Design of syntactic adaptive interactive system based on human perception state estimation within scenario context

Dian Jin[1], Weiwei Yu[1,2*], Gautam Srivastava[3,4], Xinliang Yang[1]

[1*]School of Mechanical Engineering, Northwestern Polytechnical University, No. 127 West Youyi Road, Xi'an, 710072, Shaanxi, China.
[2]Unmanned System Research Institute, Northwestern Polytechnical University, No. 127 West Youyi Road, Xi'an, 710072, Shaanxi, China.
[3]Dept. of Math and Computer Science, Brandon University, Canada.
[4]Dept. of Computer Science and Math, Lebanese American University, Beirut, Lebanon.

*Corresponding author(s). E-mail(s): yuweiwei@nwpu.edu.cn;

Abstract

Adaptive interactive systems can be divided into two categories: Semantic adaptations affect the system's function, and syntactic adaptations affect the presentation of information through the operator interface without modifying the system behavior. Although automation is becoming more and more important, in many situations, human operators still play an irreplaceable role, making the development of syntactic adaptations indispensable. However, existing syntactic adaptations usually do not relate to the task context, which is a critical factor to achieve syntactic adaptations. Besides, existing studies do not fully consider the operator's information perception state under highly dynamic scenarios, thus cannot dynamically prompt accurate information to the operators. Aiming at the problems above, a syntactic adaptive interactive system architecture based on perception state estimation and context segmentation is proposed in this paper. To address the real-time context segmentation, a time series segmentation algorithm is proposed, which achieves real-time estimation of the scene and produces highly interpretable results. To address the problem of dynamic perception state estimation, for the first time, the concept of a perception stack is proposed, which can be used to estimate the perception bias in the operator's working memory based on the eye movement data and generate proper prompt information according to the current context. The proposed design method of an adaptive syntactic interactive system was verified in a flight simulator. The system improves information prompts accuracy and reduces redundancy through these innovative mechanisms mentioned above. Comparative experiments confirm significant improvements in the operator's control accuracy and control stability according to our evaluation criteria.

Keywords: Adaptive interactive system, Human-computer interaction, Perception enhancement

1 Introduction

A human-computer system refers to a system comprising humans and computers that fulfills some functions through their interaction (Yu et al, 2022b). However, human-computer interaction can be highly constrained. Numerous experimental results indicated that the human perception bandwidth is severely limited (Cohen et al, 2016), i.e., only limited information can be successfully received by the operator either because the cognitive resources conflict or due to sensory channel saturation.

Therefore, researchers have proposed the concept of adaptive interaction to enhance the effectiveness of human-computer interaction and improve manipulation performance in highly dynamic environments. Adaptive interaction can be divided into two categories (Lim et al, 2018) based on functional goals. The first category aims to adjust the system to accommodate humans, which involves affecting the system's function, behavior, and goals. This category can be thought of as modifying the system interactions and is referred to as semantic adaptations. The second category aims to enhance human capabilities by affecting the presentation of information through the operator interface without modifying the system's behavior. This category is known as syntactic adaptations.

Semantic adaptations are exemplified by the process of human-machine switching, which refers to detecting the operator's state and replacing human functionality with an automation proxy at appropriate times to enhance system efficiency. Feigh et al (2012) used human workload state with other information as triggers, where human state and additional context information were used for switching between machine automation and human operation. Dehais et al (2022) used Brain-Computer Interface to decode pilots' intention and to infer their level of attention, and achieved the simulation of an automatic avoidance maneuver when the system detected that pilots missed an incoming collision. Although automation is becoming more and more important and most automated systems are equipped with shared control systems (Wen et al, 2022; Stanton et al, 2022), human operators still play an irreplaceable role in many applications (Choi, 2016), such as aviation flight operations (Li et al, 2018). Semantic adaptations may lead to a decrease in the importance of human capabilities, potentially resulting in "automation surprise" (Barnell, 2022), therefore reducing human trust in the automated system.

As for the syntactic adaptations, they do not replace human functionality with automated systems. Instead, they enhance human capabilities by providing appropriate information at the right time, ensuring cognitive assistance for "humans in the loop". From the perspective of keeping human in the loop, syntactic adaptation provides a more suitable idea for designing human-centered interaction systems.

To achieve the syntactic adaptation, firstly, the system needs to evaluate the state of the operator. Some studies use human cognition and behavior models to guide interactive system design, such as ACT-R (Chen et al, 2021), Air-MIDAS (Corker, 2017), VUMS (Kaklanis et al, 2016) and Viual Cognitive Graph (Yu et al, 2022a). For example, ACT-R and other methods are used to study the mechanism of the human response process and evaluate the system's performance. However, these models are often based on hypothetical perception and behavior process modeling and lack personalized features. There are also studies focusing on estimating the human state using human personal data via algorithms. Pandey and Taffese (2021) proposed MWCogTool, which used eye movement data and manipulation data to realize the estimation of the operator's cognitive load. Gil et al (2019) introduced an architecture that assesses the human state by examining the person's location, attention, and hand position, and establishes corresponding rules. Following this, the task fosters cooperative engagement between the system and the human, guaranteeing a secure takeover is executed.

However, the aforementioned methods do not establish a strong connection with the contextual information of the scenario, and they do not specifically focus on the user's perception of information, which is crucial in highly dynamic systems. Another type of method based on SEEV (salience effort expectancy value) or N-SEEV (Noticing SEEV) attention model was proposed, which incorporates the information bandwidth and image saliency features. For example, Schwerd and Schulte (2021) used the eye movement data as features to assess attention allocation and situation awareness of a pilot in real time based on the SEEV theory. According to a statistical analysis of 143 aviation accidents (Stanton et al, 2001), most errors at the pilot's perceptual level occur due to a failure to timely observe specific important information (Table 1). However, the human perception state obtained from the above study is too coarse-grained to be directly used in the syntactic adaptation system.

As for perception information required in syntactic adaptation, it is fine-grained and accurate.

Table 1 Primary types of human perception errors and proportions

Error Type	Proportion (%)
Failure to timely observe information	**35.1**
Environmental lack of relevant information	13.0
Difficult-to-read information	11.1
Misinterpretation of information	8.7
Forgetting information	8.4

Only by estimating the operator's perception state associated with the scenario can the information prompt be designed, or the redundant adaptive information would disturb the perception efficiency. To achieve this, Fortmann and Mengeringhausen (2014) developed an assistive monitoring system that guides the visual attention of operators towards critical information by continuously monitoring visual cues during supervisory tasks on a display. This research considers the current scenario state as an upgraded form of interface layout adaptation and has shown effective enhancement of operator perception in certain tasks. Peysakhovich et al (2018) proposed a concept for pilot perception enhancement based on eye-tracking. The concept suggests defining maximum allowable dwell times and unmonitored maximum allowable times for each instrument in the cockpit based on flight phases (determined by real-time analysis of relevant flight parameters such as position, altitude, and speed). If any of these rules are violated, countermeasures such as auditory alerts can be implemented. However, the aforementioned human-perception-based models face challenges in accurately locating information in highly dynamic scenarios. Additionally, relying solely on a few simple rules makes it difficult to effectively estimate the operator's perception state associated with scenario context.

In summary, existing research on syntactic adaptive interactive systems often lacks focus on context-specific relevance. Alternatively, the analysis of human perception tends to have a coarse granularity, making it challenging to accurately locate prompt information in highly dynamic environments. Therefore, this paper first constructs an online time series segmentation algorithm to recognize and label the contextual scenario. Second, this paper introduces the innovative concept of a perception stack, which addresses the issue of fine-grained information prompts in a highly dynamic scenario context.

