An Adaptive Design Optimization Approach to Model-based Discrimination of Cognitive Control Mechanisms

Thesis

Presented in Partial Fulfillment of the Requirement for the Degree Master of Arts

in the Graduate School of The Ohio State University

By

Sang Ho Lee, M.A.

Graduate Program in Psychology

The Ohio State University

2017

Thesis Committee:

Jay I. Myung, Advisor

Mark A. Pitt, Co-advisor

Andrew B. Leber

Abstract

Cognitive control refers to the ability to adjust our thoughts and behaviors in order to achieve internalized goals. Experimental investigation of the underlying mechanisms often employs variations of the congruency task such as flanker, Stroop, and Simon tasks. In the past, researchers have proposed several theories of cognitive control to account for the characteristic patterns of response times observed in the tasks (e.g. Botvinick, Braver, Barch, Carter, & Cohen, 2001; Yu & Cohen, 2009). The goal of this study is to empirically discriminate two formal instantiations (models) of such theories, namely, the conflict-driven control model (M1) and the expectancy-based control model (M2). Each model is defined in terms of its own design space that can be capitalized on experimentally. This is the proportion of repetition for model M1 and the proportion congruency for model M2. To compare those models, three flanker task experiments were conducted using different design selection methods. The first experiment used the adaptive design optimization (ADO; Myung et al, 2013) to select a combination of the two design variables that optimizes model evaluation. ADO is an algorithm-based experimentation method for adaptively selecting the values of designs and stimuli on the fly in each experimental stage. The second and third experiments used pre-determined designs. The model-based approach adopted in the three experiments was shown to be efficient in discriminating the competing theories of cognitive control. Specifically, the present results indicate that each model had its own advantages in explaining individual behavior. M1 was better at explaining a reversed congruency sequence effect (CSE),

whereas M2 showed better fit to the proportion congruency effect. It suggests that there are diverse cognitive control mechanisms utilized to generate different response time patterns in the congruency task.

Dedication

I dedicate this to my family, advisors, and friends who supported and inspired me.

Vita

2015-Present	Ph.D. program in *Cognitive Psychology*, The Ohio State University (Advisors: Jay Myung & Mark Pitt)
2013-2015	M.S. program in *Cognitive Psychology*, Korea University
2007-2013	B.A. in *Psychology* and *Econonomics*, Korea University

Publications

- Kim, S., **Lee, S.H.**, Cho, Y.S. (2015). Control processes through the suppression of the automatic response activation triggered by task-irrelevant information in the Simon-type tasks. *Acta psychologica*, 162, 51-61.

- **Lee, S.H.** & Cho, Y.S. (2015). Self-concept in fairness and rule establishment during a competitive game: a computational approach. Frontiers in Psychology, 6, 1321.

Conference Presentations

- **Lee, S.H.**, Kim, S.J., Cho, Y.S. Congruency sequence effect between two different Simon congruencies, OPAM 2013, Toronto, Canada

- **Lee, S.H.**, Han, J.H., Kim, H.J., Kim, S.P., Cho, Y.S. The fairness model based on EEG hyperscanning for behaviors in competitive games, Society for Neuroscience 2014, Washington, D.C., USA

- **Lee, S.H.**, Cho, Y.S. Computational Model of Inhibitory Cognitive Control in Cross-Task Context, The Psychonomic Society 2014, Long Beach, USA

Fields of Study

Major Field: Psychology

Table of Contents

List of Tables

List of Figures

1. Introduction

Cognitive control refers to the ability to adjust our thoughts and behaviors in order to achieve internal goals (Botvinick, Braver, Barch, Carter, & Cohen, 2001; Posner & DiGirolamo, 1998; Egner, 2007). This flexible regulation is especially important in a task in which habitual responses have to be inhibited to enhance performance. For example, in an arrow flanker task where only the central arrow (e.g., < in >><>>) should be responded to, cognitive control helps us suppress the urge to respond to the other irrelevant arrows outside the center. This ability is crucial in everyday life because it allows us to focus our attention to a specific goal, in the world full of potential distractions. However, despite the importance of cognitive control in goal-directed behaviors, its exact mechanism is not fully known.

In order to investigate cognitive control processes, congruency tasks (e.g., flanker, Stroop, Simon task) are often used in experimental studies. An important characteristic that is shared by all congruency tasks is that the stimuli in those tasks contain task-irrelevant information that potentially affect the task performance. In the arrow flanker task, the stimuli are more quickly responded to when the task-irrelevant flanker arrows are the same with the task-relevant central arrow (e.g., >>>>>), than when they differ from each other (e.g., <<><<). This difference in the response time between congruent and incongruent stimuli is called the congruency effect. The activation of cognitive control is often measured by another phenomenon that is called the congruency sequence effect (CSE), which indicates a reduction of the congruency effect after an incongruent trial (Gratton, Coles, & Donchin, 1992). A typical response time pattern of the CSE is shown in Figure 1, in which the combinations of the previous and the current trial type are denoted by the combinations of c (previous congruent), i (previous incongruent), C (current congruent) and I (current incongruent). For example, cI indicates an incongruent trial after a congruent one. It is shown that the congruency effect after a congruent trial (cI – cC) is larger than that after an incongruent trial (iI – iC). This sequential modulation of the congruency effect has been replicated in many studies that use congruency tasks (e.g., Kerns, Cohen, MacDonald, Cho, Stenger, & Carter, 2004; Stürmer, Leuthold, Soetens, Schröter, & Sommer, 2002).

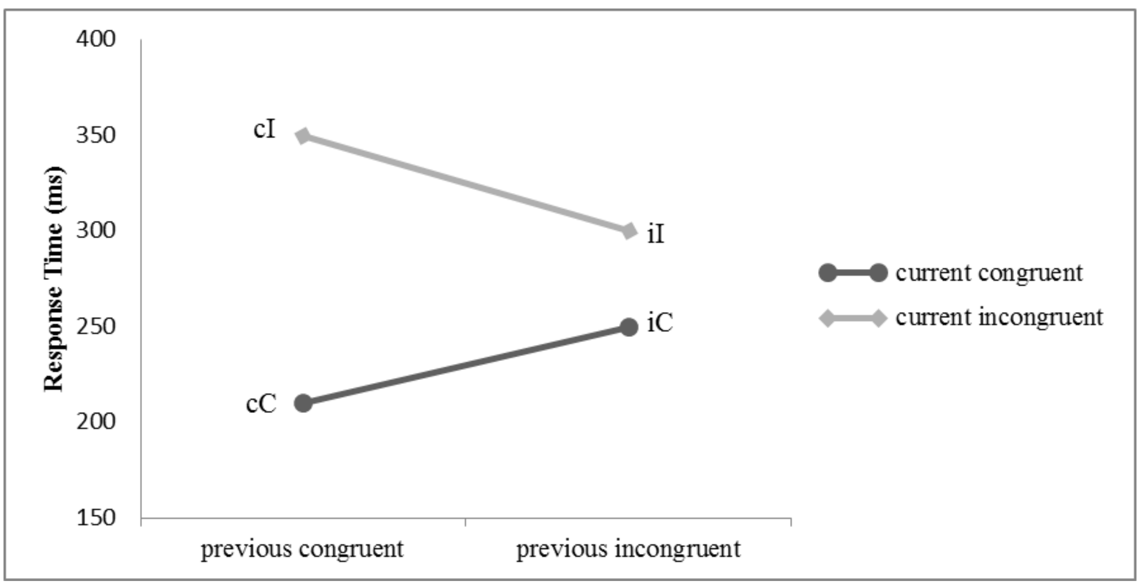


Figure 1. The congruency sequence effect (CSE).

The conflict monitoring theory (Botvinick et al., 2001) provides a framework that accounts for the CSE, by suggesting that a conflict monitoring system in the anterior cingulate cortex (ACC) detects conflict and sends out a signal to the dorsolateral prefrontal cortex (DLPFC). The top-down control signal from the DLPFC then activates cognitive control, allocating more attention to task-relevant information than to task-irrelevant information. Therefore, an elevated level of control after the detection of a high conflict trial (i.e., incongruent trial) enhances performance in the next trial, reducing the difference in response time between incongruent and congruent trials (i.e., the congruency effect). This theory is supported by fMRI studies that show stronger ACC activation after incongruent trials than after congruent trials in flanker tasks (e.g., Botvinick, Nystrom, Fissell, Carter, & Cohen, 1999; Carter et al., 2000). Due to its conceptual plausibility and neural evidence, the conflict monitoring theory has been one of the most influential frameworks for the study of control mechanisms. However, it is still unclear whether the conflict-driven control is the only process that contributes to control-related phenomena (Egner, 2007).

Gratton et al. (1992) that first reported the CSE supposed that the attentional biases causing this effect derive from the expectation of the upcoming trial type. Expectancy affects response time in a way that expected stimuli are responded to faster than unexpected stimuli (Yu & Cohen, 2009). The response to repeated stimuli (e.g., A after AAAA) are facilitated by repetition expectancy, and the response to alternated stimuli (e.g., B after BABA) benefit from alternation expectancy (Ayton, Hunt, & Wright, 1989; Lopes, 1982; Remington, 1969; Soetens, Boer, Hueting, 1985). The CSE is generated with the repetition expectancy, with which the same trial type as the previous one is expected. This prior belief reduces the congruency effect after an incongruent trial, by allocating more attention to the task-relevant information in anticipation of another incongruent stimulus (Gratton et al., 1992).

In addition to the CSE, expectancy-based control accounts for another key behavioral phenomenon, the proportion congruency effect. The congruency effect is often modulated by a long-term probability of congruency (e.g., Bugg & Chanani, 2011; Bugg, McDaniel, Scullin, & Braver 2011; Hutchison, 2011; Kane & Engle, 2003; Logan & Zbrodoff, 1979; Tzelgov, Henik, & Berger, 1992). For example, Tzelgov et al. (1992) observed that the congruency effect was larger when the trials were mostly congruent, than when they were mostly incongruent. An example of this effect of list-wide proportion congruency (LWPC) is shown in Figure 2.

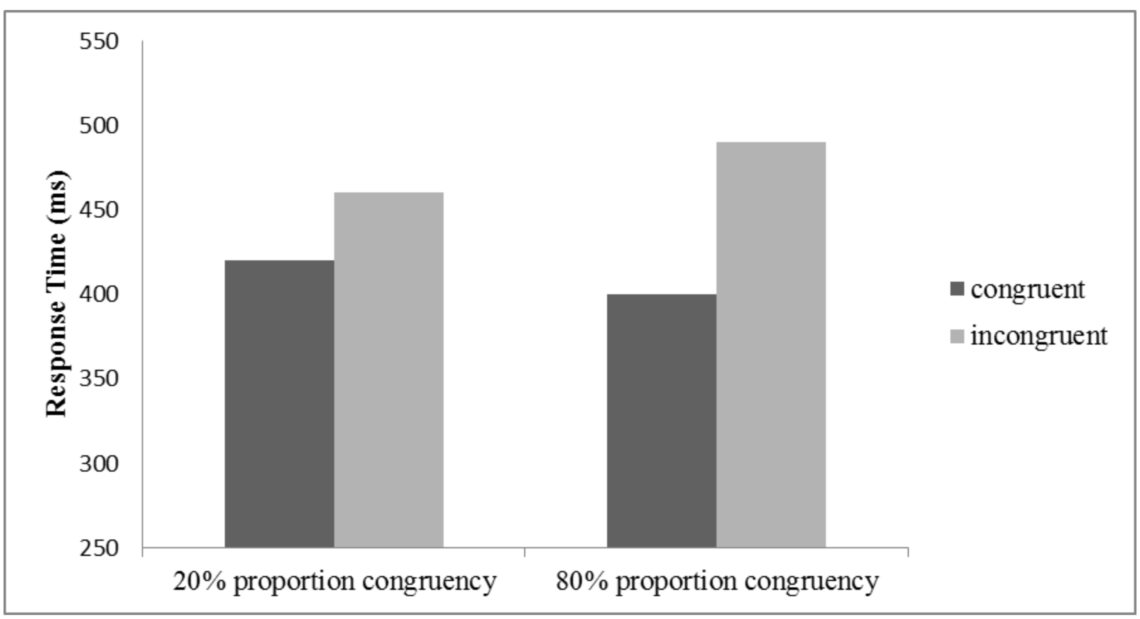


Figure 2. The list-wide proportion congruency (LWPC) effect.

Expectancy-based control provides an explanation of this phenomenon, by assuming that people form explicit probabilistic expectancies for congruency (Logan & Zbrodoff, 1979; 1982). In an experiment with mostly congruent trial, stimulus type is likely to be repeated after congruent trials, and alternated after incongruent trials. This belief in favor of congruent stimuli would increase the congruency effect by weakening the attentional bias toward task-relevant information. The LWPC effect is also consistent with Botvinick et al.'s (2001) conflict-driven account because the overall level of control would be low in mostly congruent condition where conflicts are seldom detected.

One advantage of expectancy-based control over conflict-driven control is that the former accounts for the reversed congruency effect that is regarded as an anomaly in the latter (Logan & Zbrodoff, 1979, 1982; Stürmer et al., 2002). For example, when a repetition of the trial type is strongly expected, the iI trial in Figure 1 can be responded to faster than the iC trial because the iI trial is boosted by the expectancy (e.g., Wendt, Kluwe, & Peters, 2006). Conflict-driven control models (e.g., conflict monitoring model) would not explain such a reversal, because the suppression of task-irrelevant information in those models would at best eliminate the congruency effect. In addition to the reversed congruency effect, expectancy-based control suggests a possibility of a reversed CSE. For example, if an alternation of trial types is expected for most trials, a subject will expect a congruent trial after an incongruent trial, and expect an incongruent trial after a congruent trial. The expectation of alternation leads to a stronger cognitive control and a smaller congruency effect after a congruent trial, as opposed to the CSE. Consistent with this hypothesis, the CSE was observed only when the repetition of congruencies was expected, in the experiment where the subjects explicitly reported their expectations of upcoming stimulus type (Duthoo, Wühr, & Notebaert, 2013). However, it is not clear how the subjects in this study formed their expectation, while the proportion of trial-type repetition or alternation was 50% in the experiment. In order to explicitly affect participants' expectancies, Jiménez and Méndez (2013) manipulated the proportion of trial-type repetition. For example, in the experiment with high proportion of repetition, 70% of the trials were repetition trials (i.e., cC and iI in Figure 1) and the remaining 30% of the trials were alternation trials (i.e., cI and iC in Figure 1). The CSE was observed when the proportion of repetition was high (70%), but not when the probability was low (30%). Jiménez and Méndez (2013) concluded that the effects of expectancies are not significant when considering second-order and third-order repetition trials (e.g., two or three congruent trials in a row before a trial),but it is still possible that the differences in the first-order CSE between the repetition expectancy and the alternation expectancy reflect expectancy-based control mechanisms.

For a methodological standpoint, a possible limitation of the previous studies is that the majority of them relied on null hypothesis significance testing (NHST) to investigate the expectancy-based control effects. NHST compares the averaged data from the limited number of qualitative conditions (e.g., congruent vs incongruent, previous congruent vs previous incongruent), but this method is not likely to capture precisely the gradual adaptation to conflict or gradual formulation of expectancies that are assumed by the models of cognitive control. For example, repetition expectancy is updated based on the proportion of repetition in previous trials, getting closer to the actual probability as the information is accumulated (e.g., Yu & Cohen, 2009). This continuous update of belief and its effect on response time should be included in the analysis, in order to test the effect of expectancy. Assuming that the belief of repetition is adjusted over time, it is possible that the difference in the response time between high and low proportion of repetition does not reach a significant level when we merely compare the averaged data using NHST.

The present study aims to overcome this limitation by using formal mathematical models to discriminate between conflict-driven control and expectancy-based control. The models that represent each account continuously calculate critical variables such as the level of conflict and the expected proportion of repetition, and predict the behavior based on those values. Plausibility of such models and the corresponding theories would be tested by their fit to behavioral data, rather than by rejecting or failing to reject the null hypothesis. We believe that this model-based approach can help capture the specific variances of behavior that reflect cognitive processes of interest.

To evaluate the models, it is essential to have a task that measures the behaviors that reflect cognitive control. We will use a version of the flanker task (Eriksen & Eriksen, 1974), in which the participants are asked to distinguish the direction of the central arrow, while ignoring the other flanker arrows on the sides. For example, the correct response is "right" for >>>>> or <<><<, and it is "left" for >><>> or <<<<<. The stimuli are called incongruent when the target arrow and the flanker arrows are pointing to different directions (>><>> and <<><<), and are called congruent if all the arrows are the same (>>>>> and <<<<<). Examples of experimental stimuli are shown in Figure 3. The flanker task is very simple, yet feasible of manipulating the designs such as proportion congruency and proportion of repetition (e.g., Gratton et al., 1992). Mathematical models' predictions can depend upon the designs selected, so it is important to choose a design appropriate for the purpose of the study.

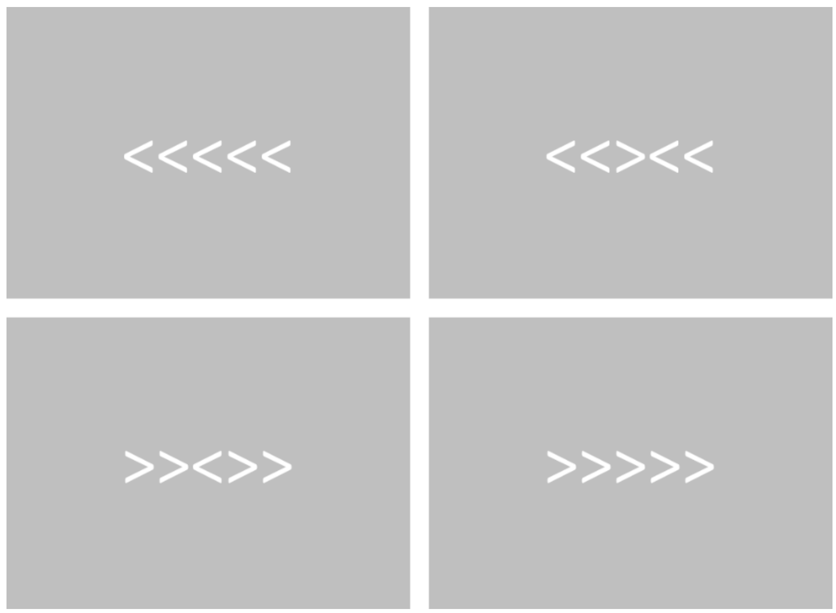


Figure 3. Four possible experimental stimuli of an arrow flanker task.

In discriminating between conflict-driven control and expectancy-based control, the optimal design is relatively straightforward because the overlap between the two accounts in predicting the CSE occurs only when the repetition expectancy is high, as discussed earlier. Therefore, a low repetition rate that yields to different predictions between the two models would be appropriate for the discrimination. However, designs useful for model discrimination may not necessarily help estimating model parameters (Myung, Cavagnaro, & Pitt, 2013). Efficiency of parameter estimation should not be ignored because the parameter values often reflect critical characteristics of cognitive control such as the sensitivity to conflict and the speed of learning the proportion of repetition. Identifying the optimal design for parameter estimation is a non-trivial undertaking because the utility of a design could differ according to the posterior distribution of the parameters.

A statistical method called the adaptive design optimization (ADO) resolves this problem by continuously adjusting experimental designs based on the posterior distributions updated during the experiment (Myung & Pitt, 2009). However, this adjustment is complicated when there are multiple models to be compared and each model requires different design. For example, estimation of conflict-driven control model and expectancy-based model respectively benefit from the manipulation of proportion congruency and proportion of repetition. A reasonable solution that minimizes the loss of information would be to optimize the designs in favor of a model that is more likely to be true. This goal can be achieved by a “hybrid” ADO that facilitates the discrimination of the models at the earlier stage of experiments, and then focuses on evaluating a favorable model at the later stage.

2. Model-based Discrimination of Cognitive Control Mechanisms

For the purpose of the present study, a model-based discrimination of cognitive control mechanisms, it is important to have decent mathematical models that represent each account. The ADO that optimizes experimental designs also requires formal models that have explicit likelihood functions. In this section, the details of the ADO and the models of cognitive control are introduced.

2.1. Adaptive design optimization (ADO)

ADO is a Bayesian statistical framework for optimally and adaptively selecting an experimental design (e.g., stimulus type, presentation time, sequence of stimuli) for each stage of experiments (Cavagnaro, Myung, Pitt, & Kujala, 2010; Myung et al., 2013). Due to its efficiency and flexibility, ADO has been popular in many fields of research. For example, it has been applied to the estimation of visual psychometric functions (Gu et al., 2016; Kujala & Lukka, 2006; Lesmes et al., 2006), to the discrimination of decision making models (Cavagnaro, Gonzalez, Myung, & Pitt, 2013; Cavagnaro, Aranovich, McClure, Pitt, & Myung, 2016), and in clinical trials using experimental drugs (Haines, Perevozskaya, & Rosenberer, 2003; Ding, Rosner, & Müller, 2008).

In the ADO framework, the most informative design is chosen for the next stage of the experiment using the computational models updated by preceding observations. This process is repeated until the end of the experiment, as described in

Figure 4. In design optimization phase, an experimental design for a mini-experiment is determined using the model. This mini-experiment could be a trial, several trials, a block or more of trials. The data is then collected from the mini-experiment using the design selected. The data collected in the experiment phase subsequently updates the model so it can be used to choose the design for the next mini-experiment.

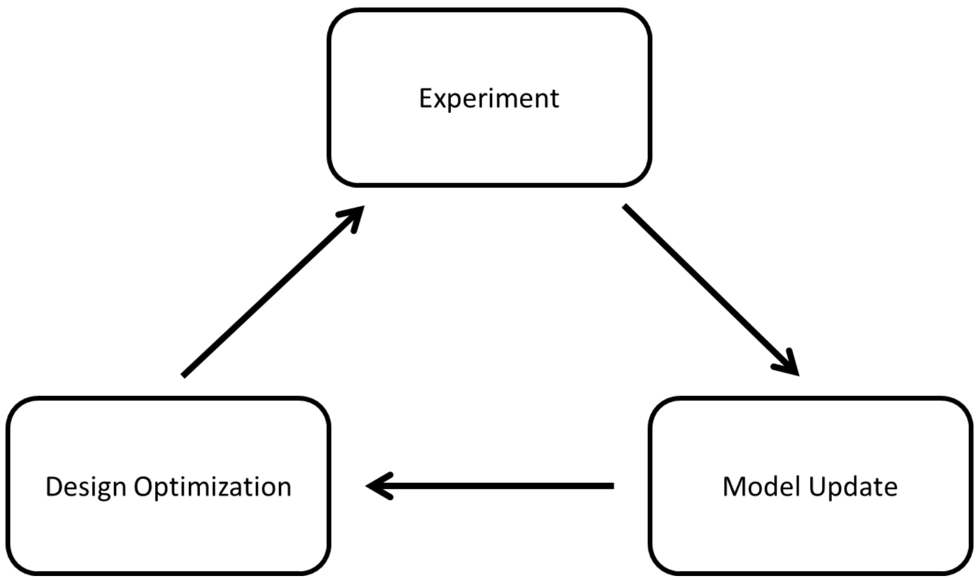


Figure 4. The updating scheme of adaptive design optimization (ADO).

Formally, the optimal design d^* is identified as

$$d^* = \underset{d}{\operatorname{argmax}}\, U(d), \tag{1}$$

where $\mathrm{U}(d)$ is the utility function of the designs. The criterion for the utility or the informativeness of designs is dependent on the specific goal of the study. ADO is compatible with two goals of model-based experiments, which are parameter estimation and model discrimination. The parameter estimation, on the one hand, indicates finding accurate estimates of the model parameters, which can be achieved by updating the model using the data. On the other hand, the model discrimination is to find the model that fits best to the data, by comparing multiple models.

When the purpose of the study is to estimate model parameters, the utility function is defined as

$$U(d) = \int\int u(d, \theta, y)p(y|\theta, d)p(\theta)dyd\theta, \tag{2}$$

where y is the outcome vector resulting from an experiment conducted with design d, θ is the model parameter, $p(y|\theta, d)$ is the model likelihood, and $p(\theta)$ is the parameter prior (Chanloner & Verdinelli, 1995). The local utility function, $u(d, \theta, y)$, is the log ratio of posterior to prior distributions of the parameters,

$$u(d, \theta, y) = ln\frac{p(\theta|y, d)}{p(\theta)}. \tag{3}$$

The basic idea here is to find the design that yields to the largest gain in information about the parameters under the possible outcome y, measured by the Shannon entropy.

For the purpose of model discrimination, the utility function $U(d)$ should be modified to consider multiple models, as follows:

$$U(d) = \sum_m p(m)\int\int u(d, \theta_m, y_m)p(y_m|\theta_m, d)p(\theta_m)dy_m d\theta_m, \tag{4}$$

where m = {1, 2, ..., K} is one of K different models being considered, $p(m)$ is the prior probability that m is the true model that generates behaviors, y_m is the outcome vector in an experiment under design d and model m, and θ_m is a parameter vector of model m.

The local utility function in the above equation reflects possible increase in certainty about model probability as follows:

$$u(d, \theta_m, y_m) = ln \frac{p(m|y, d)}{p(m)}, \tag{5}$$

where $p(m|y, d)$ is the posterior model probability of model m updated by the outcome y under design d.

