Motor sequence execution while counting tones and using an unpracticed hand configuration after extensive practice: Does stimulus-response translation matter when other processing strategies are inhibited?

Master Thesis by

Steffen Neumeister

Faculty of Behavioural, Management and Social Sciences

Master of Human Factors and Engineering Psychology

University of Twente

Supervisor: Prof. Dr. Ing. Verwey

2nd Supervisor: Dr. van der Lubbe

26th May 2022

Abstract

The present study investigated whether it is likely that stimulus-response translation contributes significantly to sequence execution in the discrete sequence production (DSP) task after extensive practice. This was done by inhibiting two other processing strategies, namely, the use of motor chunks and central-symbolic representations. In a DSP study, 24 participants counted tones and used an unpracticed hand configuration for sequence execution after extensive practice. Based on the assumption that each manipulation slowed responses by inhibiting a processing strategy, we expected the combined slowing of tone counting and using an unpracticed hand configuration to be larger than the added slowing of each manipulation alone. However, the results showed an additive increase of mean reaction times (RTs) when combining the two manipulations. To determine whether this can be explained by significant contribution of stimulus-response translation, we used the data of the DSP study to compare a model including stimulus-response translation with a model excluding it. The simulation of both models showed that stimulus-response translation likely contributed to sequence execution, but that it cannot explain the additive increase of RTs either. This could mean that more processing strategies were used or that other factors such as biomechanical differences affected mean RTs.

Keywords: discrete sequence production task, automated movement sequences, motor chunking, secondary task, sequence learning, cognitive processing, stimulus-response translation

Table of contents

1. Introduction

How humans automate a sequence of movements through practice is a topic that has been studied by a considerable number of researchers. When people automate behaviors like typing on a keyboard, driving a car, or lacing a shoe, they can use their processing resources for something else like observing the traffic in the case of the car example (Verwey et al., 2015). This begs the question of why it is possible for a person to engage in this kind of multi-tasking despite the sequence of movements, which is being executed, potentially being quite complex. Additionally, one might wonder if and possibly why a secondary task impairs sequence execution in certain ways, for example, by decreasing the speed or accuracy of sequence execution.

An attempt to explain the underlying processes in movement sequence learning is made by the cognitive framework for sequential motor behavior (C-SMB) developed by Verwey et al. (2015). The C-SMB distinguishes between a central and a motor processor that are responsible for sequence execution and operate in parallel. Both processors are initially needed to execute a sequence of movements in the sense that the central processor initiates and triggers each movement, so the motor processor can execute it. Through practice, motor chunks develop which reduce the complexity of executing a sequence of individual actions to executing one or more motor chunks of actions that summarize the whole sequence. This enables the motor processor to execute a sequence on its own while the central processor only has to initiate the sequence but not trigger each individual movement. Hence, if the central processor is unavailable for sequence execution, for example, due to a secondary task, the motor processor can still execute the sequence. This might explain, for example, why it is possible, after extensive driving practice, to shift gears quickly while still observing the traffic or having a conversation with another passenger.

However, it has also been observed that the speed of sequence execution increases through practice and that it can decrease again after practice if a secondary task is being performed while executing the sequence (Verwey et al., 2015). Here, the C-SMB argues that, after extensive practice, the central and motor processors are assumed to be racing to trigger each next movement of the sequence. Each processor builds up its own representation of the response and whichever processor is faster at providing a complete representation is also the processor responsible for triggering the response. A theoretical implication of this race model is that adding another processor to the race would generally increase the speed of sequence execution while removing a processor would generally decrease the speed of sequence execution.[1] For example, when motor chunks have formed, the motor processor can race with the central processor which should increase the speed of sequence execution in addition to the use of motor chunks generally being faster than other processing strategies which further increases the speed of sequence execution. Similarly, eliminating the central processor from the race by, for example, engaging in a secondary task while executing a sequence should decrease the speed of sequence execution even when motor chunks have already formed. This brings us to the focus of this paper.

1.1 Aims

The general purpose of this paper was to affirm the race assumptions of the C-SMB in the context of executing sequences in the discrete sequence production (DSP) task (Abrahamse et al., 2013). To be more precise, we wanted to investigate to what extent responses are slowed in the DSP task after extensive practice when one or even two processors are eliminated from the race of processors. One of those two processors is the motor processor and the other one is a processor related to central processing which will be explained in more detail in the following paragraphs. Several studies with the DSP task have already suggested that responses are slowed when a processor is presumably eliminated from the race (De Kleine & Verwey, 2009; Verwey et al., 2010; Verwey et al., 2014; Verwey & Clegg, 2005; Verwey & Wright, 2004). However, how much responses are slowed if two processors are inhibited from contributing to sequence execution has not been studied yet. This is the central question that this paper will address along with the question of which two processors are likely being inhibited from contributing to sequence execution in the DSP task.

The latter question is related to the idea that the central processor itself can use different processing strategies for sequence execution (Verwey et al., 2015). To understand how this might be relevant for the present study, it is first necessary to understand how the involvement of the central processor in sequence execution evolves even before motor chunks have formed. When a sequence of movements is initially presented, an individual can only use stimulus-response (S-R) translation for sequence execution which is rather slow (Verwey et al., 2015). S-R translation describes the notion that the central processor triggers individual movements of a sequence which is characterized by reacting to key-specific stimuli in the

[1] While this is generally true, the extent to which the speed of sequence execution increases or decreases depends on how often the added or removed processor wins or won the race. A processor, that rarely wins the race and hence does not contribute much to sequence execution, also has a small impact on the speed of sequence execution when added to or removed from the race of processors.

DSP task. After a bit more practicing, the central processor uses so-called central-symbolic representations, and the individual can respond faster.[2] The C-SMB assumes that a secondary task inhibits the use of central-symbolic representations by the central processor causing the slowing of responses after practice. With more repetition, motor chunks develop in long-term memory which means that individuals can load these chunks to execute the sequences rapidly with limited effort.

While it was initially believed that S-R translation would be outpaced by the other sequencing systems suggesting that the central processor would mainly use central-symbolic representations after extensive practice, Verwey et al. (2020) have shown that participants continue reacting to key-specific stimuli as long as their luminance changes. This means that S-R translation can still race with the other sequencing systems (i.e., the use of motor chunks and central-symbolic representations). Moreover, Verwey et al. (2014) suggested that the central processor can be split up to perform different parallel processes. Hence, rather than performing one task at a time and switching between two tasks in a serial fashion, the central processor could perform two tasks parallelly by distributing central processing resources to both tasks in a graded fashion. Therefore, based on the findings of Verwey et al. (2020) and Verwey et al. (2014), it would be possible to partition the central processor into an *S-R translation processor* and a *central-symbolic processor*. These two processors are assumed to work in parallel racing against each other and the motor processor.

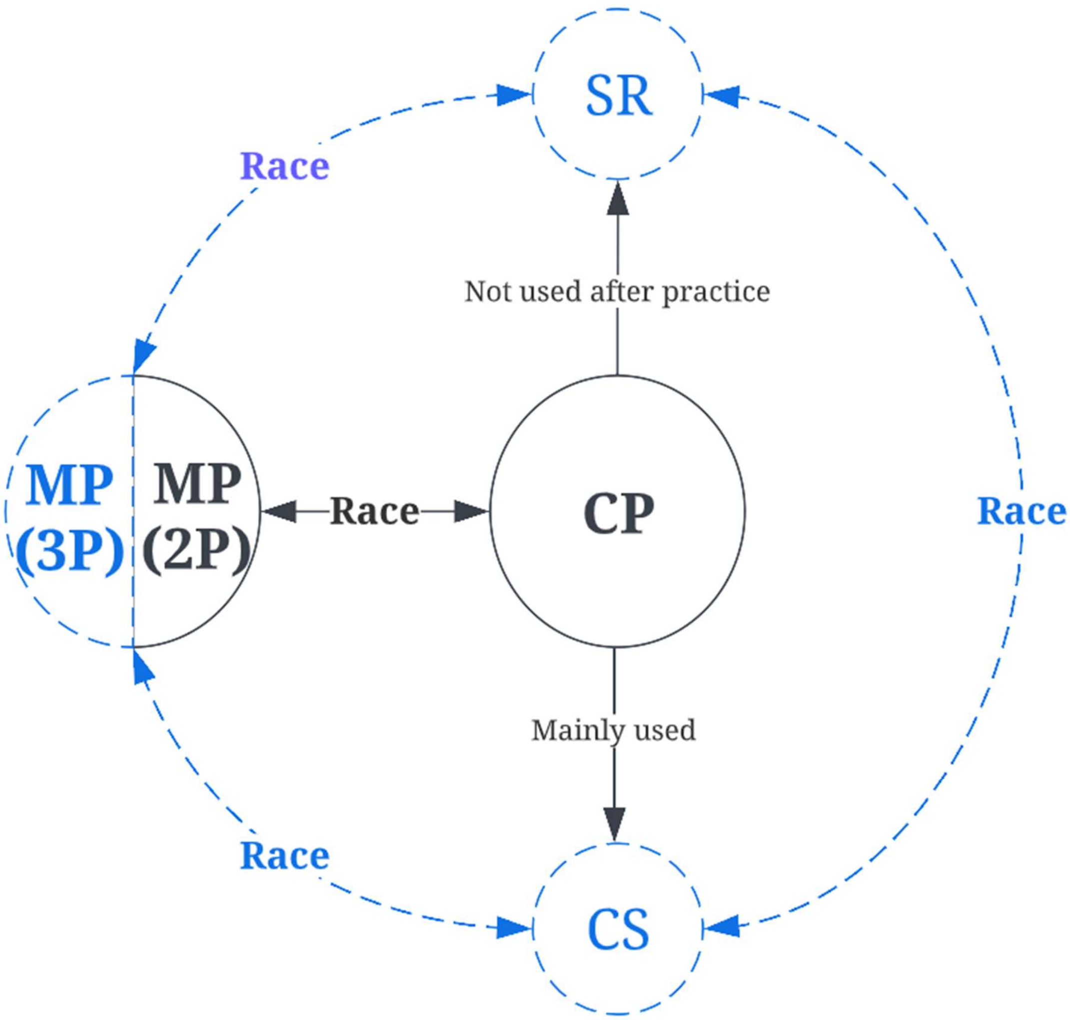
Hence, such a three-processor model basically assumes the S-R translation (SR) processor, the Central-Symbolic (CS) processor, and the Motor Processor (MP). The previous two-processor model assumes the Motor Processor and the Central Processor (CP) that reads central-symbolic representations and performs the S-R translation which would not be relevant for sequence execution after extensive practice. Figure 1 summarizes the differences between the two models.

