International Journal of Innovative Computing, Information and Control
Volume **9**, Number **12**, December **2013**

ICIC International ©2013 ISSN 1349-4198
pp. 4605–4618

RIM: RISK INTERACTION MODEL AND ITS APPLICATIONS

Lixia Meng[1], Linmi Tao[2] and Fuchun Sun[2]

[1]Army Aviation College
[2]Department of Computer Science and Technology
Tsinghua University
Haidian District, Beijing 100084, P. R. China
menglixia2000@yahoo.com.cn; { linmi; fcsun }@tsinghua.edu.cn

Received August 2012; revised January 2013

Abstract. *In human computer, machine or robot interactions, human errors during operation or interaction could lead to risks. Existing computer systems and applications are designed to reduce the losses due to human interactive errors via reversible operations. Unfortunately, reversal operation is not applicable for many interactive systems, such as vehicle driving, aircraft control. In these systems, operation errors often lead to serious consequences. A novel human-computer interaction model, termed risk interaction model (RIM), is proposed for quantitative evaluation of the risk of interaction for interactive systems. Factors for analyzing risks in interaction are summarized and the risks are structured as original risk, basic risk, operational risk and task risk. The risk of interaction in a system is formalized based on the analysis, and the parameters are calculated in experiments. Two application instances are used to validate RIM.*
Keywords: Human robot interaction, Risk interaction model, Auto-driving system

1. Introduction.

Human error occurs during action and decision-making in human computer interactions. Many studies have investigated human error in the field of cognitive science [4, 7, 8]. According to James Reason, user operation errors (mistakes, lapses and slips) can lead to risks, but he did not quantitatively model the consequences of human errors. To the best of our knowledge, there have been no studies on interaction risk for human robot interaction (HRI) and human computer interaction (HCI), since computer systems are designed to reduce losses due to human interactive errors via reversible operations. Unfortunately, reversal is not applicable for many interactive systems, such as vehicle driving, aircraft control, and management of large factories. In these systems, operation errors can lead to serious consequences. Therefore, the interaction risk for these systems has to be modeled and carefully calculated. In other interaction systems, risks caused by human error are relatively low, but quantitative calculation and comparison of risks are necessary.

In current HCI theories, models such as GOMS [5], PIE [2], Fitts Law [3, 9] and keystroke-level models [1, 6] focus on the operation efficiency for interaction and do not consider the consequences of interaction. Therefore, a theoretical model is needed to evaluate risks, independent of the interactive mode. A risk interaction model (RIM) for quantitative assessment of system risk is proposed.

We propose two examples of applications of the model. The first is an interactive vehicle driving system based on BCI (brain-computer interface). The system risk is irreversible and catastrophic. Detailed risk analysis and calculation using model are given in this paper. The second is the image annotation task based on mouse operation. The example shows that the interactive operation in general will cause errors and risks, and the method of errors and risks analysis using model is given in detail.

RIM is defined and formalized in Section 2. Two application instances are proposed in Sections 3 and 4. In each instance, a detailed system risk calculation using RIM is compared with the experimental results. Conclusions are presented in Section 5.

2. Risk Interaction Model.

RIM is a task-oriented interaction model. Operation risk is calculated after task decomposition into a series of operations, and the system risk can be theoretically assessed through the analysis of task risks.

2.1. Risk analysis.

The main factors in generating and influencing risks are:

- **System state $\boldsymbol{s}$**: Each system state has an inherent risk, and each operation may produce state transformation. Transformations between different states will lead to additional risks.
- **Risk probability $\boldsymbol{p}$**: This is the probability of risk generated by one operation, in other words, the probability of operation error. Its value is decided in the system design or interface design.
- **Operation urgency $\boldsymbol{u}$**: This is the time between an operation request and its performance. The shorter the time is, the more difficult is successful completion of the operation and the more prone it is to risk. Operations requiring a higher response speed (such as brake operation) are urgent.
- **Cognitive load $\boldsymbol{c}$**: This is the number of continuous operations executed before a new operation. The greater this number is, the more fatigued the user is, and the greater the risk is.

The risk structure of RIM is shown in Figure 1.

The various risks are as follows:

- **Original risk $\boldsymbol{R_0}$**: inherent risk for a correct operation in normal system operation. It is usually defined by an expert according to the system state and the actual situation.
- **Basic risk $\boldsymbol{R_1}$**: risk value for an operation due to operation error or system error. The risk is decided by R_0 and the corresponding risk probability p.

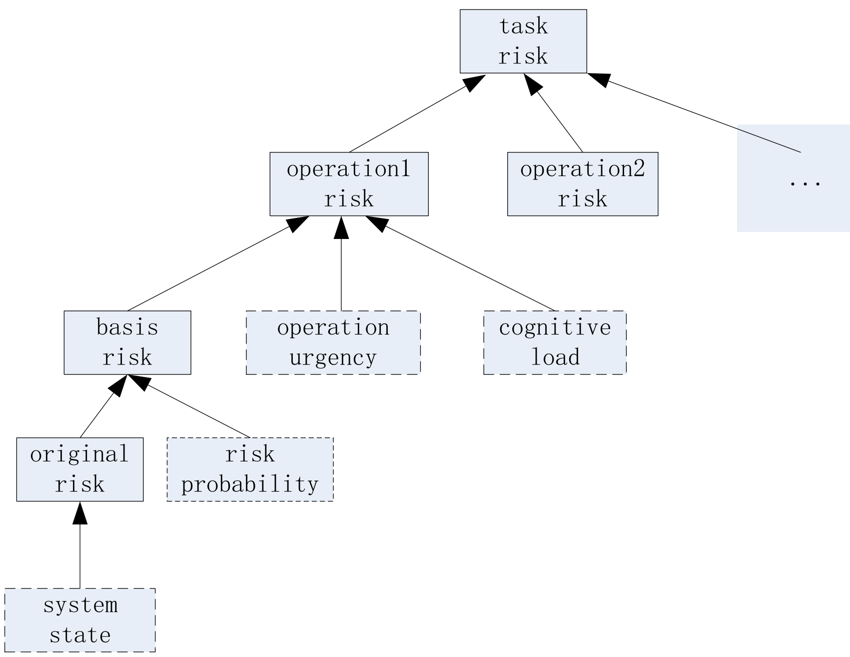


FIGURE 1. Risk structure of RIM

- **Operation risk R_2**: final risk value for an operation. It is affected by operation urgency u and the user's cognitive load c based on R_1.
- **Task risk R_T**: risk value for the entire task, determined by all operations arising from task decomposition.