The model and parameter priors are updated by Bayes rule and Bayes factor, using the data observed from the previous stage of experiment:

$$p_{s+1}(m) = \frac{p_s(m)}{\sum_{k=1}^{K} p_s(k) BF_{(k,m)}(z_s|d_s^*)} \tag{6}$$

$$p_{s+1}(\theta_m) = \frac{p(z_s|\theta_m, d_s^*) p_s(\theta_m)}{\int p(z_s|\theta_m, d_s^*) p_s(\theta_m)\, d\theta_m} \tag{7}$$

where s = {1, 2, …} denotes an ADO stage, m = {1, 2, ..., K}, and z_s is the outcome observed in the stage s. An ADO stage could be a single trial or a sequence of trials, depending on how frequently the experimenter wishes to adjust the design. The Bayes factor, $BF_{(k,m)}(z_s|d_s^*)$ in Equation (6) is defined as

$$BF_{(k,m)}(z_s|d_s^*) = \frac{\int p(z_s|\theta_k, d_s^*) p_s(\theta_k)\, d\theta_k}{\int p(z_s|\theta_m, d_s^*) p_s(\theta_m)\, d\theta_m}, \tag{8}$$

The posteriors obtained in Equation (6) and (7) are then used as the priors to identify the optimal design d^* for the next stage of the ADO experiment (See Myung et al., 2013 for more details).

2.2. Formal models of cognitive control

Formal mathematical models are required for ADO-based experiments where the designs are chosen based on the model likelihood. Since the purpose of our study is to discriminate between the expectancy-based control and the conflict-driven control, it would be appropriate to use the models that reflect those two mechanisms. There are various models of cognitive control that represent each control module (e.g., Blais, Robidoux, Risko, & Besner, 2007; Botvinick et al., 2001; Yu & Cohen, 2009; Verguts & Notebaert, 2008), but they usually do not explicitly provide a likelihood function that is required for the ADO implementation. Therefore, we developed our own models that have an explicit expression of the likelihood function, based on the assumptions and structures of the models previously suggested by other researchers.

2.2.1. Expectancy-based control model

The expectancy-based control model is built upon the model suggested by Yu and Cohen (2009) that explained sequential effects in congruency tasks as the effect of repetition expectancy. There are two trial types in a congruency task, a congruent type and an incongruent type. The trial types can be repeated or alternated over two consecutive trials. Let X_t be a set of binary observations (x_1,…, x_t), where x_t = 1 if the congruencies are repeated, and x_t = 0 if the congruencies are alternated in the t^{th} trial (see Table 1).

Table 1. Observable repetitions or alternations represented by x_t.

$(t-1)^{th}$ trial type	t^{th} trial type	x_t
Congruent	Congruent	1
Congruent	Incongruent	0
Incongruent	Congruent	0
Incongruent	Incongruent	1

It is assumed that subjects believe that there is a fixed probability u of observing a repetition of either trial type (congruent or incongruent), that is, with u being equal to $\mathrm{p}(x_t = 1)$. The model assumes that u is updated as,

$$u = \lambda u + (1-\lambda)\frac{r_t}{t} \tag{9}$$

where, r_t denotes the observed number of repetitions of trial types (i.e., x_t=1) up to the t^{th} trial in a trial block. The initial belief u_0 at trial $t = 0$ is assumed to follow Normal (0.5, 0.1).

Equation (9) can be approximated as

$$\hat{u} \approx \lambda u + (1-\lambda)d_r \tag{10}$$

where d_r is a design variable representing the proportion of repetition in a block of trials.

It is important to note that d_r as a single design variable would not fully control the trial sequences in a block. For example, a list of trials where the proportion of repetition is 30% for congruent trials and 70% for incongruent trials would show 50% d_r as like a list of trials with 50% proportion of repetition for both the trial types, assuming that both the lists have 50% proportion of congruency. It would be problematic if the effects of repetition differ in the two lists above, because they are considered to have the same effect according to Equation (10). To prevent these uncontrolled effects of repetition, we assumed separate d_r and u values,

$$\hat{u}_k \approx \lambda u_k + (1-\lambda) d_{r,k} \tag{11}$$

where k indicates the trial types, k = {incongruent, congruent}.

According to the model of Yu and Cohen (2009), the expectation about an upcoming stimulus has a linear relationship with the response time, in a way that expected stimuli are responded to faster than unexpected stimuli. Incorporating this assumption to the present model, we assume that the belief $\mathrm{p}(x_t|X_{t-1})$ about the t^{th} trial is linearly transformed into the response time RT_t as follows:

$$\begin{gathered} RT_t(x_t) = \beta_0 + \beta_1[1 - \mathrm{p}(x_t|X_{t-1})] + \beta_2 I(incong) \\ +\beta_3[1 - \mathrm{p}(x_t|X_{t-1})] I(incong) + \varepsilon \end{gathered} \tag{12}$$

where $\beta_0 > 0,\ \beta_1 > 0,\ \beta_2 > 0,\ \beta_3 > 0,\ x_t = \{0,1\}$, $\mathrm{p}(x_t = 1|X_{t-1}) = u$, and $\mathrm{p}(x_t = 0|X_{t-1}) = 1 - u$. In this equation, $I(incong)$ is equal to 1 for an incongruent trial, and 0 for a congruent trial, and ε is a normal error distributed as Normal $(0,\sigma^2)$.

Putting these together, the probability density function of the response time RT_t in trial t is given by

$$f_{RT}(RT_t|\mu,\sigma) = \frac{1}{\sqrt{2\sigma^2\pi}} e^{-\frac{(RT_t-\mu)^2}{2\sigma^2}} \tag{13}$$

where $\mu = \beta_0 + \beta_1[1 - \mathrm{p}(x_t|X_{t-1})] + \beta_2 I(incong) + \beta_3[1 - \mathrm{p}(x_t|X_{t-1})] I(incong)$.

The likelihood function given an observation $y_t = RT_t$ is then derived as follows:

$$\mathrm{L}\ (y_t\ |\theta = (\beta_0, \beta_1, \beta_2, \beta_3, \sigma, u_0), t, n, d_r) = \prod_{t=1}^{n} f_{RT}(RT_t|\mu,\sigma) \tag{14}$$

2.2.2. Conflict-driven control model

Botvinick et al. (2001) provides a representative model of conflict-driven control, but their model is not amenable to ADO, which requires an explicit expression of the likelihood function. We therefore developed a new ADO compatible model as an implementation of the conflict monitoring theory, with the following assumptions described below.

The level of conflict from the previous trial, c_t, is simplified to a binary value in the current model, as c_t is 0 if the previous $t-1^{th}$ trial was congruent, and 1 if it was incongruent. The perceived level of conflict in the current trial is defined as

$$C_t = \lambda c_t + (1-\lambda)(\overline{c} + a) \quad (15)$$

where $\overline{c}$ is the average of c_t values up to the current trial, and a is a scaling value that represents the base level of conflict.

Equation (15) can be approximated as

$$\widehat{C_t} \approx \lambda c_t + (1-\lambda)(d_c + a) \quad (16)$$

where d_c is the proportion congruency in a block of trials.

In the conflict monitoring model (Botvinick et al., 2001), a high level of conflict accelerates the response to an incongruent trial, and decelerates the response to a congruent trial, on average. The response time RT_t in the current model follows the same concept:

$$RT_t = \beta_0 + \beta_1 C_t + \beta_2 I(incong)(1-\beta_3 C_t) + \varepsilon \quad (17)$$

where $\beta_0 > 0,\ \beta_1 > 0,\ \beta_2 > 0,\ 1 > \beta_3 > 0,$ and $1 > C_t > 0$. As in the expectancy-based model, $I(incong)$ is 1 for an incongruent trial, and 0 for a congruent trial, and ε is a normal error following Normal $(0,\sigma^2)$.

The probability density function and the likelihood function are then derived as,

$$f_{RT}(RT_t|\mu,\sigma) = \frac{1}{\sqrt{2\sigma^2\pi}} e^{-\frac{(RT_t-\mu)^2}{2\sigma^2}} \tag{18}$$

$$\mathrm{L}\ (y_t\ |\theta = (\beta_0, \beta_1, \beta_2, \beta_3, \sigma, a), t, n, d_c) = \prod_{t=1}^{n} f_{RT}(RT_t|\mu,\sigma) \tag{19}$$

where $\mu = \beta_0 + \beta_1 C_t + \beta_2 I(incong)(1 - \beta_3 C_t)$.

2.3. Design variables to be optimized

Each of the two models above has a design variable, the proportion of repetition (D1) for the expectancy-based model (M1) and the proportion congruency (D2) for the conflict-driven model (M2). To fully control the number of the four possible trial types (cC, cI, iC, and iI), D1 was separately manipulated for congruent and incongruent trials. The task of ADO is to select an optimal combination of the two design variables at each block of an experiment. One issue is that some of the combinations are impossible because the two variables are not completely independent with each other (see Figure 5). For example, it is unviable to make a stimuli list with 30% D1 for congruent trials and 70% D2. At lease about 57% (40/70) of the congruent trials should be repeated when 70% of the trials are congruent. Such impossible designs were excluded from ADO. A list of ADO-feasible designs used in the present study is shown in Table 2. For computational simplicity, both D1 and D2 were discretized into a grid ranging from 30% to 70%, with 10% intervals. The variables below 30% or above 70% were not used because they only made a limited number of design combinations that often result in seemingly "systematic" (e.g., easily predictable) sequences of trials.

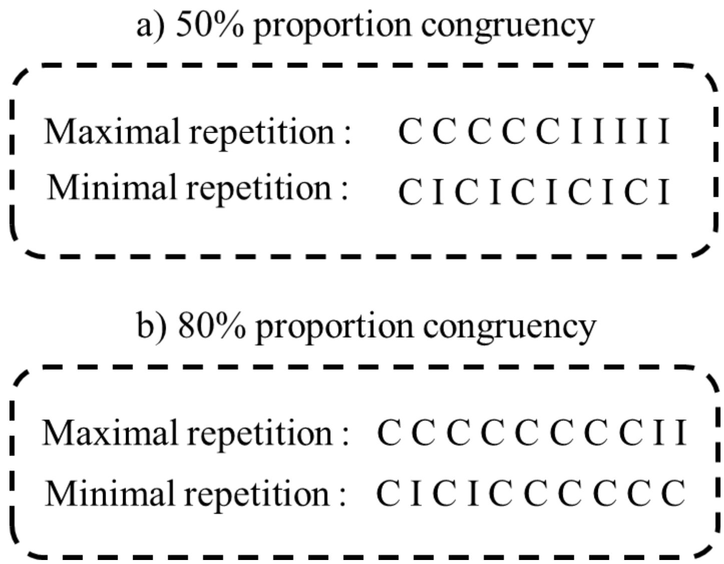


Figure 5. A dependency between the proportion of repetition (D1) and the proportion congruency (D2). The letters in the box show possible sequences of congruent (C) and incongruent (I) trials in a block of 10 trials. a) When the proportion congruency is 50%, the maximum number of repetition (i.e., CC or II tirals) is 9, and the minimum is 0. b) With 80% proportion congruency, the maximum is still 9, but the minimum is 4. The range of possible proportion of repetition is dependent on the proportion congruency.

Table 2. The designs used in the ADO selection.

Design	D1 for congruent	D1 for incongruent	D2
30-30-50	30%	30%	50%
30-50-40	30%	50%	40%
30-60-40	30%	60%	40%
30-70-30	30%	70%	30%
40-40-50	40%	40%	50%
40-60-40	40%	60%	40%
40-70-30	40%	70%	30%
50-30-60	50%	30%	60%
50-50-50	50%	50%	50%
50-70-40	50%	70%	40%
60-30-60	60%	30%	60%
60-40-60	60%	40%	60%
60-60-50	60%	60%	50%
60-70-40	60%	70%	40%
70-30-70	70%	30%	70%
70-40-70	70%	40%	70%
70-50-60	70%	50%	60%
70-60-60	70%	60%	60%
70-70-50	70%	70%	50%

2.4. Hybrid ADO

The primary purpose of the present study is the comparison between conflict-driven control and expectancy-based control in an ADO-based experimental study. The ADO for model discrimination would satisfy this goal, but this method alone does not guarantee convergence of model parameter estimates, which is the secondary goal of the study. To elaborate, fixing the proportion of repetition (D1) at a low level is likely to be advantageous for the discrimination, but it is necessary to manipulate that design variable during the experiment in order to estimate the learning rate parameter λ in the expectancy-based model. As a way to find a balance between two possibly conflicting goals, we propose a method that combines model discrimination and parameter estimation in a judicious manner, dubbed "hybrid ADO". The basic idea of hybrid ADO is to use the designs that maximize the differences in prediction between the models at first over a certain number of experimental trials, and then switch to the optimization for parameter estimation once a model is decisively favored in some defined sense. An issue is then the timing of such transition. If the model discrimination phase is too short, there would not be enough information for comparing the models. If it is too long, the data collected after a certain point would be unnecessary. In the hybrid ADO, the timing is determined by the posterior model probability (i.e., p(m) where m = M1 or M2) in the model discrimination ADO. Once the p(m) of a model reaches the threshold (e.g., 0.95), the parameter estimation ADO begins with the selected model as the target. The loss of information is minimized because the discrimination phase stops at the moment a model is proven to be superior to the others. This efficiency is not achievable with experiments with pre-determined designs, especially when it is unknown which model fits better with participants before we run the experiments (see Figure 6).

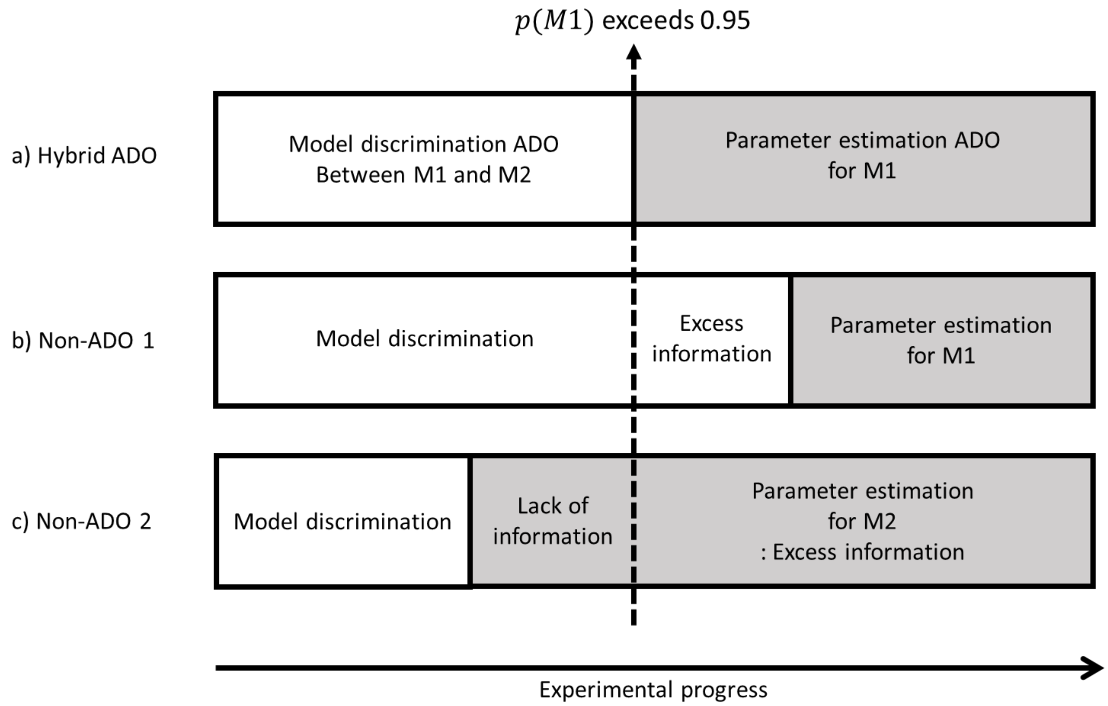


Figure 6. Examples of a Hybrid ADO experiment and prescheduled non-ADO experiments. The white area contains the experimental trials for model discrimination, and the grey area contains the trials for parameter estimation. Vertical dotted line in the center indicates the timing that the models are sufficiently discriminated. M1 and M2 indicate two models to be compared in the experiments. a) A hybrid ADO experiment switches from the model discrimination ADO to the parameter estimation ADO for M1 as soon as the model probability p(M1) reaches the threshold of 0.95. b) A non-ADO experiment in favor of M1. The predetermined number of trials in the model discrimination phase is longer than necessary in this experiment. c) A non-ADO experiment in favor of M2. The length of model discrimination phase is too short in this case, leading to a premature transition to the parameter estimation for M2, which is inefficient because the data supports M1.

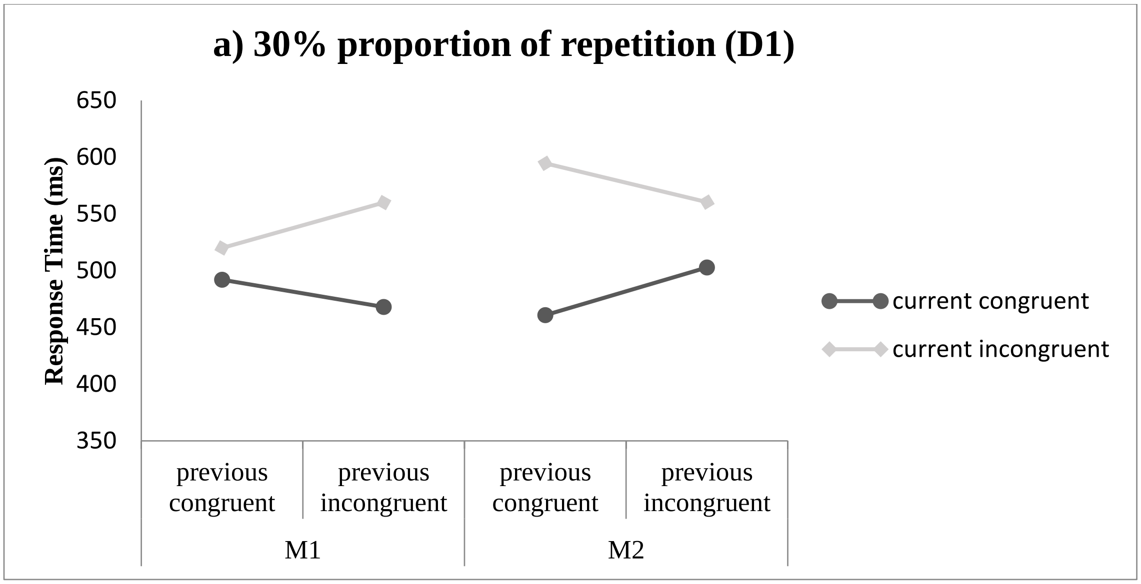
For a non-adaptive design experiment to be efficient, there should be a certain model generalizable to every participant behavior, along with reasonably good fixed designs to evaluate the model. If a single model could account for the behavior of any individual, the hybrid ADO would not be necessary. However, it is possible that there is a trait-like tendency for each individual to prefer a certain type of control strategy (Braver, 2012), among multiple strategies each represented by a model. The hybrid ADO would reach its full potential in such situation, where there are significant individual differences so that different models are required for each participant.

3. Simulational Studies

We conducted simulations to demonstrate the efficiency of the hybrid ADO for discriminating between the two models of cognitive control. The flanker task experiment that consisted of 6 blocks of 40 trials each was iteratively simulated. The design variables were adaptively optimized after the completion of each block. Having a small number of trials in each block was necessary for more frequent adaptation, but the number of trials had to be large enough to observe the response time data from all of the four possible trial types (cC, cI, iC, and iI) even when the design values are extremely large or small. For example, when the proportion of congruency (D2) is 30% and the proportion of repetition (D1) is 30%, only about 9% of the trials are congruent trials that are repeated (cC). With 40 trials in a block, we can observe such infrequent stimulus at least three times from every block. Parameter updating could be problematic with a smaller number of trials since there would be just too few observations for some trial types.

The ADO simulations started with uninformative, uniform prior on parameters, except for λ about which we had explicit assumptions in the models. The λ in the expectancy-based model (M1) indicates the rate with which the prior belief about D1 is maintained after an update. In the conflict-driven model (M2), λ denotes relative influence of the current conflict and the average conflict on the perceived level of conflict. We assumed that the belief in M1 is updated gradually rather than abruptly, and the current and previous conflicts are equally important in M2. Therefore, the prior $p(\lambda)$ was set to be relatively high for M1, $p(\lambda)_{M1}$ ~ Beta(8, 3), and medium for M2, $p(\lambda)_{M2}$ ~ Beta(5, 5). The prior distributions of the other parameters were $p(\beta_1)$ ~ Uniform(0,150), $p(\beta_2)$ ~ Uniform(0,150), and $p(\beta_3)$ ~ Uniform(0,150) for M1, and $p(\beta_1)$ ~ Uniform(0,80), $p(\beta_2)$ ~ Uniform(0,200), and $p(\beta_3)$ ~ Uniform(0,1) for M2. The parameters of less interest, β_0 and σ, were fixed as β_0 = 450 and σ = 50 to reduce the computational load. The initial model probability, $p_0(m)$, was set to 0.5 for both the models. In the hybrid ADO, the initial, model-discrimination phase is switched to the parameter estimation phase once the model probability of either model exceeds 0.95. To simplify the numerical computations required for ADO, the design space that includes parameter values, design variables, and response time was discretized into a finite grid consisting of equally spaced values for each dimension (Myung et al., 2013). The utility U(d) in Equation (1) was estimated for each design in the discretized space, to find the design d* with the highest utility.

The response time in the simulations were generated by either of the models using fixed parameter values. The parameters of the data-generating model were set as u_0 = 0.6, β_0 = 450, β_1 = 60, β_2 = 40, β_3 = 40, λ = 0.8, and σ = 50 for M1, and as a = 0.1, β_0 = 450, β_1 = 60, β_2 = 150, β_3 = 0.7, λ = 0.7, and σ = 50 for M2. These parameter values were chosen such that expected patterns are similar between the two models when D1 is high, but are clearly different when the probability is low. The response time patterns predicted by those fixed models are depicted in Figure 7.



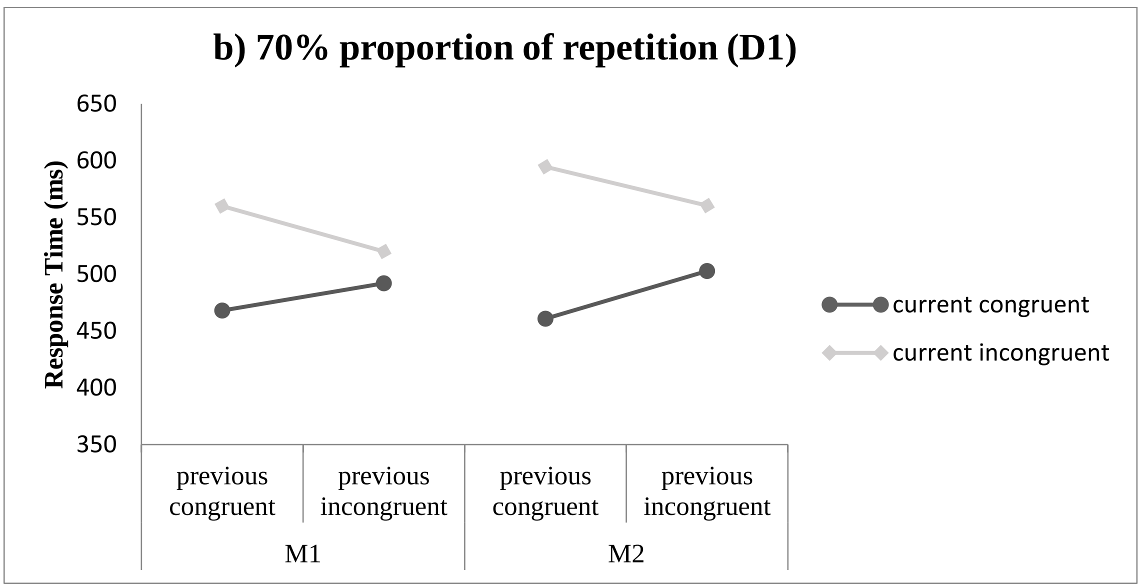


Figure 7. Expected response time patterns generated by the expectancy-based model (M1) and the conflict-driven model (M2). a) is the patterns with 30% proportion of repetition. b) is the patterns with 70% proportion of repetition. The proportion congruency was fixed to 50%.

The goal of the simulations was to compare relative performance of three different design selection methods. They were the hybrid ADO (H-ADO), the model comparison ADO (M-ADO), and predetermined fixed designs. The parameter estimation ADO was not separately simulated because it requires a specific target model to be estimated, while we assumed that the two models are equally probable at the initial stage of the simulations. Instead of the parameter estimation ADO, two predetermined designs that focus on the effect of a target design variable were tested. In the first fixed designs (F1), D1 that directly affect the model prediction of M1 was manipulated. D1 was 30% in the first three blocks, and was 70% in the latter three blocks, for both congruent and incongruent trials. D2 was fixed to 50%. In the second fixed designs (F2), on the other hand, the effect of D2 was of our main interest. The trials were 30% congruent in the first three blocks and were 70% congruent in the latter three. D1 could not be fixed, because it is not independent from D2 (see Figure 5). We set up D1 as 30% for congruent and 70% for incongruent in 30% congruent blocks, and as 70% for congruent and 30% for incongruent in 70% congruent blocks. The resulting designs were similar to the prescheduled, balanced designs used in the previous studies that compare the behaviors from different conditions (e.g., Carter et al., 2000; Duthoo et al., 2013; Jiménez & Méndez, 2013). By including the designs similar to those conventional ones, we expected to find from the simulations the difference in efficiency between the ADO and the predetermination.