In the following part, Section 2 gives an overview of our syntactic adaptative interactive system architecture. Section 3 proposed an explainable online segmentation method. Section 4 introduced the perception stack and proposed the perception bias estimation method. In Sections 5, our proposed system is realized based on a flight simulator, several experiments are conducted, and the effectiveness of our system is validated by operator performance in Sections 6. In Sections 7 and 8, the discussion and conclusion of our proposed method are discussed.

2 Syntactic adaptative interactive system architecture

The overall syntactic adaptative interactive system architecture is shown in Fig.1. The system includes five subsystems:

(1) Scenario context recognization module: This module estimates the current scenario based on system parameters. Different scenarios have different objectives, and the scenario context associated parameters were recorded. The Scenario context was recognized online and labeled in this module. A detailed introduction of the proposed algorithm is provided in Section 3 of this paper;

(2) AOI localization module: This module receives the operator's eye movement data and undergoes a series of processing steps to generate an AOI (Area of Interest) sequence. Since the eye movement information itself does not contain the content of the operator's perception, the eye gaze coordinates are mapped to the interactive information

through AOI. Based on the recognition of various functional areas in the interface layout, each eye movement point in the denoised eye movement sequence is labeled to display the current information perceived by the operator, facilitating subsequent processing. This is discussed in detail in Section 4.1 of this paper;

(3) Perception bias estimation module: This module utilizes AOI sequence data to estimate the operator's perception state and assess perception biases in conjunction with current system parameters. Differences may exist between the pilot's current situational perception information and actual parameter information, particularly noticeable among novice operators. A perception stack is constructed in this module to dynamically model the operator's perception state, enabling real-time updates and online state estimation. Detailed explanations of this model are provided in Section 4.2 of this paper;

(4) Perception demand generator: This module generates perception information requirements based on the current scenario and operational manual. The perceived bias estimation module generates perceived bias information, some of which is meaningless as it is not currently used. However, certain perceived bias information is crucial and has specific requirements for tasks. Detailed explanations are provided in Section 4.2 of this article;

(5) Prompt information generator: This module generates information prompts by receiving perceived requirements and perceptual biases. After understanding the crucial information in the current stage and the information indicating the operator's current perception failure, the corresponding prompt content is generated by combining these two pieces of information. This includes importance level, information content, information parameters, and so on. Detailed explanations are provided in Section 4.3 of this paper.

The system estimates the perception state of the operator through eye movement data. In addition, according to the current scenario parameters, it assesses the operator's current perception state, analyzes the missing and outdated information, and then highlights these information pieces through the display interface. By perceiving this

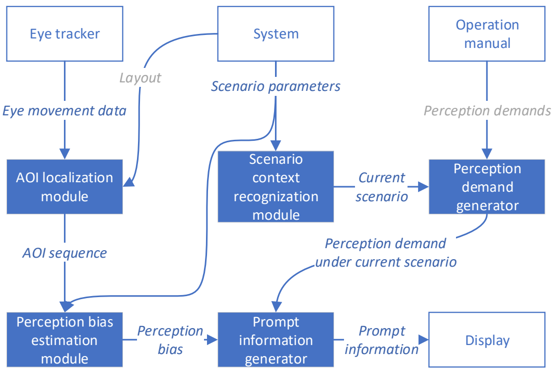


Fig. 1 System architecture: grey represents statistic data/ parameters

information, operators can adjust their operations in time, to ensure safety and improve performance.

3 Online scenario context recognization

Since the operator's perception state is closely related to the current scenario, the system should speculate the current task stage of the operator through the scenario context data (Kim and Yoon, 2018). Toeplitz Inverse Covariance-based Clustering (TICC (Hallac et al, 2017), as shown in Fig.2) is a very effective method for unsupervised clustering of time series data. It can segment the generated data into stages adaptively.

However, TICC uses the EM method which bases on global optimization to iteratively generate the optimal partitioning, thus cannot achieve online segmentation. Therefore, the original TICC method cannot be applied to recognize the current scenario during adaptive interaction. In this paper, the idea of TICC is absorbed, and the Toeplitz Inverse Covariance based - Online Smooth Segmentation (TIC-OSS[1]) is proposed. The TIC (Toeplitz Inverse Covariance) of each task is constructed by a supervised learning method, and then the online smooth segmentation of online time-series data is realized by setting the time continuity penalty.

The question can be abstracted as below:

[1]Github Repository: https://github.com/CLaSLoVe/TIC-OSS

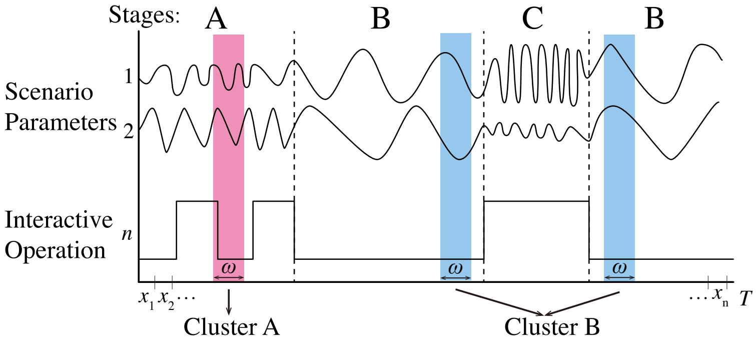


Fig. 2 Main concept of TICC method

Training set $\boldsymbol{X}_{m\times n} = [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_m]^{\mathrm{T}}$ is the high-dimensional time-series data whose data length is m and data dimensions is n. $\boldsymbol{y}_{m\times 1}$ is the training set label. Online data flow $\boldsymbol{s} = [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_i, \cdots]^{\mathrm{T}}$. Predict the labels at each moment of the data stream.

The high-dimensional time series data $\boldsymbol{X}_{m\times n}$ presents the context parameters derived from scenario sensor information.

The overall process of solving this problem is shown in Fig.3.

3.1 Algorithm implementation

Academic translation: As shown in Fig.3, the algorithm consists of an offline component during training and an online component during runtime. The following two sections will introduce the Offline process and Online process, respectively.

3.1.1 Offline process

Firstly, the scenario was manually divided and the ground truth value was constructed. Then, through the analysis of experts and dimensionality reduction methods, some parameters are selected from the high-dimensional scenario parameters as the training set. Thirdly, the data were normalized to reduce the impact of data magnitude, and the relevant normalization parameters were recorded. A graphical lasso was used to obtain the optimized mean value vector and the TIC of each stage. (Shown in Fig.3, grey part.)

Normalization

To prevent variable magnitude affect the estimation results, the data should be normalized before training. The general normalization method is max-min normalization, which magnifies the data to the range of $[0, 1]$. However, for online data, we do not know the maximum and minimum data, so the max-min normalization method cannot be used. Here, we use the tanh function to approximate the max-min normalized linear function:

$$f(x) = \frac{b-a}{2}\tanh(\frac{2(x-c)}{\alpha}) + \frac{a+b}{2}$$

where (a, b) are the scaling lower and upper limit, $alpha$ denotes the 98% quantile, c is the function center.

This function shrinks the maximum value of the training set to about 0.98 and the minimum value to about 0.02. Meanwhile, in the online training scenario, α and c of the training set are used to estimate $\hat{\alpha}$ and $\hat{c}$ of the test set. If the online data (test data) contain larger or smaller data, this function allows for a difference while maintaining normalization.

TIC obtaining

We conduct a windowing operation on the original data to construct time-interleaved TIC, to discover the association between sensors and time slices.