2.2. Formalization of RIM.

Definition 2.1. *RIM is a tuple $M = (S, O, R_T, T)$ where S is a set of states of the system, T is the whole task that needs to be completed by the interaction system, R_T is the task risk and O is a set of atomic operations arising from task decomposition, defined as:*

$$O = \left\{(O_l, O_p, O_r)^i | i = 1, 2, \dots\right\} \tag{1}$$

where O_l is the operation label, $O_p = \{p, u, c\}$ is a set of properties of the operation, $O_r = \{R_0, R_1, R_2\}$ is a set of risk values of the operation.

For instance, an interactive auto-driving car is running on a small road straightly, the current state S is "running straightly". At a time, the user wants to stop the car, and sends a command to the auto-driving system. When the system received and confirmed this "stop" command, the system current task T is "to stop the car at some place". Unfortunately, the car cannot be stopped immediately on the center of a small road. The nearest parking lots is on the left, only a hundred meters away which is estimated based on the current location and in-car map. Therefore, the task T is decomposed into a set of atomic operations O, which includes "go straight", "turn left", "slow down", etc. Finally, the car is stopped, and the state S is "still" now.

Risk values are calculated as follows:

$$R_1^i = \sum_j (R_0^j \times p_i^j) \quad \left(\sum_j p_i^j = 1\right) \tag{2}$$

where p_i^j is the probability that the wrong operation j will be carried out instead of the intentional operation i.

$$R_2 = R_1 \times K_1(u) \times K_2(c) \tag{3}$$

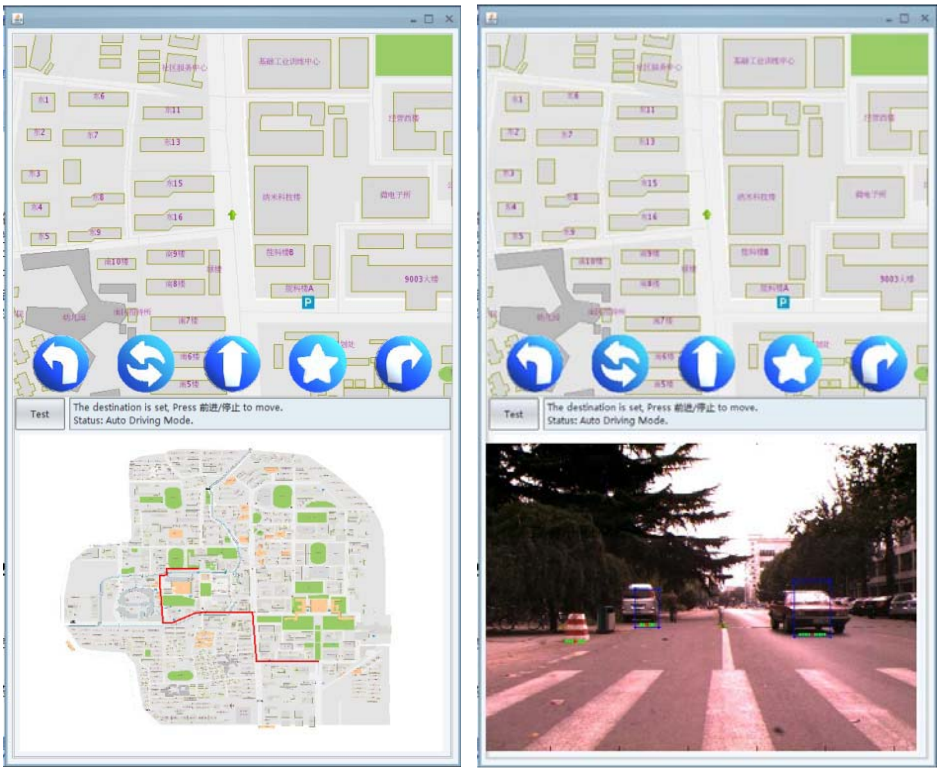
where $K_1(u)$ is a penalty function for operation urgency u and $K_2(c)$ is a penalty function for cognitive load c.

$$R_T = \frac{\sum R_2^i}{n} \tag{4}$$

where n is the number of operations arising from task decomposition.

3. Application Instance 1.

3.1. Application background. An interactive auto-driving system is an outdoor vehicle system for disabled and elderly individuals. Users can interactively control the system by defining the destinations and indicating the paths along which to travel. The vehicle will automatically drive itself to the destination via the path planned by the system. Users can also operate the automatic driving system via the interactive interface at any time. For instance, users can change their destinations, or want to stop at some place. In this system, operation errors can lead to serious consequences. RIM is used for quantitative assessment of system risk bellow.



(a) (b)

FIGURE 2. Interaction interface

3.2. **Interaction operation and interface.** The interaction interface of the system is shown in Figure 2. The driver inputs their intention by BCI (brain-computer interface) system. Only five buttons are designed on screen to complete the whole operation for the limited input of BCI. The buttons are "mode switch", "forward/stop", "left", "right", "destination" and "display switch".

To reduce operation risks caused by human error, we designed two driving modes, auto mode (automatic driving by the system based on global path planning) and free mode (the driver may intervene during travel). The user can switch between them by choosing "mode switch". Only "forward/stop" operation is allowed in auto-mode driving. The vehicle will automatically turn at the next junction when the user chooses "left" or "right".

If "destination" is chosen, the intelligent vehicle system maps out a minimum-risk path instead of the normal shortest path based on the vehicle's current location and the electronic map, as shown in Figure 2(a). The vehicle will reach the destination along this path in auto-mode driving. To change the path, the driver must choose "stop" first and then "destination", or else choose "mode switch" to select free driving mode, in which turning operation is allowed. Meanwhile, the video from the camera mounted on the vehicle is shown in the interaction interface (Figure 2(b)).