3.1. Simulation 1: Expectancy-based model

A total of 150 independent simulation runs were performed for each of the three design selection methods described in the previous section, with the artificial data generated by the expectancy-based model (i.e., model M1) defined in Eq. (13). The parameter values of M1 that generated the data were as follows: u_0 = 0.6, β_0 = 450, β_1 = 60, β_2 = 40, β_3 = 40, λ = 0.8, and σ = 50.

The frequency of design selections and the estimated posterior model probability, p(M1), in each of the six blocks are shown in Tables 3 and 4, for the hybrid ADO (H-ADO) method and the model-discrimination ADO (M-ADO) method, respectively. The first thing to note from the results is that only four out of a total of 19 designs listed in Table 2 were chosen consistently and repeatedly by both H-ADO and M-ADO. In particular, in block 1, only one design (i.e., 30-30-50, see Table 2) with a low proportion of repetition (D1) was always selected under both design selection methods. This is probably because this particular combination of 30% D1 for both congruent and incongruent trials, and 50% proportion of congruency (D2) is the most informative one among the 19 designs for discriminating between the two models, M1 and M2, as discussed earlier (see Fig. 7). After block 1, the designs with higher repetition probability were also selected by H-ADO, increasingly more so as the experiment moved from the model discrimination phase to the parameter estimation phase. The most frequently selected design in the last block was the one with the highest D1 (i.e., 70-70-50). Apparently, a high D1 is good for parameter estimation of model M1, because response time differences between low and high D1 is essential in estimating the λ parameter. λ in M1 indicates the learning rate for repetition expectancy, which can be estimated by manipulating D1 to see how quickly the behavior changes.

The results for M-ADO in Table 4 depict a pattern somewhat different from that of H-ADO. That is, only two out of the 19 possible were selected across the six blocks; most of the time, the 30-30-50 design with the lowest D1 and occasionally, the 30-70-30 design with the lowest D2. The superiority of the 30-30-50 design in model discrimination was as observed with H-ADO, but the selection of the 30-70-30 design was not as expected. The 30-70-30 design with 30% D2 mostly produces incongruent trials, while some of the parameters (i.e., β_2, β_3 in both M1 and M2) can only be updated by observing incongruent trials. It seems that this design is preferred to the 30-30-50 design according to the posterior parameter distributions.

Table 3. Number of times each design was selected by the hybrid ADO (H-ADO) method in Simulation 1. The combinations of the numbers in the first column indicate “D1 for congruent trial - D1 for incongruent trial - D2”. For example, 30-30-50 design has 30% D1 for both congruent and incongruent trials, and 50% D2.

	Block					
Design	1	2	3	4	5	6
30-30-50	150	139	126	92	52	26
30-70-30	0	10	17	20	20	20
70-30-70	0	0	2	18	34	37
70-70-50	0	1	5	20	44	67
Mean p(M1)	0.50	0.41	0.58	0.74	0.84	0.91

Table 4. Number of times each design was selected by the model discrimination ADO (M-ADO) method in Simulation 1.

Design	Block					
	1	2	3	4	5	6
30-30-50	150	139	136	136	136	136
30-70-30	0	11	14	14	14	14
70-30-70	0	0	0	0	0	0
70-70-50	0	0	0	0	0	0
Mean p(M1)	0.50	0.38	0.55	0.71	0.85	0.93

Figure 8 shows the posterior model probability of M1 as a function of trial block for each of the four methods of design selection. The model discrimination performance of H-ADO was similar to that of M-ADO. Apparently, the transition scheme from model discrimination to parameter estimation under H-ADO didn't matter much, as far as model discrimination is concerned. As for the non-adaptive, F1 method of design selection, its model discrimination performance was virtually indistinguishable from those of H-ADO and M-ADO. The model probabilities were especially similar among them in the first three blocks, where F1 method chose the 30-30-50 design as like H-ADO and M-ADO did in most of the early blocks. Finally, the non-adaptive, F2 method that by construct is not utilizing the low D1 designs such as 30-30-50 fared the worst in model discrimination, especially in the mid-block regions (i.e., blocks 2-4).

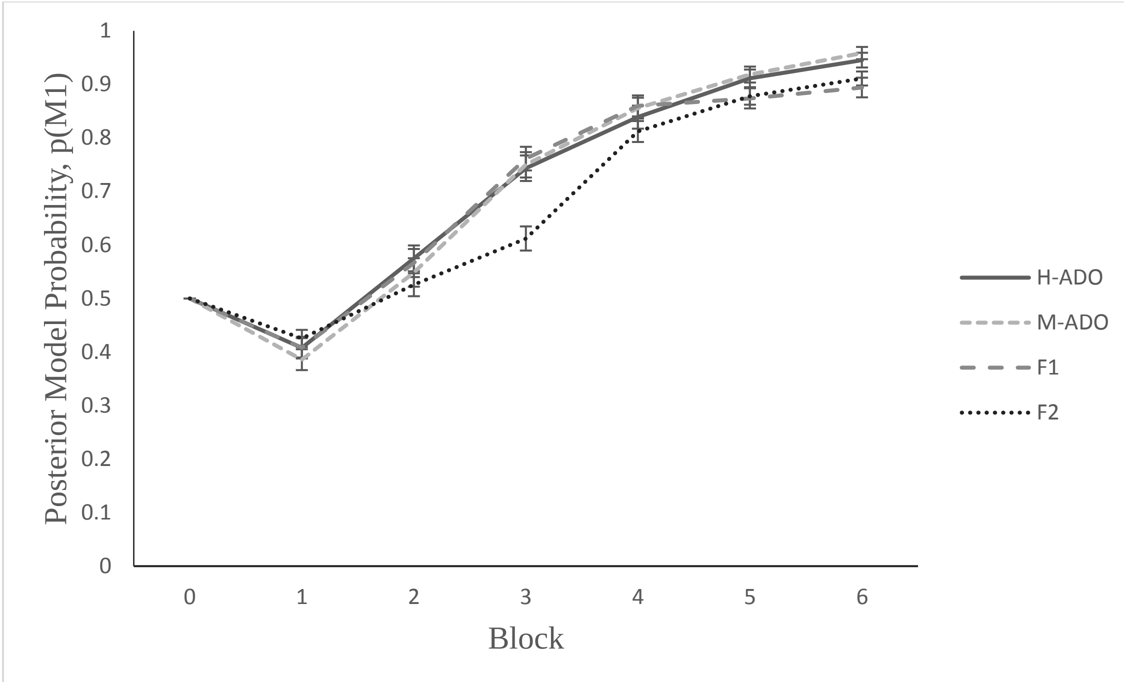


Figure 8. Model probability of the target model (M1) in Simulation 1. The vertical axis indicates the model probability of the expectancy-based model, p(m1). The horizontal axis indicates the block number, which is 0 at the initial stage before the simulation. The error bars represent one standard error at each block.

In addition to the model discrimination, estimating the model parameters as quickly and accurately as possible was an important goal of the design selection methods. The efficiency of parameter estimation for the target model M1 was compared in Figures 9 and 10. Figure 9 shows the root-mean-square error (RMSE) where the errors indicate the difference between the posterior mean of the parameter estimates and the true parameter values used for data generation. The RMSE for each parameter was calculated every block as follows:

$$RMSE = \sqrt{\frac{\sum_{t=1}^{n}(\hat{\psi}_t - \psi)^2}{n}} \quad (20)$$

where $\hat{\psi}_t$ is the posterior mean of the parameter in t^{th} trial, ψ is the true parameter value, and n is the total number of trials in the block. The estimates are considered more accurate if the errors are closer to zero. The method with the highest RMSE overall was the F2 that focuses on observing the effect of D2. The differences in the RMSE among the other methods were generally small. A noticeable pattern was that the RMSE for λ was reduced faster with H-ADO than M-ADO at the later blocks. H-ADO is the same as M-ADO until the models are discriminated, so it is reasonable that the performance of the two methods differs only at the later blocks. The manipulation of D1 that was done by H-ADO at the parameter estimation phase is critical for the estimation of λ, because this parameter indicates the speed with which D1 is reflected to repetition expectancy. It is probable that the utility of the designs and thus the design selections of H-ADO were largely influenced by the amount of information for λ.

Despite some differences in the RMSE, the posterior parameter distributions described in Figure 10 were in similar shape across the methods. The parameter estimates are considered accurate if the posterior distributions peak near the true values marked by the vertical lines in the figure. Overall, the parameter values were recovered accurately, except for one parameter, β_3, of which the posterior distribution did not peak at the true value. β_3 appeared to be the hardest one to estimate, as its influence on the response time was smaller than that of the other parameters given the parameter values used to generate the data.

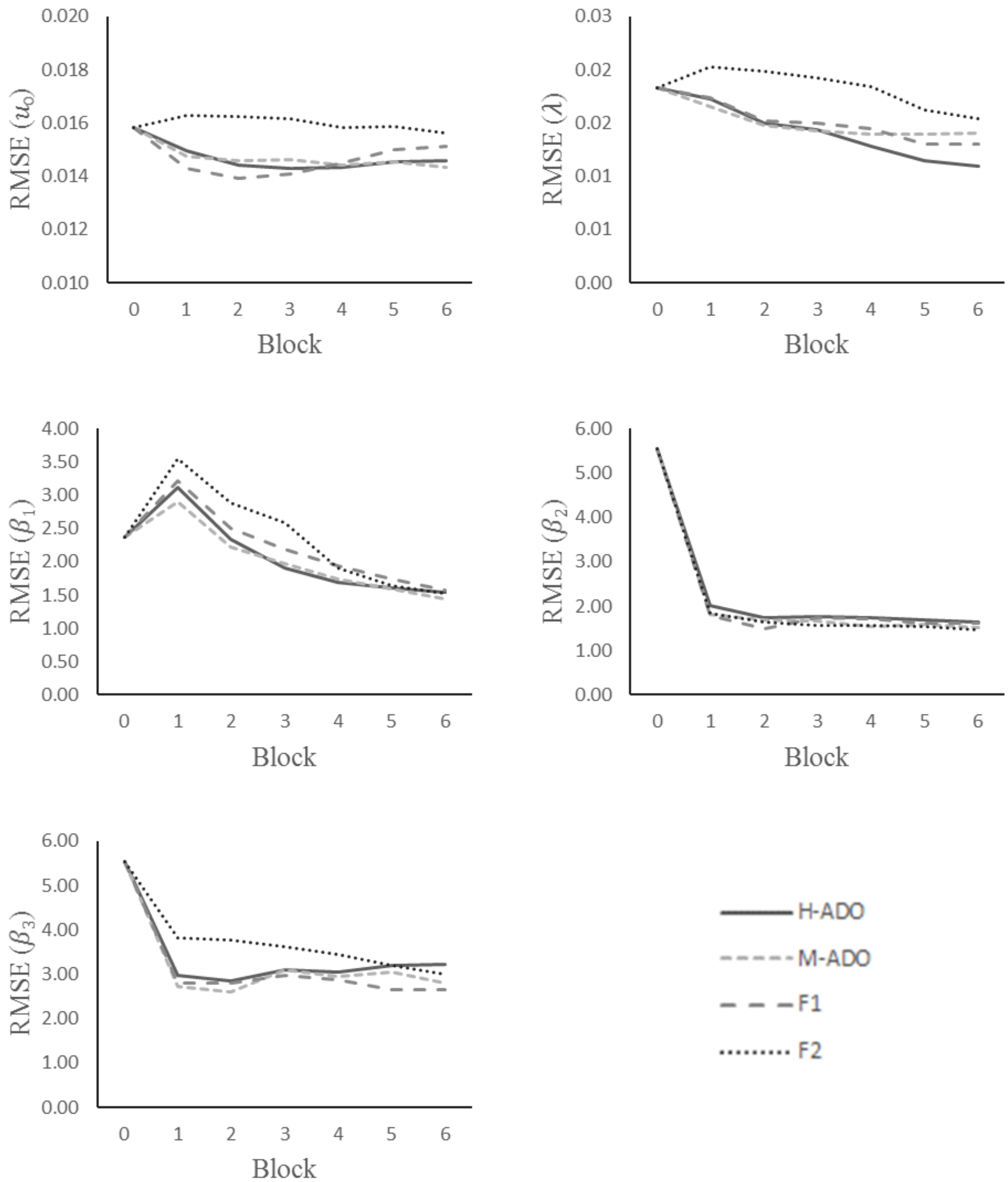


Figure 9. The root-mean-square error (RMSE) of the parameters in the expectancy-based model.

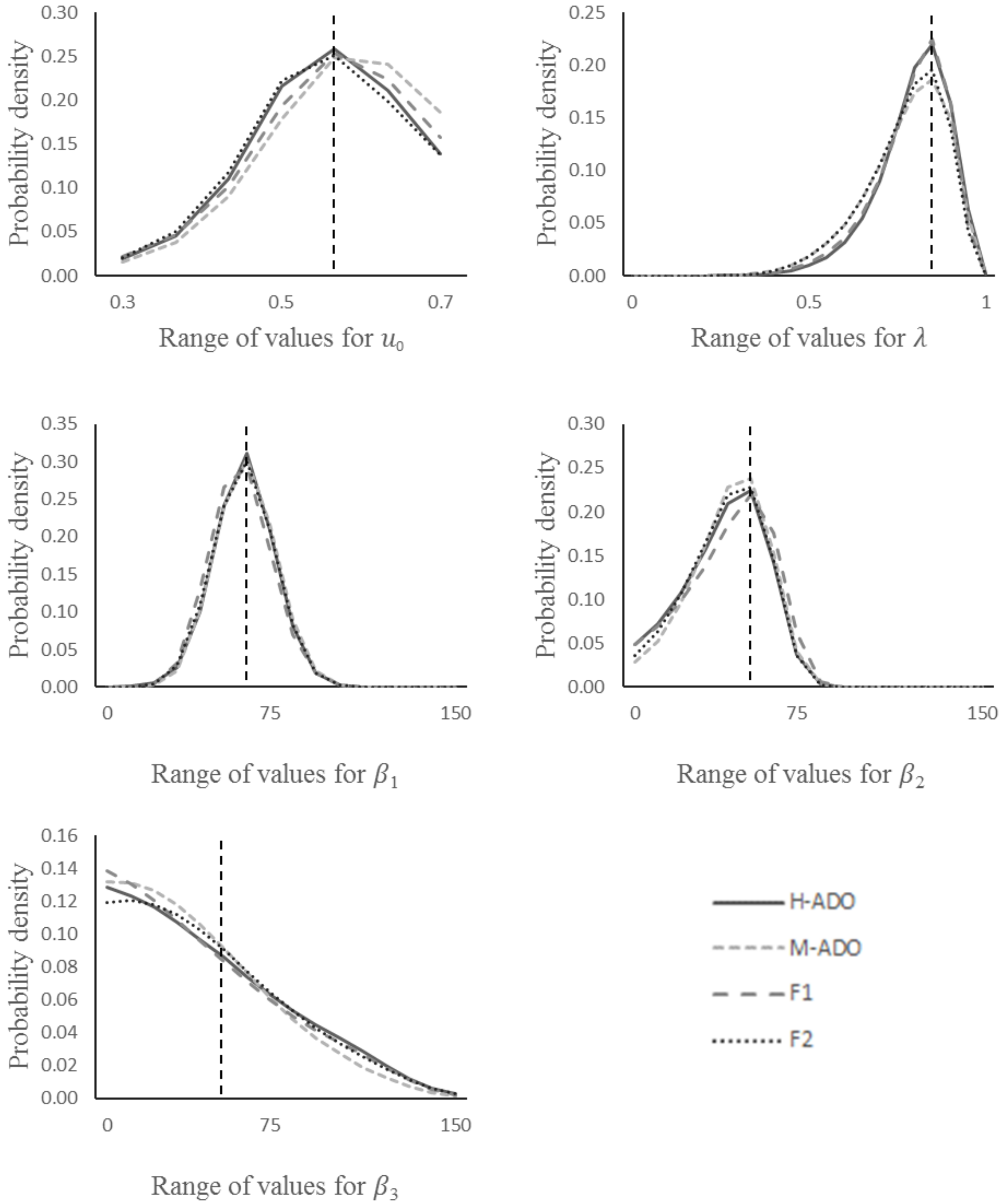


Figure 10. Posterior distributions of the parameters in the expectancy-based model after the end of the simulations. The vertical axis indicates the probability density of the distribution. The horizontal axis indicates the parameter values within their range. The vertical lines in the plots indicate the true values used to generate the data.

In conclusion, the results of Simulation 1 showed that the two models are discriminable, as the posterior model probability for the data generating model (i.e., M1) was significantly higher than 50%. H-ADO at the parameter estimation phase mostly manipulated D1, being consistent with the model structure of M1 where D1 is critical for the parameter estimation. However, Simulation 1 used M1 exclusively as the data generating model, leaving the possibility that the data generated by M2 would not discriminate the models well enough for the H-ADO to move to the parameter estimation phase. In order to confirm that the models are discriminable also with the data generated by M2, another set of simulations was required. This is done and described in the next section.

3.2. Simulation 2: Conflict-driven model

The data in Simulation 2 was generated by M2, with all the other settings the same as Simulation 1. The parameter values used for the data generation were as follows: α = 0.1, β_0 = 450, β_1 = 60, β_2 = 150, β_3 = 0.7, λ = 0.7, and σ = 50.

Tables 5 and 6 show the frequency of the design selections and the posterior model probabilities in H-ADO and M-ADO simulations. The general pattern of design selections for the model discrimination was much similar to that in Simulation 1, as shown in Table 6. M-ADO chose either the 30-30-50 or the 30-70-30 design, suggesting that they are indeed the optimal designs no matter what the data generating model. A difference from the previous simulation occurred with H-ADO, as it mostly selected the 30-70-30 design at the later blocks where the model probability for M2 was high (see Table 5). A design with lower D2 is likely to provide more information about λ, β_2, and

β_3 in M2, because conflict from incongruent trials is essential to estimate those parameters. Hence, it was predictable that H-ADO would choose the design with 30% D2 in the parameter estimation phase.

Table 5. Number of times each design was selected by the hybrid ADO (H-ADO) in Simulation 2.

	Block					
Design	1	2	3	4	5	6
30-30-50	150	112	15	5	2	0
30-70-30	0	34	134	145	148	150
70-30-70	0	1	1	0	0	0
70-70-50	0	0	0	0	0	0
Mean p(M2)	0.50	0.825	0.972	0.993	0.996	0.998

Table 6. Number of times each design was selected by the model discrimination ADO in Simulation 2.

	Block					
Design	1	2	3	4	5	6
30-30-50	150	145	144	141	139	135
30-70-30	0	5	6	8	11	15
70-30-70	0	0	0	1	0	0
70-70-50	0	0	0	0	0	0
Mean p(M2)	0.50	0.863	0.978	0.996	0.998	0.996

The design selections in H-ADO swiftly switched from the 30-30-50 for the model discrimination to the 30-70-30 for the parameter estimation, due to a rapid increase of the model probability for M2. This increasing trend is illustrated in Figure 11 where the model probability is compared among the four methods. The model probability for M2 reached nearly 100% on average after only three or four blocks of simulations, even with F2 method that again showed the worst performance. This seemed quicker than the discrimination in Simulation 1 where M1 generated the data. It suggested that there might be a bias in model probability toward M2, at least under the uniform prior distributions used in the simulations. This bias was also observed in Figure 8, as the model probability for the true model went below the 50% baseline after the first block.

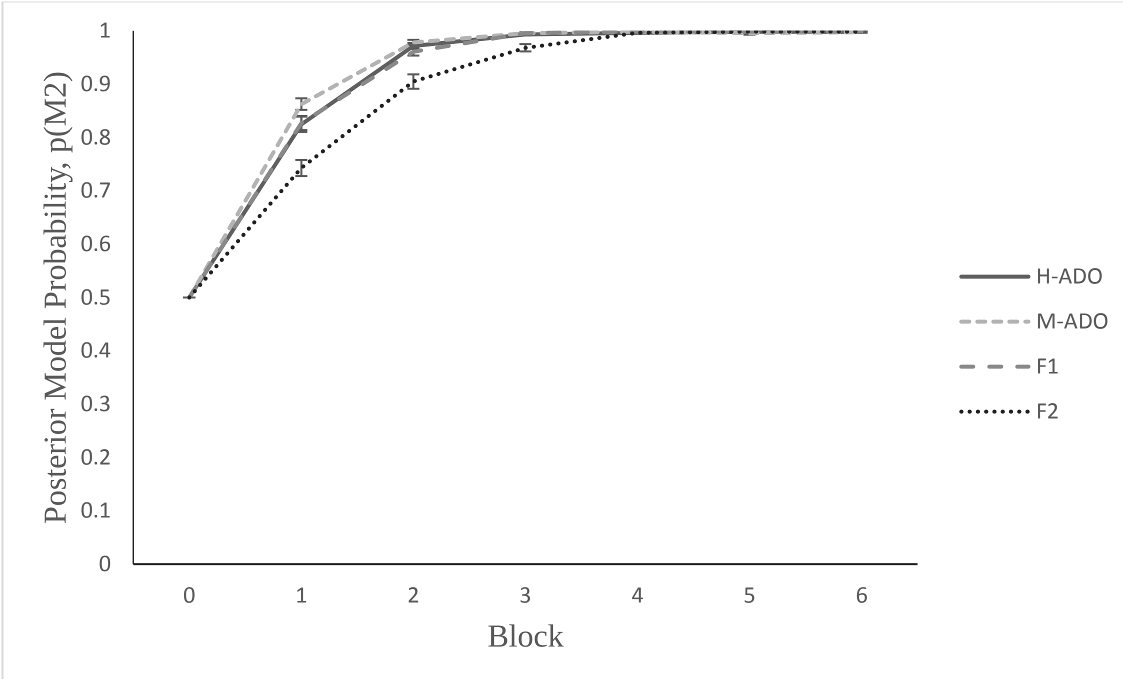


Figure 11. Model probability of the conflict-driven model (M2) in Simulation 2. The vertical axis indicates the model probability of the conflict-driven model, p(M2). The horizontal axis indicates the block number, which is 0 at the initial stage before the simulation. The error bars represent one standard error at each block.

In spite of clear model discrimination, parameter estimation was somewhat problematic in Simulation 2. The RMSE of the parameters plotted in Figure 12 suggested that no method could accurately estimate the parameter a, as the RMSE of a increased over time. The effect of this scaling parameter on the response time is relatively small, and is confounded with the effect of λ. The parameter a is conceptually meaningful as it represents the base level of conflict, but its estimation from limited amount of response time data seems to be difficult, especially when the value of λ is large. The inaccurate estimation of a is also shown in the posterior parameter distributions illustrated in Figure 13. The posterior distributions of a had high variance, and did not peak around the true value. This was especially true for F2 method that showed the highest RMSE in Figure 12. F2 was designed to manipulate D2, which is the design variable of M2, but the simulations suggest that the parameter estimation for M2 does not benefit much from this prescheduled design. A possible solution to the inaccuracy of parameter estimation is the use of informative prior for the parameter, instead of the uniform distribution used in the simulations. However, since we aimed to use the models for human experiments, it was considered premature to set up an informative prior before we actually obtain the information from human subjects. Besides the parameter a, the other parameters relatively converged well around the true value with any design selection method.

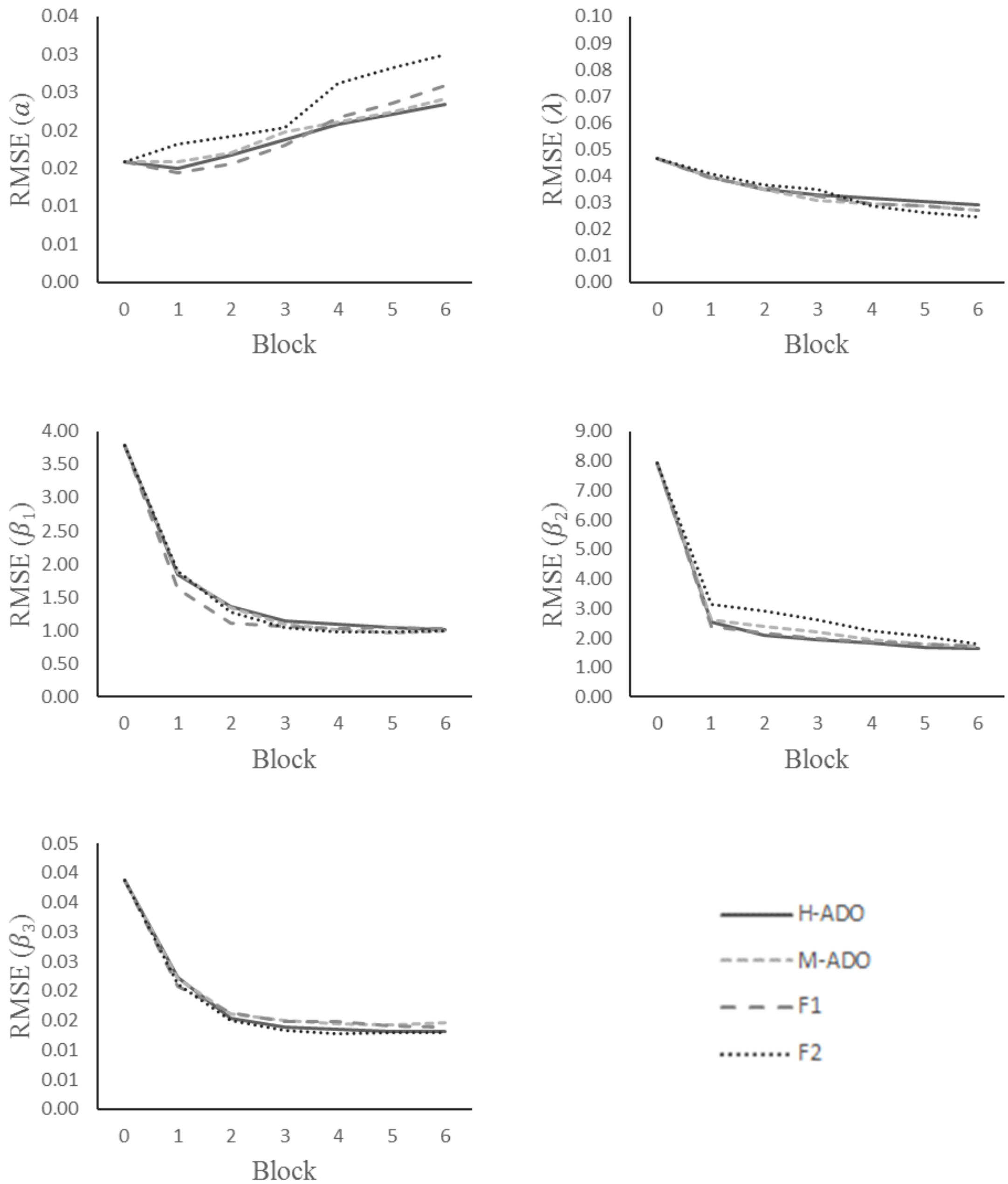


Figure 12. The root-mean-square error (RMSE) of the parameters in the conflict-driven model.