Toeplitz inverse covariance matrix was constructed:

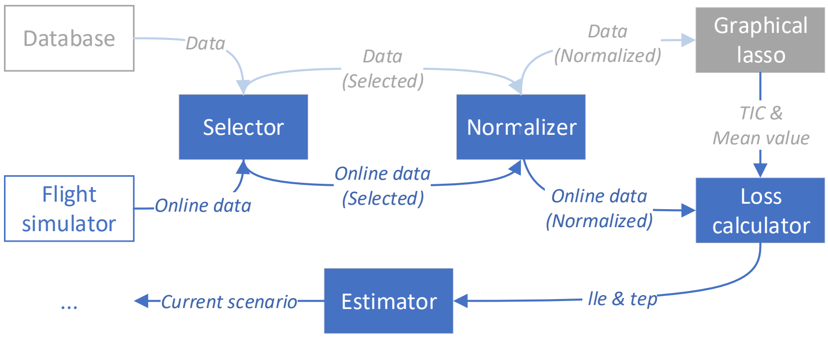


Fig. 3 Algorithm implementation: Grey represents offline process

$$\Theta_i = \begin{bmatrix} A^{(0)} & \left(A^{(1)}\right)^T & \left(A^{(2)}\right)^T & \cdots & \left(A^{(w-1)}\right)^T \\ A^{(1)} & A^{(0)} & \left(A^{(1)}\right)^T & \ddots & \vdots \\ A^{(2)} & A^{(1)} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \left(A^{(1)}\right)^T & \left(A^{(2)}\right)^T \\ \vdots & \ddots & A^{(1)} & A^{(0)} & \left(A^{(1)}\right)^T \\ A^{(w-1)} & \cdots & A^{(2)} & A^{(1)} & A^{(0)} \end{bmatrix}$$

Here $A^{(\tau)}$ denotes the TIC between flight parameters with time interval τ.

To obtain the matrix, we have a basic assumption: the matrix is sparse (see Hallac et al (2017)). We can solve it using graphic lasso:

$$\sum_{X_t \in P_i} -\ell\ell\left(X_t, \Theta_i\right) = -|P_i| \left(\log\det\Theta_i + \operatorname{tr}\left(S_i\Theta_i\right)\right) + C$$

According to (Friedman et al, 2008), the problem can be transformed into minimizing the Eq.(1):

$$\begin{aligned} &\min\ -\log\det\Theta_i + \operatorname{tr}\left(S_i\Theta_i\right) + \tfrac{1}{|P_i|}\left\|\lambda \circ \Theta_i\right\|_1 \\ &\text{s.t.}\quad \Theta_i \in \mathcal{T}. \end{aligned} \tag{1}$$

Among them, the $|P_i|$ is the points in class i, $S_i$is the experience covariance matrix, C is an irrelevant constant to Θ_i.

The sparse matrix Θ_i or TIC is obtained by using the alternating direction multiplier method (ADMM), which is regarded as the estimation of MRF (Markov Random Field).

As the estimated TIC can be regarded as a Markov Random Field for the selected flight parameters at both the same and different time intervals (temporal interleaving), visualizing the TIC provides insights into the algorithm's basis for identifying different stages.

In this study, seven flight parameters were selected, and a time window of 3 was set. The TIC was analyzed for four tasks, as shown in Fig.4. Each point represents the value on the diagonal of the corresponding TIC submatrix. The blue points represent the normalized average values, indicating the relative magnitudes among the data. The orange nodes represent the relative rate of change within the time interval. If the data at the corresponding position on the diagonal of the TIC submatrix is 0, it is represented by a small dot for easy observation.

The connecting lines (including straight and curved lines) between the points indicate the values on the off-diagonal of the TIC submatrix, reflecting the data correlation under temporal interleaving. The green lines represent positive correlations between flight parameters, while the red lines represent negative correlations. In Fig.4(a), during the takeoff stage, the engine speed and throttle exhibit a positive correlation and both increase, which aligns with the gradual increase in throttle in this task. In Fig.4(b), the algorithm did not discover any specific correlations between the data, but the altitude changed (increased), consistent with the ascent task where there is relatively little parameter variation but an increase in altitude. In Fig.4(c), representing the level flight task, the description aligns with the relatively stable flight parameter situation. In Fig.4(d), the altitude decreases, the flaps increase, and the engine speed decreases, consistent with the landing task.

3.1.2 Online process

Due to the different training conditions, such as the length of the training set, the obtained $\sum_j \sum_k \Theta_i(j,k)$ for each class i can own a large difference. Thus, we need to regulate the obtained TIC, to make them equal.

For the regular TIC, denoted $\hat{\Theta}_i$, the negative log-likelihood loss (LLE) is calculated:

$$-\ell\ell\left(X_t, \hat{\Theta}_i\right) = \frac{1}{2}\left(X_t - \mu_i\right)^{\mathrm{T}} \hat{\Theta}_i \left(X_t - \mu_i\right) - \frac{1}{2}\log\det\hat{\Theta}_i + \frac{n}{2}\log(2\pi)$$

where $\frac{n}{2}\log(2\pi)$ is a constant.

In addition, in online prediction, since the next moment is more likely to be in the same task state as the last moment, the time continuous penalty β is set. The total penalty of task i is:

$$\rho_i = -\ell\ell\left(X_t, \Theta_i\right) + \beta \cdot \mathbb{1}\left\{X_{t-1} \notin P_i\right\} \cdot \sum_{X_t \in P_i} -\ell\ell\left(X_t, \Theta_i\right)$$

The one with the smallest penalty can be regarded as the estimation of the current task.

4 Perception state estimation

To realize the prompt of situational awareness information missed by the operator while reducing interference, we need to understand:

1. What information did the operator perceive?
2. What information is required for the current scenario?
3. Was the operator's information outdated and unreliable?

4.1 AOI localization

To determine the operator's perceived information, an eye tracker can be employed. However, the process of visual fixation is usually accompanied by small eye movements, such as tremors, drifts, and small non-voluntary eye movements, which have little significance for higher-level analysis. In addition, eye trackers may also have some noise, which can cause the coordinates collected by the eye tracker to fluctuate and have high noise. Therefore, the second step is to denoise the collected gaze point coordinates. For the identified gaze points, a sliding average is used to calculate the average value of the eye movement coordinates with $k = 11$ within 100 ms as the current coordinates $(\bar{x}_t, \bar{y}_t)$:

$$(\bar{x}_t, \bar{y}_t) = \sum_{i=0}^{k}(x_{t-i}, y_{t-i})$$

After capturing the gaze point, the gaze point is mapped onto the information interface area based on functional segmentation, thereby semantically representing the eye-tracking coordinates.

4.2 Perception stack construction

To address the issue of outdated and unreliable operator information, we propose utilizing a perception stack for accurate perception state estimation.

The process of perception state estimation is shown in Fig.5. A perception stack is constructed by the AOI of the current attention calculated by the eye movement of the operator. We use the perception stack to describe the operator's current perception state of scenario information (scenario parameters). Since each gaze is located in the AOI, whose corresponding information is believed to be processed by the operator. The scenario information in the AOI is added to the current operator's perception stack, and the new perception state will cover the old one. Meanwhile, at each moment, the difference between the current perception state and the current scenario parameters is calculated as the current perception bias.