3.3. **System risk analysis.** In this system,error operation can lead to serious consequences. For example, the vehicle execute command "turn left" and crash instead of exactly "turn right" because of operation error. In these navigation tasks, risks come mainly from the wrong input of the BCI system. In addition, continuous operation will increase cognitive load, and it will also increase risks.

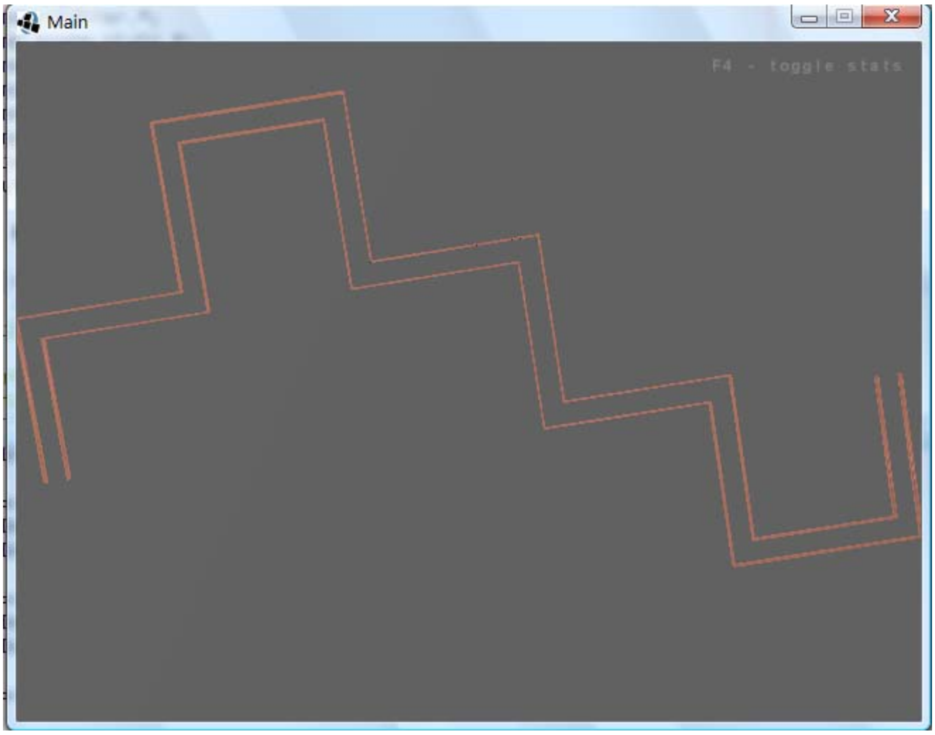


FIGURE 3. Road platform for $K_1(u)$ analysis

3.3.1. *Function parameter analysis in RIM.* The characteristics of the two penalty functions $K_1(u)$ and $K_2(c)$ were further analyzed experimentally.

We implemented a Java 3D vehicle driving simulation program. The user controls the vehicle via driving operations on a road displayed on a screen. A road map can be drawn, including inflexions, as read from a text file. The road bends are 90 and can be used to preserve as much familiarity with the turning angle as possible.

Driving operation involves clicking the up-arrow button to control vehicle movement or to stop. The left-arrow button is clicked to turn left, and clicked again to complete a vehicle turn and to continue in a straight line. Turning right is similar to turning left.

Function $K_1(u)$ analysis:

As shown in Figure 3, the road for $K_1(u)$ analysis had 10 turns. The straight parts are long enough to ensure that users have enough time to rest after turning each corner to avoid a cognitive burden. The turning speed is variable, the same as for the straight part, to ensure a perfect driving experience. In experiments, 10 subjects simulated driving at different turning speeds (i.e., different u). A score of 10 was assigned for successful completion of the road network and a score of $n-1$ for failure at the nth turn. The car is reset if an error occurs. Each subject performed five tests and the mean results are shown.

The experimental results are shown in Figure 4. The rate of decrease in the score is relatively small as the turning speed increases (decerasing u), so the risk is less. As the turning speed continues to increase, the score then rapidly decreases at an inflexion point corresponding to $u = 1.0s$, corresponding to a significant increase in risk. When u is greater than a threshold, the operation time is sufficient, the operation error rate is low and the operation risk is small. For times less than the threshold, the operation time is insufficient, the operation error rate greatly increases, and the operation risk is greatly affected by u.