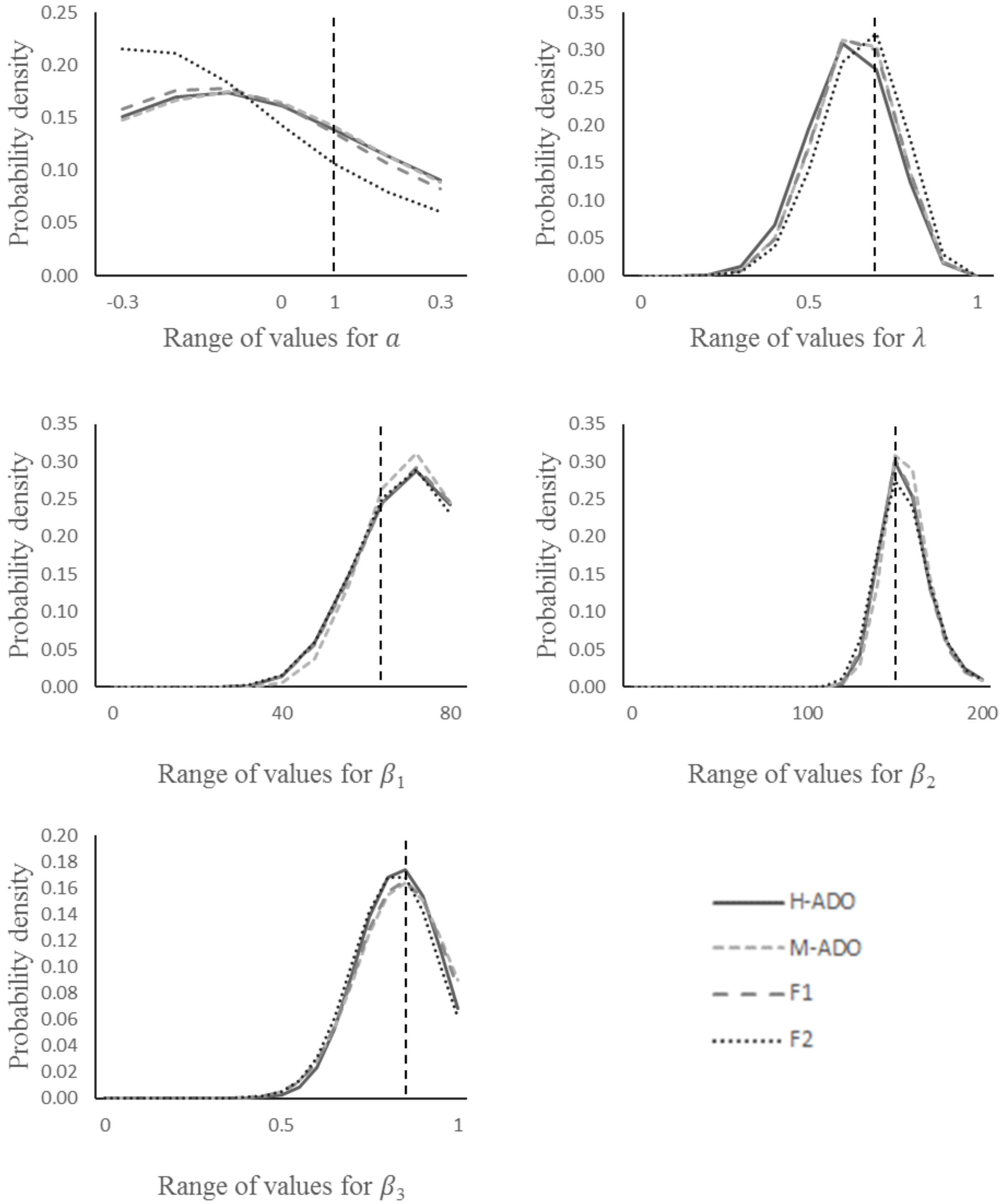


Figure 13. Posterior distributions of the parameters in the conflict-driven model. The vertical axis indicates the probability density of the distribution. The horizontal axis indicates the parameter values within their range. The vertical lines in the plots indicate the true values used to generate the data.

To summarize, the results from Simulation 2 showed that the data generated by M2 could discriminate the models as well as the data from M1 did in Simulation 1. The H-ADO did not show a clear advantage over the other methods in both the model discrimination and the parameter estimation, except for the F2 method that showed the worst performance. The posterior distributions of M2 seemed to be relatively invariant to the design selection methods, showing little benefit of H-ADO. However, it was still possible that H-ADO would perform better than the other methods in human experiments, because estimating the model parameters is likely to be more difficult with the data that is not generated by the model itself. Selecting an optimal design would be more important when the discrimination and the estimation of the models are not easily achieved. We therefore conducted human experiments to see how well the models could be discriminated using the design selection methods in the simulation.

4. Experimental Studies

The arrow flanker task described in Figure 3 was used for the experiments. We assumed that the response time data from the flanker task could discriminate between the two models of cognitive control. However, the model discrimination was likely to be slower than in simulations, because the data would not be completely consistent with the model predictions. As in the simulations, the ADO experiments were compared to the experiments with predetermined designs. A total of three experiments (Experiment 1, 2, & 3) were conducted using the same flanker task stimuli, but with different design selection methods. Experiment 1 used the hybrid ADO, whereas Experiment 2 and 3 used fixed designs that are appropriate to observe the effect of D1 and D2 on response time, respectively.

4.1. Experiment 1

In the simulations, there was a clear distinction between the model probabilities of the two models because either of them generated the data. H-ADO performed well under this condition where the model probability for the data generating model easily exceeds 0.95 during the simulation. However, it was not sure if the response time data observed from human subjects could clearly discriminate the models as in the simulations. Experiment 1 was conducted using H-ADO, in order to see if the participants show discriminable response time patterns that reflect the predictions of the computational models.

4.1.1. Participants

20 undergraduate students at the Ohio State University participated in the experiment (18 - 22 years old, 7 males and 13 females). All subjects had normal or corrected-to-normal vision.

4.1.2. Stimuli and procedure

Stimuli were controlled by PsychoPy (Peirce, 2007) module in Python. The stimuli presented were similar to those in Figure 3. At the beginning of each trial, a white fixation cross (+) was presented at the center of the screen for 800 ms. White colored target stimulus appeared at the same location after 200 ms from the disappearance of the fixation cross, with white flanker arrows on the sides. The arrows remained on the screen for 200 ms. The next trial started after 2 seconds from the onset of the stimulus. All stimuli were presented on a grey background on a LCD monitor.

Subjects performed the experiment in a dimly lit, soundproof room. They had to press the “x” key on the keyboard with their left index finger if the target arrow was pointing to the left, and the “.” key on the keyboard with their right index finger if the target arrow was pointing to the right. They were instructed to focus on the location of the central fixation, and to make a response as quickly and as accurately as possible. If they failed to respond within 1500 ms after the presentation of the stimuli, or if they made an incorrect response, they heard a beep sound indicating the error. There was a practice block consisting of 20 trials, followed by 6 experimental blocks of 40 trials each. The practice block had 50% proportion of repetition (D1) and 50% proportion congruency (D2). There was a one minute break between blocks, which could be prolonged because participants could control when to proceed to the next block by pressing a key in the keyboard. The design variables in each block were selected by the hybrid ADO algorithm used in the simulations.

4.1.3. Technical details

Computational settings of the hybrid ADO were basically the same as those in the simulations, but there were several differences. The discretized grid for response time was equally spaced between 200 ms and 1,300 ms in the simulation. However, because the models predict that the response time follows a normal distribution around a mean, the grid points far from the mean were less likely to be used. For more accurate estimation of response time, we redistributed the grid points after each block, using the normal distribution with the mean and the variance of response time observed in the previous block. The minimum 200 ms grid point was set as 0 percentile and the maximum 1300 ms was set as 100 th percentile. The grid points were then allocated every 2 percentile interval (i.e., 2nd, 4th, 6th, ..., 98th percentile). Therefore, the closer the grid points were to the mean, the higher their density was.

Another difference is that the data from the previous block were also used to estimate β_0 and σ that were fixed in the simulations. For the estimate of σ, the pooled sample variance of the response time observed from the four possible trial types (i.e., cC, cI, iC, and iI) was used.

$$\hat{\sigma} = S_p = \sqrt{\frac{(n_{cC}-1)S_{cC}^2 + (n_{cI}-1)S_{cI}^2 + (n_{iC}-1)S_{iC}^2 + (n_{iI}-1)S_{iI}^2}{n_{cC}+n_{cI}+n_{iC}+n_{iI}-4}} \quad (21)$$

β_0 was estimated using the sample mean $\bar{y}$ and the posterior mean of β_1. We simplified the response time function of M1 in Equation (12) as,

$$RT_t(x_t) = \beta_0 + \beta_1[1 - \mathrm{p}(x_t|X_{t-1})], \tag{22}$$

by excluding the response time for incongruent stimuli. In this equation, the expected value of β_0 given the data is,

$$\mathrm{E}(\beta_0|y) = \bar{y}(congruent) - E(\beta_1[1 - \mathrm{p}(x_t|X_{t-1})]|y). \tag{23}$$

Because $1 - \mathrm{p}(x_t|X_{t-1})$ was either u or 1- u, we assumed that $E(1 - \mathrm{p}(x_t|X_{t-1}))$ is the average of u and 1- u, which is 0.5. Therefore, Equation (23) became

$$\mathrm{E}(\beta_0|y) = \bar{y}(congruent) - (\textstyle\int \beta_1\, p(\beta_1|y)d\beta_1)/2, \tag{24}$$

using the posterior mean of β_1 as the value of $E(\beta_1)$.

Applying the same procedure to M2, we obtained the same equation,

$$\mathrm{E}(\beta_0|y) = \bar{y}(congruent) - (\textstyle\int \beta_1\, p(\beta_1|y)d\beta_1)/2, \tag{25}$$

for M2 as well, assuming that the expected level of conflict is 0.5.

The expected value of β_0 was used as the estimate of β_0,

$$\widehat{\beta_0} = \mathrm{E}(\beta_0|y). \tag{26}$$

4.1.4. Results and discussion

Since the models used for ADO did not consider the accuracy of responses, trials with incorrect responses were excluded from analyses. Further, the data from one participant who showed low accuracy rate (below 80%) was excluded from the study. The average accuracy rate of remaining 19 participants was 96.56%, which means that on average, they made less than two erroneous responses in each block of 40 trials.

The primary goal of the H-ADO used in Experiment 1 was to discriminate the two models based on the response time data. Therefore, the model probabilities that indicate the likelihood of each model were of our main interest. We first used 0.95 model probability as an indicator of model discrimination, because that was the threshold value required for the H-ADO to switch from the M-ADO to the parameter estimation ADO (P-ADO). There were 9 participants who exceeded 0.95 model probability for M2 in at least one ADO stage, while only 2 participants reached 0.95 threshold for M1 during the entire course of the experiment (i.e., 6 blocks of 160 trials). Those 11 participants switched from M-ADO to P-ADO once the threshold was met. Among the other 8 participants that were not fully discriminated, 6 participants showed 0.8 or higher model probability at least once, three for M2 and the other three for M1. The participants were grouped into three categories for detailed analyses, according to the model probabilities. The 9 participants with $p(M2) \geq 0.95$, and the 5 participants with $p(M1) \geq 0.8$ were categorized as the conflict-driven group (C-group) and the expectancy-based group (E-group), respectively. The remaining 5 participants formed the neutral group (N-group). The model probability criterion was set lower for the expectancy-based group than for the conflict-driven group because only a few participants showed $p(M1)$ being higher than the 0.95 threshold.

The C-group and the E-group were expected to show distinct response time patterns that correspond to the prediction of the models shown in Figure 7. We compared the response time patterns between the two groups to clarify that the model probability reflects participants' behavior as expected. The response time pattern averaged over all the 19 participants is shown in Figure 14. The congruency sequence effect (CSE) was not observed because the previous trial type did not affect the congruency effect. The congruency effect was 83.4 ms after congruent, and 83.7ms after incongruent. However, sequential effects were found when the response time data was averaged separately by the E-group and the C-group (see Figure 15). The results were similar to the model predictions in Figure 7-a, where the expectancy-based model predicted a reversed CSE under 30% proportion of repetition (D1). This result suggests that the model probability in ADO successfully reflected the response time pattern expected by the models. The similarity between the observed data and the model prediction under 30% D1 is reasonable because the ADO selected 30% D1 in most blocks for the E-group (see Table 7) in this experiment. The two groups showed the patterns that cancel out each other, resulting in the averaged pattern in Figure 14 without a sequential effect.

The differences between the E and C groups were also supported by a frequentist analysis. A three way ANOVA analysis with the factors of current congruency (congruent vs incongruent), previous congruency (congruent vs incongruent), and group (expectancy based, conflict based, neutral) showed the main effect of current congruency ($F(1, 16) = 150.437$, $p < 0.001$), and the three-way interaction of current congruency × previous congruency * group ($F(2, 16) = 4.843$, $p = 0.0227$). The F statistic of the interaction between current congruency and previous congruency was $F(1, 16) = 0.273$, $p = 0.6083$. This two-way interaction that indicates a sequential effect (e.g., the CSE) was not significant, but the three-way interaction suggested that there are sequential effects modulated by the group factor. The results from ANOVA is consistent with the interpretation of the response time patterns based on the model discrimination results, suggesting that the CSE differed among the groups discriminated by the model probabilities.

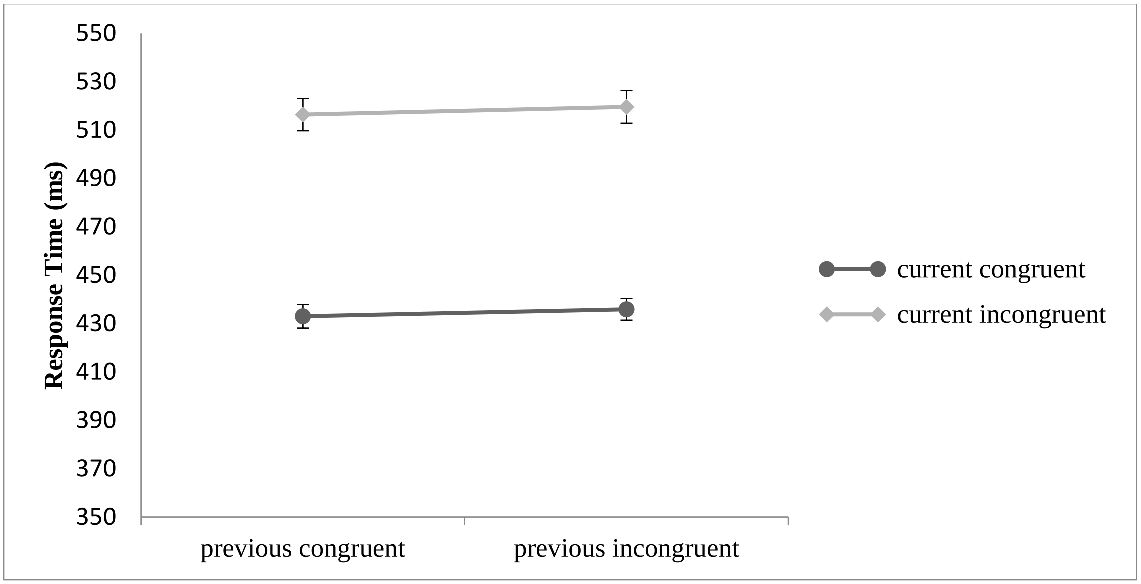


Figure 14. Response time pattern averaged over all participants in Experiment 1.

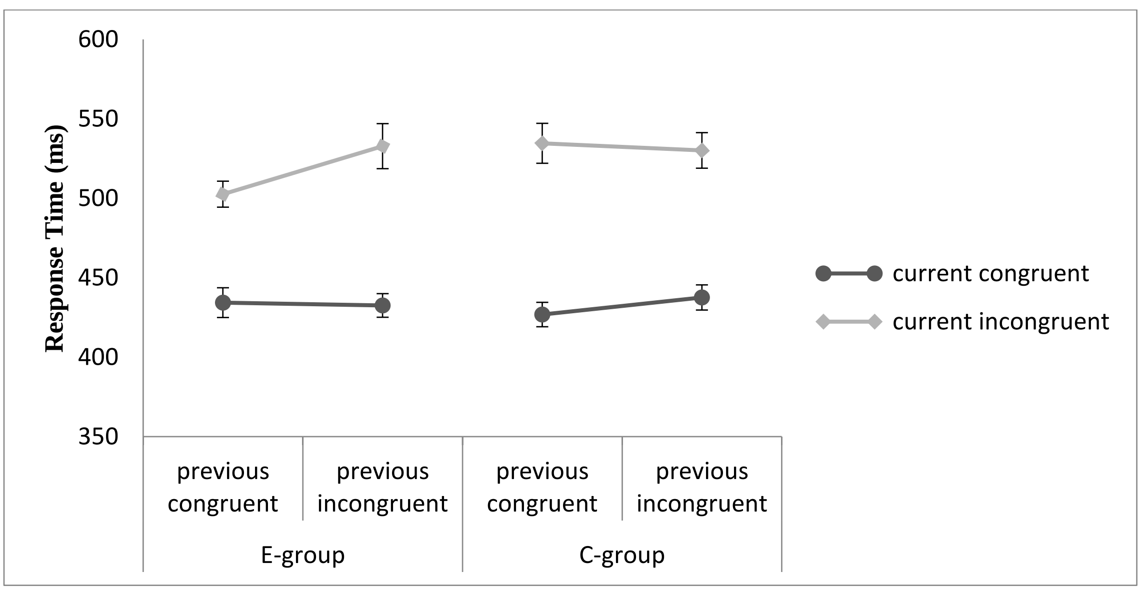


Figure 15. Comparison of the response time patterns from the expectancy-based group (E-group) and the conflict-driven group (C-group) in Experiment 1.

Table 7. Number of times each design was selected by the expectancy-based group (E-group) in Experiment 1.

Design	Block 1	2	3	4	5	6
30-30-50	5	4	4	4	4	4
30-70-30	0	1	1	1	1	1
70-30-70	0	0	0	0	0	0
70-70-50	0	0	0	0	0	0
Mean p(M1)	0.5	0.66	0.71	0.64	0.65	0.52

Although there was a clear difference in response time between the two groups, the model probability of the E-group was not highly discriminable on average. As shown in Figure 16, the model probability tended to increase in earlier blocks, but decrease in later blocks. This pattern implied that the participants who used expectancy-based control gradually switched to the conflict-driven control. Such transition was more significant in the model probability of individual participants, than in the averaged data in Figure 16. For example, participant 3 in Figure 17 exceeded 0.95 model probability by the third block, but the probability had constantly decreased from then. The response time pattern in Figure 18 was also changed accordingly, showing the transition from a reversed to a normal CSE. The other participants in the E-group showed similar switching patterns, except for participant 16 whose model probability constantly increased.

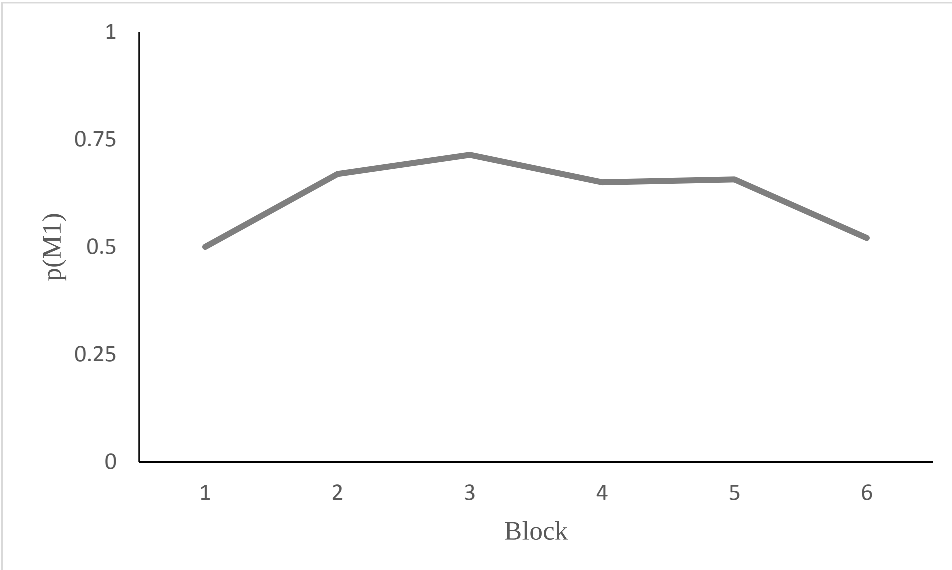


Figure 16. Model probability of the expectancy-based group (E-group) in Experiment 1.

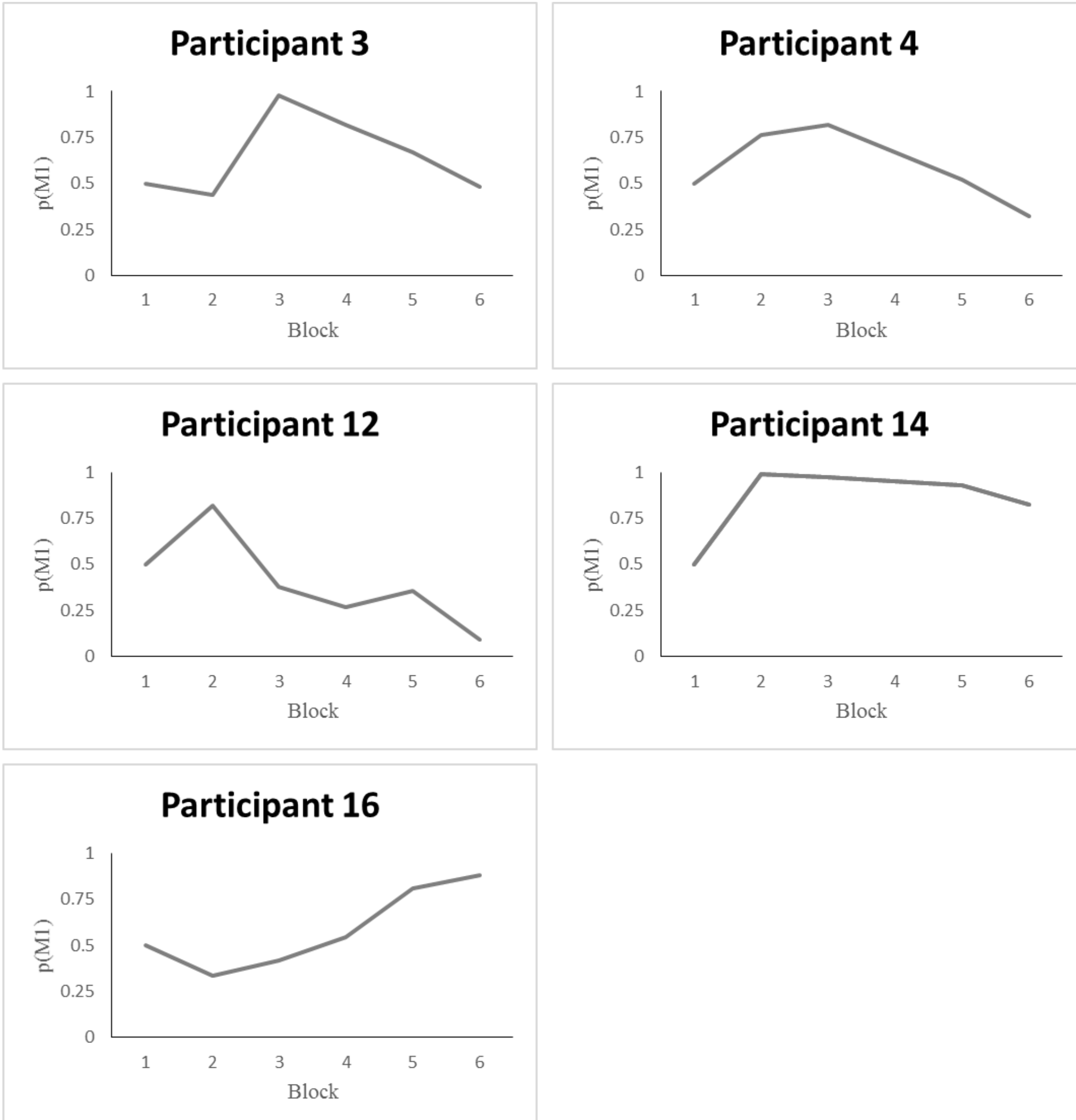


Figure 17. Model probabilities of the individual participants in the expectancy-based group (E-group) in Experiment 1.

