only the information in the two-dimensional array is read and correctly interpreted, so that the logic conversion control function can be completed. Compared with the traditional algorithm, the matrix method can simplify the logic in programming complexity, easy to design, modify and expand the function, and can form a standard program algorithm is extended to state control, such as mobile phone equipment and the automatic teller machine. This paper will combine MFD display and control system simulation, from the matrix modeling, program design and performance analysis, in-depth study and description of this algorithm.

^a Corresponding author :kingliyu@sina.com;

^b Corresponding author : An_gang@163.com;

^c Corresponding author :370713464@qq.com

2. System modeling

2.1. Analysis of complex logic control relations

In the design of MFD display and control system of aircraft avionics system, it is an extremely critical functional requirement to realize the accurate and accurate transformation between multi task and multi operating modes. MFD provides 19 main operating modes such as MENU, HSD, DATA, BIT, PDR, and so on. Each of the main operating modes contains several sub modes of operation. As an example, take part of the working mode and its conversion relation as shown in figure 1. The meaning expressed in Figure 1 is:

For example, in MENU mode, pressing the peripheral key L2 enables the MFD to switch to HSD mode, and press the peripheral key R5 in HSD mode to switch the mode to TCN.

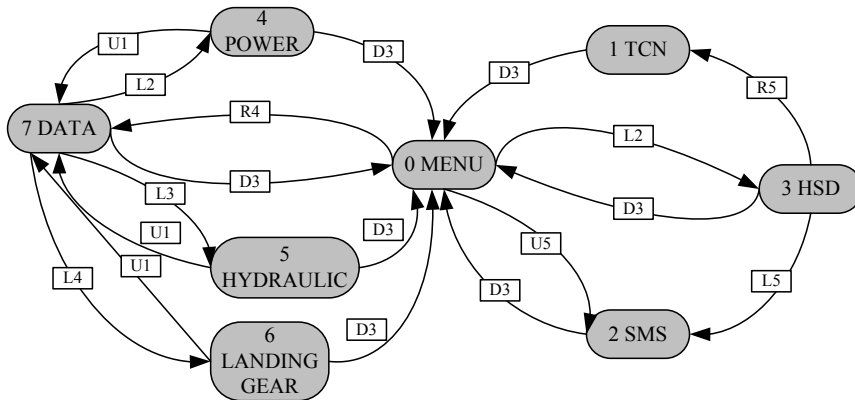


Figure 1. Work mode switching

When dealing with this kind of problem, the traditional multi branch conditional transfer statement can be implemented by the multi layer nesting algorithm[2]. The algorithm is a kind of algorithm without data structure support, and it has many limitations:

- From analysis to design to test the entire development process, it is not suitable for large-scale complex logic processing.
- Requirements change and function expansion are unavoidable problems in software development, and traditional algorithms are difficult to adapt.
- Due to different programming habits and different understanding of the problem, the traditional algorithm is usually designed code differences, it is difficult to form a standard algorithm to packaging, promotion and code reuse.
- For large-scale problems with complex logic relations, programs written by traditional algorithms have large amount of code, complicated nested relationships, and poor readability and maintainability of source programs.

To solve the above problems, an algorithm based on matrix modeling is designed:

- The large scale complex logic relation is represented by the data structure of the two-dimensional matrix.
- The complex logic control program based on matrix method is designed by accessing the program of two-dimensional matrix.

2.2. Matrix modeling of complex logic control relations

When modeling the working mode switch shown in Fig. 1, the operation mode and the press operation of the function keys need to be digitally abstracted so that the computer can recognize and process the operation. The digital process is:

- The 8 states shown in Figure 1 are numbered in accordance with 0~7.

- Press the peripheral keys U1~U5, L1~L5, R1~R5, D1~D5 when 20 key values in turn according to 0~19 number.
- This gives you the MFD state transition table as shown in table 1.
- Line represents the current working state of the MFD. The column represents the key value when pressing the peripheral key.
- The content determined by rows and columns is the number of MFD target states that should be switched to when a peripheral key is pressed in the current working state.
- "-1" stands for the corresponding peripheral key in this work state without the operation definition.

Table 1. State transition table

Working condition	The key value of the function key presses operation																			
	U 1	U 2	U 3	U 4	U 5	L 1	L 2	L 3	L 4	L 5	R 1	R 2	R 3	R 4	R 5	D 1	D 2	D 3	D 4	D 5
0	-1	-1	-1	-1	2	-1	3	-1	-1	-1	-1	-1	-1	7	-1	-1	-1	-1	-1	-1
1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	-1	-1
2	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	-1	-1
3	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	1	-1	-1	0	-1	-1
4	7	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	-1	-1
5	7	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	-1	-1
6	7	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	-1	-1
7	-1	-1	-1	-1	-1	-1	4	5	6	-1	-1	-1	-1	-1	-1	-1	-1	0	-1	-1

3. Software design

3.1. Matrix data structure

In programming, you should first solve the data structure problem when the matrix model is represented in the computer[3]. The simplest and direct method is to use a two-dimensional array, and the C language pseudo code is described as:

```
Int stateTable [total number of work modes] [key number].
```

In this case, it is represented as stateTable[8][20]. The data in the array is the contents of table 1. Thus, the stateTable array represents such a two-dimensional matrix as shown in figure 2.