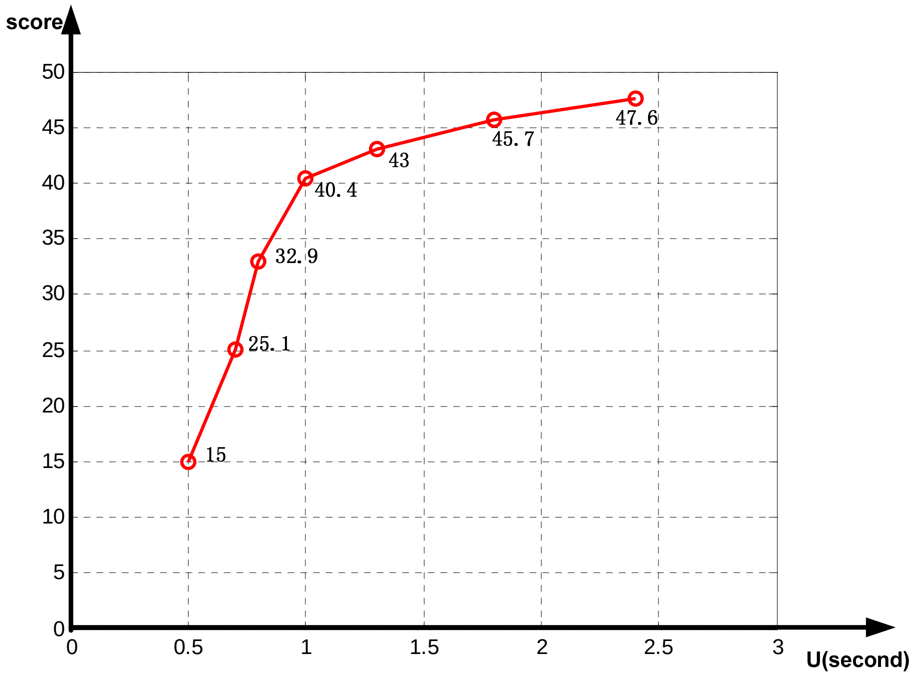


FIGURE 4. Experimental results for $K_1(u)$ analysis

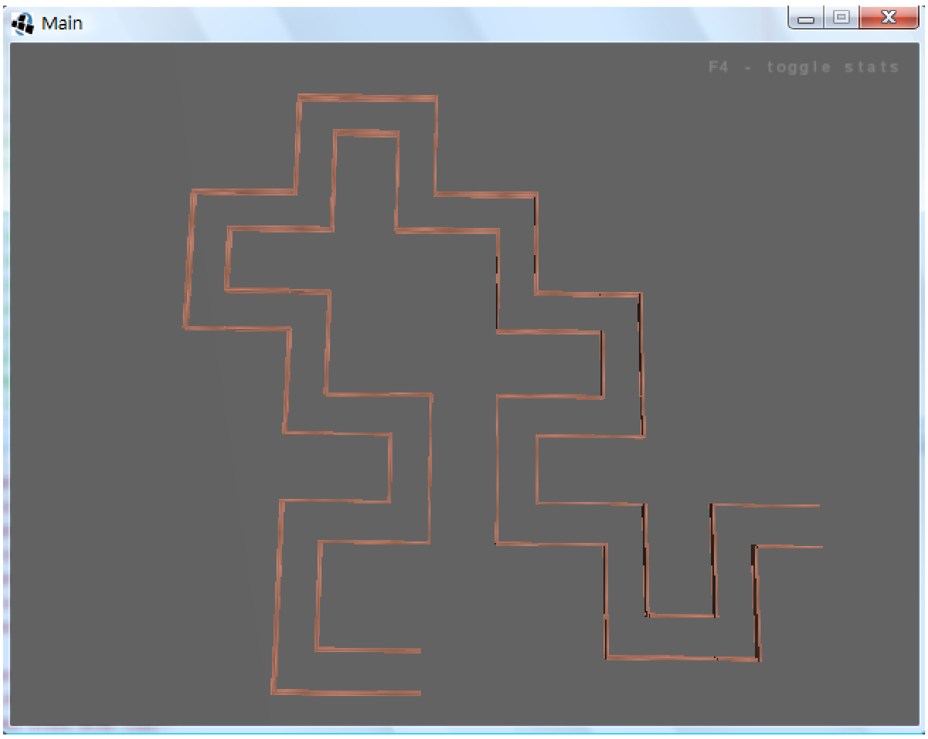


FIGURE 5. Road platform for $K_2(c)$ analysis

Function $K_2(c)$ analysis:

As shown in Figure 5, the road for $K_2(c)$ analysis had 22 corners. The length of the straight parts is short enough to ensure that users have to turn immediately after the previous turn to increase the cognitive load. The turning speed is fixed (i.e., u fixed) at a moderate level for continuous operation and a good driving experience. In experiments,

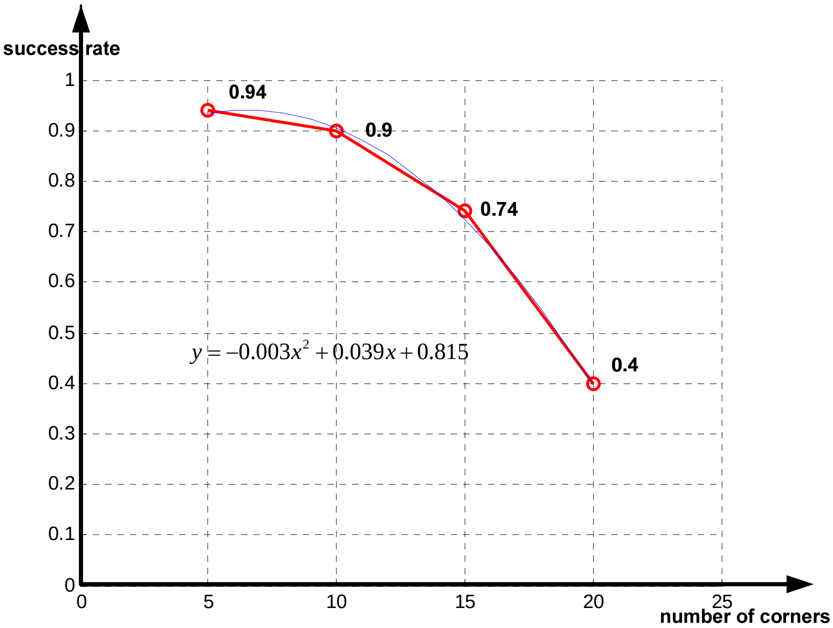


FIGURE 6. Experimental results for $K_2(c)$ analysis

10 subjects simulated driving and the number of successes were recorded at the fifth, 10th, 15th, 20th and 25th corners. The car is reset if an error occurs. Each subject performed five tests and the mean results are shown.

In Figure 10, the horizontal axis represents the number of corners successfully turned and the vertical axis represents the success rate. It is evident that because of the continuous operation required, the user's cognitive load and the operation error rate both increase; consequently, the risk also increases. In addition, as can be seen from the graph, a curve is fitting the points very well. This proves that the increase in cognitive load is not linear, but is similar to a power law function. Thus, further risk is created as the cognitive load increases.

3.3.2. *Interaction tasks in experiments.*

Task 1: Starting from the east gate of Tsinghua University, proceed to the west gate and stop at ERXIAOMEN intersection once using auto-mode driving;

Task 2: Starting from the west gate of Tsinghua University, proceed to the north gate of the library using auto-mode driving.

Task 3: Starting from the north gate of the library, switch to free-mode driving at the EASTMAINBULDING intersection (the driver may choose his own route, but if the time taken for the task is too long, this results in task failure).

The three tasks are representative since they include all possible operations in the system using both auto-mode and free-mode driving with minimal cost.

The interaction flow chart is shown in Figure 7. Solid arrows indicate a short time interval between two operations, which can be regarded as a continuous operation for which the cognitive load will gradually increase. Dotted arrows indicate discontinuous operations for which there is no increase in cognitive load.