If the prompt information reduces user interactivity instead, it should not take place (Miraz et al, 2021). Perception bias acts as a switch when a scenario parameter is inconsistent with the intended target. If the operator's perception deviation of the current parameter is within the threshold, we assume that the operator understands the current status of the scenario. At this time, the deviation of scenario parameters may be caused by emergencies, temporary changes in tasks, etc., so it is not necessary to prompt the operator, to avoid

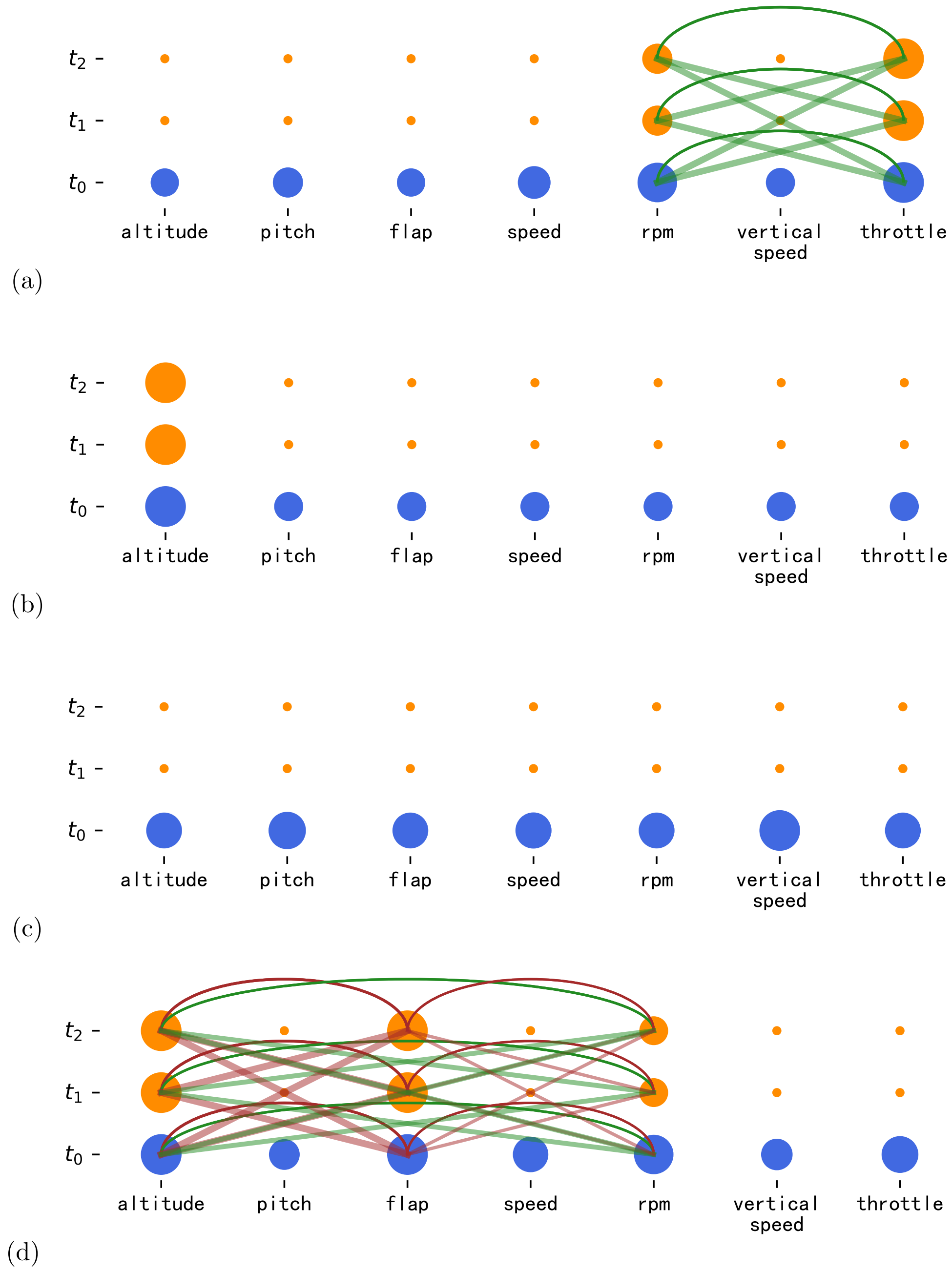


Fig. 4 TIC for each task:(a)take-off;(b)climb;(c)level-off;(d)land

redundant prompts which may waste the operator's limited attention resources, thus reducing the performance.

A perception stack is used to record what the operator perceives. The operator's eye movements were recorded by an eye tracker. After noise reduction processing and AOI localization, we can obtain the AOI that the operator paid attention to at each moment. Each AOI on the system interface has its function. Take an example of a simulated flight mission, AOI Altitude is used to reflect the current Altitude of the aircraft. When the pilot has fixations on this AOI, he perceives altitude information. For each moment, the pilot's

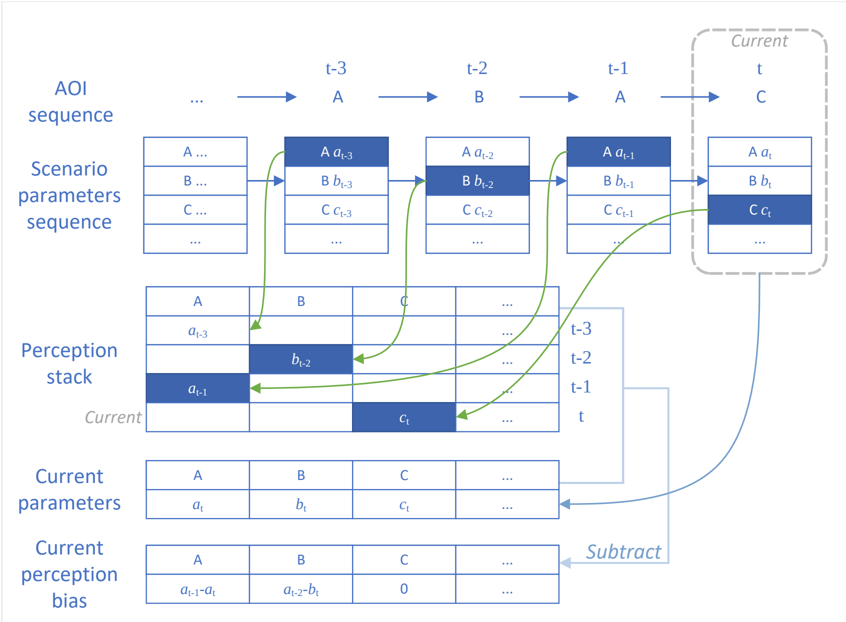


Fig. 5 An example of perception bias estimation

perception of each parameter at each moment in the perception stack was recorded. When the pilot repeatedly perceives a certain AOI, the new perception data will replace the old data.

Perception bias represents the deviation between the user's perceived information and the real value due to information changes and omissions. We use the latest perception data in the perception stack to subtract the current scenario parameters to estimate the bias of the operator's current perception of each parameter.

When the scenario parameters are not consistent with the intended target and the operator's perception deviation of the scenario parameters is large, we believe that the operator is not skilled, and attracted too much attention to other information, resulting in the error of the current situation awareness. At this point, we will guide the operator's attention to remind him/her. On the contrary, when a scenario parameter is inconsistent with the intended target, but the operator's perception deviation of the scenario parameter is within the threshold, we assume that the operator understands the current status of the scenario. At this time, the deviation of scenario parameters may be caused by emergencies, temporary changes in tasks, etc., to avoid redundant prompts which may waste the operator's limited attention resources, thus reducing the task performance.

Take a flight mission as an example, as shown in Fig.6, when the pilot first perceives Speed information, he/she obtained that the current aircraft speed is 71 knots (kn). When the pilot second perceives Speed information, he knew that the current aircraft speed changed to 74 kn.