---

[2] In this phase, the central processor still selects individual responses, but these are primed by execution of the preceding responses at all processing levels enabling the individual to respond faster (Verwey et al., 2015). This is a further component of sequence learning called ‘associative learning’ which is also responsible for decreasing RTs through practice. The possible effect of associative learning on sequence execution is being largely ignored in this paper.

Figure 1

Relationships between the processors in terms of racing with differentiation between the two-processor (2P) model (solid black lines in the center) and the three-processor (3P) model (dashed blue lines forming the outer circle). For the 2P model, the central processor (CP) mainly uses central-symbolic (CS) representations after extensive practice whereas S-R translation (SR) is only used initially with little or no practice. The motor processor (MP) is part of both models and requires extensive practice, so it can use motor chunks to participate in the race to trigger the next response. For the three-processor model, one parallel race including the MP, CS, and SR processors is assumed rather than three individual races between each pair of processors like the figure might indirectly suggest.



While Verwey et al. (2020) found evidence for participants continuing to react to key-specific stimuli after extensive practice, it was not clearly known whether the contribution of S-R translation to sequence execution would affect reaction times in a meaningful way when two processors are presumably eliminated from the race. Therefore, we also could not know beforehand how relevant the contribution of an SR processor would be for the present study. This was an issue because our predictions regarding the slowing of responses were different depending on how much the SR processor contributes. If the contribution of the SR processor is small after extensive practice, only the motor and central processors are assumed. Furthermore, when both processors are inhibited, no processor would initially be available for sequence execution. While it can be assumed that the central processor switches between a secondary task and sequence execution to perform both tasks at "roughly" the same time, (Verwey et al., 2015) this would most likely slow the central processor significantly due to the switching interrupting the execution of the sequence. Hence, an increased slowing of responses would be expected if the motor and central processors are inhibited from contributing to sequence execution.

On the other hand, significant contribution of the SR processor after extensive practice might not lead to this increased slowing of responses when two processors are inhibited. If S-R translation continues after extensive practice, the central-symbolic processor would be inhibited by a secondary task while the SR processor could still contribute to sequence execution and possibly, but not necessarily, prevent an increased slowing of responses that the two-processor model would expect.

Hence, while a reaction time study (Study 1) had the purpose of investigating to what extent responses of a DSP sequence are slowed after extensive practice when one or two processors are eliminated from the race of processors, a follow-up simulation study (Study 2) was designed to test whether the assumed processors can be eliminated from the race. This was done by modeling the processors and races of processors of the two- and three-processor models based on the RT data of the DSP study to determine which model was more likely to be true. However, before going into more detail regarding Study 2, we first have to discuss how the assumed processors could be inhibited from contributing to sequence execution in the DSP task by discussing prior research.

1.2 Previous research with the DSP task

In the DSP task, participants are required to execute keying sequences on a simple computer keyboard while their reaction time (RT) is being measured (Abrahamse et al., 2013). Multiple studies with the DSP tasks have shown that a secondary task slows responses and suggested that a secondary task can eliminate a processor related to central processing from the race to trigger the next response (Verwey et al., 2010; Verwey et al., 2014). For example, Verwey et al. (2010) used a tone counting task as a secondary task to investigate the role of the central processor during sequence execution. The secondary tone counting task had the purpose of occupying the central processor so that its contribution to sequence execution was inhibited. When participants had to count tones next to executing the sequences, their RTs were longer. Moreover, responses slowed down less for familiar sequences compared to unfamiliar ones when comparing sequence executing with and without tone counting. The lessened slowing of responses for familiar sequences can be explained by the reduced involvement of the central processor in sequence execution once motor chunks have formed. This would support the assumption that a secondary task eliminates a processor related to central processing from the race.

Next, there are also several studies which might suggest that the use of motor chunks by the motor processor can be inhibited by a change of effectors given that the sequence has been extensively practiced (De Kleine & Verwey, 2009; Verwey & Clegg, 2005; Verwey & Wright, 2004). Verwey and Wright (2004) examined whether performance in the DSP task is effector-dependent. The effectors in this case were the hand configurations which differed between the use of fingers from one or two hands in practice. Verwey and Wright (2004) compared sequence execution with practiced and unpracticed hand configurations. Their results suggest that learning in the DSP task involves an effector-dependent component in the sense that participants executed sequences with their practiced hand configuration faster than with the unpracticed hand configuration. This slowing could be explained by the motor processor not being able to participate in the race anymore. Nonetheless, executing a familiar sequence with an unpracticed hand configuration was faster than executing an unfamiliar sequence with it suggesting an effector-independent learning component. Verwey and Wright (2004) concluded that sequence learning starts in an effector-independent way and gradually involves an effector-dependent component with more practice. This could imply that the use of motor chunks by the motor processor is effector-dependent, as motor chunks develop in later stages of sequence learning and require extensive practice.

To conclude, both tone counting and using an unpracticed hand configuration after extensive practice are assumed to be manipulations that eliminate a processor, namely the motor processor and a processor related to central processing, from the race to trigger the next response of a sequence. In that sense, the studies from Verwey and Wright (2004) and Verwey et al. (2010) affirmed the race assumptions of the C-SMB by suggesting that using only one processor led to longer RTs relative to using two processors. However, an issue is that we cannot truly know whether tone counting and using an unpracticed hand configuration actually eliminated a processor from the race and that this elimination caused RTs to increase. For example, Verwey et al. (2016) pointed out that the effector-dependent component of sequence learning could alternatively be explained by adjusting to the biomechanical properties of the used effectors while motor learning might be mostly effector-independent. This would imply that using an unpracticed hand configuration does not directly affect the use of motor chunks by the motor processor.

As it was not fully clear whether the RT increases in previous studies can be attributed to the elimination of a processor from the race, we tested whether the presumed elimination of two processors from the race would be consistent with the race assumptions of the C-SMB in terms of RTs increasing. Hence, by investigating how much RTs increase if we combine the use of tone counting and an unpracticed hand configuration, we attempted to generate further evidence supporting the notion of parallel processors that are racing to trigger each next response and that can be eliminated from the race by the manipulations used in our and previous studies.

2. Study 1

The aim of Study 1 was to examine to what extent tone counting and/or using an unpracticed hand for sequence execution would increase RTs in the DSP task after extensive practice. To that end, we incorporated a practice phase into the DSP task which served the purpose of building up central-symbolic representations and motor chunks through practice. In this practice phase, one of two hand configurations was used which did not change until the following test phase. Here, a tone counting task and a switch of hand configurations from the practiced to the unpracticed hand configuration were implemented to study the effects of tone counting and using an unpracticed hand configuration on sequence execution.

This led to four different conditions in the test phase. One condition included the tone counting task (Tone Counting condition), a second one involved the use of an unpracticed hand configuration (Hand Switch condition), and a third one included both of these manipulations at the same time (Hand Switch/Tone Counting condition). Finally, the fourth condition did not include any manipulations relative to practice (Control condition).

As we assumed that each manipulation would eliminate one processor from the race to trigger the next response, and the C-SMB postulates that eliminating a processor from the race slows responses, we expected tone counting and using an unpracticed hand configuration to each increase RTs similar to previous studies (De Kleine & Verwey, 2009; Verwey et al., 2010; Verwey et al., 2014; Verwey & Clegg, 2005; Verwey & Wright, 2004). Consequently, mean RTs in the Tone Counting and Hand Switch conditions were predicted to be longer than mean RTs in the Control condition.

Next, given that both the use of tone counting and an unpracticed hand configuration slowed responses in previous studies and each of them is assumed to eliminate one processor from the race, it could also be expected that combining the two manipulations would further slow responses. The question is how large this increase in RTs would be relative to the slowing with one manipulation. The two-processor model would predict an increased slowing of responses due to the assumed unavailability of both central and motor processors for sequence execution. Statistically, the increased slowing of responses would be reflected in an interaction of tone counting and using an unpracticed hand configuration significantly increasing mean RTs in the Hand Switch/Tone Counting condition.

However, the slowing of responses might not be further increased when three processors are involved, and S-R translation continues to contribute to sequence execution after extensive practice. An additive effect would be predicted if the slowing of responses is not further increased, so the sum of the slowing effects of tone counting and using an unpracticed hand configuration would be roughly the same as their combined slowing effect in the Hand Switch/Tone Counting condition. However, as we did not know before Study 2 whether the three-processor model would predict an additive or interactive increase of RTs, both types of RT increases were theoretically possible and could be an indication of significant contribution of the SR processor to sequence execution. Hence, we limited our explicit expectations for Study 1 to the two-processor model and the assumption that S-R translation contributes little to sequence execution after extensive practice. Based on that assumption, we expected the RT increase when tone counting and using an unpracticed hand configuration are combined to be larger than the sum of RT increases when only one manipulation is applied.

Lastly, Verwey et al. (2010) have shown that the effect of tone counting has only been observed for the three responses following a tone and that ensuing responses until the tone of the next sequence were not affected by the secondary task anymore. Consequently, we also expected the three responses following a tone to be slowed more than the responses preceding the tone. This analysis excluded the initial response of a sequence which is usually significantly slower than all following responses of a sequence even after extensive practice (Verwey et al., 2015).

3. Methods

3.1 Participants

Twenty-four participants between the age of 18 and 29 years took part in this experiment (M_{age} = 20.71 years, SD_{age} = 2.40 years). They were all students from the University of Twente. Only right-handed participants who did not smoke and had not drunk any alcohol in the 24 hours prior to the experiment were eligible to participate. The reward for participating in this study were 3.5 credit points. The ethics committee of the University of Twente approved this study.

3.2 Materials

The experiments were conducted in a small room on a Dell PC using Windows 10 with a Logitech keyboard using the QWERTY layout as well as a PS/2 connection. The AOC G2460PF monitor had a resolution of 1920x1080, a size of 24 inches, and a refresh rate of 144 Hz. A pair of Sennheiser HD 400s headphones was used for the test phase to present high- and low-pitched tones. An informed consent, a participant instruction paper, and a paper version of the awareness task were given to the participant. The blocks for the practice phase, test phase, and a computerized awareness task were created using E-prime 2.0 (E-Prime® Legacy Versions, 2018).