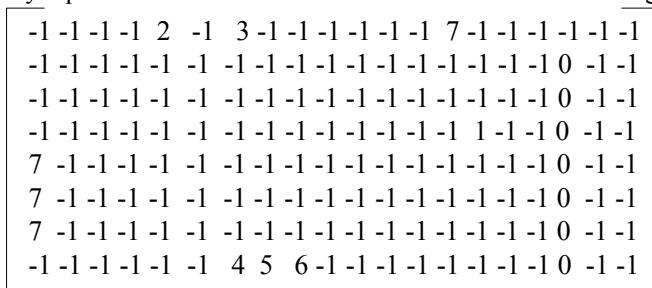


Figure 2. Two-dimensional matrix

The first dimension of each element subscript in the array represents the current working mode number of the MFD, and the second dimension represents the key value for the operation of the peripheral key. For example, the element stateTable[0][6] in an array has the meaning of:

In the 0 mode press the 6 key, MFD will switch to the working mode of stateTable[0][6]=3.

3.2. Design of complex logic control algorithm based on matrix mode

Different from the traditional multi branch program process to solve complex logic control problem, using matrix to deal with the problem, only the two-dimensional array stateTable storage MFD mode switch table release read and correct according to working mode and keys around the current operation is a realization way is very simple and effective. Algorithms using C language pseudo code are described below:

```
int stateAnswer( int curState, int keyValue)
{
    if ( stateTable [curState][keyValue] != -1 )
    {
        return stateTable[curState][keyValue];
    }
    else
    {
        Return curState;
    }
}
```

Description: stateAnswer function through the operation of the stateTable array, the corresponding switch to achieve the working mode. The parameter curState indicates the current working mode number, and keyValue represents the key number for the current push of the peripheral key operation. The return value indicates the target working mode number to be switched to.

```
void stateExplain( int stateID)
{
    switch(stateID)
    {
    case 0:
        MENU Program entry for work mode dependent processing;
        break;
    case 1:
        TCN Program entry for work mode dependent processing;
        break;
    .....;
    }
}
```

Description: stateExplain function, the completion of a variety of work mode number translated into the corresponding specific functions. The parameter stateID indicates the number of working states that need to be converted.

When the MFD display logic handler starts, you should set the default mode to the current mode, and then begin to receive instructions for the peripheral key. The default mode of operation is performed without operation, and the stateAnswer function is called to get the number of the next working mode. Finally, call the stateExplain function to perform the specific function of the new working mode.

3.3. Design method

According to the summary and abstraction of the simulation modeling and program design process and method of MFD display and control system, the matrix modeling and programming method is refined, as shown in figure 3.

System requirements analysis and function definition process, focusing on the complex logic transformation of the status and conversion conditions are analyzed in detail, resulting in a state transition diagram[4].

According to the state transition diagram, all States and trigger conditions are digitally abstracted and encoded, and a state transition table is formed, which is the basic work of software design[5].

According to the state transition table, the data structure of the matrix is defined by using the two-dimensional array.

Design control logic software to access the matrix.

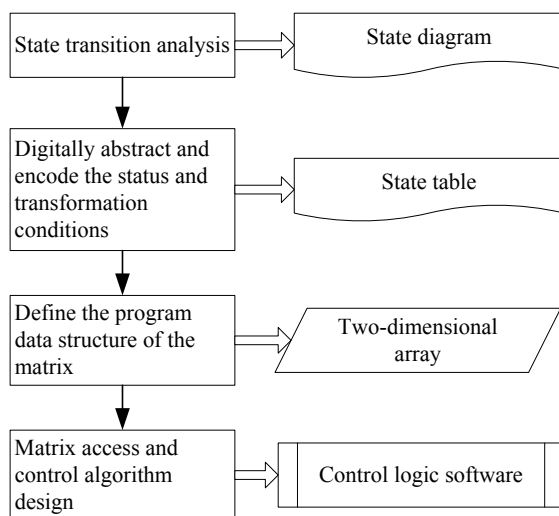


Figure 3. Matrix modeling and programming method

4. Performance analysis

4.1. Simplify the complexity of problem analysis

Complex logic transformation control relationships usually exhibit complex mesh structures (Figure 1), and it is often difficult for a programmer to make a correct judgment because of the limitation of thinking ability and scale of the problem in system analysis[6]. When using a matrix for system analysis, no matter how complex the logical relationship is, the programmer just needs to clear up each state and its trigger conditions, and then fill in the table of status transitions shown in table 1. Without the logical relationship between the depth and breadth of consideration and analysis, always at the same level to make the whole system analysis, the scale and complexity of problems, can greatly simplify the system analysis difficulty for solving large-scale complex logic control problem is particularly advantageous.