Then, at this moment, the pilot thinks the current speed is 74 kn, whereas the current speed is 71 kn, which means the pilot's perception bias is +3 kn.

4.3 Prompt information generation

To solve the problem of "What information is required for the current scenario?", the perceived requirements must align with the flight requirements and flight manuals established prior to the mission. This process relies on expert experience and necessitates manual completion. For example, for the Cessna Skyhawk aircraft, according to the flight manual, the speed during takeoff should be around 55 kn, and the speed during climbing should be stable around 74 kn. In addition, for specific missions, this paper also sets specific target parameters. For instance, for this flight mission, the level flight altitude should be 1500 m, so the pilot should climb to 1500 m and stabilize at that altitude.

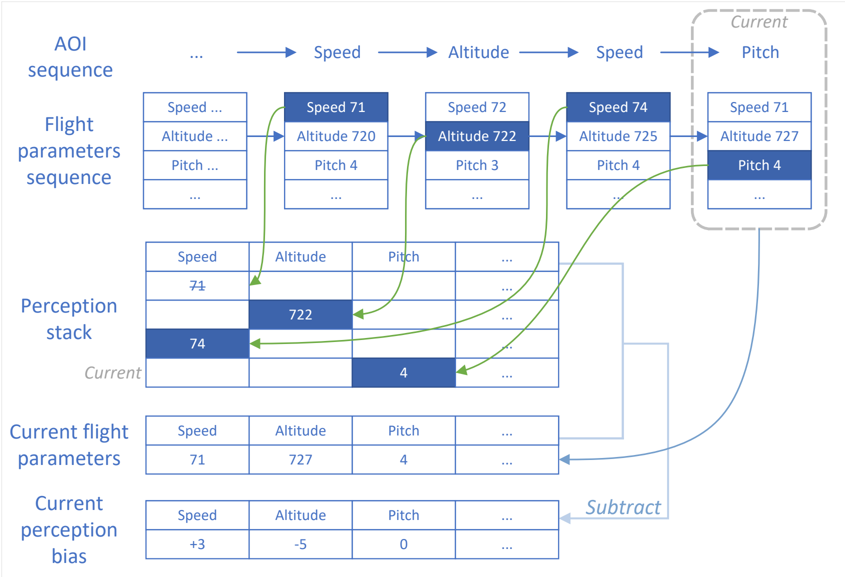


Fig. 6 An example of perception bias estimation under a simulated flight mission

However, flight parameters are typically dynamic, and due to factors such as airflow and control precision (Yu et al, 2022b), it is nearly impossible to maintain them accurately at the target values. Therefore, this paper sets the tolerance threshold for perceived deviations. The task requirement Example is shown in Table 2.

For each flight parameter, a safety threshold is set, and when the perceived deviation is within the tolerance threshold set for the perceived requirements, the current perceived deviation is considered normal. When the perceived deviation exceeds the tolerance threshold set for the perceived requirements, the system determines that the pilot cannot perceive important situational information in the current flight scenario and triggers a prompt.

5 Experiment

5.1 System realization

Our experiment is based on a simulated flight control task, which involves various requirements that users need to fulfill, with a high demand for timely information acquisition. The system is composed of software and hardware platforms.

5.1.1 Software platform

Our experiment platform is based on X-plane 11 simulator. To make the information collected by the eye tracker more accurate, we rearranged the interface layout. Then we divided the interface into several functional areas, which are called AOIs. The AOI layout is shown in Fig.7.

Fig. 7 AOI layout

We built a system whose interface was superimposed on the simulator interface. When no prompt information is generated, our system interface remains transparent. When a piece of certain information needs to be prompted, the corresponding AOI of our interface will be highlighted

Table 2 Task Requirement Example

Task Scenario	Parameter	Target Type	Target Value	Tolerance
Take-off	Heading	keep	50	10
	Speed	to	55	5
Climb	Speed	keep	74	4
	Heading	keep	50	10
	Altitude	to	1500	100
Level-off	Heading	keep	50	10
	Altitude	keep	1500	50
Land	Climb Rate	keep	-500	500

Fig. 8 Highlight functional area (AOI)

to attract the pilot's attention (Information highlighting is an effective intervention (Carenini et al, 2014)), as shown in Fig.8. After the pilot obtains the information, the highlighted area will disappear with a delay of 1 second.

5.1.2 Hardware platform

Eye movement data is obtained in real time through an eye tracker. The eye tracker used in this study is the Tobii X3-120 Professional Eye Tracker produced by Tobii, a Swedish company, with a sampling rate of 120Hz. The eye tracker is calibrated before each experiment with each participant to obtain more accurate data.

Manipulation of input devices includes throttle lever, joystick, and foot pedals. The throttle lever and joystick are part of the A-10C Warthog Flight Stick set produced by Thrustmaster in the United States. The foot pedals are produced by Logitech and are called Flight Rudder Paddles. All devices were calibrated prior to the experiment.

5.2 experiment setup

Three novice pilots were invited as our subjects. All the subjects had learned basic flight control instructions and could complete a group of flight experiments independently. To prevent the pilot's proficiency from influencing the conclusion, we switched to the experiment with and without our proposed system. In addition, to prevent the influence of pilot fatigue on the conclusion, all the subjects were asked to take a few minutes to rest after each experiment.

(1) take-off: Release the brakes, and push the throttle. When the speed reaches 55kn, pull up the nose and enter the climb.
(2) climb: The control speed is stable at 74kn, and the azimuth angle (psi) is controlled. When the altitude is close to 1500m, the nose is gradually flattened and enters level-off.
(3) level-off: Control altitude stabilized at 1500m. Control aircraft on course.
(4) land: Approach the target, reduce the height, timely reduce the throttle, timely open the flaps, maintain the landing speed around -500fpm, and land on the runway.

Each participant performed three sets of experiments, for a total of 18 experiments. Each of them consisted of four task scenarios: take-off, climb, level-off, and land, and lasted about 8 minutes.

6 Results and analysis

6.1 Algorithm evaluation

6.1.1 Accuracy evaluation

This paper's classifier distinguishes four stages for scene recognition problems. The precision and recall rates for each stage and the overall accuracy rate for each algorithm (Logistic Regression, K Nearest Neighbor, Support Vector Machine, Decision Tree, and our proposed method TIC-OSS) are shown in Table 3. It can be seen that TIC-OSS has higher precision and recall rates in each stage, with the highest overall accuracy rate.

6.1.2 Real-time performance evaluation

As for the proposed interactive system, it is important to ensure the algorithm can run in real time. Therefore, an experiment was conducted to achieve real-time performance evaluation.

For a set of test data with a length of 29687, the time taken to complete the overall calculation is 3.237 seconds, thus the average value is 0.0001 seconds. This paper took 100 data points from the dataset as a group, for a total of 296 groups, and conducted tests separately. The test results reveal that the majority of the computation time is around 0.0001 seconds, with the maximum not exceeding 0.0002 seconds.

On the other hand, the flight platform's sampling frequency for flight parameters in this paper is 60Hz, i.e., one sampling is done every 0.0167 seconds. The time taken to process the flight parameters and generate the phase results is much less than this time, so the algorithm can run in real time on the designed platform without interfering with the system or causing delays.

6.2 System evaluation

6.2.1 Results visualization

According to the control requirement in the flight manual, the flight task was divided into two categories: those requiring the pilot to control the vehicle parameters close to a certain value (goal-oriented control), and those requiring the pilot to maintain the vehicle stable around a certain value (stability-oriented control).