3.3 Task

The version of the DSP task used in this study started with a practice phase consisting of six blocks. Each block consisted of 240 trials with a 20-second break halfway through each block and a four-minute break after each block, yielding 720 practice trials per sequence. Four empty squares were shown on the screen in horizontal order. The squares corresponded to the keys C, V, B, N as spatially compatible placeholders that were ordered in the same horizontal way as the keys. To indicate which of the four keys had to be pressed, the corresponding placeholder was filled with green color while the other three placeholders remained white like the background and did not illuminate.

Two 7-key sequences were executed in each practice session and counterbalancing was used to distribute those two out of four sequences to each participant. The sequences used in this experiment were VBCNVNC, CNVBCBV, BVNCNCB, and NCBVBVN. Which one of the two sequences was shown on each trial was randomized, but the total number of trials with each sequence was equal at the end of the practice phase. There was a 1500 ms break between each trial. In case of a mistake, the message “error, wait” was shown for 1 second and the current sequence was skipped. The same happened if a response was not quick enough which was followed by a “no response” message.

The test phase consisted of one block including four subblocks with 48 trials each. At the third key position of a sequence in all subblocks, a tone was played for 100 ms. This tone was either low pitched (440 Hz) or high pitched (698 Hz), and it was random which one was played. The task was to count the low pitch tones and ignore the high pitch tones. However, this was only required for half of the subblocks whereas the others demanded ignoring all tones. To be more precise, the subblocks were divided into a 2 (hand configuration: practiced vs unpracticed) x 2 (tone counting: count vs ignore) design. The order of the subblocks was counterbalanced across participants. There was a 20-second break between each subblock which was followed by an instruction page for the next subblock.

3.4 Procedure

Before the experiment started, participants filled out the informed consent and read the participant instructions. They gave their cell phones to the researcher to avoid distraction. Then, the participant read the description of the experiment shown on the computer screen. It was checked whether the participant used the correct fingers and that these were placed on the keys C, V, B, N before starting. Half of the participants started with the left hand whereas the other half used fingers from both of their hands. Participants starting with the left hand used all of their fingers except the thumb, so the little finger was on the far left key and the index finger on the far right. The other half started with pinky and ring finger from their left hand on the two leftmost keys and index finger plus middle finger from their right hand on the two rightmost keys. If the participant did not have any further questions, the researcher left the room and the participant started with the first block. The experimenter could observe the participant through a camera in the room. After each block and its respective break, the researcher came back into the room to start the next block. This was done for all six blocks of the practice phase.

After the sixth and last practice block, the experimenter explained what was expected from the participant in the seventh block. Written instructions followed on the computer screen, and the researcher made sure that everything was understood. The participant put on the headphones and listened to an example of the low- and high-pitched tones. Then, the experimenter left again, so the participant could start the first of four subblocks. Before each subblock, the participant read an instruction page that indicated what had to be done. Regardless of whether low tones had to be counted or not, during the test block high- and low-pitched tones were presented to the participant in all subblocks. After a subblock that included tone counting, the participant indicated how many low tones they had counted.

Following the test block, a computerized awareness task was used to assess explicit sequence knowledge by testing whether participants could reproduce the sequences executed during the DSP task without seeing the keyboard. A slightly different paper version of the awareness task was also used. However, the results of these awareness tasks are not reported in this paper and therefore, the awareness tasks are not further described.

3.5 Data Analysis

For the practice phase, a Hand(s) used in Practice (2: one vs two) x Block (6) x Key (7) repeated measures ANOVA was used with the Hand(s) used in Practice as the between-subjects variable and the other two variables as within-subjects variables. This was done for both the mean reaction times and the error proportions that were transformed to arcsine proportions before submission. For the test phase, a Hand(s) used in Practice (2: one vs two) x Hand configuration (2: unpracticed vs practiced) x Tone counting condition (3: distractor tone vs target tone vs no counting) x Key (7) repeated measures ANOVA was used with the Hand(s) used in Practice as the between-subjects variable and the other three as within-subjects variables. Again, arcsine proportions of the errors were submitted to this ANOVA design as well as the mean reaction times.

Moreover, a Hand(s) used in Practice (2: one vs two) x Hand configuration (2: unpracticed vs practiced) x Tone counting condition (3: distractor tone vs target tone vs no counting) x Key (3: 3-5) repeated measures ANOVA was done with the Hand(s) used in Practice as the between-subjects variable and the other three as within-subjects variables. This analysis served the purpose of focusing only on the three responses following a tone, as Verwey et al. (2010) have shown that still later responses after a tone were not slowed by it anymore. For all repeated measures ANOVAs, Greenhouse-Geisser values were reported unless the assumption of sphericity was violated in which case the results of the multivariate test were reported. Lastly, a Hand(s) used in Practice (2: one vs two) x Hand configuration (2: unpracticed vs practiced) x Keys before/after tone (2: 2,6,7 vs 3,4,5) x Tone counting condition (3: distractor tone vs target tone vs no counting) repeated measures ANOVA was used with Hand(s) used in Practice as the between-subjects variable and the other three as within-subjects variables. This last ANOVA was used for planned comparisons regarding whether responses following a tone were slowed more than those preceding it as observed by Verwey et al. (2010).

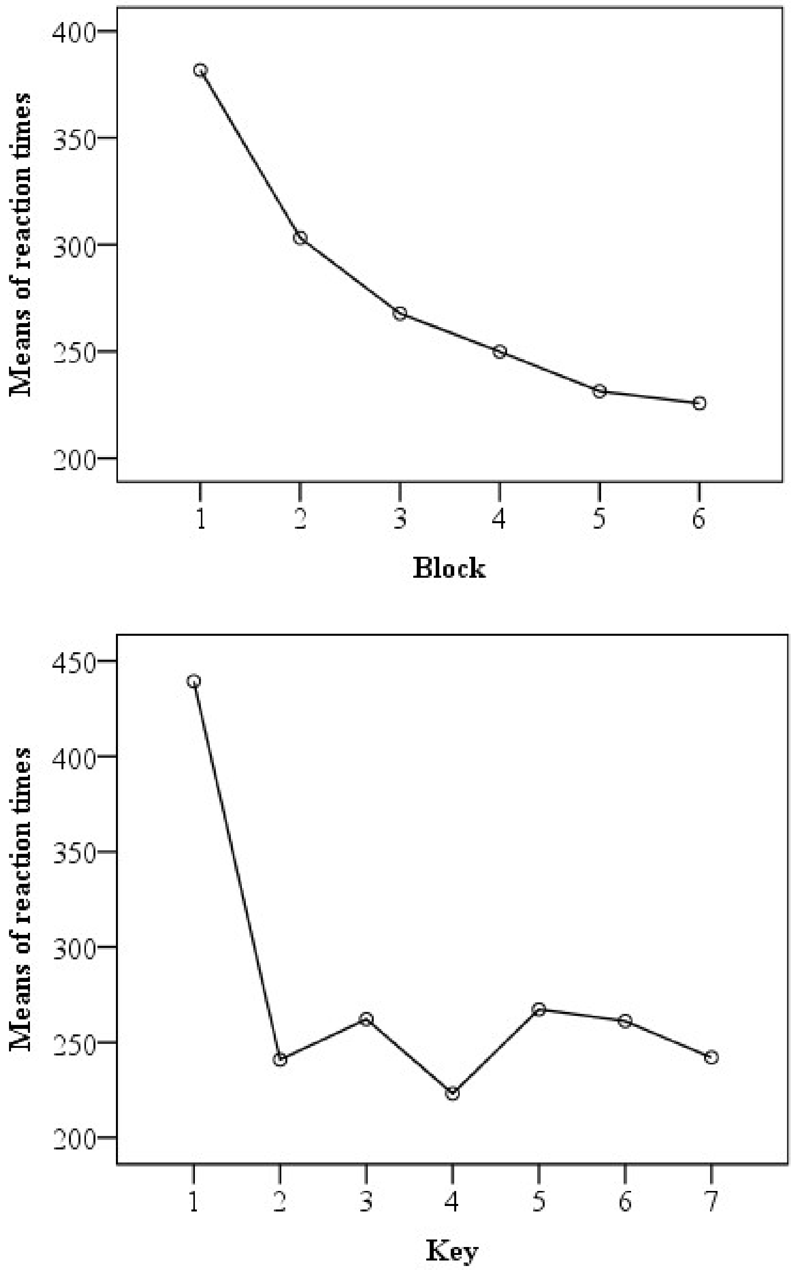
4. Results

4.1 Practice phase

The repeated measures ANOVA for the mean reaction times showed that participants became faster with more practice, $F(2.41,53.11) = 131.22$, $p < 0.01$, $\eta_p^2 = 0.85$, and the mean reaction times for key positions within a sequence differed as well, $F(2.38,52.54) = 136.52$, $p < 0.01$, $\eta_p^2 = 0.86$. Figure 2 shows the mean reaction times for all blocks and key positions. Furthermore, a Block x Key interaction indicated that some key positions experienced a higher decrease in mean reaction times through practice compared to others, $F(6.68,147.16) = 9.43$, $p < 0.01$, $\eta_p^2 = 0.30$. Lastly, it did not seem to matter whether participants practiced with one ($M = 270.95$ ms) or two hands ($M = 282.23$ ms), $F(1,22) = 0.18$, $p = 0.66$, $\eta_p^2 = 0.01$.

Figure 2

Mean reaction times (ms) for each of the six blocks (upper frame) as well as each of the seven key positions of a sequence across all of these blocks (lower frame).



Next, the repeated measures ANOVA for arcsine-transformed error proportions showed that the proportions differed based on the key position within a sequence, $F(3.81,83.97) = 21.88$, $p < 0.01$, $\eta_p^2 = 0.49$ ($M_1 = 0.005$, $M_2 = 0.010$, $M_3 = 0.019$, $M_4 = 0.003$, $M_5 = 0.021$, $M_6 = 0.025$, $M_7 = 0.008$). Moreover, a Block x Key interaction indicated that certain key positions differed in terms of the development of error proportions through practice, $F(9.26,203.90) = 2.24$, $p = 0.02$, $\eta_p^2 = 0.09$. Lastly, a Block x Key x Hand(s) used in Practice interaction showed that the deviations of the previous interaction look different when comparing whether one or two hands were used, $F(9.26,203.90) = 1.94$, $p = 0.046$, $\eta_p^2 = 0.08$.