3.3.3. *Calculation of various risks.* Risk does not arise when the vehicle is immobile in this system. Thus, risks are only generated by system commands via the main interface. Therefore, we only discuss risks for the five buttons on the main menu.

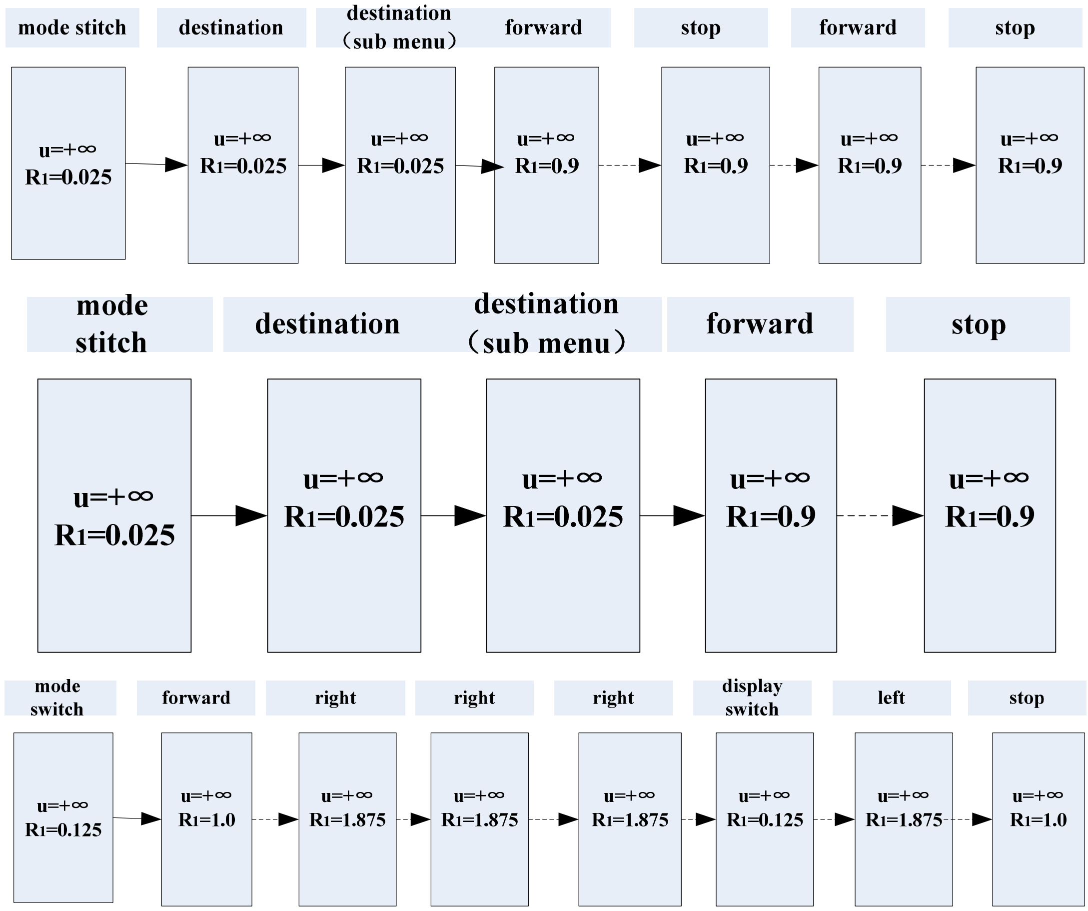


FIGURE 7. Interaction flow chart of Task 1, Task 2 and Task 3

- **Original risk R_0:** A transition in the system state occurs when "forward/stop", "left" or "right" is clicked. We define $R_0 = 1$ for "forward/stop" because the system transition is between immobile and mobile, $R_0 = 2$ for "left" and "right" because the system transition is between mobile and mobile, and $R_0 = 0$ for other commands.
- **Basic risk R_1:** The input accuracy is approximately 90% for the brain-computer interface, and the remaining 10% is recognized as an error, randomly assigned to the other four commands. Basic risks arising from the command operation via the interface buttons can be calculated according to Equation (2), which are shown in Table 1.
- **Operation risk R_2:** In the system design there are no time limits for any operations. The BCI is assigned a long response time (2-4s seconds) and urgent operations are handed over to the sensors (radar, cameras and other intelligent devices) on the vehicle. Thus, $K_1(u) = 1$ and $K_2(c) = a^c$ in Equation (3) for any operation in the system, where $a = 1.1$ in our experiments.
- **Task risk R_T:** The task risk depends on the specific task and is calculated according to Equation (4).

The risk for the three tasks described above (R_{T1}, R_{T2}, R_{T3}) was calculated. The results are shown in Table 2.

The results show that R_{T1} and R_{T2} are small and of similar magnitude. This is because the main operations are static operations with low risk and forward/stop operations with medium risk in tasks 1 and 2. Furthermore, continuous static-state operations occur with less risk. R_{T1} is slightly greater than R_{T2} because task 1 has more forward/stop operations

TABLE 1. Basis risk of five commands

R_1	immobile	mobile (auto-mode)	mobile (free-mode)
forward/stop	0.9	0.9	1.0
left	0.025	0.025	1.875
right	0.025	0.025	1.875
mode switch	0.025	0.025	0.125
destination	0.025	0.025	0.125

TABLE 2. Task risk

Task	**Risk**
1	0.5687
2	0.4361
3	1.2313

TABLE 3. Operation errors and its probability

	Task 1	Task 2	Task 3
Number of error operation Per capita	0.6	0.4	1.1
Number of error operation Per capita with risk	0	0	0.9
probability of Error operation	0.086	0.08	0.138
probability of Error operation with risk	0	0	0.113

with relatively high risk than task 2, which reflects the high sensitivity of the proposed model.

R_{T3} is much greater than R_{T2} and R_{T1} and is close to three times R_{T2}. The reasons are (i) the main operations in task 3 are high risk, such as turn right and turn left and (ii) task 3 involves free-mode driving. The basic risk in free-mode driving is much greater than that in auto-mode driving, as shown in Table 1. The quantitative results reflect the qualitative analysis of the actual situation.