Fig.9 indicated the psi information of an experiment. In this figure, we can understand quite clearly how the system works for operating stability-oriented control:

(1)The aircraft veered off the course;

(2)The system detects that pilots have a large perception bias in psi information;

(3)The system highlights the psi information;

(4)The pilot heeded the message;

(5)The pilot steered to the intended target.

For operating goal-oriented control, see Fig.10:

(1)Altitude close to the intended target;

(2)The system detected that there was a large perception bias in the altitude information of pilots;

(3)The system highlights the altitude information;

(4)The pilot heeded the message;

(5)The pilot steered to the intended target.

Fig.11(a) records the heading information (psi) operated by the pilot during the whole flight mission. Bias means the deviation from a preset course (here, a straight line). Fig.11(b) and Fig.11(c) show the control ability of subjects on speed in climb task and altitude in level off task. Since manipulation cannot be completely stable, we set a tolerant bias, within which range, the error is not counted. These three are the parameters that we require novice pilots to keep in corresponding tasks.

It can be found that with our system, the subjects can better perceive the change in flight parameters to control them better.

6.2.2 Performance evaluation

To address the effectiveness of the adaptive interaction system, this paper conducts flight control performance evaluation through two sets of comparative experiments with and without the adaptive interaction system. These experiments were conducted in complex scenarios, such as storms, with high perceptual load and difficulty in control. Three subjects performed two sets of experiments, each containing two trials.

Two aspects were used to evaluate flight performance. The first aspect is the accuracy of flight control, denoted as **accuracy performance score** A:

$$A = \frac{1}{n}\sum_{i}^{n} \alpha_i \tag{2}$$

where:

$$\alpha_i = \begin{cases} o_i - p_i & p_i < L \\ p_i - o_i & p_i > U \\ 0 & L < p_i < U \end{cases} \tag{3}$$

where α_i represents the control credibility. If the recorded flight parameter p_i is lower than the lower limit value L or higher than the upper

Table 3 Algorithms comparison

Algorithms	Precision				Recall				Accuracy
	Take-off	Climb	Level-off	Land	Take-off	Climb	Level-off	Land	
LR	0.922	0.708	0.823	0.658	1.0	**0.644**	0.728	0.943	0.765
KNN	0.892	0.731	0.799	0.608	1.0	0.635	0.670	**0.977**	0.741
SVM	0.787	0.732	0.832	0.758	1.0	0.606	0.794	0.961	0.789
DT	**0.937**	0.738	**0.975**	0.238	1.0	0.280	0.340	0.912	0.477
TIC-OSS	0.896	**0.786**	0.768	**1.0**	1.0	0.611	**0.908**	0.779	**0.814**

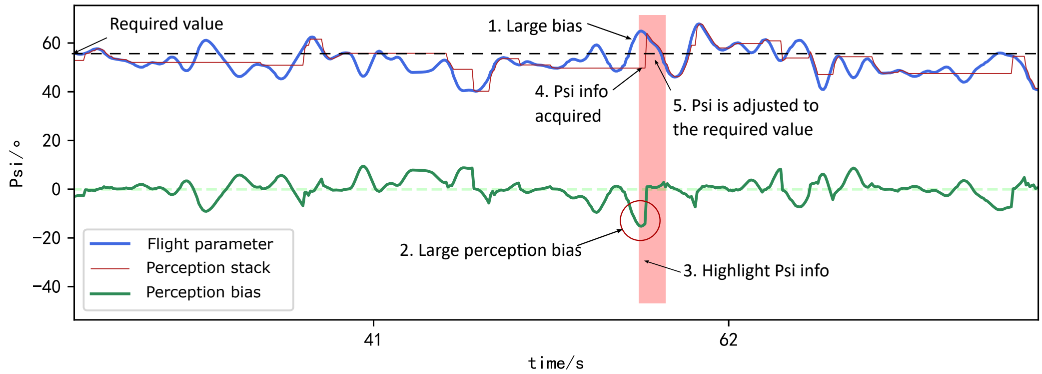


Fig. 9 An example of prompt information: Psi (Flightpath), the red column represents the prompt information (The highlighted area disappears with a delay of about 1 second)

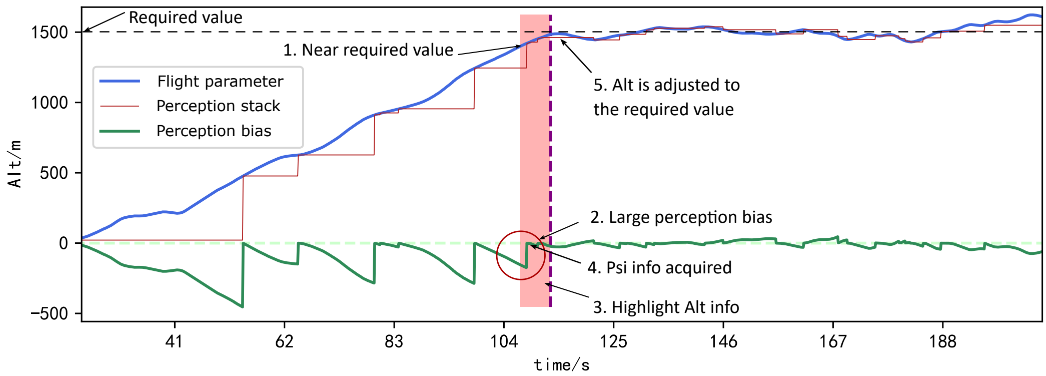
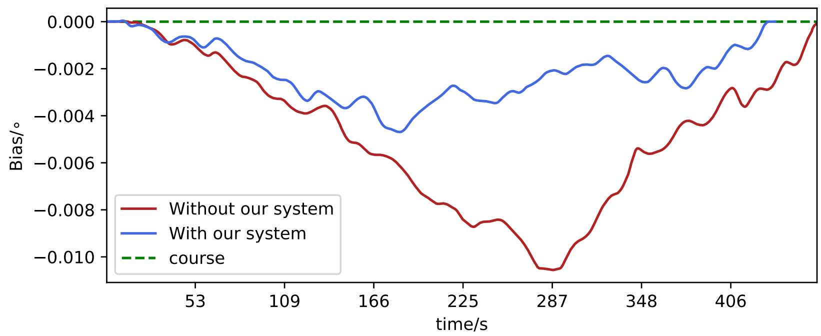
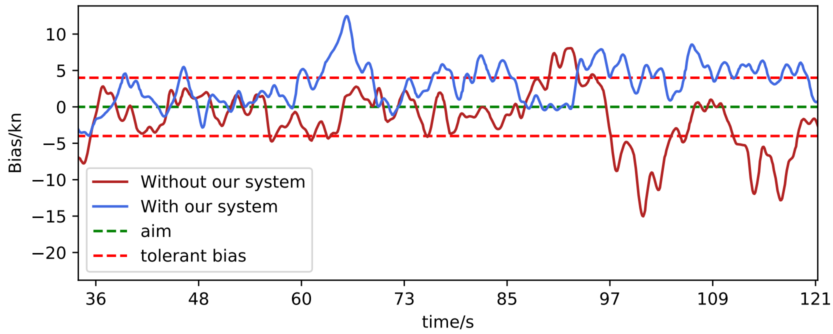


Fig. 10 An example of prompt information: Altitude, the red column represents the prompt information