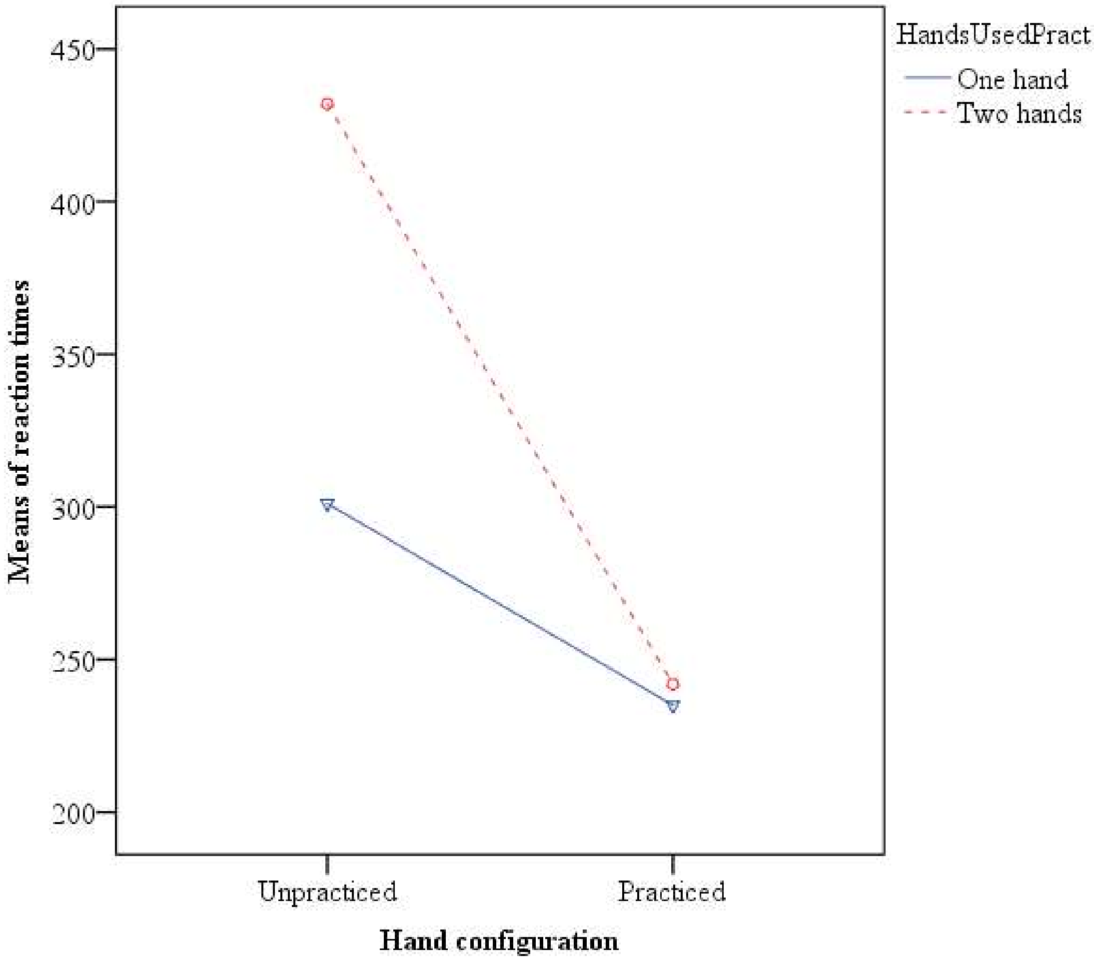
4.2 Test phase

The repeated measures ANOVA for the mean reaction times including all responses showed that participants were faster when they used the same hand configuration that they had previously practiced with ($M = 238.49$ ms) compared to using an unpracticed hand configuration ($M = 366.56$ ms), $F(1,22) = 88.32$, $p < 0.01$, $\eta_p^2 = 0.80$. Participants were fastest in the no-counting condition ($M = 271.01$ ms), followed by sequences with a distractor tone ($M = 311.21$ ms), and lastly, sequences with a target tone ($M = 325.35$ ms), $F(1.55,34.17) = 26.67$, $p < 0.01$, $\eta_p^2 = 0.54$. A Hand configuration x Tone counting condition interaction was not found, $F(1.23,27.13) = 0.05$, $p = 0.86$, $\eta_p^2 = 0.003$ (Practiced: $M_{DistractorTone} = 247.53$ ms, $M_{TargetTone} = 260.07$ ms, $M_{NoCounting} = 207.86$ ms; Unpracticed: $M_{DistractorTone} = 374.88$ ms, $M_{TargetTone} = 390.62$ ms, $M_{NoCounting} = 334.17$ ms).

Additionally, key positions within a sequence differed in terms of their mean reaction time, $F(2.80,61.81) = 63.30$, $p < 0.01$, $\eta_p^2 = 0.74$ ($M_1 = 461.27$ ms, $M_2 = 255.80$ ms, $M_3 = 315.57$ ms, $M_4 = 244.38$ ms, $M_5 = 290.87$ ms, $M_6 = 285.54$ ms, $M_7 = 264.24$ ms). Participants who practiced with two hands were in general slower than participants who practiced with one hand, $F(1,22) = 4.78$, $p = 0.04$, $\eta_p^2 = 0.17$. Next, a Hand configuration x Hand(s) used in Practice interaction showed that participants who practiced with two hands were slowed more when they switched to one hand than vice versa, $F(1,22) = 20.72$, $p < 0.01$, $\eta_p^2 = 0.48$ (Figure 3). A Hand configuration x Key interaction indicated that the difference between the hand configurations used was not the same for all key positions, $F(3.82,84.04) = 8.50$, $p < 0.01$, $\eta_p^2 = 0.27$, and that this effect was also dependent on the hand(s) used in practice as seen in the Hand configuration x Key x Hand(s) used Practice interaction, $F(3.82,84.04) = 9.58$, $p < 0.01$, $\eta_p^2 = 0.30$.

Figure 3

Mean reaction times (ms) of all responses in the test phase for using a practiced or unpracticed hand configuration based on the hand(s) used in practice

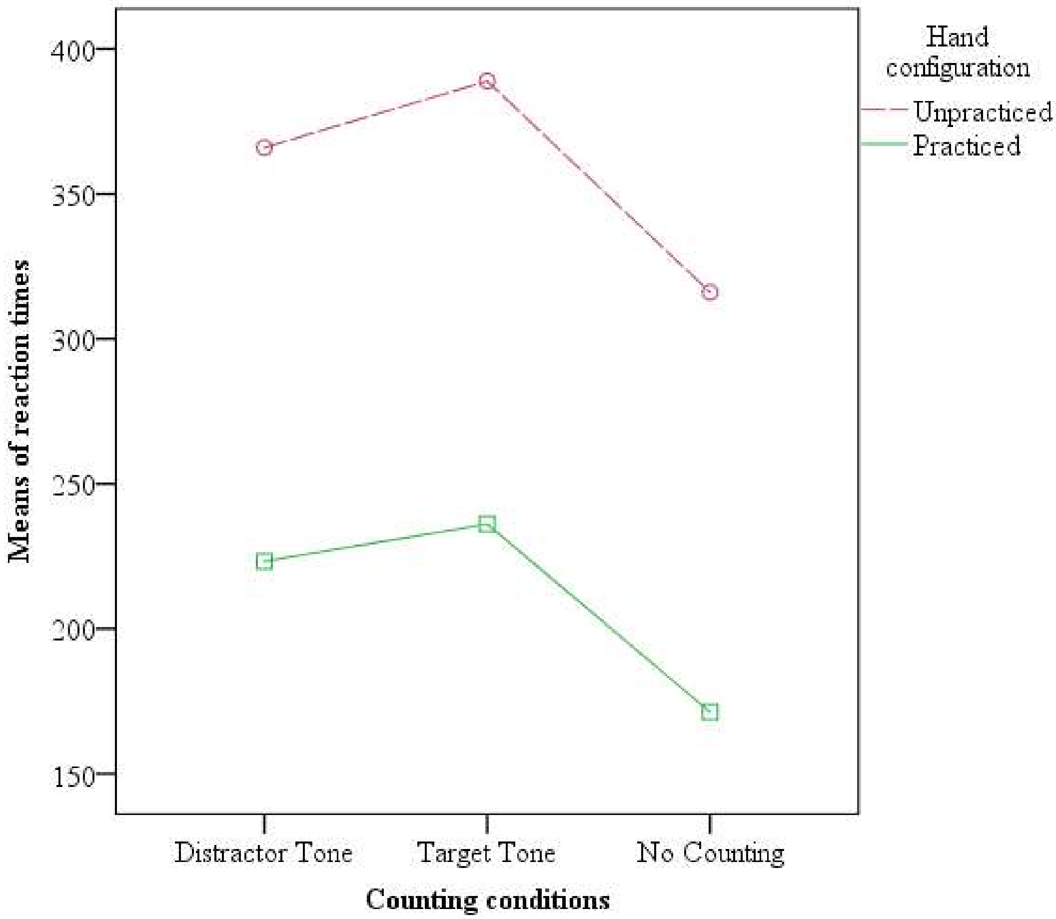


Conducting the same repeated measures ANOVA with only the mean reaction times for responses three to five affirmed the main effect of Tone counting condition, $F(1.59,34.98) = 20.44$, $p < 0.01$, $\eta_p^2 = 0.48$, and the main effect of Hand configuration, $F(1,22) = 78.37$, $p < 0.01$, $\eta_p^2 = 0.78$. Nonetheless, a significant Hand configuration x Tone counting condition interaction was not found in this analysis either, $F(1.19,26.38) = 0.19$, $p = 0.71$, $\eta_p^2 = 0.01$. Figure 4 shows that the increases in mean reaction times following a tone were additive when tones had to be counted and an unpracticed hand configuration was used. The major differences in this analysis compared to the analysis including all responses were that it is questionable whether the difference between practicing with one (M = 247.89 ms) or two hands (M = 319.33 ms) made a significant impact on mean reaction times of the test phase,

$F(1,22) = 4.19$, $p = 0.053$, $\eta_p^2 = 0.16$, and that a Hand configuration x Key x Hand(s) used Practice interaction was not observed, $F(1.81,39.94) = 0.36$, $p = 0.67$, $\eta_p^2 = 0.01$.

Figure 4

Mean reaction times (ms) for using a practiced or unpracticed hand configuration in all three counting conditions for responses three to five



Planned comparisons also showed that the three responses following a tone ($M_{DistractorTone} = 294.62$ ms, $M_{TargetTone} = 312.49$ ms, $M_{NoCounting} = 243.69$ ms) were slowed more than responses two, six, and seven ($M_{DistractorTone} = 277.32$ ms, $M_{TargetTone} = 289.57$ ms, $M_{NoCounting} = 238.69$ ms), $F(1,22) = 4.67$, $p = 0.04$, $\eta_p^2 = 0.17$.