4.2. Easy to design, maintain, modify and extend function

Traditional procedural algorithms use complex branch conditional transfer to solve complex logic problems. The condition of judgment, the number of branches, the depth of nested hierarchy, and the length of program vary with the complexity of the problem. Thus, programmers have to write different programs according to different problems, even if the same problem has to be changed massively when the requirements change, the software upgrades or the functions are expanded. At the same time, the state is large and the logic relation is complex, which brings difficulties to the test. Using the matrix, you only need to modify the status response table and the stateExplain function. Thus, it is very easy to design, maintain, modify and extend the program. For example, suppose that MFD display control program was originally designed only shown in Figure 1 of the right part of "original design mode", this state response is shown in Table 1 of the 0~3 state, when the program needs to expand the left part of Figure 1 shows the functional expansion increased working mode "when simply fill in Table 1 can be (in Table 1 with shading to modify the content of the function

expansion, then) will be associated with the increase of the working mode of the program can be added to the stateExplain function in the entrance. The test switch just to check "state response table", and the test function can be stateAnswer, because the number of stateAnswer function statement, the algorithm is simple, therefore, both white box and black box testing are very easy.

4.3. Increase execution speed

In order to improve the efficiency of CPU implementation, modern computer architecture is designed using pipelining technology. A pipelining is a task decomposition technique that divides an instruction into several steps that are executed by different execution units in parallel[7]. Due to pipelining processing instruction set is different from the processing conditions for the transfer of the sequential execution of the instruction set, only when the instruction is executed to the "operation", in order to determine the next step should jump to the first few instructions, since this preprocessing work instructions will be adjacent to the waste, but also disrupted the normal flow of pipeline. Using the traditional multi branch conditional transfer nested algorithm, it is easy to waste the preprocessing because the conditional transfer statement is many and the nesting level is deep. The complex logic control algorithm based on matrix modeling is directly positioned to the target state by array index, so it can avoid the occurrence of this situation and improve the execution speed.

4.4. Analysis and solution of the contradiction between execution speed and storage space

In the design of computer algorithms, time and space are always a pair of contradictions. The complex logic control algorithm based on matrix modeling improves the efficiency of program execution. But it must be admitted that due to the use of two-dimensional matrix to represent the logical conversion relationship, the computer storage space has to be increased. Therefore, the algorithm is a programming algorithm designed to sacrifice space for efficiency[8]. However, with the rapid development of computer hardware technology, no matter the speed of CPU execution or memory storage capacity, it is no longer the key to the development of software. Single board computer or mobile phone even as automatic teller machines that, for space utilization is strict, the two-dimensional representation of state transition matrix usually appears in sparse array form, so it can take compression array storage, which fall in the value of -1 compression matrix without operation defined in space and space efficiency in contradiction to seek compromise. Or change the data type, change the plastic number to bit mode, and save space.

5. Conclusions

Through practical application in an aircraft engineering simulator and theoretical analysis of its performance, it is demonstrated that: The complex logic control algorithm based on matrix modeling is very effective for solving large-scale complex logic control problems. It makes every link of analysis, design and test program design simplified and standardized. The algorithm is simple and readable, and can be used as a standard program algorithm for dealing with such problems.

References

1. Soares M D S, Vrancken J. Model-Driven User Requirements Specification Using SysML[J]. Journal of Software, 2008,3(6).
2. J Fisher. Model-Based Systems Engineering: A New Paradigm[J]. INCOSE Insight, 1998,1(3).
3. Thomas Johnson, Christian J. J. Paredis. Integrating Models and Simulations of Continuous Dynamics Into SysML[J]. Journal of Computing and Information Science in Engineering, 2012,12(3).

4. Cai Shubin, Ming Zhong, Li Shixian, Liu Xianming. ontology based model integration. Journal of Electronic Science and technology, 2009 (4).
5. One day, Zhang Yan, Wang Linzhang, Li Xuandong, in science, design pattern modeling and model transformation based on MDA. Journal of software engineering, 2008.19 (9).
6. Zhou Yi, Liu Lei, fan Ren hong. Ontology mapping method based on pattern structure classification [J]. Journal of electronics. 2011 (04).
7. Li Lanying, Wang Feng. Research on scheduling algorithm of reconfigurable system [J]. Computer engineering and application. 2014 (22).
8. Zhang Jiafeng. Modeling and verification of reconfigurable discrete event control systems [D]. Xi'an Electronic and Science University. 2015.