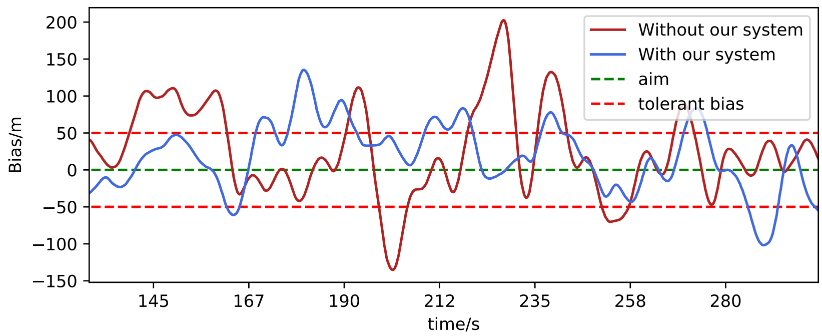
limit value U of the target flight parameter o_i, the credibility is the absolute difference between the recorded parameter p_i and the target parameter value o_i. As the control error increases, the penalty value will increase. In simulated flight control, if sufficient information is correctly obtained, the control personnel can easily keep the aircraft within a range. However, if there is a perceptual error in some parameters, there may be a significant deviation from the task requirements. For example, if the pilot ignores altitude information, the altitude of the aircraft may exceed the required altitude. However, it is not realistic or necessary to control the flight parameters to a perfect value. Therefore, this paper's algorithm



(a)



(b)



(c)

Fig. 11 (a)Psi stability comparison;(b)Speed control comparison during climb;(c)Altitude control comparison during level-off

considers the balance between these two aspects when calculating performance.

The second metric is the stability of flight control, denoted as **stability performance score** S is calculated as follows:

$$S = \frac{1}{n}\sum_{i=1}^{n}(p_i - \bar{p})^2 \tag{4}$$

where $\bar{p}$ is the mean value of the flight parameters recorded during the phase.

After obtaining the scores for the accuracy and stability performance score of flight control in each stage, the scores obtained using the system constructed in this paper will be compared with those obtained without using the system. In the comparison process, the score obtained without using the system will be used as the baseline to calculate the relative score. A higher score here indicates better performance. Fig.12 shows the comparison of the accuracy performance score of flight control, and Fig.13 shows the comparison of the stability performance score of flight control. Note that since performance scores are calculated as the reciprocal of penalty values, a higher performance score will be obtained under more precise control, leading to a larger ratio.

Through comparative analysis of the experimental results, it can be concluded that using the system presented in this paper has improved the participant's ability to control various flight parameters, including the accuracy and stability of flight control to a certain extent. The improvement in the accuracy of control during the climb phase was not significant, possibly due to the relatively simple nature of the task. However, there was a significant improvement in the control of parameters in the other tasks. In general, the control accuracy performance score improved by about 80%, and the control stability performance score improved by 30% under our experiment settings and our evaluation criteria.

7 Discussion

7.1 Conceptual contribution

For adaptative interactive system design, on the one hand, most recent studies mainly focus on the operator's state, such as workload or fatigue, while what information people need is rarely studied. On the other hand, if the adaptive interactive system is achieved by changing the interface layout, this may lead to the operator's confusion (in SEEV theory, the user's expectation of the interface is also an important part of the attention (Wickens et al, 2009)), resulting in lower performance. Furthermore, in a highly dynamic scenario, such as flight operation, if the prompt information is only based on the deviation of flight parameters, it is likely to lead to redundant prompts and disturb the operator.

In order to address the problem of user perception estimation, for the first time, the concept of a perception stack is proposed and a prototype is developed. This prototype dynamically estimates user perception bias, enabling the system to acquire real-time knowledge of the user's perceptual state. Previous methods have rarely considered user perception states, making it difficult to locate missing information.

To address the real-time segmentation of temporal data and the weak interpretability issues, the TIC-OSS algorithm is proposed in this paper, which achieves real-time estimation of the scene and produces highly interpretable results.

The system constructed in this study is based on these innovative mechanisms, which enhance the accuracy of information prompts while reducing redundancy. Through multiple comparative experiments, the effectiveness of the proposed system has been verified.

The system proposed in this study is not restricted to aviation flight control but can be applied to other interactive devices that deal with highly dynamic information. Examples of such devices include nuclear power plant monitoring (Ahn et al, 2022) and more.

7.2 Limitations and future work

The limitations of our study can be summarized as follows: Firstly, in certain application scenarios, an AOI may contain a substantial amount of information. Due to the limitations of the eye tracker's accuracy, accurately locating AOIs in such cases becomes challenging. Secondly, the eye tracker used in our study is only capable of capturing a limited range, typically confined to a single screen. However, for larger scenes, a glasses-type eye tracker may be employed, while the accuracy will be reduced. Thirdly, our system relies on established rules that are determined by manuals and expert knowledge. However, the creation of these rules is often a complex task and subject to challenges. Lastly, the layout issue of the interface was not taken into consideration when multiple pieces of information needed to be displayed. This omission may impact the effectiveness and usability of the system in certain scenarios.

In future work: Firstly, in terms of the system, additional prompt methods can be designed, including text prompts and voice prompts. Moreover, incorporating more operator states, such as working memory and fatigue rating, can be used collectively to enhance the system's functionality. Furthermore, exploring more complex tasks with detailed task demands will be investigated to further improve the system's utility. Additionally, the TIC matrix, being a high-dimensional and sparse matrix, can undergo further treatment using various methods, such as latent factor analysis (Wu et al, 2022, 2021). This approach can extract more features to uncover deeper underlying patterns. Thirdly, experiments will be included in future research to evaluate the system's performance under various conditions, such as different

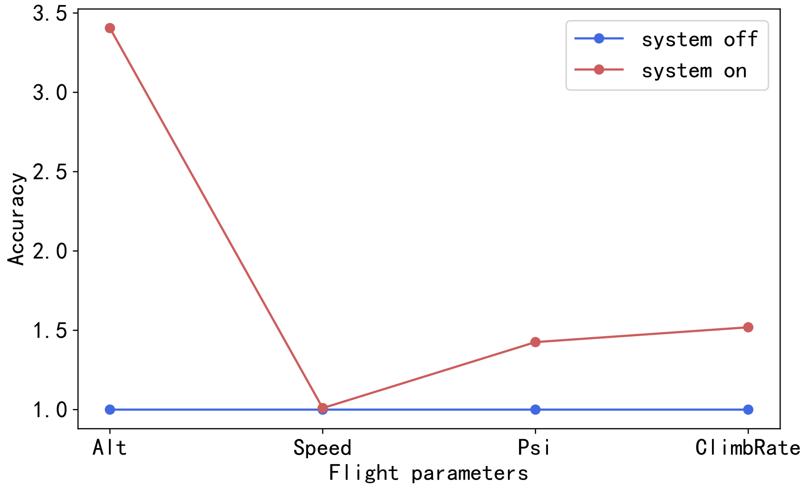


Fig. 12 Pilot control accuracy performance score

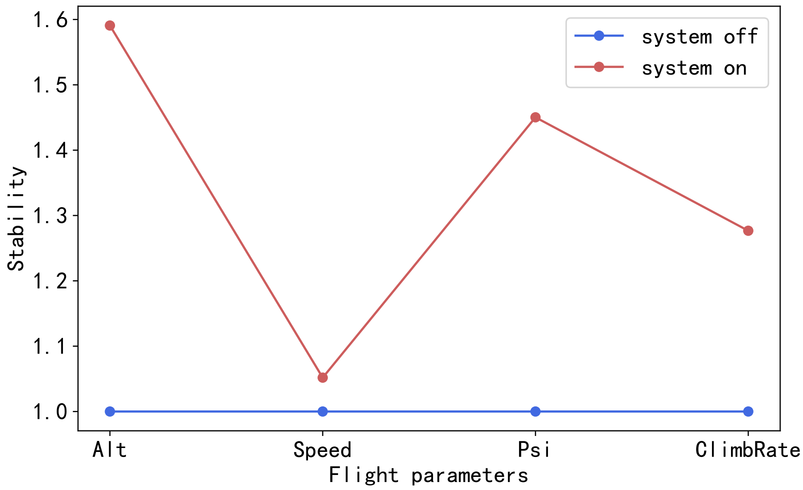


Fig. 13 Pilot control stability performance score

user states, task difficulty levels, and task durations. Fourthly, it is crucial to achieve automatic prioritization of context-relevant information during system implementation. When multiple pieces of information require simultaneous notification, ensuring the correct sequence of notifications becomes highly significant.