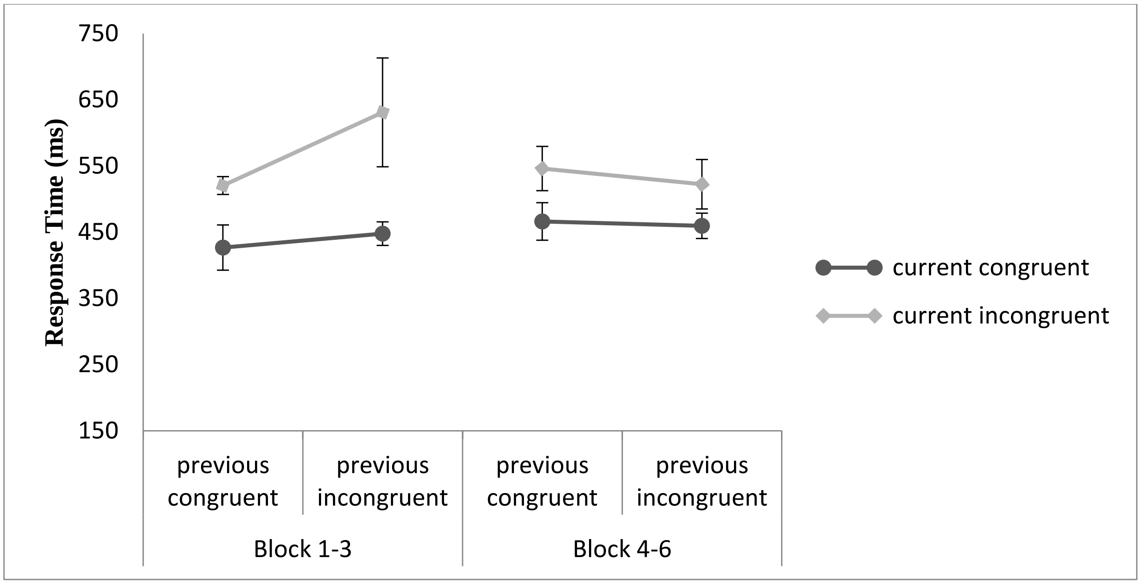


Figure 18. A transition pattern in the response time of Participant 3.

The participants in the C-group were more easily discriminated, as shown in Figure 19. The probability of the conflict-driven model, $\mathrm{p(M2)}$, that is equal to 1- $\mathrm{p(M1)}$, constantly increased throughout the experiments. As the models were discriminated, the hybrid ADO mostly selected 70% D2 for parameter estimation (see Table 8). This was not consistent with the design selections in Simulation 2, where the ADO chose 30% D2 in most of the iterations. This inconsistency was not surprising because the design selections are dependent on the response time from the participants, which is distinct from the data generated by a model. The selection of the design with more congruent trials indicates that the model required more information from congruent trials than from incongruent trials given the posterior distributions updated by the response time data.

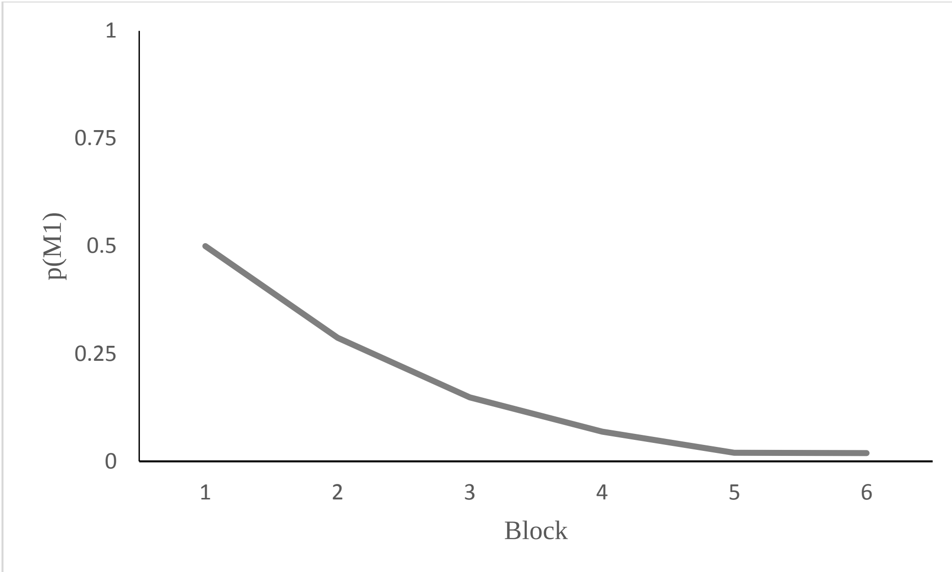


Figure 19. Model probability of the conflict-driven group (C-group) in Experiment 1.

Table 8. Number of times each design was selected by the conflict-driven group (C-group) in Experiment 1.

	Blocks					
Designs	1	2	3	4	5	6
30-30-50	9	7	6	2	0	1
30-70-30	0	0	0	1	0	0
70-30-70	0	2	3	6	9	8
70-70-50	0	0	0	0	0	0
Mean p(M2)	0.5	0.71	0.85	0.93	0.98	0.98

In summary, the results from Experiment 1 show that the models of cognitive control can be discriminated based on the response time data in the flanker task. The model probabilities successfully reflected the response time patterns that are predicted by the models. Also importantly, we observed significant individual differences in terms of model probability, implying that each participant has different preference on the types of cognitive control. The H-ADO would be efficient in this case, where a single model is not generalizable to every participant.

4.2. Experiment 2

Recall that Experiment 1 showed that the models are discriminable based on the response time data. However, the H-ADO in Experiment 1 did not choose the 70-70-50 design for the E-group, unlike in Simulation 1 where the expectancy-based model (M1) generated the data. Therefore, the response time patterns under low and high proportion of repetition (D1) were not directly compared with each other. A low D1 would be sufficient for model discrimination because the predictions of the two models differ significantly when D1 is low (see Figure 7). However, without comparing the response time patterns with low and high D1 values, it would not be possible to see if the congruency sequence effect (CSE) of each subject is actually influenced by D1. For example, a participant who shows a reversed CSE with a low D1 would be classified as using expectancy-based control, but if he/she still shows the same pattern with a high D1, it would not be fully consistent with the prediction of M1. Experiment 2 was conducted to fill this gap by explicitly manipulating D1. The designs in Experiment 2 were the same as those in non-adaptive, fixed 1 (F1) method in the simulations, except that the order of the blocks was counterbalanced across participants. We used 30% D1 for 3 blocks, and 70% D1 for the other three, to investigate the effect of D1 on the response time patterns. According to the model predictions in Figure 7, we hypothesized that the participants with a high probability of M1 would show the congruency sequence effect (CSE) only when D1 is high. The participants who are inclined to the conflict-driven model (M2) were expected to be insensitive to D1. We aimed to observe these differences in response time patterns and then classify the participants according to the model fit.

4.2.1. Participants

12 undergraduate students at the Ohio State University participated in the experiment (18 - 22 years old, 5 males and 7 females). All subjects had normal or corrected-to-normal vision.

4.2.2. Stimuli and procedure

Experimental settings were identical to those in Experiment 1, except that the design variables to be used in the experimental blocks were predetermined. For a half of the participants, the combinations of the design variables were 30-30-50 for the first three blocks, and 70-70-50 for the latter three. To counterbalance the order of the combinations, the other half of the participants conducted the task in the reversed order (i.e., 70-70-50 before 30-30-50).

4.2.3. Results and discussion

The accuracy rate was 97.83% on average, without any participant excluded from the data. The trials with a wrong response were excluded from the analysis as like in Experiment 1, because the models we used do not consider the accuracy of the response.

The purpose of Experiment 2 was to observe the effect of D1 on the response time patterns that were predicted differently by the two models (see Figure 7). We assumed that the model probability would capture the response time patterns that fit well with the models, because the model probability successfully reflected participants' behavior in Experiment 1. For the comparison of the response time patterns based on the model probability, the participants were grouped using the same criteria as in Experiment 1. There were 3 participants in the expectancy-based group (E-group), and 3 participants in the conflict-driven group (C-group). Two participants in the expectancy-based group showed higher than .95 model probability in an experimental block. The remaining 6 participants were categorized as the neutral group (N-group).

In spite of the fact that only half of the participants were included in either the E-group or the C-group, those participants showed distinguishing model probabilities. Figure 20 describes the average model probability of the E-group after each block of trials. There is an increasing trend without a clear transition pattern as in Experiment 1. Participant 5 was the only one who showed a decreasing model probability at the later blocks, whereas the other two participants each reached the highest probability at the end of the experiment. The C-group was well-discriminated as well, showing the mean model probability increased to .98 for M2 by the end of the experiments (see Figure 22).

Given the high model probabilities, it was likely that the two groups would show distinct response time patterns that fit well with the model predictions.

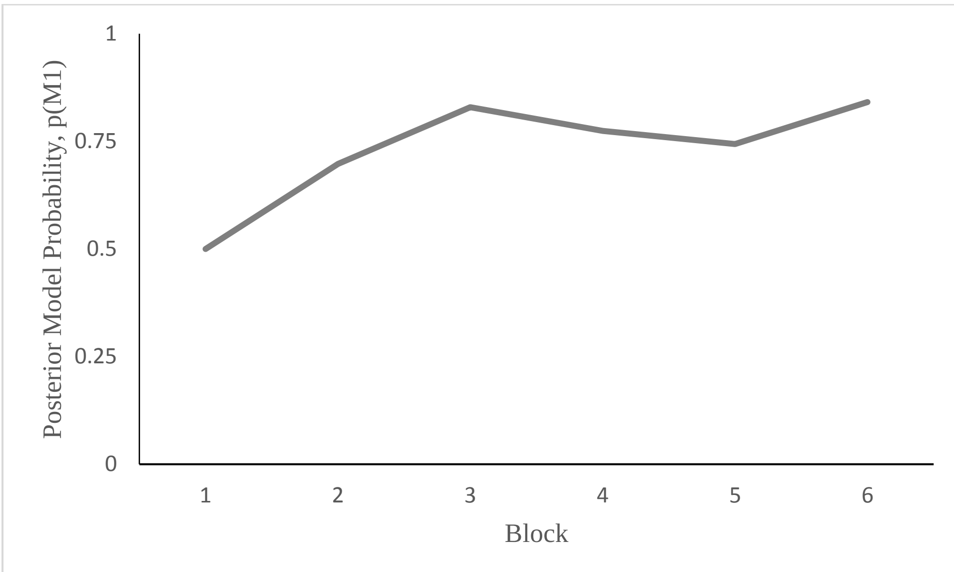


Figure 20. Model probability of the expectancy-based group (E-group) in Experiment 2.

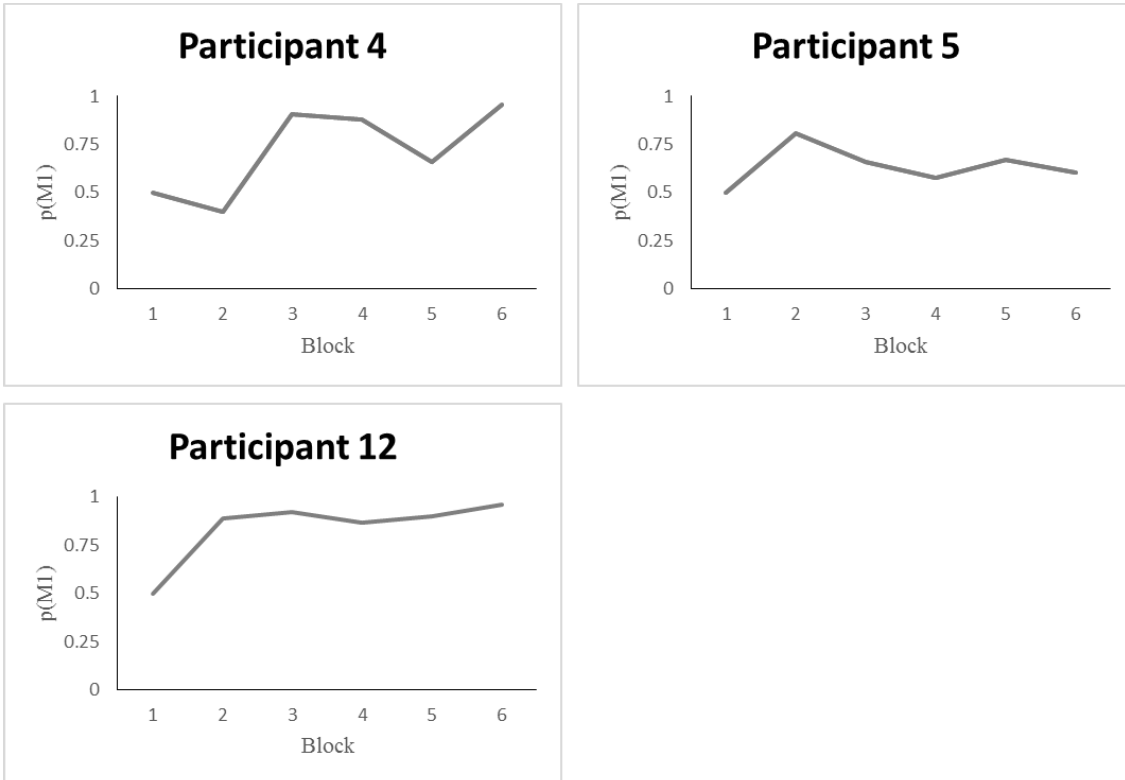


Figure 21. Model probabilities of the individual participants in the expectancy-based group (E-group) in Experiment 2.

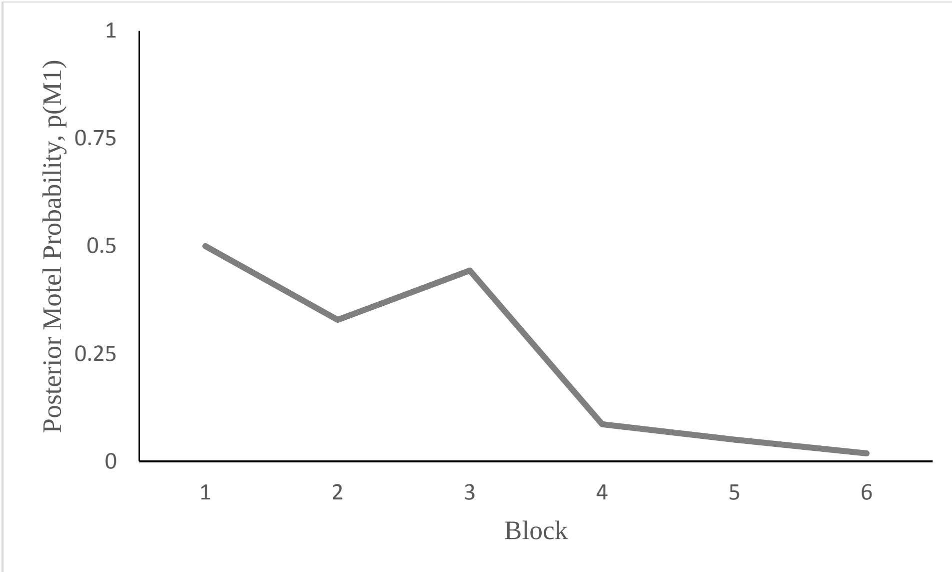


Figure 22. Model probability of the conflict-driven group (C-group) in Experiment 2.

Prior to a group comparison, we examined the sequential effects in the response time data averaged over all the participants. Unlike in Experiment 1 (see Figure 14), the congruency sequence effect (CSE) was observed from the averaged data, as shown in Figure 23. However, the comparison of the response time patterns under different D1 suggested that the effect is dependent on D1. Figure 24 shows that the CSE is relatively strong with 70% D1, but is weaker with 30% D1. A stronger CSE under 70% D1 is consistent with the model predictions because both M1 and M2 predict the CSE under a high D1. The behavioral data under 30% D1 was similar to the averaged data in Experiment 1 that showed no interaction between current and previous congruencies (see Figure 14). It is reasonable because the H-ADO selected 30% D1 in the majority of blocks in Experiment 1, mostly for the model discrimination. The absence of a sequential effect in Experiment 1 was due to the offsetting effects found from the two participant groups discriminated by the model probability. To see if the similar individual differences would be found in Experiment 2, we again compared the response time patterns of the E-group and the C-group.

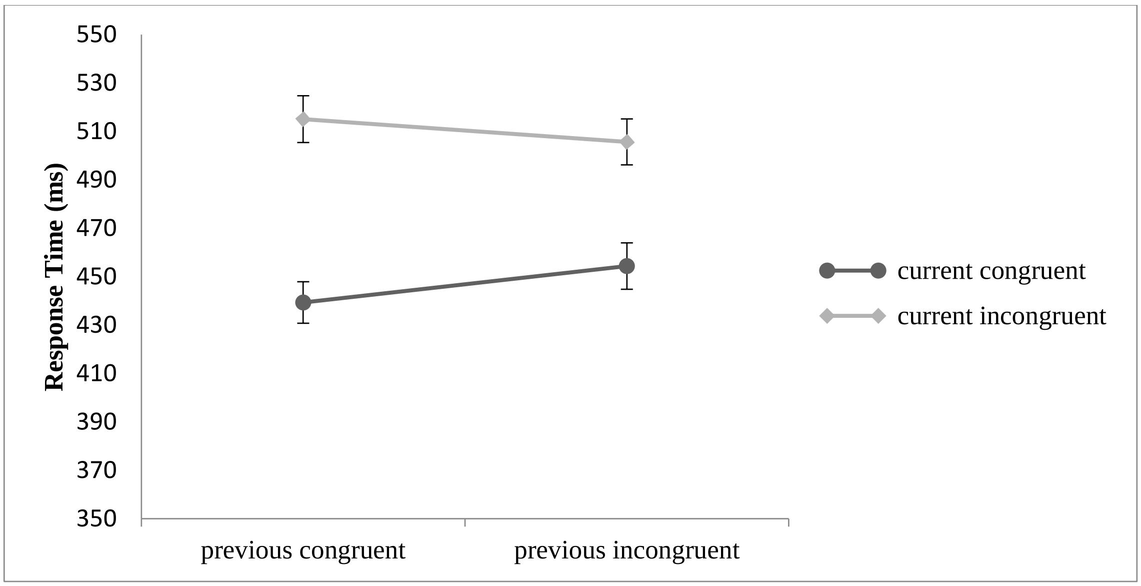


Figure 23. Averaged response time data in Experiment 2.

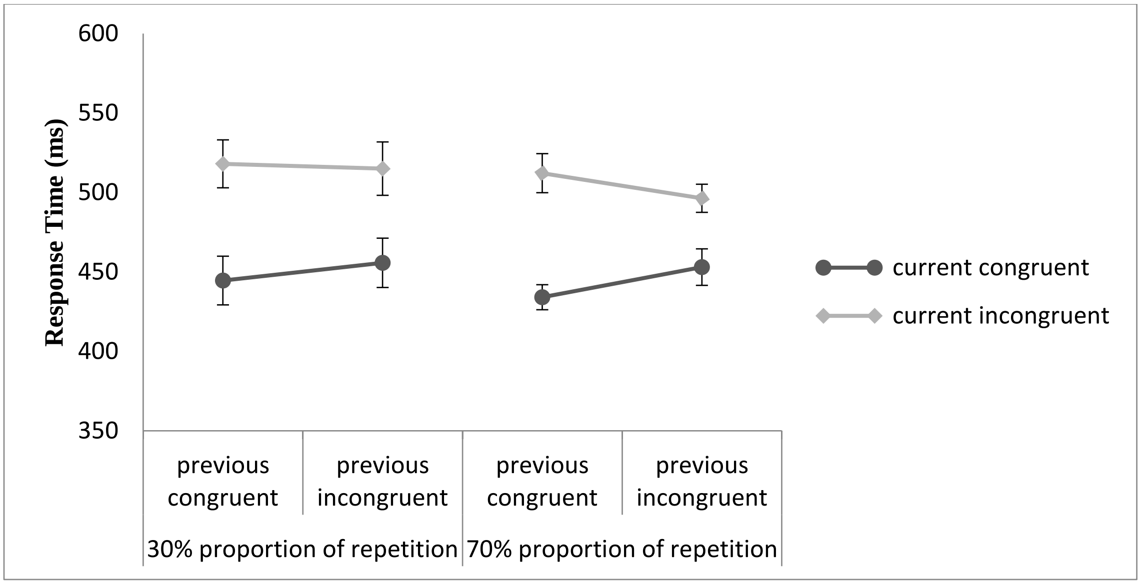


Figure 24. Comparison of the response time data under different proportions of repetition (D1) in Experiment 2.

The behavioral data of the E-group described in Figure 25 was consistent with the prediction of M1, which expects the CSE when D1 is high, and a reversed CSE when D1 is low (see Figure 7). There appeared to be the CSE under 70% D1, but this effect tended to be slightly reversed under 30% D1. However, the results were inconclusive because the variance of the data was significantly high. The response time patterns of the C-group shown in Figure 26 was more stable, but the patterns themselves were not as expected. The behavior of the C-group seemed to be modulated by D1, although M2 assumes that response time is not directly dependent on D1. The data seemed to show the CSE with 30% D1, but not with 70% D1. This result did not fully support M2, but was opposite to M1's prediction that the CSE is stronger with higher D1. The model probabilities of the C-group were high for M2 (see Figure 22), but it was probably because of very low likelihood of M1, rather than a good fit of M2.

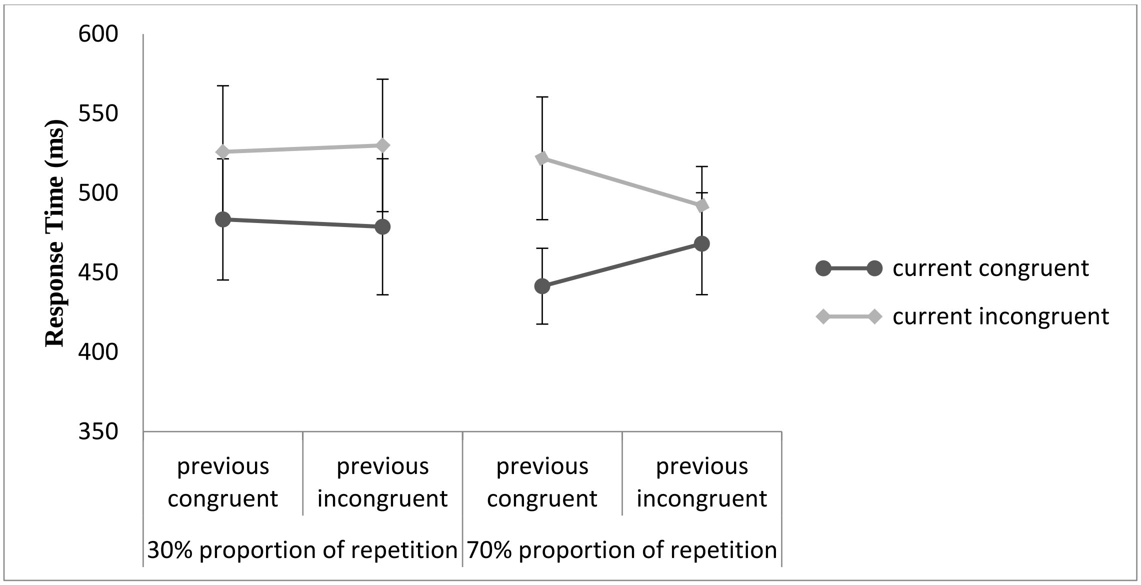


Figure 25. Comparison of the response time data of the expectancy-based group (E-group) in Experiment 2 under different proportions of repetition (D1).

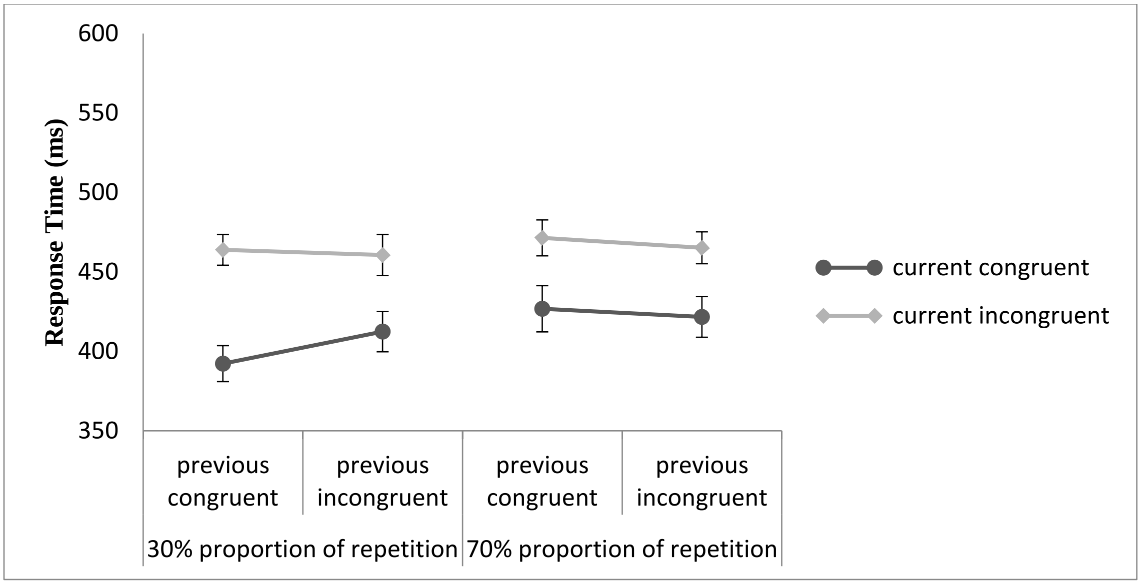


Figure 26. Comparison of the response time data of the conflict-driven group (C-group) in Experiment 2 under different proportions of repetition (D1).