In practice, free-mode driving will occasionally be used when somewhere is not reachable by auto-mode driving. Thus, auto-mode driving (similar to tasks 1 and 2) is the predominant driving mode. Therefore, our interactive system has lower risk and better interactive safety according to model comparisons.

3.3.4. *Simulation experiment and analysis.* To assess the usability of the model, we carried out a user test and compared the results with model calculations. For the risks in real automatic driving system, we designed a simulation system by collecting the real situation of the environment in the test driving tasks.

Ten users performed simulated driving to complete the three tasks mentioned above. The goal was to identify possible operation errors leading to risk by observing user operations, especially errors. During the tests, the number and position of user operation errors were recorded, together with whether the errors involved risk. For example, in the three tasks, when an operation occurs during a static operation, it does not lead to risk. On the contrary, when an operation error occurs during a dynamic operation, it leads to risk. Test results for the 10 users are shown in Table 3.

As shown in Table 3, the number of operation errors per capita and the probability of operation errors are relatively low for all the three tasks. Operation errors with risk are basically avoided during auto-mode driving (tasks 1 and 2), while a few operation errors

Table 4. User survey

Factor	Score
Safety	4.25
Convenience	4.07
Fatigue	4.85

occur during free-mode driving (task 3). However, since the probability of free-mode use is low in practice, this risk has little effect on system safety. In addition, the number of incorrect operations increases with road unfamiliarity.

In addition, a user survey of subjective feelings (safety, convenience, fatigue) provided important information. The score results (range 0-5) are shown in Table 4. The users assigned high scores to all three parameters.

According to our test results, the interactive auto-driving system is safe and has a user-friendly interaction interface.

Model analysis and users test proved that the interactive auto-driving system has relatively low risk and high safety according to risk calculations and statistical error probability. Model analysis revealed that risk is lower in auto-mode than in free-mode driving, which echoes the probability distribution of operation errors with risk from the user tests. This confirms that RIM is a valuable application.

4. **Application Instance 2.**

4.1. **Application background.** Classification is a nature intelligence of human beings. In areas of computer engineering, a classifier is trained with labeled samples. Annotation of training samples is a precondition of pattern recognition while we have a large number of unlabeled samples. The task of this experiment is to annotate images as training samples for image classification. The pictures used in this experiment are downloaded from the Corel image library (http://wang.ist.psu.edu). The annotation task is to draw the images into 10 categories, 100 pictures per category. The subjects are asked to classify all the images in a limited time. RIM is used to analyze and calculate the value of risks generated from error operation.

4.2. **Interaction operation and interface.** The task of annotation in this experiment is to click and drag a certain number of pictures to the corresponding category folders in a certain period of time. The interface of the operation is as shown in Figure 8. Here the operation is the click and drag mouse operation. The task of annotation of 1000 pictures requires 500 such operation orderly. So the task was decomposed into 500 operations.

4.3. **System risk analysis.** If the picture annotation is a mistake, and take these wrong samples for training, the accuracy rate of the classifier will decline. Therefore, the operation error consequences will bring risks. We use the wrong picture ratio to measure the risk value of task. In these tasks, risks come mainly from the following aspects. First, due to the user's error operation, the picture is dragged to adjacent folder. Second, as a result of the operation time requirements, the emergency degree of operation will increase the wrong probability. Finally, the increase of quantity of task makes cognitive load increases, it will also increase risks.

4.3.1. *Function parameter analysis in RIM.* In this section, we experimentally determined $K_1(u)$ and $K_2(c)$ in Formula (3).

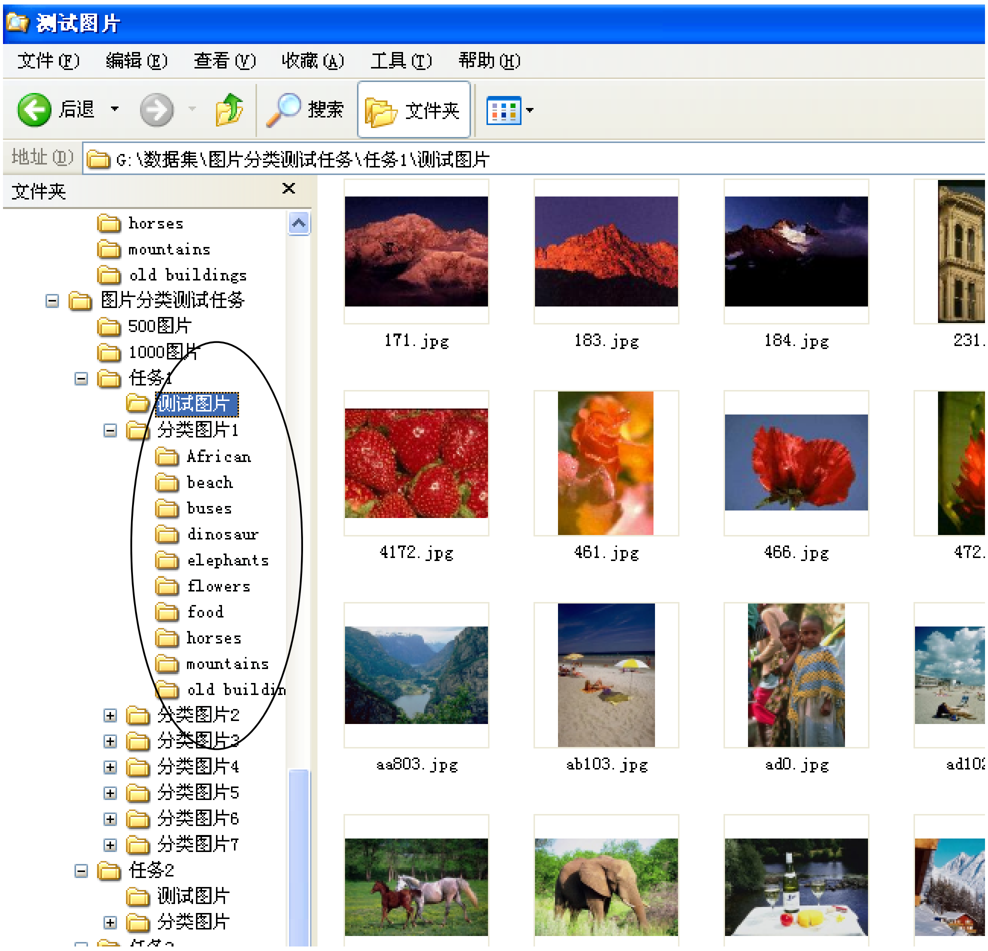


FIGURE 8. The interface and operation

Function $K_1(u)$ analysis:

Experimental process: Use the timer to limit task time, drag 500 pictures to the 10 category folders respectively by mouse, record the number of wrong pictures manually and calculate picture error ratio to measure the risk value. The same task is made 6 times, and each time shorten the task time constantly in order to increase the operation urgent degree. The mean test results of the five subjects are as shown in Figure 9.