Next, the repeated measures ANOVA for arcsine-transformed error proportions showed that participants made more errors when they used an unpracticed hand configuration ($M = 0.023$) compared to the practiced hand configuration ($M = 0.013$), $F(1,22) = 20.10$, $p < 0.01$, $\eta_p^2 = 0.47$, and that participants made significantly more errors when they had to count a tone compared to only identifying a distractor tone or ignoring all tones, $F(2,21) = 11.24$, $p < 0.01$, $\eta_p^2 = 0.51$ ($M_{DistractorTone} = 0.015$, $M_{TargetTone} = 0.022$, $M_{NoCounting} = 0.018$). Furthermore, the error proportions regarding a key position within a sequence differed as well, $F(4.13,90.90) = 13,22$, $p < 0.01$, $\eta_p^2 = 0.37$ ($M_1 = 0.014$, $M_2 = 0.009$, $M_3 = 0.023$, $M_4 = 0.005$, $M_5 = 0.029$, $M_6 = 0.034$, $M_7 = 0.015$). Lastly, a Tone counting condition x Key interaction revealed that differences between the counting conditions do not seem to be the same for all key positions, $F(7.32,161.22) = 3.12$, $p < 0.01$, $\eta_p^2 = 0.12$.

5. Discussion

The purpose of Study 1 was to test the predictions that, first of all, reaction times (RTs) in the DSP task would get longer when tones have to be counted or an unpracticed hand configuration is used for sequence execution and secondly, that counting tones and using an unpracticed hand configuration at the same time would lead to an increase in RTs that is larger than the sum of RT increases when only one manipulation is applied. The results confirmed the prediction that counting tones during sequence execution increases RTs relative to executing sequences without counting tones. A similar effect was observed for the use of an unpracticed hand configuration in the sense that RTs increased relative to using the practiced hand configuration. Furthermore, as expected, the three responses following a tone were slowed more than the ones preceding a tone. These results are in line with previous research studying the effects of tone counting and using an unpracticed hand configuration after extensive practice (De Kleine & Verwey, 2009; Verwey et al., 2010; Verwey et al., 2014; Verwey & Clegg, 2005; Verwey & Wright, 2004). Still, the expected interaction of tone counting and using an unpracticed hand configuration could not be observed, as the increase in RTs was additive rather than interactive when both manipulations were applied relative to applying only one manipulation (Figure 4).

Another notable observation in this study was that it mattered which hand configuration was used in practice. Figure 3 shows that this was relevant when participants used an unpracticed hand configuration, as participants who practiced with two hands were slower when using one hand than vice versa. This is also what Verwey and Wright (2004) and Verwey et al. (2016) found in their studies. A possible explanation for the difference between the use of one or two hands as unpracticed configurations might be the adjustment hypothesis which suggests that more practice is needed to adjust to the biomechanical properties of fingers from one hand compared to fingers from two hands (Verwey & Clegg, 2005). As participants who practiced with two hands did not have any practice with one hand, this biomechanical difference might explain why they were slower using their unpracticed hand configuration than participants who practiced with one hand and then used two hands in the test phase.

Overall, despite affirming the slowing effects of tone counting and using an unpracticed hand configuration after extensive practice, Study 1 failed to generate further evidence supporting the race assumptions of the C-SMB given that the manipulations were assumed to eliminate the motor processor and a processor related to central processing from the race to trigger the next response. A possible explanation for the missing interaction of tone counting and using an unpracticed hand configuration could be that this prediction was based on the assumption that S-R translation would contribute little after extensive practice. We have already mentioned that a three-processor model including the SR processor does not necessarily predict an increased slowing of responses when two processors are presumably eliminated from the race of processors. Hence, it would be possible that continued S-R translation in the test phase of Study 1 caused RTs in the Hand Switch/Tone Counting condition to not increase as much as initially expected. Study 2 investigated whether this assumption was likely true or not.

6. Study 2

The two-processor model assumed that S-R translation would contribute little to sequence execution after extensive practice. This assumption might have been wrong given the results of Study 1 did now show an interaction of tone counting and using an unpracticed hand configuration increasing mean RTs in the DSP task as expected. However, this did not necessarily imply that the three-processor model including the SR processor was correct either. Hence, the purpose of Study 2 was to investigate (1) whether it is likely that S-R translation significantly contributed to sequence execution in the test phase of Study 1 and (2) whether significant contribution of S-R translation to sequence execution can explain the additive effect of tone counting and using an unpracticed hand configuration in the Hand Switch/Tone Counting condition. We will first recap the theoretical assumptions of the three-processor model (Model 3P) in contrast to the two-processor model (Model 2P) by putting them into the context of Study 1 and then, discuss the basic procedure used in Study 2 to answer the described research questions.

As recent evidence indicates that participants still react to stimuli using S-R translation after extensive practice when key-specific stimuli involve a luminance change (Verwey et al., 2020), and Verwey et al. (2014) have shown that the central processor can be split up to perform different parallel processes, Model 3P assumes a partitioned central processor in which an SR processor and central-symbolic (CS) processor are working in parallel racing against each other and the motor processor (MP) (Figure 1). This partitioned central processor stands in contrast to the central processor of Model 2P which behaves like a single unit switching between tone counting and sequence execution, for example. While Model 2P assumed that tone counting would eliminate the central processor from the race of processors as a whole, Model 3P assumes that the central-symbolic processor is eliminated from the race while the SR processor can continue contributing to sequence execution.

Table 1 shows how this logic is applied to the four test conditions of Study 1 with regards to which processor is assumed to contribute to sequence execution according to each model. As the luminance of key-specific stimuli changed in all four conditions, we expected the SR processor of Model 3P to contribute to sequence execution in all four conditions.

Table 1

Processors involved in each test condition according to both models with 'vs.' implying that a race of processors took place (MP = Motor processor, CP = Central processor, CS = Central-symbolic processor, SR = Stimulus-response translation processor).

Model	Control	Tone Counting	Hand Switch	Hand Switch/Tone Counting
Model 2P (Study 1)	MP vs. CP	MP	CP	CP switching
Model 3P	MP vs. CS vs. SR	MP vs. SR	CS vs. SR	SR

Study 2 examined which of these models, if any, fits the mean RTs observed in the four conditions of the test phase in Study 1. This required the simulation of processors and races assumed in each test condition as shown in Table 1. By modeling the processors and races of each model, we tried to determine which model is better able to predict the results of Study 1 and hence, also whether Model 3P could replicate the unexpected additive effect in the Hand Switch/Tone Counting condition.

On the basis of the RTs in Study 1 we first estimated the processing time distributions for each assumed processor that could then be used to model the assumed races of processors for each condition.[3] To be more precise, we started off using the RT distributions obtained in the Tone Counting, Hand Switch, and Hand Switch/Tone Counting conditions of Study 1 to estimate the distributions of each processor from Model 3P. For example, as the SR processor was the only processor contributing to sequence execution in the Hand Switch/Tone Counting condition according to Model 3P, its processor distribution should theoretically be roughly the same as the RT distribution of the Hand Switch/Tone Counting condition. Similarly, the distribution resulting from the race of SR and CS processors should be roughly the same as the RT distribution of the Hand Switch condition.

For Model 2P, the RT distributions obtained in the Tone Counting and Hand Switch conditions were used to estimate the distributions of motor and central processors. However, we did not simulate the switching central processor of Model 2P for the Hand Switch/Tone Counting condition, as that would have required concrete prior knowledge about how exactly this switching works. Hence, we just assumed, but could not reliably test, that the switching central processor would replicate the mean RT of the Hand Switch/Tone Counting condition.

Lastly, all two or three processors of the respective models raced in the Control condition, and we determined which, if any, of the models would be able to show the respective pattern of mean RTs from Study 1. Ultimately, only a model producing similar mean simulated RTs (simRTs) for all four conditions would be successful in reproducing the pattern of mean RTs from Study 1. Given that Verwey et al. (2020) have shown that participants continue reacting to key-specific stimuli as long as their luminance changes and the results of Study 1 were not in line with the predictions of the two-processor model, we expected Model 3P to be able to reproduce the pattern of mean RTs suggesting an additive effect in the Hand Switch/Tone Counting condition. This would show that significant contribution of S-R translation to sequence execution can explain the additive effect in the test phase of Study 1.

[3] The processing time refers to the time a processor needs to build up a complete representation of a response, so the winner of a race between two or more processing times is the processor which can more quickly transmit a completed representation, hence having a shorter processing time. The build-up of the representation can be seen in a similar way as described by Brown and Heathcote (2008) who developed the linear ballistic accumulator (LBA) model. Evidence for a response is accumulated in multiple separate evidence accumulation processes until a certain response threshold is met. The first accumulator (similar to processor) that reaches this threshold provides the response.

However, although we tried to reproduce the mean RTs of the four test conditions from Study 1, the basis for the simulations were not the mean RTs from Figure 4. The mean RTs from Figure 4 also included the biomechanical slowing due to the use of fingers of a single hand as the unpracticed hand configuration (Figure 3). As we did not want the biomechanical slowing to skew mean RTs, we only included data of two-handed sequence execution for the Hand Switch and Hand Switch/Tone Counting conditions. This led to two different modeling approaches which were based on two different datasets that only included part of the data for each condition that is summarized as mean RTs in Figure 4. More details regarding the datasets can be found in Section 7.1.

A within-subjects approach (including the data of one half of participants) allowed us to reduce individual differences between participants through standardization which further reduced skewness in the data. For a between-subjects approach, this was not possible, so individual differences remained high. However, the pattern of mean standardized RTs from the within-subjects approach suggested an interactive effect in the Hand Switch/Tone Counting condition.[4] Hence, the between-subjects approach, which included a pattern of mean RTs suggesting an additive effect, was still needed to test whether Model 3P can explain the additive effect observed in the test phase of Study 1.

As both modeling approaches had certain advantages and disadvantages, we considered it to be most useful to simulate Models 2P and 3P for both approaches and test whether the models could reproduce the pattern of results suggesting an interactive and/or additive effect. While the within-subjects approach could not be used to test whether any of the models can reproduce the additive effect observed in Study 1, a replication of the data from one half of the participants could still indicate whether S-R translation likely contributed significantly to sequence execution or not. Therefore, a separation of the two research questions of Study 2 existed given that no model could explain the additive effect in the Hand Switch/Tone Counting condition, but we could still determine which model is generally more likely than the other.