For an additional statistical indicator of the data patterns observed, we conducted a four-way ANOVA analysis with the factors of current congruency (congruent vs incongruent), previous congruency (congruent vs incongruent), group (expectancy based, conflict based, neutral), and design (30% vs 70% D1). The interaction effect between current congruency and previous congruency was significant, $F(1, 9) = 8.034$, $p = 0.0196$, supporting the CSE shown in Figure 23. The three-way interaction of current congruency × previous congruency × design showed lower F-value, $F(1, 9) = 1.631$, $p = 0.234$, indicating that the sequential effects were not significantly modulated by the design, as shown in Figure 24. The F-statistic for the four-way interaction of current congruency × previous congruency × group × design was $F(2, 9) = 1.839$, $p = 0.214$. A direct comparison between the expectancy-based group and the conflict-based group was not reliable because of the low sample size, but the F-value for the four-way interaction was higher, $F(1, 4) = 4.822$, $p = 0.0931$, if we exclude the neutral group from the analysis. This result is in line with our finding that the two groups discriminated by the models show different dependencies between the design and the CSE (see Figure 25 & 26).

To summarize, the results from the group comparison analysis in this experiment were inconclusive due to the small sample size and the high variance of the observed data, but the response time patterns of each group seemed similar to the predictions of a corresponding model. The E-group and C-group showed different sequential effects especially with 30% D1, as predicted by the models (see Figure 7). This result provides further support for the hypothesis that there are much individual differences in cognitive control mechanisms that are represented by the two models.

4.3. Experiment 3

Experiment 1 and 2 showed the differences between the E-group and the C-group, but those differences were mostly derived from the manipulation of the proportion of repetition (D1). Therefore, we could not collect enough data to reproduce the LWPC effect (see Figure 2). This effect in which the congruency effect is larger with higher proportion congruency (D2) is predicted by both M1 and M2. Given this, the discrimination between the models is not likely to be clear based on the manipulation of D2. However, it would still be worthwhile to show that the models actually capture the effect of D2 from the data. It was especially important for M1 because the LWPC effect was not usually explained by an expectancy-based account. Thus, we conducted Experiment 3 to reproduce the LWPC effect, using non-adaptive designs comparable to those in fixed 2 (F2) method in the simulations. We expected that both E-group and C-group would produce the LWPC effect because both M1 and M2 predict it. However, there are some differences between the predictions because M1 explains the LWPC effect as an indirect effect of D1 that is constrained by D2. It was possible that the model probability captures this subtle difference in the predicted response time patterns. The result of model discrimination in this experiment was likely to indicate whether D2 directly or indirectly affects the congruency effect.

4.3.1. Participants

Ten undergraduate students at the Ohio State University participated in the experiment (18 - 22 years old, 4 males and 6 females). All subjects had normal or corrected-to-normal vision.

4.3.2. Stimuli and procedure

The designs were predetermined to manipulate the proportion congruency (D2). For a half of the participants, the combinations of the design variables were 30-70-30 for the first three blocks, and 70-30-70 for the latter three. To counterbalance the order of the combinations, the other half of the participants conducted the task in the reversed order (i.e., 70-30-70 before 30-70-30). Experimental stimuli for the flanker task were identical to those in Experiment 1 & 2.

4.3.3. Results and discussion

One participant whose accuracy rate was lower than 80% was excluded from the study. The average accuracy rate of the other participants was 98.48%. The trials with inaccurate responses were excluded from the analysis.

We hypothesized that the models would not be clearly discriminated in Experiment 3 because the two models both expected the LWPC effect we aimed to reproduce. This hypothesis was tested by investigating the model probabilities of the participants. Three participants satisfied the 0.80 model probability threshold for the E-group, and four participants satisfied the 0.95 threshold for the C-group, in at least one block during the experiment. However, there was an overlap between the two groups because there were two participants included in both groups. Those two participants showed strong transition of model probability, from the expectancy-based model to the conflict-driven model. Therefore, there were only 3 participants who were exclusively discriminated as either of the two groups. The group discrimination was not as clear as in Experiment 1 and 2, presumably because the 30-30-50 combination of the designs that is optimal for the model discrimination was not used.

The transition pattern found from the participants is clearly shown in Figure 27 that describes the model probability of the E-group. The model probability for M1 tends to increase until the third block, but rapidly decrease at the later blocks. This pattern was more obvious than the similar pattern observed from the E-group in Experiment 1, as the mean model probability for M1 reached 0.056 at the end in Experiment 3. This result suggests that M2 was better than M1 when accounting for the effect of D2, considering that D2 was manipulated after the third block. It actually seemed somewhat misleading to say that the E-group in Experiment 3 supports M1. In contrast, the model probability of the C-group shown in Figure 28 showed the pattern of constant decrease of the probability of M1.

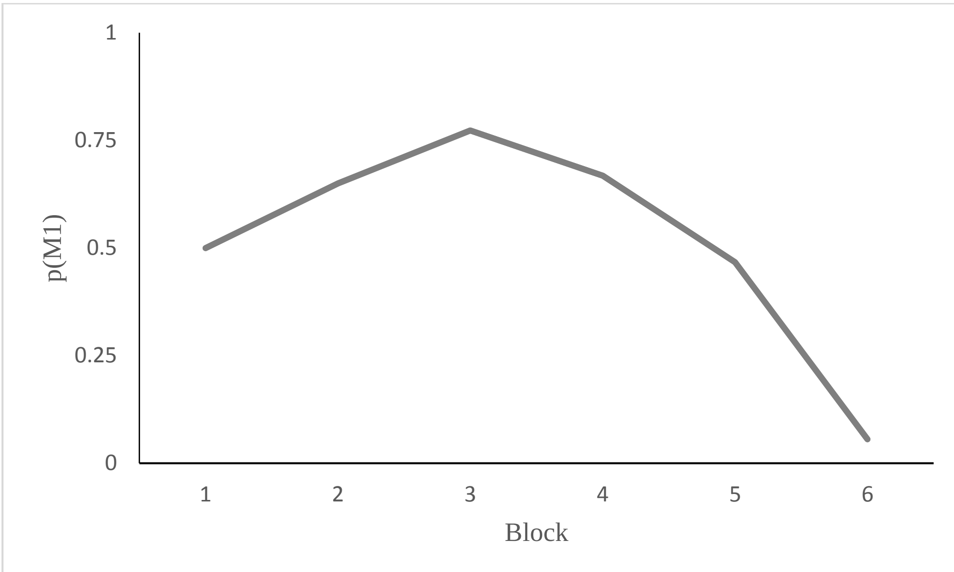


Figure 27. Model probability of the expectancy-based group (E-group) in Experiment 3.

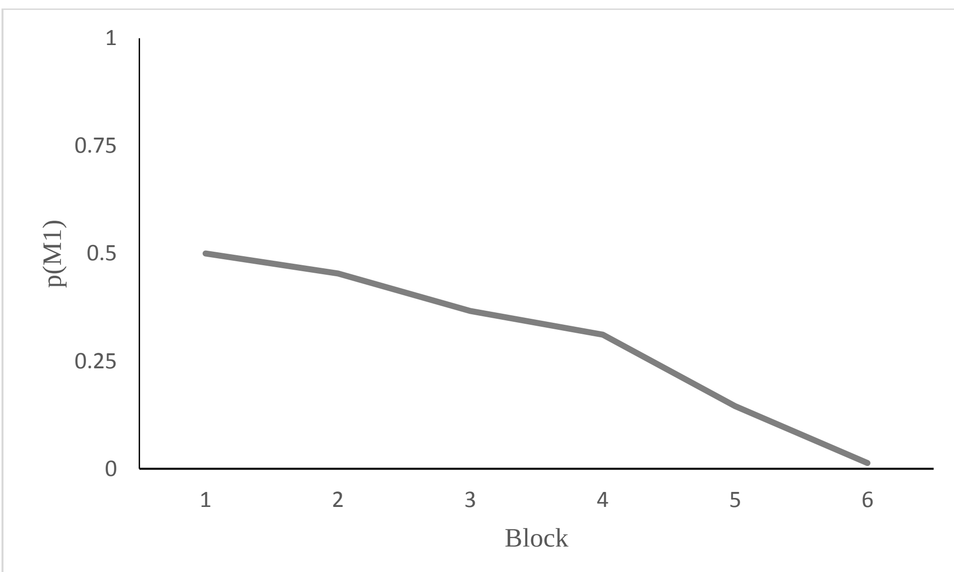


Figure 28. Model probability of the conflict-driven group (C-group) in Experiment 3.

The participants were grouped based on the model probabilities, but we believed that the LWPC effect would be observed regardless of the group. Figure 29 shows the effect of D2 on the congruency effect observed from all nine participants in Experiment 3. The LWPC effect seemed to be reproduced, as the congruency effect appeared to be larger with 70% D2 than with 30% D2. Figure 30 shows this pattern more in detail, including the influence of the previous congruency. The response time pattern with 70% D2 seemed to show the congruency sequence effect (CSE), but no sequential effect was observed with 30% D2. A group comparison between the E-group and the C-group was required to interpret this result because the two models provide different predictions about sequential effects.

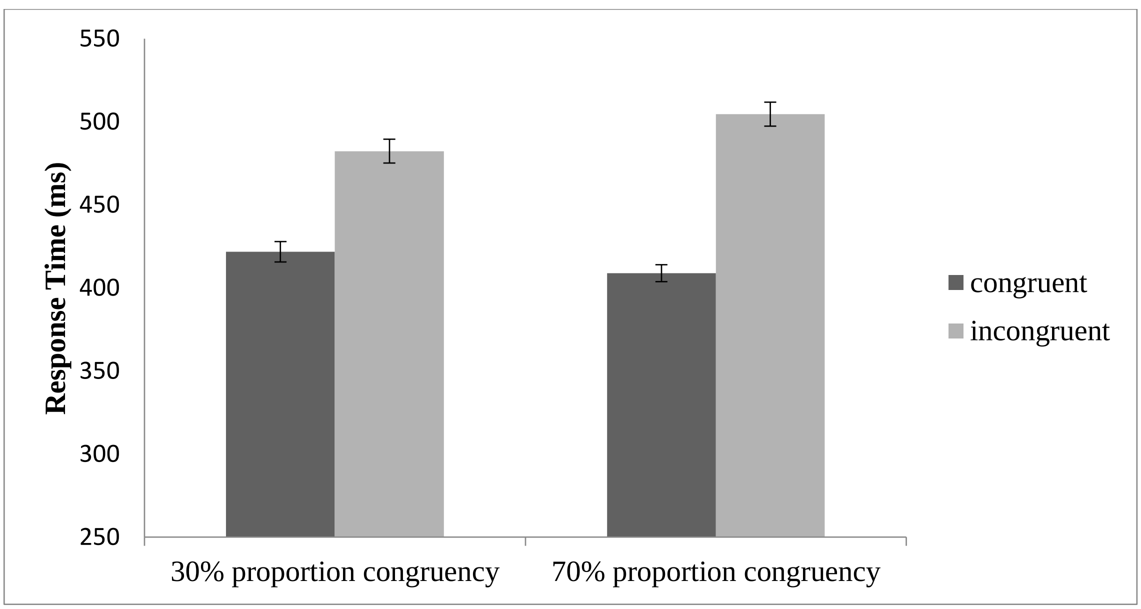


Figure 29. Proportion congruency effect observed from all participants in Experiment 3.

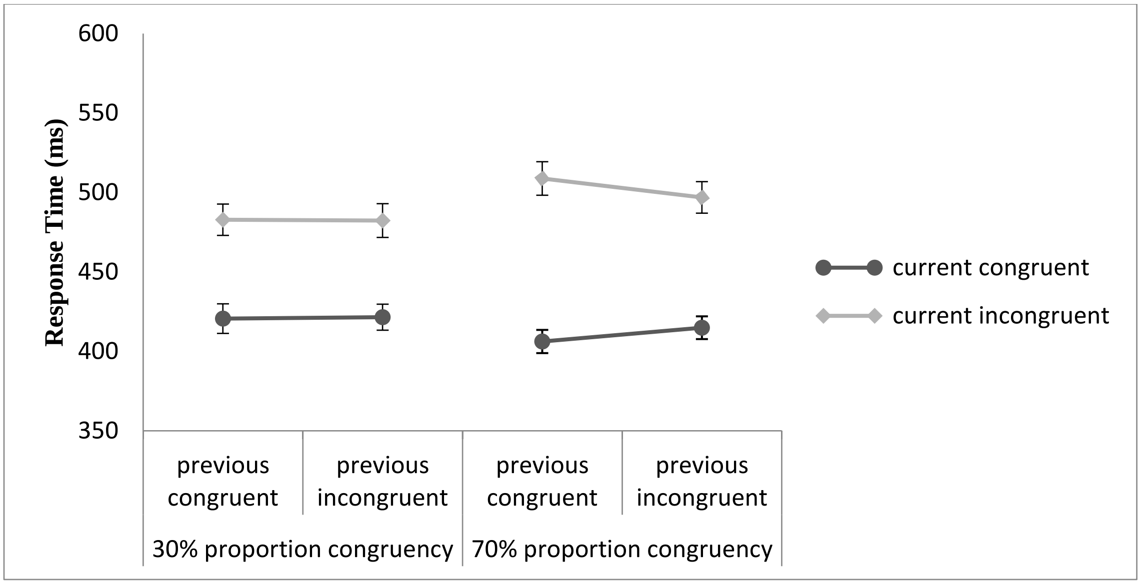


Figure 30. Comparison of the response time data from all participants in Experiment 3 under different proportion congruency (D2).

The predictions of M1 depend on D1, which is 30-70 for 30% D2, and 70-30 for 70% D2. D1 is dependent on D2 in those designs (see also Figure 5), so the sequential effects for each trial type are expected to differ according to D2. However, the data from the E-group described in Figure 31 suggests that only congruent trials are dependent on the designs. The response time patterns for the congruent trials with 30% and 70% D2 were similar to the model prediction of M1 with 30% and 70% D1, respectively (see Figure 7). This was plausible because D1 values for congruent trials were identical to D2 values in Experiment 3. This suggested that repetition expectancy is formulated separately for congruent and incongruent trials as assumed by M1, but it was premature to draw a conclusion because the sample size was small and the participants in the E-group showed a strong transition pattern as shown in Figure 27.

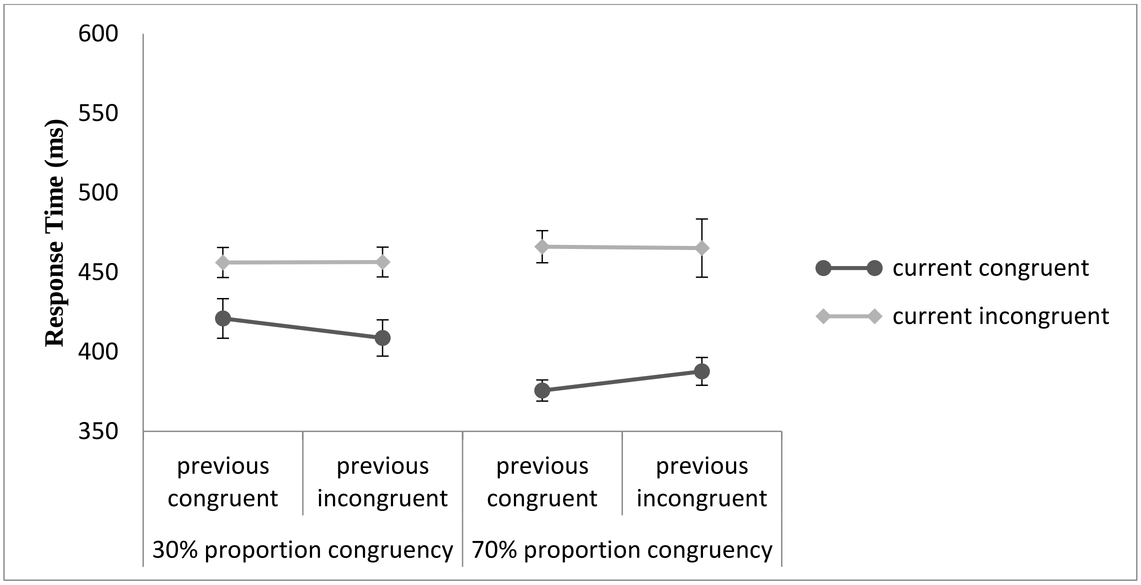


Figure 31. Comparison of the response time data of the expectancy-based group (E-group) in Experiment 3 under different proportion congruency (D2).

The observations from the E-group were not consistent with the prediction of M2 that the CSE occurs with any D2. The C-group was likely to produce such pattern, but the response time data shown in Figure 32 was somewhat different from the prediction. There appeared to be the CSE with 70% D2, but the data with 30% D2 showed little sequential effect. It was probably because the 2 among the 4 participants in the C-group showed high probability of the expectancy-based model at earlier blocks. We can see that the response time pattern in Figure 32 with 30% D2 resembles the pattern of the E-group in Figure 31. It seemed not reliable to investigate the response time patterns specific to a group, due to the vague distinction between the E-group and the C-group.

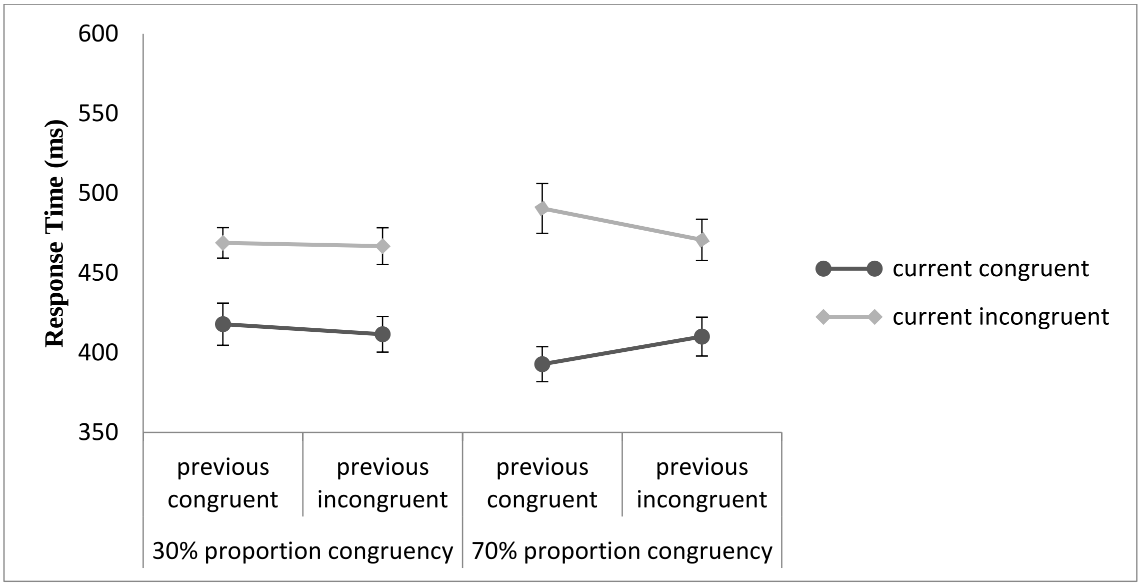


Figure 32. Comparison of the response time data of the conflict-driven group (C-group) in Experiment 3 under different proportion congruency (D2).

Besides the sequential effects, both the groups seemed to show the LWPC effect as expected (see Figure 33 & 34). The results supported our hypothesis that the same effect of D2 on the congruency effect would be observed from any group that supports either of the models. It is unlikely that the models are discriminated based on the LWPC effect, because both models predict larger congruency effect with higher D2. However, high model probabilities for M2 after a manipulation of D2 after the third block (see Figures 27 & 28) suggested that the LWPC effect fits better with M2 than M1.

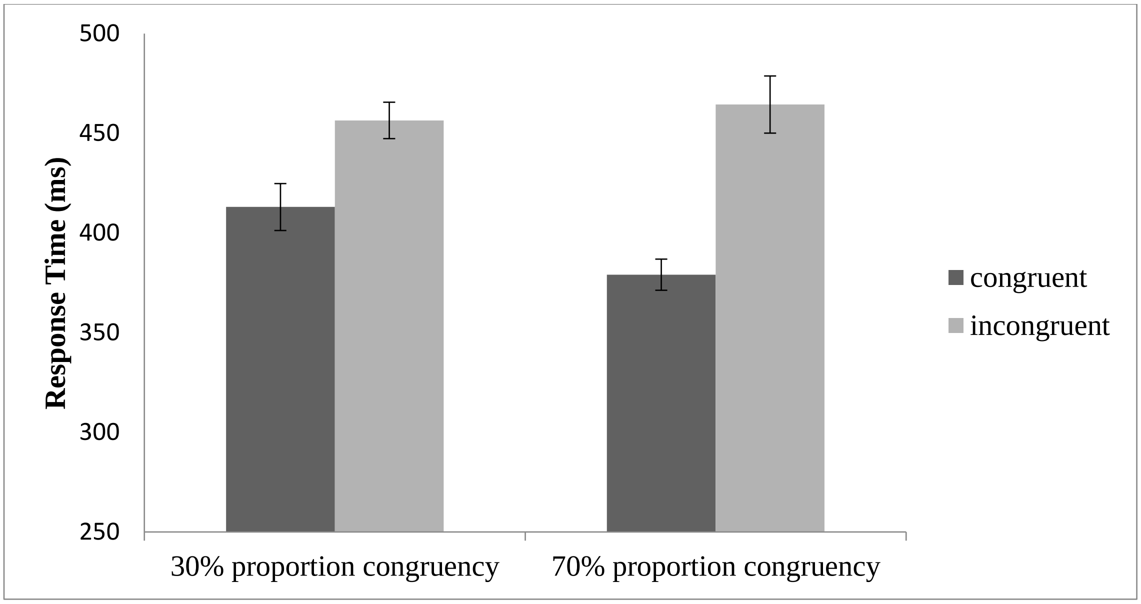


Figure 33. Proportion congruency effect of the expectancy-based group (E-group) in Experiment 3.

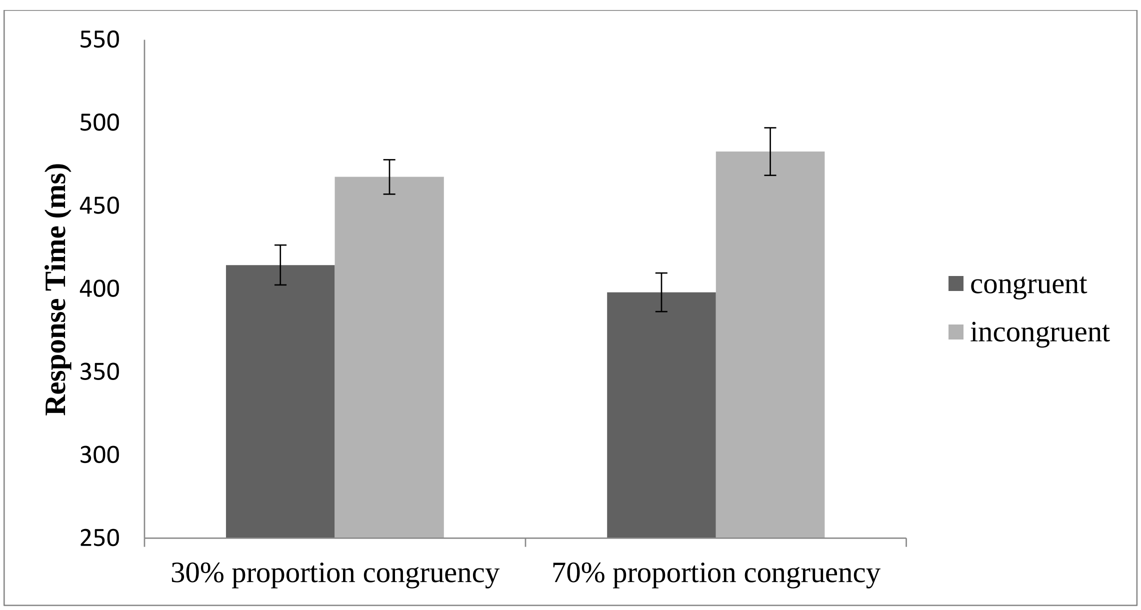


Figure 34. Proportion congruency effect of the conflict-driven group (C-group) in Experiment 3.

We also conducted a three-way ANOVA analysis with the factors of current congruency (congruent vs incongruent), previous congruency (congruent vs incongruent), and design (30% vs 70% D2). The group factor was excluded because of the overlap between the expectancy-based group and the conflict-driven group. The proportion congruency effect was supported by the interaction effect of current congruency × design, $F(1, 8) = 29$, $p < 0.0001$. The F-statistic was $F(1, 8) = 3.602$, $p = 0.0943$ for current congruency × previous congruency interaction, and $F(1, 8) = 2.123$, $p = 0.183$ for the three-way interaction of current congruency × previous congruency × design. The results suggested that D2 affected the congruency effect, but not the interaction between the current and the previous trial types. This is consistent with our finding that the LWPC effect is significant with any group, while the effects of D2 on the sequential effects are unclear.

