The test of Particle Filter Simulator software aims to discover the functional and service performance-related problems through variety test approaches. According to apposite approaches, basic errors and drawbacks on functions and logic can be detected and reduced. Ensure that software can implement its functions as expected.

In this software, the whole functions will be tested including importing a json file, exporting data, drawing chart of algorithm running result, refreshing the chart and altering the parameter value of Particle Filter algorithm. The correctness of algorithm implementation will be tested particularly. Plus, parts of non-functional features which are significant to the software will be tested as well, for instance, system compatibility, installation and uninstallation, and performance in response time.

Pass/Failed criteria

For functional features, the pass criterion applies to all of them: implement correctly without any error and warning. And if the condition do not reach, the feature is failed.

For non-functional features, pass or fail criteria should be considered separately.

* System compatibility: The criterion of system compatibility is whether the software can implement its functions well as a cross-platform software. Two mainstream operating systems must be satisfied, Windows and Mac OS.
* Installation: installation pass criterion is verification that software works properly after installation. It is failed if installation error occurs.
* Uninstallation: the pass criterion of uninstallation is the software can be uninstalled clearly and has no harmful effects on the system.
* Performance(response time): Miller(1968) indicated there are three levels of response time for users. 0.1 to 0.2s: users may think it is immediate response.1~5s: The users feel that the interaction is basically smooth. The users will notice the delay. 8s or more: Users will follow the dialog. A prompt or progress bar is required to confirm that the system is still in process. Treat the information as a reference, the pass criterion is set to most response time of operations should be under 5s, otherwise, a progress bar should be showed to tell users the computer does work. Over 5s and no progress bar or other prompt will be considered as failure.

Approach

The test will use black box testing to check functions with equivalence partition testing and boundary testing, while white box testing and unit testing to check the source code. Matlab will be applied to support mathematical principles.

Performance will be tested by Chrome devtools to present the response time of every operation. All test cases are manual tests.

Test case

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ID | Category | Aim | Test steps | Expected result | Actual result | Pass/Fail |
| 1 | Code correctness | Check the correctness of randn function in Algorithm.js | 1. Execute randn function by variety parameter. 2. Output every data in returned matrix mat. 3. Input these data into Matlab as a matrix. 4. Execute jbtest in Matlab. | Jbtest shows ans = 0 | As expected | Pass |
| 2 | Code correctness | Check the correctness of other function in Algorithm.js | 1. Insert a line of code to output the return value to console. 2. Give actual value of parameters and call the function. | Output data is the same as expected | As expected | Pass |
| 2 | Install | Check if software can work after installation | 1. Double click installation package and install the software. 2. Open PF simulator. | 1. Install successfully. 2. Software can work well. | As expected | Pass |
| 3 | System compatibility | check if software is supported by Windows and MacOs | 1. Install the software on Windows and MacOs. 2. Open software check the UI and functions. | 1. Software can be installed on different systems. 2. UI and functions work correctly. | As expected | Pass |
| 4 | Functions | Software is able to implement PF algorithm and show result with a chart | Click ‘start’ button to running the algorithm by presetting data. | 1. Two lines in the chart. 2. There are different charts after clicking ‘start’. | As expected | Pass |
| 5 | Changing ‘Particles’ value in ‘Parameter Setting’ will change data running in algorithm | 1. Insert console.log(data) in source code to show the number of particles. 2. Change particles value to 200 and click ‘Start’ in software. | Data showing in console is 200. | As expected | Pass |
| 6 | Changing ‘Initial Noise Covariance’ value in ‘Parameter Setting’ will change data running in algorithm | 1. Insert console.log(data) in source code to show the value of Initial Noise Covariance. 2. Change Initial Noise Covariance value to 50 and click ‘Start’ in software. | Data showing in console is 50. | As expected | Pass |
| 7 | Changing Process Noise Covariance value in ‘Parameter Setting’ will change data running in algorithm | 1. Insert console.log(data) in source code to show the value of Process Noise Covariance. 2. Change Process Noise Covariance value to 20 and click ‘Start’ in software. | Data showing in console is 20. | As expected | Pass |
| 8 | Changing Measurement Noise Covariance value in ‘Parameter Setting’ will change data running in algorithm | 1. Insert console.log(data) in source code to show the value of Measurement Noise Covariance. 2. Change Measurement Noise Covariance value to 30 and click ‘Start’ in software. | Data showing in console is 30. | As expected | Pass |
| 9 | Save chart | Click the button ‘Save as ’on the top of chart. | Chart can be saved as a png file in local. | As expected | Pass |
| 10 | Export data | Click ‘Export data’ in menu ‘File’. | Related data will be save as a json file. | As expected | Pass |
| 11 | Import a json file | 1. Click ‘Import a json file’ in menu ‘File’. 2. Click ‘Refresh’. | Chart will be drawn by given data in json file. | As expected | Pass |
| 12 | Software window always on top | 1. Click ‘Always on Top’ in menu ‘View’. 2. Open other windows in the screen. | Particle Filter Simulator will always on the top and not be covered by other windows. | As expected | Pass |
| 13 | Show ‘Help’ window | Click ‘Help’ in menu ‘Help’ | A new window is created and show help document. | As expected | Pass |
| 14 |  | Show ‘About’ message dialog | Click ‘About’ in menu ‘Help’ | A message dialog shows including brief introduction and version information | As expected | Pass |
| 15 | Quit the software | Click ‘Quit’ in menu bar | Quit the software | Under MacOs system, window is closed but not quit. | Fail |
| 16 | Performance | Test the response time of drawing chart with default data | 1. Open chrome-devtools and choose ‘Performance’ to start record. 2. During recording, click ‘Start’ button and waiting the chart. 3. Repeat step 2 for 5 times. | The response time is less than 5s, otherwise there should be a prompt. | Average time is 1530ms. | Pass |
| 17 | Test the response time of drawing chart with max data | 1. Set all parameter values to maximum. 2. Open chrome-devtools and choose ‘Performance’ to start record. 3. During recording, click ‘Start’ button and waiting the chart. 4. Repeat step 2 for 5 times | The response time is less than 5s, otherwise there should be a prompt. | Average time is 7963.6ms and no prompt. | Fail |
| 18 | Uninstallation | uninstall the software | Uninstall | No components remain and no effect to system | As expected | Pass |
|  |  |  |  |  |  |  |

To sum up above test cases, there are no logic and functions error in the software. The major issue is that it is slow to deal with complex data. Due to the data from chrome-devtools, the main time is for scripting. After investigation and analysis, this kind of slow is limited by JavaScript. There is no library for complex mathematical computation, therefore, amounts of basic computation must be written in code. It leads to the slow running time.

To solve the issue, reducing the number of calculations is the main direction. In order to achieve this goal, we are try to get a more concise algorithm and find a powerful mathematical library is needed. Maybe in next update, it will have a better running time. Other approaches like changing the way of declaration or loop to reducing calling time have a role as well and we are changing them gradually.

Miller, R. B. (1968). Response time in man-computer conversational transactions. *national computer conference*.