In Figure 9, horizontal axis represents the time required for each operation. The ordinate is the classification accuracy rate, the corresponding error rate could be calculate by 1 minus accuracy rate and it is directly proportional to the risk. Apparently, the urgent the operation is, the less the accuracy rate is and the greater the error rate is. The fitting curve is $y = -0.1044 \times 10^{-8} \times x^2 + 0.6696 \times 10^{-5} \times x - 0.1284$.

Function $K_2(c)$ analysis:

Experimental process: drag 1000 pictures to the 10 category folders respectively by mouse in about 30 minutes, record the number of wrong pictures manually and calculate the corresponding picture accuracy ratio of the first $100, 200, \ldots, 1000$ pictures. The increasing number of pictures will increase the user's cognitive load and increase risks too. The mean test results of the five subjects are as shown in Figure 10.

In Figure 10, horizontal axis represents the number of pictures, the ordinate is the corresponding accuracy rate. The curve shows that the accuracy rate is slowly declining with the increase of the number of the pictures. The fitting curve is $y = 2.6048 \times 10^{-8} \times x^2 - 4.8335 \times 10^{-5} \times x + 0.9328$.

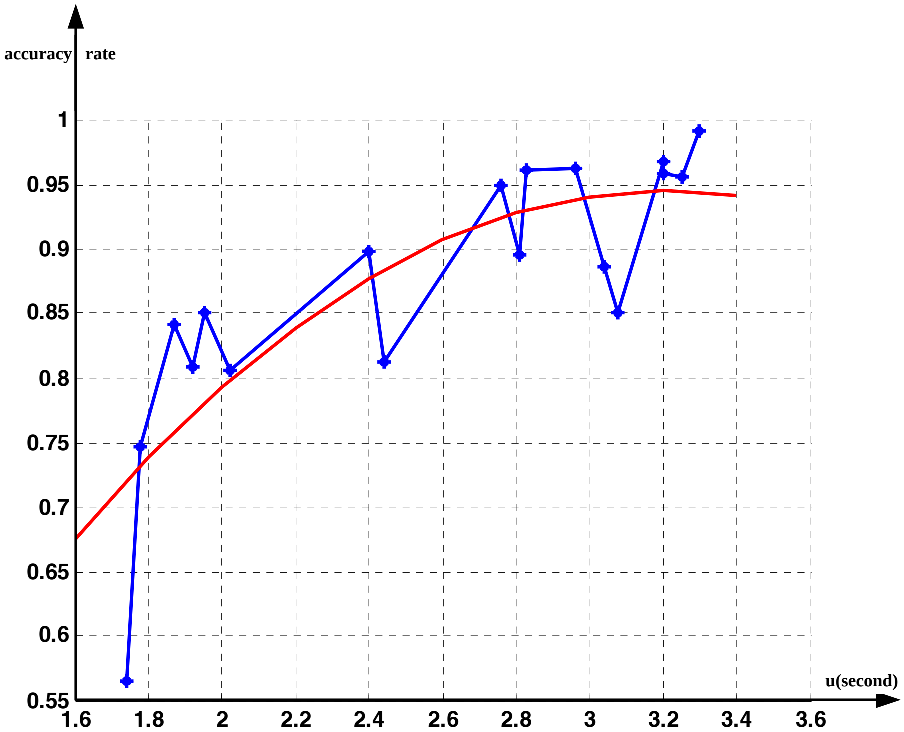


FIGURE 9. Experimental results for $K_1(u)$ analysis

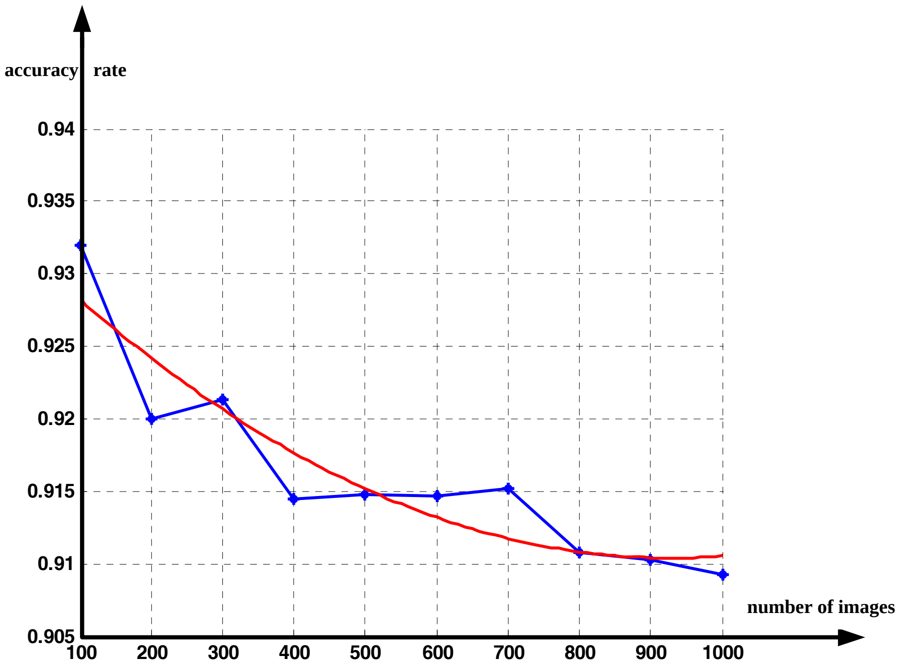


FIGURE 10. Experimental results for $K_2(c)$ analysis

4.3.2. *Interaction tasks in experiments.*

Task 1: drag 500 pictures to the 10 category folders respectively by mouse in about 15 minutes;

Task 2: drag 1000 pictures to the 10 category folders respectively by mouse in about 30 minutes;

TABLE 5. The calculation results of task risks

Task	Risk
1	0.0218
2	0.0230
3	0.0316

TABLE 6. The average error rate

Task	error rate
1	0.1232
2	0.1511
3	0.1916

Task 3: drag 1000 pictures to the 10 category folders respectively by mouse in about 25 minutes.