To summarize, in Experiment 3, by manipulating D2, we reproduced the LWPC effect that is predicted by both models. In addition to the LWPC, the sequential effects in response time seemed to be modulated by D2, although they were only partially interpreted by the models. A notable finding was that the participants in the E-group showed a strong transition pattern toward high model probability for M2 after the manipulation of D2, suggesting that the effect of D2 is better explained by M2 than M1.

5. General Discussion

The main purpose of the present study was to discriminate between the expectancy-based control and the conflict-driven control using the response time data from the flanker task. Those two control mechanisms were usually considered as different ways to explain control related effects such as the congruency sequence effect (CSE) and the list-wide proportion congruency (LWPC) effect. Discriminating those control mechanisms has been challenging because they often lead to similar predictions of behavior, despite theoretical differences in internal processes. Hence we deliberately manipulated experimental designs to investigate the circumstances in which the predictions of the two theories differ from each other. The two competing theories were directly compared using the mathematical models that represent each theory, rather than relying on null hypothesis significance testing (NHST) that test hypotheses in an indirect manner. This model-based approach also allowed us to adaptively optimize experimental designs in order to speed up the evaluation of the models.

To compare the two models with human data, three types of an arrow flanker experiment were conducted using different design selection methods. Experiment 1 used hybrid ADO (H-ADO) that selects an optimal design either for model discrimination or for parameter estimation, mainly to group the participants according to their fit to the two models of cognitive control. Experiment 1 successfully classified the participants, but it was not sufficient to investigate the effect of the design variables to the full extent. For example, the ADO design selection did not yield a within-participant comparison of the response time patterns under 30% and 70% design variables. Experiment 2 was conducted to compare the behaviors under 30% and 70% proportion of repetition (D1) in a balanced design. The participants in the group with a high probability of the expectancy-based model showed the interaction between D1 and the sequential effects in response time, as predicted by the model. Experiment 3 compared the response time patterns in the blocks with 30% and 70% proportion congruency (D2). Both the expectancy-based model (M1) and the conflict-driven model (M2) predict the effect of D2, but the observed list-wide proportion congruency (LWPC) effect was better explained by M2. Taken together, the results from the three experiments suggest that the model-based flanker task experiment is efficient in discriminating between the two theories of cognitive control. To elaborate, the main findings from those experiments can be summarized in terms of the following three phrases: model discrimination, transition pattern, and individual differences.

5.1. Model discrimination

The present study used the two models of which the relative likelihood was reflected by the model probability. A major benefit of this model-based study was that we could directly compare the competing theories represented by the models. The most useful design for model discrimination was the one with the lowest D1. H-ADO in Experiment 1 usually selected this design until the models were fully discriminated, resulting in the highest proportion of the participants classified into the two groups that correspond to the two models. The 74% of the participants were classified as either the expectancy-based group (E-group) or the conflict-driven group (C-group) in Experiment 1, while the proportion was 50% and 56% for Experiment 2 and 3, respectively.

One difficulty in model discrimination was that the model probability of the expectancy-based model rarely exceeded the 0.95 threshold for H-ADO to switch to the parameter estimation. The inclination toward the conflict-driven model was also found in Simulation 1, although the data was generated by the expectancy-based model. A possible solution to this problem is the use of informative priors that are closer to the posterior distributions updated by the participants. Using more realistic priors instead of the uniform priors used in the present study is likely to enhance the model discrimination by reducing the trials required for estimating the models. However, there was another reason for the low model probability for the expectancy-based model, which is a transition pattern frequently observed from the participants in the E-group.

5.2. Transition pattern

Many of the participants who initially exhibited a high model probability for M1, p(M1), showed a rapid increase of p(M2) at the later blocks. This transition pattern was somewhat unexpected, but there is a possible explanation for such observation. The expectancy-based control model assumes that participants maintain and update their repetition expectancy, and expect the upcoming trial to control their attention. The conflict-driven control is relatively effortless because the participants just respond to the stimuli given the level of conflict. The distinction between the two control processes resembles the differences between automatic and strategic control processes. Automatic control is effortless and unintentional (Logan, 1988; Wells & Matthews, 2014), while strategic control is intentional, effortful, and fully conscious (McNally, 1995; Sternberg, 1996). According to this description, the conflict-driven control is relatively automatic, and the expectancy-based control is relatively strategic. A notable aspect of this distinction is that the control for a task tends to move from the strategic to the automatic processes through practice (Beck & Clark, 1997; Sternberg & Sternberg, 2016). In short, the participants who showed the transition pattern may have learned to use an automatic, conflict-driven control as they became accustomed to the task.

5.3. Individual differences

The results of model discrimination showed that there are individual differences in the preference for the cognitive control strategies, at least at the earlier blocks where the task is new to the participants. The participants in E-group and C-group were assumed to prefer the expectancy-based control and the conflict-driven control, respectively. This distinction was also shown in the response time patterns that reflect the prediction of the models. The two patterns of our interest were the congruency sequence effect (CSE) and the list-wide proportion congruency (LWPC) effect.

There appeared to be a significant difference in the CSE between the two groups, mainly with a low proportion of repetition (D1). In Experiment 1 where 30-30-50 design was frequently used, the E-group showed a reversed CSE, while the C-group showed the CSE. The average response time data showed no CSE (see Figure 14), because the two groups cancelled out each other's sequential effects. This result provides another possible interpretation of the data that do not show the CSE. For example, Duthoo et al. (2013) observed no CSE when participants expected alternations, but it is possible that the participants who exploit expectancy-based control cancelled out the sequential effect of the other conflict-driven participants. A model comparison is advantageous to acknowledge this kind of variance in the data because an ANOVA analysis does not usually capture the individual differences that generate different effects unless the participants were arbitrarily grouped in advance. The studies of sequential effects in congruency tasks usually report the response time averaged over every participant (e.g., Duthoo et al., 2013; Jiménez and Méndez, 2013; Kim, Lee, & Cho, 2015), but it might be advantageous to focus more on individual differences.

The LWPC effect, on the other hand, did not show much difference between the two groups. More importantly, the participants in the E-group showed strong transition patterns toward M2, suggesting that M2 is better than M1 at explaining the effect of D2. This suggests that people prefer to use conflict-driven control when they experience a significant change in the D2 that is correlated with the average level of conflict. However, this result should be considered inconclusive at this point, primarily because of the small sample size, as discussed in a later section.

5.4. Implications

The findings in the present study are inconclusive, but they still may have important implications for the nature of cognitive control. Among the various theories of cognitive control mechanisms, the conflict monitoring theory (Botvinick et al., 2001) has been supported by numerous empirical studies. However, most of those studies relied on ANOVA as an evidence to support their hypotheses, comparing the aggregated response time data from different conditions (Akçay & Hazeltine, 2007; Hazeltine, Akçay, &

Mordkoff, 2011; Kerns et al., 2004; Notebaert, Gevers, Verbruggen, & Liefooghe, 2006; Ullsperger, Bylsma, & Botvinick, 2005).

Null hypothesis significance testing (NHST) such as ANOVA is an indirect way to compare competing theories or models, because it merely tests whether the data shows significant differences from the null hypothesis that usually assumes no difference among different conditions. For the purpose of model discrimination, evaluating "how" the data differ from the null hypothesis is more important than knowing whether there is a difference or not. The model-based approach in the present study, on the other hand, directly compares the two theories using the quantitative models that predict specific response time patterns assumed by the theories. Instead of rejecting or failing to reject a null hypothesis, this approach tests which model fits better with observed data by calculating Bayes factors. Although there are various models of cognitive control mechanisms (e.g., Botvinick et al, 2001; Blais et al., 2007; Verguts & Notebaert, 2008; Yu & Cohen, 2009), those models were not frequently compared in experimental studies.

The results in the present study were in some sense consistent with the previous studies that support the conflict monitoring theory, as the majority of participants tended to show higher model probability for the conflict-driven model (M2) as the experiment proceeds. However, there still were some participants who seemed to exploit expectancy-based control. They showed a reversed congruency sequence effect (CSE) when the proportion of repetition (D1) is low, as expected by the expectancy-based model (M1). This reversal of the CSE would not be seen in the average response time pattern because of the larger number of participants who showed normal CSE. In the ANOVA, the participants with the reversed CSE might be considered as meaningless outliers as long as the interaction between the current and the previous congruency is significant in the aggregated data (see the ANOVA results in Experiment 2). It implies that the experimental studies using NHST may ignore meaningful individual differences that reflect distinct cognitive control mechanisms.

5.5. Limitations

There are two major limitations of the present study that make some results inconclusive at this time. One is the small sample size, especially for Experiment 2 and Experiment 3. There were only five or six participants that were classified into either E-group or C-group in those two experiments. Each group had only three or four participants, so the comparison between the groups was problematic. It was mainly because we could not recruit more participants in time. In addition, the proportion of participants who showed a high model probability was smaller in Experiment 2 and 3 than in Experiment 1. The lack of data could simply be resolved by collecting more participant data.

Another limitation is the limited grid size for the ADO calculation. The size and the range of the response time grid and the parameter grid had to be small enough so that all the calculations could be completed within the limited time between the trial blocks. We used dynamic grid for the response time, and the point estimates for some parameters to improve the grid resolution for the other parameters as much as possible, but the shape of the posterior parameter distributions was still not smooth (see Figures 10 and 13). Therefore, the accuracy of parameter estimates was constrained by the grid resolution. This is a non-trivial problem because the model probability and the utility of the designs are also influenced by the posterior parameter distribution. However, given limited computational capacity, there would not be a straightforward solution to this problem. This will need to be considered in future studies that implement the ADO.

Besides the technical limitations, there is an inherent limitation of model-based studies, which is the plausibility of the models. We assumed that either of the models would reflect the cognitive control mechanism of the participants, but it is possible that there is another model that better captures their behaviors. This suggests that the neutral groups that were not discriminated by the two models in the present experiments might have been explained better by a third model. This problem would be found in any model-based study because it would not be possible to have a model that perfectly reflects human cognition. A realistic solution would be to thoroughly test the models to see if they fit well enough to the data. If there is a model that possibly account for the data better than the models we used, it would be worthwhile to evaluate the additional model in future studies.

5.6. Future studies

We compared only two models of cognitive control in the present study, but there are other theories that provide alternative interpretations of the CSE or the LWPC effect (e.g., Hommel, Proctor, & Vu, 2004; Mayr, Awh, & Laurey, 2003; Schmidt, 2013). The models based on those theories could be evaluated in future studies using the ADO. For example, a repetition priming account provides an alternative explanation for the congruency sequence effect, based on item-specific priming (Mayr et al., 2003). This account assumes that a response is enhanced when the stimulus features are completely repeated, resulting in faster responses when a trial type (i.e., congruent or incongruent) is repeated than when it is alternated. This resembles the expectancy-based control account with repetition expectancy in some sense. The distinction between the effects of priming and expectancy has not been clear in theory, but a model comparison between a priming-based model and the expectancy-base model would help distinguish between those theories.

In addition, there are models that predict a new response time pattern, the item-specific proportion congruency (ISPC) effect, by considering item-specific control (Blais et al., 2007; Verguts & Notebaert, 2008). The ISPC effect is similar to the LWPC effect, except that it occurs at an item level (Jacoby, Lindsay, & Hessels, 2003; Trainham, Lindsay, & Jacoby, 1997). That is, if the items in a congruency task (e.g., central arrows > and < in the flanker task) have separate proportions of congruent trials, the congruency effect for each specific item is modulated by its own proportion congruency. The models in the present study do not reproduce the ISPC effect because they assume that the control is imposed on a task as a whole, rather than on individual items. Future studies might investigate the distinction between the list-wide control and the item-specific control, by a model comparison that includes the models of item-specific control.

6. Conclusion

In the present study, the two models of cognitive control were compared using the arrow flanker task. The most important finding was that none of the models turned out to be strongly favored over the other in accounting for the data patterns observed in the present experiments. The expectancy-based model (M1) had advantage in explaining the reversed congruency sequence effect (CSE), whereas the conflict-driven model (M2) performed better when there was the list-wide proportion congruency (LWPC) effect. This result suggests that there may be multiple control mechanisms that operate in different contexts. Previous studies have made extensive efforts to develop a generalizable model of cognitive control, but it would also be beneficial to emphasize the diversity of cognitive control. In conclusion, we believe that the model-based approach of the present study in which that computational instantiations of theoretical hypotheses were compared directly using an adaptive experimentation algorithm (i.e., ADO) could help identify and determine the roles of the underlying control mechanisms.

References

Akçay, Ç., & Hazeltine, E. (2007). Conflict monitoring and feature overlap: Two sources of sequential modulations. *Psychonomic Bulletin & Review*, 14(4), 742-748.

Ayton, P., Hunt, A. J., & Wright, G. (1989). Psychological conceptions of randomness. *Journal of Behavioral Decision Making, 2*(4), 221-238.

Beck, A. T., & Clark, D. A. (1997). An information processing model of anxiety: Automatic and strategic processes. *Behaviour Research and Therapy*, *35*(1), 49-58.

Botvinick, M. M., Braver, T. S., Barch, D. M., Carter, C. S., & Cohen, J. D. (2001). Conflict monitoring and cognitive control. *Psychological Review*, *108*, 624-652.

Botvinick, M., Nystrom, L., Fissell, K., Carter, C. S., & Cohen, J. D. (1999). Conflict monitoring versus selection-for-action in anterior cingulate cortex. *Nature*, *402,* 179-181.

Blais, C., Robidoux, S., Risko, E. F., & Besner, D. (2007). Item-specific adaptation and the conflict-monitoring hypothesis: A computational model. *Psychological Review*, *114*, 1076-1086.

Braver, T. S., Gray, J. R., & Burgess, G. C. (2007). Explaining the many varieties of working memory variation: Dual mechanisms of cognitive control. *Variation in Working Memory*, 76-106.

Bugg, J. M., & Chanani, S. (2011). List-wide control is not entirely elusive: Evidence from picture-word Stroop. *Psychonomic Bulletin & Review, 18,* 930–936.

Bugg, J. M., McDaniel, M. A., Scullin, M. K., & Braver, T. S. (2011). Revealing list-level control in the Stroop task by uncovering its benefits and a cost. *Journal of Experimental Psychology: Human Perception And Performance, 37*(5), 1595–1606.

Carter, C. S., MacDonald, A. M., Botvinick, M., Ross, L. L., Stenger, A., Noll, D., & Cohen, J. D. (2000). Parsing executive processes: Strategic versus evaluative functions of the anterior cingulate cortex. *Proceedings of the National Academy of Sciences*, *97*, 1944-1948.

Cavagnaro, D. R., Aranovich, G. J., McClure, S. M., Pitt, M. A., & Myung, J. I. (2016). On the functional form of temporal discounting: An optimized adaptive test. *Journal of Risk and Uncertainty*, *52*(3), 233-254.

Cavagnaro, D. R., Gonzalez, R., Myung, J. I., & Pitt, M. A. (2013). Optimal decision stimuli for risky choice experiments: An adaptive approach. *Management Science*, *59*(2), 358-375.

Cavagnaro, D. R., Myung, J. I., Pitt, M. A., & Kujala, J. V. (2010). Adaptive design optimization: A mutual information-based approach to model discrimination in cognitive science. *Neural Computation*, *22*(4), 887-905.

Chaloner, K., & Verdinelli, I. (1995). Bayesian experimental design: A review. *Statistical Science*, 273-304.

Ding, M., Rosner, G. L., & Müller, P. (2008). Bayesian optimal design for phase II screening trials. *Biometrics*, *64*(3), 886-894.

Duthoo, W., Wühr, P., & Notebaert, W. (2013). The hot-hand fallacy in cognitive control: repetition expectancy modulates the congruency sequence effect. *Psychonomic Bulletin & Review*, *20*(4), 798-805.

Egner, T., Delano, M., & Hirsch, J. (2007). Separate conflict-specific cognitive control mechanisms in the human brain. *Neuroimage*, *35*, 940-948.

Eriksen, B. A., & Eriksen, C. W. (1974). Effects of noise letters upon the identification of a target letter in a nonsearch task. *Attention, Perception, & Psychophysics*, *16*(1), 143-149.

Gratton, G., Coles, M. G., & Donchin, E. (1992). Optimizing the use of information: Strategic control of activation of responses. *Journal of Experimental Psychology: General*, *121*, 480-506.

Gu, H., Kim, W., Hou, F., Lesmes, L. A., Pitt, M. A., Lu, Z. L., & Myung, J. I. (2016). A hierarchical Bayesian approach to adaptive vision testing: A case study with the contrast sensitivity function. *Journal of Vision*, *16*(6), 15-15.

Haines, L. M., Perevozskaya, I., & Rosenberger, W. F. (2003). Bayesian optimal designs for phase I clinical trials. *Biometrics*, *59*(3), 591-600.

Hazeltine, E., Akçay, Ç., & Mordkoff, J. T. (2011). Keeping Simon simple: Examining the relationship between sequential modulations and feature repetitions with two stimuli, two locations and two responses. *Acta Psychologica*, 136(2), 245-252.

Hommel, B., Proctor, R. W., & Vu, K. P. L. (2004). A feature-integration account of sequential effects in the Simon task. *Psychological Research*, *68*, 1-17.

Hutchison, K. A. (2011). The interactive effects of listwide control, item-based control, and working memory capacity on Stroop performance. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 37*(4), 851–860.

Jacoby, L. L., Lindsay, D. S., & Hessels, S. (2003). Item-specific control of automatic processes: Stroop process dissociations. *Psychonomic Bulletin & Review*, *10*, 638-644.

Jiménez, L., & Méndez, A. (2013). It is not what you expect: dissociating conflict adaptation from expectancies in a Stroop task. *Journal of Experimental Psychology: Human Perception and Performance*, *39*(1), 271-284.

Kane, M. J., & Engle, R. W. (2003). Working-memory capacity and the control of attention: The contributions of goal neglect, response competition, and task set to Stroop interference. *Journal of Experimental Psychology: General, 132*(1), 47–70.

Kerns, J. G., Cohen, J. D., MacDonald, A. W., Cho, R. Y., Stenger, V. A., & Carter, C. S. (2004). Anterior cingulate conflict monitoring and adjustments in control. *Science*, *303*, 1023-1026.

Kim, S., Lee, S. H., & Cho, Y. S. (2015). Control processes through the suppression of the automatic response activation triggered by task-irrelevant information in the Simon-type tasks. *Acta Psychologica*, *162*, 51-61.

Kujala, J. V., & Lukka, T. J. (2006). Bayesian adaptive estimation: The next dimension. *Journal of Mathematical Psychology*, *50*(4), 369-389.

Lesmes, L. A., Jeon, S. T., Lu, Z. L., & Dosher, B. A. (2006). Bayesian adaptive estimation of threshold versus contrast external noise functions: The quick TvC method. *Vision Research*, *46*(19), 3160-3176.

Logan, G. D. (1988). Toward an instance theory of automatization. *Psychological Review*, *95*, 492-527.

Logan, G. D., & Zbrodoff, N. (1979). When it helps to be misled: Facilitative effects of increasing the frequency of conflicting stimuli in a Stroop-like task. *Memory & Cognition*, *7*(3), 166–174.

Logan, G. D., & Zbrodoff, N. J. (1982). Constraints on strategy construction in a speeded discrimination task. *Journal of Experimental Psychology. Human Perception and Performance*, *8*(4), 502-520.

Lopes, L. L. (1982). Doing the impossible: A note on induction and the experience of randomness. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *8*(6), 626-636.

Mayr, U., Awh, E., & Laurey, P. (2003). Conflict adaptation effects in the absence of executive control. *Nature Neuroscience*, *6*, 450-452.

McNally, R. J. (1995). Automaticity and the anxiety disorders. *Behaviour Research and Therapy*, *33*(7), 747-754.

Myung, J. I., Cavagnaro, D. R., & Pitt, M. A. (2013). A tutorial on adaptive design optimization. *Journal of Mathematical Psychology*, *57*(3), 53-67.

Myung, J. I., Tang, Y., & Pitt, M. A. (2009). Evaluation and comparison of computational models. *Methods in Enzymology*, *454*, 287-304.

Notebaert, W., Gevers, W., Verbruggen, F., & Liefooghe, B. (2006). Top-down and bottom-up sequential modulations of congruency effects. *Psychonomic Bulletin & Review*, 13(1), 112-117.

Peirce, J. W. (2007). PsychoPy—psychophysics software in Python. *Journal of Neuroscience Methods*, *162*(1), 8-13.

Posner, M. I., & DiGirolamo, G. J. (1998). Conflict, target detection and cognitive control. *The Attentive Brain*, 401-423.

Remington, R. J. (1969). Analysis of sequential effects in choice reaction times. *Journal of Experimental Psychology*, *82*(2), 250-257.

Schmidt, J. R. (2013). Temporal learning and list-level proportion congruency: conflict adaptation or learning when to respond? *PLoS One*, *8*(11), e82320.

Soetens, E., Boer, L. C., & Hueting, J. E. (1985). Expectancy or Automatic Facilitation? Separating Sequential Effects in Two-choice Reaction Time. *Journal of Experimental Psychology: Human Perception and Performance*, *11*(5), 598-616.

Sternberg, R. J., & Sternberg, K. (2016). *Cognitive Psychology*. Nelson Education.

Stürmer, B., Leuthold, H., Soetens, E., Schröter, H., & Sommer, W. (2002). Control over location-based response activation in the Simon task: behavioral and electrophysiological evidence. *Journal of Experimental Psychology: Human Perception and Performance, 28,* 1345-1363.

Trainham, T. N., Lindsay, D. S., & Jacoby, L. L. (1997). Stroop process dissociations: Reply to Hilstrom and Logan (1997). *Journal of Experimental Psychology: Human Perception and Performance*, *23*, 1579-1587.

Tzelgov, J., Henik, A., & Berger, J. (1992). Controlling Stroop effects by manipulating expectations for color words. *Memory & Cognition*, *20*(6), 727–735.

Ullsperger, M., Bylsma, L. M., & Botvinick, M. M. (2005). The conflict adaptation effect: It's not just priming. *Cognitive, Affective, & Behavioral Neuroscience,* 5(4), 467-472.

Verguts, T., & Notebaert, W. (2008). Hebbian learning of cognitive control: Dealing with specific and nonspecific adaptation. *Psychological Review*, *115*, 518-525.

Wells, A., & Matthews, G. (2014). *Attention and Emotion (Classic Edition): A Clinical Perspective*. Psychology Press.

Wendt, M., Kluwe, R. H., & Peters, A. (2006). Sequential Modulations of Interference Evoked by Processing Task-irrelevant Stimulus Features. *Journal of Experimental Psychology: Human Perception and Performance*, *32*(3), 644-667.

Yu, A. and Cohen, J. (2009) Sequential effects: superstition or rational behavior? In *Advances in Neural Information Processing Systems* (Koller, D. *et al.*, eds), pp. 1873–1880.