[4] In Study 1, a Hand configuration x Tone counting condition interaction was not found for participants who practiced with one hand including responses three to five, $F(1.19,13.08) = 1.78$, $p = 0.21$, $\eta_p^2 = 0.14$. However, the pattern of mean RTs was already closer to an interaction than the results shown in Figure 4. In Study 2, reducing individual differences mainly through standardization, but also through cutting off the right tail of the RT distributions (reported in Section 7.1), directed the pattern of mean standardized RTs towards an interactive effect in the Hand Switch/Tone Counting condition, $F(1,11) = 6.33$, $p = 0.02$, $\eta_p^2 = 0.36$. Due to a missing value for one participant where a target tone was played on the third key position of the sequence, we did not differentiate between distractor and target tones for the analysis including the standardized RTs leading to a Hand configuration (2: unpracticed vs practiced) x Tone counting condition (2: tone counting vs no counting) x Key (3: 3-5) repeated measures ANOVA where all three variables were within-subjects variables.

7. Methods

7.1 Data preparation

The statistical programming language R was used to conduct the RT simulations, using as a basis the RT distributions of the four test conditions from Study 1. An initial modeling approach included the data of all participants across all conditions similar to Figure 4 (Appendix B). This modeling approach showed that the biomechanical slowing of using fingers of a single hand as the unpracticed hand configuration most likely slowed the simulated central, CS, and SR processors so much that they could almost never win the race against the simulated motor processor. As one processor almost always won the race, it basically did not matter whether two or three processors were racing in total.

Hence, we decided to reduce biomechanical differences in the data from Study 1 by excluding sequence execution with one hand as the unpracticed hand configuration. This led to a within-subjects approach and a between-subjects approach which were based on different RTs for the Control and Tone Counting conditions while the used RTs for Hand Switch and Hand Switch/Tone Counting conditions were the same for both approaches. The between-subjects data included RTs of the Tone Counting and Control conditions from participants who practiced with two hands and RTs of the Hand Switch and Hand Switch/Tone Counting conditions from participants who practiced with one hand. Hence, the between-subjects approach only included data of sequence execution with two hands. On the other hand, the within-subjects data included RTs of participants, who practiced with one hand, across all test conditions.

Furthermore, we did not include all observed RTs of the chosen dataset for each modeling approach. As counting a tone affected only the three responses following it, we only considered responses three to five of each sequence (Verwey et al., 2010). Additionally, responses faster than 50 ms were removed because they were considered unrealistically fast. We also did not differentiate between distractor and target tones for the sake of simplicity.

Next, as RT distributions usually have an exponential upper tail making them ex-Gaussian rather than Gaussian (Galloway-Long & Huang-Pollock, 2018), we cut off the right tails of the RT distributions in each condition enabling us to use Gaussian distributions instead.[5] Hence, we only had to set the means and standard deviations of the simulated processor distributions rather than using three parameters of ex-Gaussian distributions. Additionally, reducing the right skew of the distributions had the advantage that the mean RTs were less skewed by the right tail as well. The removal of the long right tails was done manually for the RT distributions of each test condition. There was no strict rule used for choosing the individual cut-off points for each RT distribution, as the RT distributions of the four conditions differed in several ways not making it possible to choose a reasonable rule that would fit all distributions across both modeling approaches.

Another possibly skewing factor that we tried to reduce in Study 2 were individual differences between participants. However, this was not possible for the between-subjects approach, as standardizing the RTs of participants who practiced with two hands would have indirectly reflected the biomechanical slowing observed when these participants used one hand as the unpracticed hand configuration. Hence, no further changes were made to the between-subjects data after the removal of the long right tails, so we only reduced individual differences in the within-subjects data. Before standardizing the RTs of the within-subjects approach, we further reduced the right tail of the distributions on an individual participant level. This second cut-off operation was necessary, so the standardization process reducing individual differences would not lead to the occurrence of a new right tail. We divided the RTs of each participant into the four test conditions and cut off all data points that exceeded the mean of a participant's RTs in the respective condition by 2.5 standard deviations.

Then, the standardization process followed in which we standardized the RTs per participant across all four conditions. This was done through the method of z-score normalization which resulted in a mean of 0 and a standard deviation of 1 for each participant across all four test conditions.

Afterwards, the data of each participant was put back into one dataset including all participants which was also done for the between-subjects data where no standardization occurred. Finally, we split up the remaining (standardized) RTs into the four test conditions (Appendices C & D) as basis for determining the distributions of the processors of Models 2P and 3P. The mean standardized RTs of the within-subjects approach suggested an interactive effect in the Hand Switch/Tone Counting condition like expected in Study 1 whereas the

[5] It was seen as unproblematic to use Gaussian distributions, as we assumed that most of the right tail was caused by participants not concentrating or being distracted leading to a certain number of slow responses for each condition.

mean RTs of the between-subjects approach suggested an additive effect like observed in Study 1.

7.2 Simulation of processor distributions for Model 3P

The rnorm function, which is a built-in function of R, was used to produce 10000 random samples for each processor after mean and standard deviation had been set. The basic procedure for both modeling approaches was almost exactly the same. We started with the distribution of the SR processor, as it was the only processor assumed to be involved in the Hand Switch/Tone Counting condition. The mean and standard deviation were extracted directly from the (standardized) RT distribution of the Hand Switch/Tone Counting condition.

Then, we continued with the Hand Switch condition using a slightly different procedure, as the CS processor distribution could not be directly extracted from the (standardized) RT distribution of the Hand Switch condition. Model 3P assumed a race of the CS and SR processors, so the end product of that race had to match the (standardized) RT mean of the Hand Switch condition. Initially, the distribution of the CS processor was given two basically random parameters, as we did not know how its distribution looked like yet. But as the distribution of the SR processor was already known through the (standardized) RT distribution of the Hand Switch/Tone Counting condition, we could determine the distribution of the CS processor by tweaking its parameters until the simulated race of SR and CS processors yielded the (standardized) RT mean and standard deviation of the Hand Switch condition. This was an indirect way of determining the unknown distribution of the CS processor through testing whether the simulated processor fulfilled the required condition that the (standard) simRT mean resulting from a race with the already known SR processor reproduces the (standardized) RT mean of the Hand Switch condition.

Next, the distribution of the MP was determined by having it race with the simulated SR processor until the (standard) simRT mean resulting from that race matched the (standardized) RT mean of the Tone Counting condition. There was no difference between the approach of simulating the MP and CS processor except that they were based on data of different conditions. Appendix C shows the distributions of all three simulated processors for the within-subjects approach while Appendix D shows the three processor distributions for the between-subjects approach.

Lastly, Appendix E presents the R-code that was used to carry out the races between the simulated processors. Basically, from the list of 10000 random samples of each processor distribution, one value was randomly picked and competed against the random value of another processor distribution. The winner of the race was the processor with the shortest simulated processing time of two or three processors. This procedure was repeated until all 10000 values of a processor had competed against the value(s) of the other processor(s), and the resulting (standard) simRT distribution was determined.

7.3 Comparison of means

For the Control condition, in which sequences were executed as practiced, no further changes were made to the individual processor distributions of Model 3P. Here, the race including all three processors was carried out and compared to the (standardized) RT distribution of the Control condition. After that, the (standard) simRT mean of each condition was compared to the (standardized) RT mean of each condition and it was possible to say whether Model 3P was able to replicate the pattern of results found in Study 1 in terms of mean (standardized) RTs. The model was considered successful in replicating the pattern of results if each (standard) simRT mean fell into the range of a 95 percent confidence interval for the mean (standardized) RT of each condition.

Next, the motor and central processors of Model 2P were simulated based directly on the mean and standard deviation of the (standardized) RTs from the Tone Counting and Hand Switch conditions, respectively. We did not simulate the switching central processor for the Hand Switch/Tone Counting condition. Then, both simulated processors raced in the same way that the processors of Model 3P raced and the resulting (standard) simRT mean was compared to the (standardized) RT mean of the Control condition to see whether Model 2P could replicate the pattern of results in Study 1 in terms of mean (standardized) RTs.

7.4 Goodness-of-fit

While the mean was the main criterion to determine the accuracy of the simulated processor distributions, we also compared the resulting (standard) simRT distributions of Model 3P with the (standardized) RT distributions of the test conditions as a goodness-of-fit criterion. However, the (standardized) RT distributions were still not Gaussian after cutting off the right tail, so a more typical goodness-of-fit test like the two-sample Kolmogorov-Smirnov test (Smirnov, 1939) basically always indicated that the simulated Gaussian distribution and standardized RT distribution were significantly different.

As the two-sample Kolmogorov-Smirnov test was identified as being too sensitive for our skewed data, we chose to use the location-scale Cucconi test which can be used for two-sample location-scale problems including skewed distributions (Marozzi, 2013). The Cucconi test is a rather unknown nonparametric test that can determine whether the location and scale of two distributions are equal (Marozzi, 2009). We used the Cucconi test to determine whether the location and scale of the resulting (standard) simRT distributions of Model 3P differed from the (standardized) RT distributions of each test condition. Additionally, the goodness-of-fit for each condition was determined graphically with the expectation that the (standard) sim RT distribution should roughly fit the (standardized) RT distribution it was compared to.

8. Results

8.1 Interaction: Standardized RTs of within-subjects approach

8.1.1 Data preparation

The within-subjects data included responses three to five from participants who practiced with one hand and hence, used one hand for sequence execution in Control and Tone Counting conditions while using two hands in Hand Switch and Hand Switch/Tone Counting conditions. From the initial dataset of participants who practiced with one hand, a total of 11.2 percent of data points were removed across all four conditions. Excluding all RTs below the threshold of 50 ms led to the removal of 1.1 percent of data points across all conditions. The first cut-off operation increased this percentage of removed data points across all conditions to 10.1 percent and the second cut-off operation (using the exclusion criterion of data points that exceed the mean of a participant's condition by 2.5 standard deviations) increased the percentage to 11.2 percent.