The three tasks is representative. The operation emergency degree is the same in Task 1 and Task 2, but the cognitive load of Task 2 is two times of Task 1. The cognitive load is the same in Tasks 2 and Task 3, but the operations of task 3 are more urgent than those in Task 2.

4.3.3. *Calculation of various risks.* In this section we calculated task risks of the above 3 tasks using the risk interaction model. For each operation, assuming that the original risk $R_0 = 1$ and a probability of 1.5% for dragging into the wrong folder, then basis risk is $R_1 = 0.985 \times 1 + 0.015 \times 1 = 1$, $R_2 = R_1 \times K_1(u) \times K_2(c)$, wherein, $K_1(u) = -0.1044 \times 10^{-8} \times u^2 + 0.6696 \times 10^{-5} \times u - 0.1284$, $K_2(c) = 2.6048 \times 10^{-8} \times c^2 - 4.8335 \times 10^{-5} \times c + 0.9328$. In three tasks, $u_1 = 1.8$, $c_1 = 500$; $u_2 = 1.8$, $c_2 = 1000$; $u_3 = 1.5$, $c_3 = 1000$. Because each operation has the same R_2, $R_T = R_2$. The calculation results of task risks are shown in Table 5.

4.4. **Experiment and analysis.** The average error rates for 5 subjects to complete the tasks 1, 2, 3 are shown in Table 6.

Comparing Table 5 with Table 6, we found that the actual experimental results are in good agreement with the calculated results using the model. The error rate of Task 3 is the biggest as a result of the most urgent operation emergency and the heaviest cognitive load. Because the effects of operation emergency on error rate are greater than that of cognitive load on error rate, so the risks of task 1and task 2 are relatively close, while the risks of task 2 and task 3 are relatively large difference. The experimental results show that the calculation results by risk interaction model are a reflection of actual risks, so the presented model is reliable and useful. The model can be used for rational allocation of annotation tasks.

5. **Conclusion and Future Work.** User operation error can caused risks in human computer, machine and robot interaction. RIM is proposed for quantitative evaluation of system risk in operation or interaction. The risks can be calculated by RIM and its parameters estimated from experiment. The influence of two important factors of the model, operating urgency and cognitive load, on errors was analyzed experimentally. Trends for the penalty functions were obtained; these will be important for application of RIM. Two applications, a BCI-based interactive auto-driving system and a image annotation system, are used as examples for quantitative calculation of system risks based on the model. RIM

was evaluated in comparisons between theoretical calculations and experimental results, which also demonstrated the usefulness of RIM.

We can draw the following conclusions through the above two instances. First of all, operation errors and risks caused by the errors are ubiquitous in human-computer interaction. It is demonstrated via the vehicle driving system by BCI and the image annotation system by mouse operation. Extended to other interactions such as keyboard, voice input, they also produce errors and risks. Risks in some systems can be avoided by inverse operation, and those in other systems are disastrous, so the latter must be quantitatively analysis. Second, risks come from human, the sources of error produced by certain cognitive load, emergency of the operation, and so on. Human will make mistakes in the process of task completion. It is very important how to measure the human error probability by these factors. Third, the risk not only exists, but also is to follow some law. It could be modeled and measured. Combined with the operation process, using our model, systematic risk or task risk could be calculated. So the interaction risks can be estimated by RIM, and the RIM can also be applied to other interaction tasks or system.

Acknowledgments. This research was jointly supported by the Tsinghua Self-innovation Project (Grant No: 20111081111), the National High Technology Research and Development Program 863 (Grant No: 2012AA011602), the National Natural Science Foundation of China (Grant Nos. 90820304, 20121302251 and 61075027) and the Tsinghua National Laboratory for Information Science and Technology (TNList) Cross-discipline Foundation (042003023). We thank Mu Yu for his help in the experiment.

REFERENCES

[1] S. K. Card, T. P. Moran and A. Newell, *The Psychology of Human-Computer Interaction*, Lawrence Erlbaum Associates, 1983.

[2] A. Dix and G. Abowd, Modelling status and event behavior of interactive systems, *Software Engineering Journal*, vol.11, no.6, pp.334-346, 1996.

[3] P. M. Fitts, The information capacity of the human motor system in controlling the amplitude of movement, *Journal of Experimental Psychology*, vol.47, pp.381-391, 1954.

[4] S. Jain and S. Bhattacharya, Predictive error behavior model of on-screen keyboard users, *Proc. of CHI*, pp.1435-1440, 2011.

[5] B. E. John and D. E. Kieras, The GOMS family of user interface analysis techniques: Comparison and contrast, *ACM Transactions on Computer-Human Interaction*, vol.3, no.4, pp.320-351, 1996.

[6] E. Rabdulin, Using the keystroke-level model for designing user interface on middle-sized touch screens, *Proc. of CHI*, pp.673-686, 2011.

[7] J. Reason, Generic error-modeling system (gems): A cognitive framework for locating common human error forms, in *New Technology and Human Error*, J. Rasmussen, K. Duncan and J. Leplat (eds.), London, UK, John Wiley & Sons Ltd, 1987.

[8] J. Reason, Human error: Models and management, *British Medical Journal*, vol.320, no.7237, pp.768-770, 2000.

[9] J. O. Wobbrock, A. Jansen and K. Shinohara, Modeling and predicting pointing errors in two dimensions, *Proc. of CHI*, pp.1653-1656, 2011.