8 Conclusion

In this paper, an architecture for a syntactic adaptative interactive system is proposed. The interactive system estimates the current scenario using scenario parameters and evaluates the operator's perception bias. Based on prior information such as perception demand, the system provides prompt information.

The novelty of our work can be summarized as follows:

Firstly, this system is designed to be explainable. Unlike deep learning-based systems, our transparent approach allows the operator to understand the system's inner workings, thus establishing trust between humans and computers.

Secondly, this system utilizes the human perception state. By analyzing eye movement data and flight parameters, we estimate the perception bias, effectively reducing redundant information.

Thirdly, this system is context-based, meaning that the system adapts its behavior according to different scenarios. This enhances the accuracy and adaptability of the interactive system.

To demonstrate the effectiveness of the proposed system, it is implemented on a flight simulator. When prompt information is generated, the corresponding functional area is highlighted to capture the operator's attention, thereby maintaining the operator's situational awareness.

Finally, we conducted several comparative experiments, which revealed that this system achieved over 30% improvements in general. These results validate the effectiveness of this system.

Acknowledgments

Supported by 111 Project [Grant no. B13044].

Declarations

This paper has not been published previously and is not under consideration for publication elsewhere. Its publication is approved by all authors and tacitly or explicitly by the responsible authorities where the work was carried out, and it will not be published elsewhere in the same form, in English, or in any other language, including electronically without the written consent of the copyright holder.

References

Ahn J, Bae J, Min BJ, et al (2022) Operation validation system to prevent human errors in nuclear power plants. Nuclear Engineering and Design 397:111,949

Barnell E (2022) Utilizing bibliometric analysis tools to investigate automation surprises in flight automation systems. In: Human-Automation Interaction: Transportation. Springer, p 111–127

Carenini G, Conati C, Hoque E, et al (2014) Highlighting interventions and user differences: Informing adaptive information visualization support. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp 1835–1844

Chen H, Liu S, Pang L, et al (2021) Developing an improved act-r model for pilot situation awareness measurement. IEEE Access 9:122,113–122,124

Choi S (2016) Understanding people with human activities and social interactions for human-centered computing. Human-centric Computing and Information Sciences 6(1):1–10

Cohen MA, Dennett DC, Kanwisher N (2016) What is the bandwidth of perceptual experience? Trends in cognitive sciences 20(5):324–335

Corker K (2017) Computational human performance models and air traffic management. In: Human factors impacts in air traffic management. Routledge, p 337–370

Dehais F, Ladouce S, Darmet L, et al (2022) Dual passive reactive brain-computer interface: A novel approach to human-machine symbiosis. Frontiers in Neuroergonomics 3:824,780

Feigh KM, Dorneich MC, Hayes CC (2012) Toward a characterization of adaptive systems: A framework for researchers and system designers. Human factors 54(6):1008–1024

Fortmann F, Mengeringhausen T (2014) Development and evaluation of an assistant system to aid monitoring behavior during multi-uav supervisory control: experiences from the d3cos project. In: Proceedings of the 2014 European Conference on Cognitive Ergonomics, pp 1–8

Friedman J, Hastie T, Tibshirani R (2008) Sparse inverse covariance estimation with the graphical lasso. Biostatistics 9(3):432–441

Gil M, Albert M, Fons J, et al (2019) Designing human-in-the-loop autonomous cyber-physical systems. International journal of human-computer studies 130:21–39

Hallac D, Vare S, Boyd S, et al (2017) Toeplitz inverse covariance-based clustering of multivariate time series data. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp 215–223

Kaklanis N, Biswas P, Mohamad Y, et al (2016) Towards standardisation of user models for simulation and adaptation purposes. Universal Access in the Information Society 15(1):21–48

Kim S, Yoon YI (2018) Ambient intelligence middleware architecture based on awareness-cognition framework. Journal Of Ambient Intelligence And Humanized Computing 9(4):1131–1139

Li F, Wang W, Qu J, et al (2018) The design of adaptive user interface based on the grey relational grade. In: Journal of Physics: Conference Series, IOP Publishing, p 012046

Lim Y, Gardi A, Sabatini R, et al (2018) Avionics Human-Machine Interfaces and Interactions for Manned and Unmanned Aircraft. Progress in Aerospace Sciences 102(August):1–46. https://doi.org/10.1016/j.paerosci.2018.05.002

Miraz MH, Ali M, Excell PS (2021) Adaptive user interfaces and universal usability through plasticity of user interface design. Computer Science Review 40:100,363

Pandey S, Taffese T (2021) Using performance predictions to evaluate two-factor authentication setup processes. In: Proceedings of the Human Factors and Ergonomics Society Annual Meeting, SAGE Publications Sage CA: Los Angeles, CA, pp 999–1003

Peysakhovich V, Lefrançois O, Dehais F, et al (2018) The neuroergonomics of aircraft cockpits: the four stages of eye-tracking integration to enhance flight safety. Safety 4(1):8

Schwerd S, Schulte A (2021) Operator state estimation to enable adaptive assistance in manned-unmanned-teaming. Cognitive Systems Research 67:73–83

Stanton NA, Chambers PR, Piggott J (2001) Situational awareness and safety. Safety science 39(3):189–204

Stanton NA, Brown JW, Revell K, et al (2022) Oesds in an on-road study of semi-automated vehicle to human driver handovers. Cognition, Technology & Work 24(2):317–332

Wen H, Amin MT, Khan F, et al (2022) A methodology to assess human-automated system conflict from safety perspective. Computers & Chemical Engineering p 107939

Wickens C, McCarley J, Steelman-Allen K (2009) Nt-seev: A model of attention capture and noticing on the flight deck. In: Proceedings of the human factors and ergonomics society annual meeting, SAGE Publications Sage CA: Los Angeles, CA, pp 769–773

Wu D, He Y, Luo X, et al (2021) A latent factor analysis-based approach to online sparse streaming feature selection. IEEE Transactions on Systems, Man, and Cybernetics: Systems 52(11):6744–6758

Wu D, Luo X, He Y, et al (2022) A prediction-sampling-based multilayer-structured latent factor model for accurate representation to high-dimensional and sparse data. IEEE Transactions on Neural Networks and Learning Systems

Yu W, Jin D, Cai W, et al (2022a) Towards tacit knowledge mining within context: Visual cognitive graph model and eye movement image interpretation. Computer Methods and Programs in Biomedicine 226:107,107

Yu W, Jin D, Zhao F, et al (2022b) Towards pilot's situation awareness enhancement: A framework of adaptive interaction system and its realization. ISA Transactions https://doi.org/https://doi.org/10.1016/j.isatra.2022.12.005, URL https://www.sciencedirect.com/science/article/pii/S001905782200636X