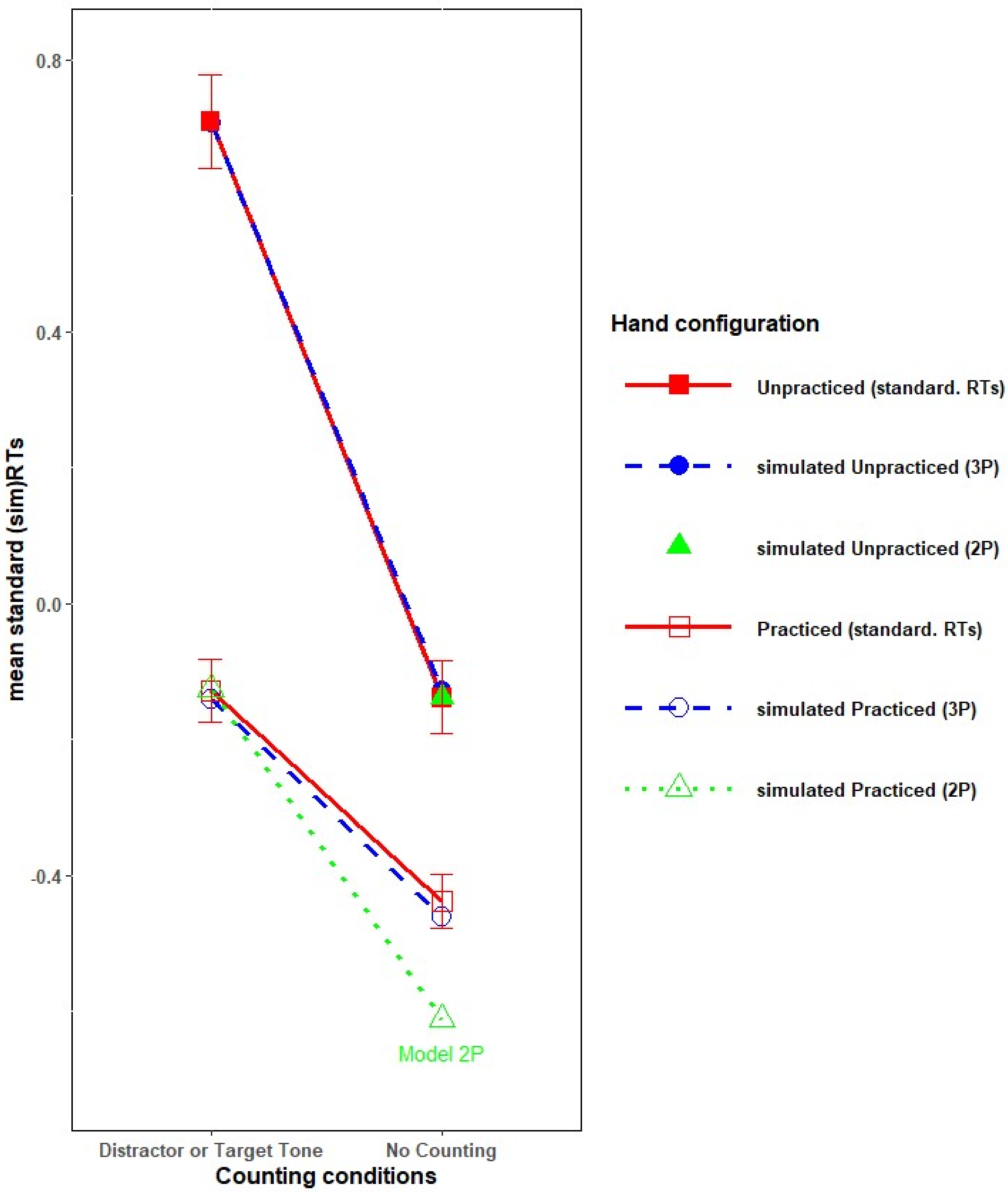
Next, all three measures combined had the following impacts on each individual condition. The reported upper thresholds are those used for the first cut-off operation. For the Hand Switch/Tone Counting condition, all RTs above the threshold of 520 ms were removed with 11.5 percent of data points excluded in total for this condition. For the Hand Switch condition, all RTs above the threshold of 430 ms were removed with 12.1 percent of data points excluded in total for this condition. For the Tone Counting condition, all RTs above the threshold of 400 ms were removed with 11.1 percent of data points excluded in total for this condition. For the Control condition, all RTs above the threshold of 330 ms were removed with 10.3 percent of data points excluded in total for this condition.

8.1.2 Comparison of means

After the SR processor distribution (M = 0.7, SD = 1.2), CS processor distribution (M = 0.3, SD = 1.0), and MP distribution (M = 0.2, SD = 0.8) had been determined and races between the processors were simulated, we compared the resulting pattern of standard simRT means to the pattern of mean standardized RTs from the test phase of Study 1. Figure 5 shows the mean standardized RTs for each test condition in comparison to the standard simRT means of Models 2P and 3P. The standard simRT means of Model 3P were all in close proximity to those found in the test phase of Study 1 and replicated the pattern of mean standardized RTs. On the other hand, Model 2P was not able to replicate the mean standardized RT of the Control condition through the simulated race of motor and central processors. Model 2P produced a different pattern of results where the standard simRT mean for the Control condition was significantly lower than observed suggesting an additive rather than the interactive effect that was found after standardizing the data of participants who practiced with one hand.

Figure 5

Mean standard simRTs of Models 3P and 2P as well as mean standardized RTs from Study 1 including participants who practiced with one hand with error bars representing 95 percent confidence intervals. Separated by the use of unpracticed and practiced hand configurations in addition to whether tones had to be counted or not. (<u>Conditions</u> (initials used): Tone/Unpracticed: HS/TC, No Counting/Unpracticed: HS, Tone/Practiced: TC, No Counting/Practiced: CC; <u>Models 2P and 3P</u>: Processors and races simulated for each condition according to Table 1 with the exclusion of the HS/TC condition for Model 2P)



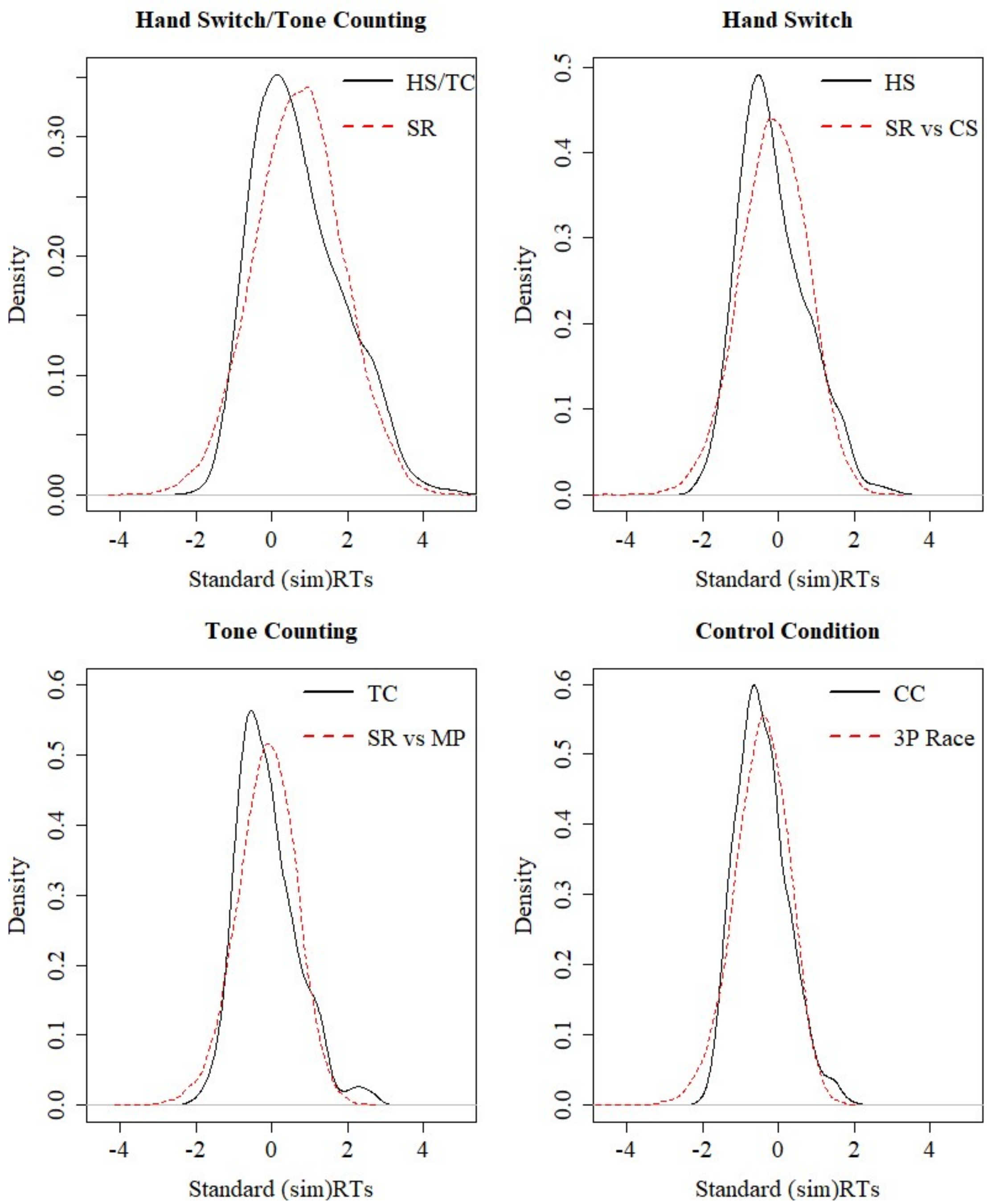
8.1.3 Goodness-of-fit

Figure 6 shows the comparison of distributions for each condition of the test phase between standardized RTs and standard simRTs of Model 3P. The resulting standard simRT distributions of Model 3P seemed to roughly fit the standardized RT distributions found in the test phase of Study 1 for participants who practiced with one hand.[6] The Cucconi test affirmed that by showing that location and scale of the resulting standard simRT distributions for Model 3P seen in Figure 6 were equal to location and scale of the standardized RT distribution of the Hand Switch/Tone Counting condition, $C = 3.6$, $p = 0.4$, the standardized RT distribution of the Hand Switch condition, $C = 4.5$, $p = 0.3$, the standardized RT distribution of the Tone Counting condition, $C = 3.4$, $p = 0.7$, and the standardized RT distribution of the Control condition, $C = 4.8$, $p = 0.3$.