Appendix

Python Module for the Simulations

This appendix includes the Python module that generated the simulation results in Figures 8-13.

```
###############################################################
# ADO_sim.py (Spring 2017)                                    #
# Python code for ADO/non-ADO simulation for a congruency task        #
# Author: Sang Ho Lee (Ohio State University), lee.7285@osu.edu        #
###############################################################

def ADO_full_module32(simulType, targetModel, simDesign_joint,
        numTarget, numDistract, include, trueSigma_m2, trueBeta3_m2,
        trueBeta2_m2, trueBeta1_m2, trueBeta0_m2, trueLamb_m2,
        trueSigma_m1, trueBeta3_m1, trueBeta2_m1, trueBeta1_m1,
        trueBeta0_m1, trueLamb_m1, trueA_m1, pre_joint, pre_joint2,
        numTrial, numSim, numBlock, lambda_m1, a, b, beta1_m1, interval,
        simDesign, lambda_m2, beta1_m2, sigma, beta2_m1, beta2_m2,
        beta3_m1, beta3_m2, pre_rtGrid, simDesign2):

    import random
    import math
    import numpy as np

    from compute import preCompute_conflict_full32,
        preCompute_conflict_full32_single, preCompute_full32,
        preCompute_full32_single
    from makeList import makeList
    from datetime import datetime
    import pickle
    import matplotlib.pyplot as plt

    rtGrid = pre_rtGrid.copy()

    design_hist = np.zeros((numBlock, numSim))
    design_hist2 = np.zeros((numBlock, numSim))

    modelProb_hist = np.zeros((numBlock+ 1, numSim)) +  0.5
```

```
#initial plot matrices
post_a = post_beta1_m1 = post_beta2_m1 = post_beta3_m1 =
      post_lamb_m1 = 0
post_b = post_beta1_m2 = post_beta2_m2 = post_beta3_m2 =
      post_lamb_m2 = 0#np.zeros(numSim).tolist()

plotError_a = plotError_lamb_m1 = plotError_beta1_m1 =
      plotError_beta2_m1 = plotError_beta3_m1 = plotError_sigma_m1 =
      0
plotError_lamb_m2 = plotError_b = plotError_beta1_m2 =
      plotError_beta2_m2 = plotError_beta3_m2 = plotError_sigma_m2 =
      0

# preCompute the likelihoods (model 1)
beta0_m1 = trueBeta0_m1
beta0_m2 = trueBeta0_m2
trueB_m2 = 0.1

repMatrix = np.zeros((len(simDesign_joint),2))
repMatrix[:,0] = np.floor(simDesign_joint/100)/100 #repRate_c
repMatrix[:,1] = (simDesign - (np.floor(simDesign_joint/100)
      *100))/100 #repRate_i

uMatrix = repMatrix.copy()

repMatrix = np.add.outer(np.zeros(len(a)),repMatrix)

u = 0.5 +  a[None,:,None,None] +
      np.zeros((len(lambda_m1),len(simDesign), 2))[:,None,:,:]
      #lambda*a*design*cong

update_interval = np.ceil(numTrial/interval)-1 # number of update per
      interval : update_interval/interval = probability of update
lambda2 = np.multiply(np.true_divide(1-
      pow(lambda_m1,update_interval),1-lambda_m1),1-lambda_m1) #
      update multiplier

baseTrial = interval*(np.ceil(numTrial/interval)-1)
post_dist, u =
      preCompute_full32(lambda_m1,u,a,beta1_m1,sigma,beta0_m1,beta2
```

```
        _m1,beta3_m1,rtGrid,baseTrial,repMatrix,update_interval)
        #interval -> update_interval
update_interval = 1 # for one interval calculation during the trials

# preCompute model 2
average_c = 1-simDesign2  # initial average conflict
divisor_c = 0
sum_c = 0

post_dist2= preCompute_conflict_full32(average_c, b, lambda_m2,
        beta1_m2, sigma, beta0_m2, beta2_m2, beta3_m2, rtGrid,
        simDesign2)

#initial utility
post_dist = post_dist * pre_joint[:, :, :, :, :, None, None, None, None]  #
        p(paramIy,d)
post_dist2 = post_dist2 * pre_joint2[:, :, :, :, :, None, None, None,
        None]  # p(paramIy,d)

like_Y = np.sum(post_dist, axis=(0,1,2,3,4)) # p(yId)
like_Y2 = np.sum(post_dist2, axis=(0,1,2,3,4))

# rarely occurs
if np.sum(like_Y < 1e-45) != 0:
    print "low like 1"
    like_Y[like_Y < 1e-45] = 1e-45 # remove 0 to prevent infinite/NaN
        bayes factor (rarely occurs)

if np.sum(like_Y2 < 1e-45) != 0:
    print "low like 2"
    like_Y2[like_Y2 < 1e-45] = 1e-45 # 1e-323 in float64

# convert like_Y2 to joint design space
tempLike = np.zeros(like_Y.shape) #d1*rep*cong*RT
tempJoint = np.subtract(simDesign_joint,simDesign) #congRate

for i in range(len(simDesign_joint)):

    loc = np.argmin(abs(simDesign2 - tempJoint[i]))
    tempLike[i,:,:,:] = like_Y2[loc,:,:,:]
```

```
like_Y2 = tempLike
del tempLike

# weight the likelihood
modelProb = 0.5 # initial p(m)

reshape_Y2 = np.zeros(like_Y2.shape)  #transform conflict dimension to
      the rep dimension
reshape_Y2[:,:,0,:] = like_Y2[:,:,0,:].copy()
reshape_Y2[:,0,1,:] = like_Y2[:,1,1,:].copy()
reshape_Y2[:,1,1,:] = like_Y2[:,0,1,:].copy()

reshape_Y = np.zeros(like_Y.shape)  #transform rep dimension to the
      conflict dimension
reshape_Y[:,:,0,:] = like_Y[:,:,0,:].copy()
reshape_Y[:,0,1,:] = like_Y[:,1,1,:].copy()
reshape_Y[:,1,1,:] = like_Y[:,0,1,:].copy()

utility = np.multiply(like_Y, -np.log(modelProb + ((1-modelProb) *
      np.true_divide(reshape_Y2, like_Y))))
utility2 = np.multiply(like_Y2, -np.log((1-modelProb) + (modelProb *
      np.true_divide(reshape_Y, like_Y2))))

utility = np.sum(utility, axis=(1, 2, 3)) # d1 * d2
utility2 = np.sum(utility2, axis=(1, 2, 3))

pre_utility_joint = (modelProb * utility) + ((1 - modelProb) * utility2)

del utility, utility2, like_Y, like_Y2, post_dist, post_dist2, reshape_Y,
      reshape_Y2

# start simulation
for iSimul in range(numSim):

   u = 0.5 +  a[None,:,None,None] +
      np.zeros((len(lambda_m1),len(simDesign), 2))[:,None,:,:]
      #lambda*a*design*cong

   average_c = 1-simDesign2  # initial average conflict
   divisor_c = 0
   sum_c = 0
```

```
prior_joint = pre_joint
prior_joint2 = pre_joint2
utility_joint = pre_utility_joint
rtGrid = pre_rtGrid.copy()

cong = np.zeros((numTrial,numBlock))
bayes_like = np.zeros((numTrial,2))

obsList = np.zeros((numTrial, numBlock))

# initialize SSE matrices
SSElog_a = np.zeros(numBlock+1)
SSElog_a[0] = pow(np.sum(np.sum(prior_joint, axis = (0, 1, 2,
  3))*a)-trueA_m1, 2)
SSElog_lamb_m1 = np.zeros(numBlock+1)
SSElog_lamb_m1[0] = pow(np.sum(np.sum(prior_joint, axis = (0, 1, 2,
  4))*lambda_m1)-trueLamb_m1, 2)
SSElog_beta1_m1 = np.zeros(numBlock+1)
SSElog_beta1_m1[0] = pow(np.sum(np.sum(prior_joint, axis = (1, 2,
  3, 4))*beta1_m1)-trueBeta1_m1, 2)
SSElog_beta2_m1 = np.zeros(numBlock+1)
SSElog_beta2_m1[0] = pow(np.sum(np.sum(prior_joint, axis = (0, 2,
  3, 4))*beta2_m1)-trueBeta2_m1, 2)
SSElog_beta3_m1 = np.zeros(numBlock+1)
SSElog_beta3_m1[0] = pow(np.sum(np.sum(prior_joint, axis = (0, 1,
  3, 4))*beta3_m1)-trueBeta3_m1, 2)

SSElog_lamb_m2 = np.zeros(numBlock+1)
SSElog_lamb_m2[0] = pow(np.sum(np.sum(prior_joint2, axis = (0, 1,
  2, 4))*lambda_m2)-trueLamb_m2, 2)
SSElog_b = np.zeros(numBlock+1)
SSElog_b[0] = pow(np.sum(np.sum(prior_joint2, axis = (0, 1, 2,
  3))*b)-trueB_m2, 2)
SSElog_beta1_m2 = np.zeros(numBlock+1)
SSElog_beta1_m2[0] = pow(np.sum(np.sum(prior_joint2, axis = (1, 2,
  3, 4))*beta1_m2)-trueBeta1_m2, 2)
SSElog_beta2_m2 = np.zeros(numBlock+1)
SSElog_beta2_m2[0] = pow(np.sum(np.sum(prior_joint2, axis = (0, 2,
  3, 4))*beta2_m2)-trueBeta2_m2, 2)
SSElog_beta3_m2 = np.zeros(numBlock+1)
```

```
SSElog_beta3_m2[0] = pow(np.sum(np.sum(prior_joint2, axis = (0, 1,
  3, 4))*beta3_m2)-trueBeta3_m2, 2)

# initialize prior distribution history
modelProb = 0.5 # initial p(m)

prior_hist = np.zeros(numBlock+1).tolist()
prior_hist[0] = prior_joint

prior_hist2 = np.zeros(numBlock+1).tolist()
prior_hist2[0] = prior_joint2

# Start simulations ####
designList = np.array([3070.3,3070.3,3070.3,7030.7,7030.7,7030.7])
  #for pre-determined design: F2

for iBlock in range(numBlock):

    print 'simulation %d block %d, %s' % (iSimul+1, iBlock+1,
  datetime.now().strftime('%H:%M:%S'))  # '%Y-%m-%d %H:%M:%S'

    if simulType == 3:
        designPick_joint = designList[iBlock]
    else:
        designPick_joint = simDesign_joint[np.argmax(utility_joint)]

    stringDesign = str(int(designPick_joint*100))
    repRate = np.float32(stringDesign[0:4])
    congRate = np.float32(stringDesign[4:6])/100

    design_hist[iBlock,iSimul] = repRate
    design_hist2[iBlock,iSimul] = congRate

    designPick1 = np.argmin(abs(simDesign - repRate))
    designPick2 = np.argmin(abs(simDesign2 - congRate))

    iterThreshold = 1000000
    stimList = makeList(designPick_joint, numTrial, iterThreshold)

    rep = 1  # no rep at the first trial
    conflict = 0  # no conflict
```

```
 prev_indicator = -1
 numCorrect = 0 #start update after the first correct response
(ignore the first one)

 for iTrial in range(numTrial):

    if stimList[iTrial][2] == 1:  # identify congruency
       cong[iTrial, iBlock] = 0  # 0 if congruent
       currentCong = 0
    else:
       cong[iTrial, iBlock] = 1
       currentCong = 1

    if iTrial > 0:  # ignore the first trial (no repetition)
       if cong[iTrial-1, iBlock] == cong[iTrial, iBlock]:
          rep = 0  # rep
       else:
          rep = 1  # non-rep

    if iTrial > 0:
       if cong[iTrial-1, iBlock] == 1:  # incongruent
          conflict = 1  # conflict
       else:
          conflict = 0

    if targetModel == 1:
       # model 1 mu
       trueU = u[np.argmax(lambda_m1 == trueLamb_m1),
np.argmax(a >= trueA_m1), designPick1, currentCong]

       if cong[iTrial-1,iBlock] == cong[iTrial,iBlock]:
          truePX = trueU
       else:
          truePX = 1-trueU

       mu = trueBeta0_m1 + (trueBeta1_m1 * (1-truePX)) +
(trueBeta2_m1 * cong[iTrial,iBlock]) + (trueBeta3_m1 * (1-
truePX) * cong[iTrial,iBlock])  # true mu
       # pick obs
```

```
        pdf_RT =
np.multiply((float(1)/math.sqrt(2*pow(trueSigma_m1, 2)*math.pi)),
                              np.exp(-
np.true_divide(np.power(np.subtract(rtGrid, mu), 2),
2*pow(trueSigma_m1, 2))))  # probability of observing RT
     elif targetModel == 2:
        # model 2 mu
        trueC = (trueLamb_m2 * conflict) + ((1-trueLamb_m2) *
average_c[designPick2])
        mu = trueBeta0_m2 + (trueBeta1_m2 * trueC) +
(trueBeta2_m2 * cong[iTrial, iBlock]) + (-trueBeta3_m2 *
trueBeta2_m2 * cong[iTrial, iBlock] * trueC)  # true mu

        pdf_RT =
np.multiply((float(1)/math.sqrt(2*pow(trueSigma_m2, 2)*math.pi)),
                              np.exp(-
np.true_divide(np.power(np.subtract(rtGrid, mu), 2),
2*pow(trueSigma_m2, 2))))  # probability of observing RT

     pdf_RT = np.true_divide(pdf_RT, np.sum(pdf_RT))  # normalize
the pdf

     obs = np.argmax(np.cumsum(pdf_RT)>random.random())  #
location of the observation in the rtGrid

     obsList[iTrial, iBlock] = rtGrid[obs]

     list_indicator = int(np.floor(iTrial/interval))

     if list_indicator != prev_indicator:

        prev_indicator = list_indicator

        baseTrial = interval*list_indicator
        preList, u =
preCompute_full32_single(designPick1,lambda_m1,u,a,beta1_m1,sig
ma,beta0_m1,beta2_m1,beta3_m1,rtGrid,baseTrial,repMatrix,update
_interval)  #interval -> update_interval
```

```
        preList2 = preCompute_conflict_full32_single(designPick2,
average_c, b, lambda_m2, beta1_m2, sigma, beta0_m2, beta2_m2,
beta3_m2, rtGrid, simDesign2)

        divisor_c = divisor_c + interval
        sum_c = sum_c + (interval * (1-simDesign2))
        average_c = sum_c/divisor_c

    ## pre_RT = (beta0,beta1,beta2,lambda,a,rep,cong,rt) likelihood
    post_dist = preList[:, :, :, :, :, rep, currentCong, obs] *
prior_joint # p(paramIy,d)
    post_dist2 = preList2[:, :, :, :, :, conflict, currentCong, obs] *
prior_joint2

    like_Y = np.sum(post_dist) # p(yId)
    like_Y2 = np.sum(post_dist2)

    if like_Y < 1e-45:  # remove 0 to prevent infinite/NaN bayes
factor (rarely occurs)
        like_Y = 1e-45  # 1e-323 in float64
    elif like_Y2 < 1e-45:
        like_Y2 = 1e-45

    bayes_like[iTrial,:] = np.log(np.array([like_Y2, like_Y]))

    ##updatd prior
    prior_joint = post_dist # posterior update given obs,design
    prior_joint2 = post_dist2 # posterior update given obs,design

    prior_joint = np.true_divide(prior_joint,np.sum(prior_joint))
#normalize the prior

    prior_joint2 = np.true_divide(prior_joint2,np.sum(prior_joint2))
#normalize the prior


  prior_hist[iBlock+1] = prior_joint
  prior_hist2[iBlock+1] = prior_joint2

  del preList, preList2
```

```
 # update SSE
 SSElog_a[iBlock+1] = pow(np.sum(np.sum(prior_joint, axis = (0,
1, 2, 3))*a)-trueA_m1, 2)
 SSElog_lamb_m1[iBlock+1] = pow(np.sum(np.sum(prior_joint,
axis = (0, 1, 2, 4))*lambda_m1)-trueLamb_m1, 2)
 SSElog_beta1_m1[iBlock+1] = pow(np.sum(np.sum(prior_joint,
axis = (1, 2, 3, 4))*beta1_m1)-trueBeta1_m1, 2)
 SSElog_beta2_m1[iBlock+1] = pow(np.sum(np.sum(prior_joint,
axis = (0, 2, 3, 4))*beta2_m1)-trueBeta2_m1, 2)
 SSElog_beta3_m1[iBlock+1] = pow(np.sum(np.sum(prior_joint,
axis = (0, 1, 3, 4))*beta3_m1)-trueBeta3_m1, 2)

 SSElog_lamb_m2[iBlock+1] = pow(np.sum(np.sum(prior_joint2,
axis = (0, 1, 2, 4))*lambda_m2)-trueLamb_m2, 2)
 SSElog_b[iBlock+1] = pow(np.sum(np.sum(prior_joint2, axis = (0,
1, 2, 3))*b)-trueB_m2, 2)
 SSElog_beta1_m2[iBlock+1] = pow(np.sum(np.sum(prior_joint2,
axis = (1, 2, 3, 4))*beta1_m2)-trueBeta1_m2, 2)
 SSElog_beta2_m2[iBlock+1] = pow(np.sum(np.sum(prior_joint2,
axis = (0, 2, 3, 4))*beta2_m2)-trueBeta2_m2, 2)
 SSElog_beta3_m2[iBlock+1] = pow(np.sum(np.sum(prior_joint2,
axis = (0, 1, 3, 4))*beta3_m2)-trueBeta3_m2, 2)

 #utility  - use updated prior, re-calculate the likelihood

 #re-calculate the likelihood (m1)
 lambda2 = np.multiply(np.true_divide(1-
pow(lambda_m1,update_interval),1-lambda_m1),1-lambda_m1) #
update multiplier

 u =
np.float32(np.multiply(pow(lambda_m1,update_interval)[:,None,Non
e,None], u) +  np.multiply.outer(lambda2, repMatrix)) # update u
for the last interval
 u = u[:,:,designPick1,:][:,:,None,:] +
np.zeros(len(simDesign))[None,None,:,None] # fix u to the current
design

 update_interval = np.ceil(numTrial/interval)-1  # update multiple
times to the last interval
 baseTrial = interval*(np.ceil(numTrial/interval)-1)
```

```
 post_dist, _ =
preCompute_full32(lambda_m1,u,a,beta1_m1,sigma,beta0_m1,beta2
_m1,beta3_m1,rtGrid,baseTrial,repMatrix,update_interval)
#interval -> update_interval

 update_interval = 1 #reset the interval

 average_c = ((average_c[designPick2] * divisor_c) + ((numTrial -
interval) * (1-simDesign2)))/(divisor_c + numTrial - interval) #
update to the last interval

 post_dist2 = preCompute_conflict_full32(average_c, b, lambda_m2,
beta1_m2, sigma, beta0_m2, beta2_m2, beta3_m2, rtGrid,
simDesign2)

 sum_c = sum_c[designPick2]
 average_c = np.add.outer(sum_c/divisor_c,
np.zeros(len(simDesign2))) #fix average c to the current design

 ##re-calculate the likelihood (m2)
 post_dist = post_dist * prior_joint[:, :, :, :, :, None, None, None,
None]  # p(param|y,d)
 post_dist2 = post_dist2 * prior_joint2[:, :, :, :, :, None, None,
None, None]

 like_Y = np.sum(post_dist, axis=(0,1,2,3,4)) # p(y|d)
 like_Y2 = np.sum(post_dist2, axis=(0,1,2,3,4))

 bayes_factor = np.sum(bayes_like,axis=0)
 bayes_factor = np.exp(bayes_factor[0] - bayes_factor[1])
 modelProb = modelProb/((1-
modelProb)*bayes_factor+ modelProb)

 print modelProb

 switch = 0

 if modelProb > 0.95:
    switch = 1
 elif modelProb < 0.05:
    switch = 1
```

```
if modelProb > 0.999:
    modelProb = 0.999
elif modelProb < 0.001:
    modelProb = 0.001

modelProb_hist[iBlock+ 1,iSimul] = modelProb

if simulType == 1 and switch == 1:

    post_dist[post_dist == 0] = 1 #log(1) = 0, 1*log(1) = 0
    post_dist2[post_dist2 == 0] = 1 #log(1) = 0, 1*log(1) = 0

    post_dist = -np.sum(np.multiply(post_dist, np.log(post_dist)),
axis=(0, 1, 2, 3, 4))  # H

    utility = -np.multiply(like_Y,post_dist)  # HY
    utility = np.sum(utility, axis = (1, 2, 3))

    post_dist2 = -np.sum(np.multiply(post_dist2,
np.log(post_dist2)), axis=(0, 1, 2, 3, 4))  # H

    utility2 = -np.multiply(like_Y2,post_dist2)  # HY
    utility2 = np.sum(utility2, axis = (1, 2, 3))

    tempUtil = np.zeros(len(simDesign_joint))
    tempJoint = np.subtract(simDesign_joint,simDesign) #congRate

    for i in range(len(simDesign_joint)):

        loc = np.argmin(abs(simDesign2 - tempJoint[i]))
        tempUtil[i] = utility2[loc]

    utility2 = tempUtil

else:

    # rarely occurs
    if np.sum(like_Y < 1e-45) != 0:
        print "low like 1"
```

```
        like_Y[like_Y < 1e-45] = 1e-45 # remove 0 to prevent
infinite/NaN bayes factor (rarely occurs)

    if np.sum(like_Y2 < 1e-45) != 0:
        print "low like 2"
        like_Y2[like_Y2 < 1e-45] = 1e-45 # 1e-323 in float64

    # convert like_Y2 to joint design space
    tempLike = np.zeros(like_Y.shape) #d1*rep*cong*RT
    tempJoint = np.subtract(simDesign_joint,simDesign) #congRate

    for i in range(len(simDesign_joint)):

        loc = np.argmin(abs(simDesign2 - tempJoint[i]))
        tempLike[i,:,:,:] = like_Y2[loc,:,:,:]

    like_Y2 = tempLike
    del tempLike

    reshape_Y2 = np.zeros(like_Y2.shape)  #transform conflict
dimension to the rep dimension
    reshape_Y2[:,:,0,:] = like_Y2[:,:,0,:].copy()
    reshape_Y2[:,0,1,:] = like_Y2[:,1,1,:].copy()
    reshape_Y2[:,1,1,:] = like_Y2[:,0,1,:].copy()

    reshape_Y = np.zeros(like_Y.shape)  #transform rep dimension
to the conflict dimension
    reshape_Y[:,:,0,:] = like_Y[:,:,0,:].copy()
    reshape_Y[:,0,1,:] = like_Y[:,1,1,:].copy()
    reshape_Y[:,1,1,:] = like_Y[:,0,1,:].copy()

    utility = np.multiply(like_Y, -np.log(modelProb + ((1-
modelProb) * np.true_divide(reshape_Y2, like_Y))))
    utility2 = np.multiply(like_Y2, -np.log((1-modelProb) +
(modelProb * np.true_divide(reshape_Y, like_Y2))))

    utility = np.sum(utility, axis=(1, 2, 3)) # d1 * d2
    utility2 = np.sum(utility2, axis=(1, 2, 3))

    del reshape_Y, reshape_Y2
```

```
        utility_joint = (modelProb * utility) + ((1 - modelProb) * utility2)

        del like_Y, like_Y2, post_dist, post_dist2

    # store SSE every simulation
    plotError_a = np.add(plotError_a,SSElog_a)
    plotError_lamb_m1 = np.add(plotError_lamb_m1,SSElog_lamb_m1)
    plotError_beta1_m1 = np.add(plotError_beta1_m1,SSElog_beta1_m1)
    plotError_beta2_m1 = np.add(plotError_beta2_m1,SSElog_beta2_m1)
    plotError_beta3_m1 = np.add(plotError_beta3_m1,SSElog_beta3_m1)

    plotError_lamb_m2 = np.add(plotError_lamb_m2,SSElog_lamb_m2)
    plotError_b = np.add(plotError_b,SSElog_b)
    plotError_beta1_m2 = np.add(plotError_beta1_m2,SSElog_beta1_m2)
    plotError_beta2_m2 = np.add(plotError_beta2_m2,SSElog_beta2_m2)
    plotError_beta3_m2 = np.add(plotError_beta3_m2,SSElog_beta3_m2)

    post_a = np.add(post_a, np.sum(prior_joint, axis = (0, 1, 2, 3)))
    post_lamb_m1 = np.add(post_lamb_m1, np.sum(prior_joint, axis = (0,
      1, 2, 4)))
    post_beta1_m1 = np.add(post_beta1_m1, np.sum(prior_joint, axis =
      (1, 2, 3, 4)))
    post_beta2_m1 = np.add(post_beta2_m1, np.sum(prior_joint, axis =
      (0, 2, 3, 4)))
    post_beta3_m1 = np.add(post_beta3_m1, np.sum(prior_joint, axis =
      (0, 1, 3, 4)))

    post_lamb_m2 = np.add(post_lamb_m2, np.sum(prior_joint2, axis = (0,
      1, 2, 4)))
    post_b = np.add(post_b, np.sum(prior_joint2, axis = (0, 1, 2, 3)))
    post_beta1_m2 = np.add(post_beta1_m2, np.sum(prior_joint2, axis =
      (1, 2, 3, 4)))
    post_beta2_m2 = np.add(post_beta2_m2, np.sum(prior_joint2, axis =
      (0, 2, 3, 4)))
    post_beta3_m2 = np.add(post_beta3_m2, np.sum(prior_joint2, axis =
      (0, 1, 3, 4)))

# average SSE over simulations
plotError_a = np.sqrt(np.true_divide(plotError_a, numSim * numTrial))
plotError_lamb_m1 = np.sqrt(np.true_divide(plotError_lamb_m1, numSim
      * numTrial))
```

```
plotError_beta1_m1 = np.sqrt(np.true_divide(plotError_beta1_m1,
      numSim * numTrial))
plotError_beta2_m1 = np.sqrt(np.true_divide(plotError_beta2_m1,
      numSim * numTrial))
plotError_beta3_m1 = np.sqrt(np.true_divide(plotError_beta3_m1,
      numSim * numTrial))
plotError_sigma_m1 = 0

plotError_lamb_m2 = np.sqrt(np.true_divide(plotError_lamb_m2, numSim
      * numTrial))
plotError_b = np.sqrt(np.true_divide(plotError_b, numSim * numTrial))
plotError_beta1_m2 = np.sqrt(np.true_divide(plotError_beta1_m2,
      numSim * numTrial))
plotError_beta2_m2 = np.sqrt(np.true_divide(plotError_beta2_m2,
      numSim * numTrial))
plotError_beta3_m2 = np.sqrt(np.true_divide(plotError_beta3_m2,
      numSim * numTrial))
plotError_sigma_m2 = 0

post_a = np.true_divide(post_a, numSim)
post_lamb_m1 = np.true_divide(post_lamb_m1, numSim)
post_beta1_m1 = np.true_divide(post_beta1_m1, numSim)
post_beta2_m1 = np.true_divide(post_beta2_m1, numSim)
post_beta3_m1 = np.true_divide(post_beta3_m1, numSim)

post_lamb_m2 = np.true_divide(post_lamb_m2, numSim)
post_b = np.true_divide(post_b, numSim)
post_beta1_m2 = np.true_divide(post_beta1_m2, numSim)
post_beta2_m2 = np.true_divide(post_beta2_m2, numSim)
post_beta3_m2 = np.true_divide(post_beta3_m2, numSim)

if targetModel == 1:
    return design_hist, design_hist2, modelProb_hist, plotError_a,
      plotError_lamb_m1, plotError_beta1_m1, plotError_beta2_m1,
      plotError_beta3_m1, plotError_sigma_m1, post_a, post_lamb_m1,
      post_beta1_m1, post_beta2_m1, post_beta3_m1, post_lamb_m2,
      post_b, post_beta1_m2, post_beta2_m2, post_beta3_m2

elif targetModel == 2:
    return design_hist, design_hist2, modelProb_hist, plotError_lamb_m2,
      plotError_b, plotError_beta1_m2, plotError_beta2_m2,
```

plotError_beta3_m2, plotError_sigma_m2, post_a, post_lamb_m1, post_beta1_m1, post_beta2_m1, post_beta3_m1, post_lamb_m2, post_b, post_beta1_m2, post_beta2_m2, post_beta3_m2