[6] A left tail can be seen throughout all distributions resulting from a race of processors, as a number of low values of the SR and CS processors (Appendix C) almost always won the race against the values of the MP and each other's higher values. Hence, the rest of each simulated distribution is shifted a bit to the right to compensate for the low values that were not present in the standardized RTs of the four test conditions.

Figure 6

Density plots of the four test conditions for the standardized RTs of Study 1 including participants who practiced with one hand (black line) and the standard simRT distributions of Model 3P (red/dashed line) with the included processors as well as races between those processors ('vs') indicated in the legend.



8.2 Additive effect: RTs (without standardization) of between-subjects approach

8.2.1 Data preparation

The between-subjects data included responses three to five of the Tone Counting and Control conditions from participants who practiced with two hands and responses three to five of the Hand Switch and Hand Switch/Tone Counting conditions from participants who practiced with one hand. From the initial dataset including sequence execution using two hands across all conditions, a total of 14.3 percent of data points were removed across all four conditions. Excluding all RTs below the threshold of 50 ms led to the removal of 1.3 percent of data points across all conditions. Cutting off the long right tails increased this percentage of removed data points across all conditions to 14.3 percent.

Participants who practiced with one hand and used their unpracticed hand configuration

Excluding extremely slow and fast responses had the following impacts on Hand Switch and Hand Switch/Tone Counting conditions. The reported upper thresholds are those used for cutting off the right tail. For the Hand Switch/Tone Counting condition, all RTs above the threshold of 450 ms were removed with 15.6 percent of data points excluded in total for this condition. For the Hand Switch condition, all RTs above the threshold of 380 ms were removed with 14.7 percent of data points excluded in total for this condition.

Participants who practiced with two hands and used their practiced hand configuration

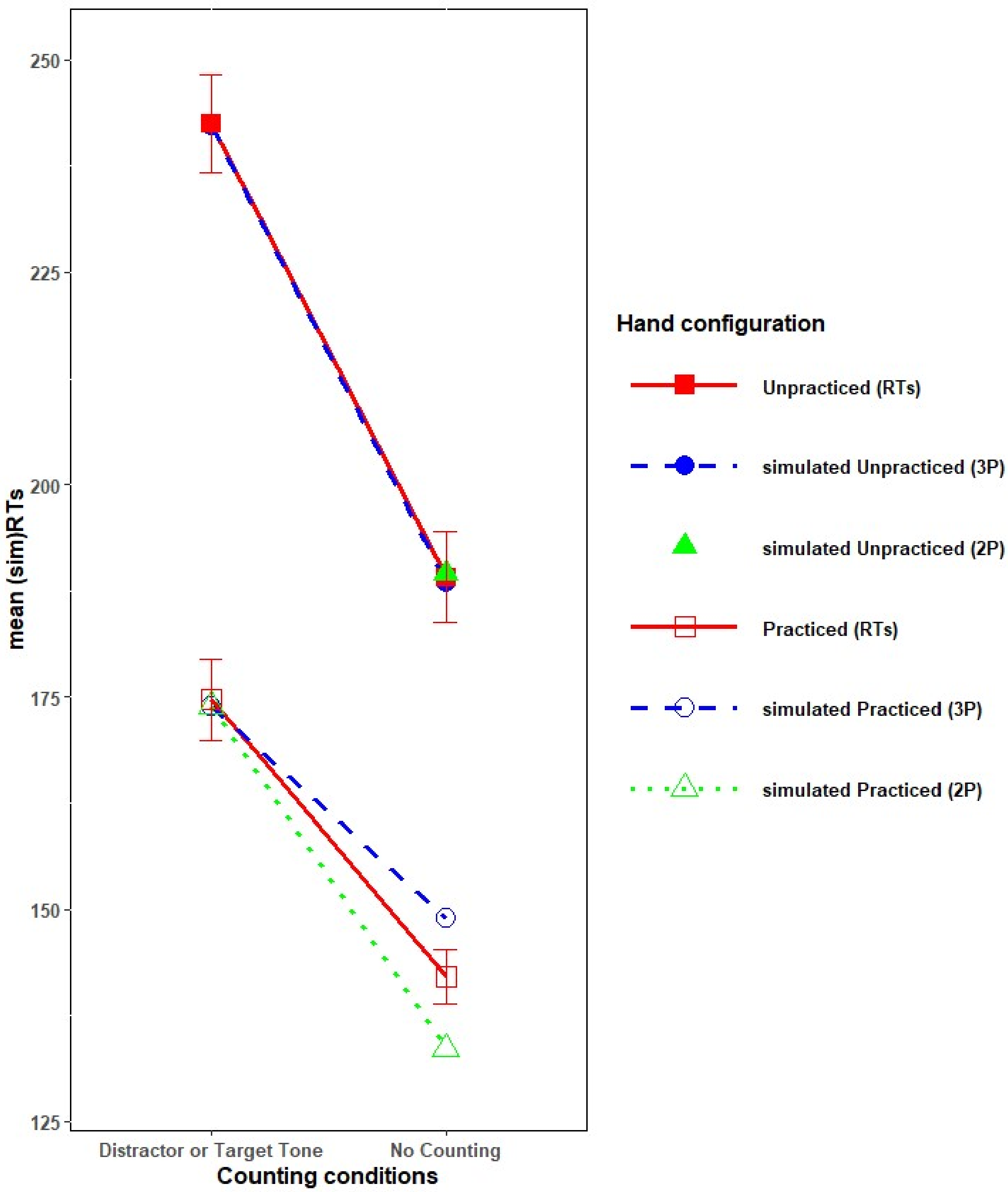
Excluding extremely slow and fast responses had the following impacts on Tone Counting and Control conditions. The reported upper thresholds are those used for cutting off the right tail. For the Tone Counting condition, all RTs above the threshold of 370 ms were removed with 17.3 percent of data points excluded in total for this condition. For the Control condition, all RTs above the threshold of 300 ms were removed with 9.6 percent of data points excluded in total for this condition.

8.2.2 Comparison of means

After the SR processor distribution ($M = 242.2$ ms, $SD = 96.4$ ms), CS processor distribution ($M = 259.7$ ms, $SD = 124.2$ ms), and MP distribution ($M = 213.9$ ms, $SD = 92.1$ ms) had been determined and races between the processors were simulated, we could compare the resulting pattern of mean simRTs to the pattern of mean RTs from the test phase of Study 1. Figure 7 shows the mean RTs for each test condition from purely two-handed sequence execution in comparison to the simRT means of Models 2P and 3P. Neither Model 2P nor Model 3P could replicate the pattern of mean RTs seen in Figure 7, as the simRT means for the Control condition differed from the observed RT mean of the Control condition. Moreover, the result of the simulated race of motor and central processors for Model 2P (M = 133.6 ms) had roughly the same distance to the RT mean of the Control condition (M = 142.1 ms) as Model 3P (M = 148.9 ms).

Figure 7

Mean simRTs of Models 2P and 3P and mean RTs (ms) from Study 1 including RTs from purely two-handed sequence execution with error bars representing 95 percent confidence intervals. Separated by the use of unpracticed and practiced hand configurations in addition to whether tones had to be counted or not. (<u>Conditions</u> (initials used): Tone/Unpracticed: HS/TC, No Counting/Unpracticed: HS, Tone/Practiced: TC, No Counting/Practiced: CC; <u>Models 2P and 3P</u>: Processors and races simulated for each condition according to Table 1 with the exclusion of the HS/TC condition for Model 2P)



8.2.3 Goodness-of-fit

The Cucconi test showed that location and scale of the resulting simRT distributions for Model 3P were equal to location and scale of the RT distribution of the Hand Switch/Tone Counting condition, $C = 9.1$, $p = 0.5$, the RT distribution of the Hand Switch condition, $C = 18.6$, $p = 0.9$, and the RT distribution of the Tone Counting condition, $C = 4.8$, $p = 0.6$. However, location and scale of the resulting simRT distribution from a race of three processors were different than location and scale of the RT distribution of the Control condition, $C = 101.4$, $p < 0.001$. A graphical evaluation of the goodness-of-fit can be found in Appendix D.

Further modeling approach

A further modeling approach can be found in Appendix B. This initial approach included the mean standardized RTs of all participants across all conditions similar to Figure 4 in Study 1. Neither model was able to replicate the pattern of results in terms of mean standardized RTs and produced standard simRT means for the Control condition that were significantly higher than the standardized RT mean. The main cause for the results was probably the biomechanical slowing that was still included for this approach and slowed all simulated processors except the motor processor of each model. As a result of that, both models produced a seemingly interactive effect for the Hand Switch/Tone Counting condition. The results of this initial approach led to the part approaches reported above.

9. Discussion

The purpose of Study 2 was to investigate (1) whether it is likely that S-R translation significantly contributed to sequence execution in the test phase of Study 1 and (2) whether significant contribution of S-R translation to sequence execution can explain the additive effect of tone counting and using an unpracticed hand configuration in the Hand Switch/Tone Counting condition. We aimed to replicate the results of Study 1 in terms of mean (standardized) RTs by simulating the distributions of the assumed processors of two different models as well as simulating races between these processors. In two modeling approaches, the 3P model including the MP, SR, and CS processors was compared to the 2P model only including central and motor processors to determine which model was better able to predict the results of each modeling approach.

The within-subjects modeling approach showed an interactive effect in the Hand Switch/Tone Counting condition after individual and biomechanical differences were reduced. Model 3P was able to replicate the pattern of results in terms of mean standardized RTs whereas Model 2P failed to do so (Figure 5). Additionally, the standard simRT distributions of Model 3P roughly fitted the standardized RT distributions suggesting that the processors and races were simulated in a roughly accurate way (Figure 6). In addition to these findings, Verwey et al. (2020) found that key-specific stimuli cannot be ignored when a luminance change occurred making a three-processor model including the SR processor more likely than a two-processor model. However, this does not necessarily mean that Model 3P can also explain the additive effect observed in the Hand Switch/Tone Counting condition, as Model 3P replicated an interactive effect for the within-subjects approach.

The between-subjects approach, where only biomechanical differences were greatly reduced, revealed that neither Model 3P nor Model 2P could replicate the pattern of mean RTs suggesting an additive effect in the Hand Switch/Tone Counting condition (Figure 7). Hence, although it is likely that S-R translation significantly contributed to sequence execution in the test phase of Study 1, significant contribution of S-R translation to sequence execution cannot explain the additive effect in the Hand Switch/Tone Counting condition. Moreover, the pattern of mean simRTs that Model 3P produced, seemed to suggest an interactive effect in the Hand Switch/Tone Counting condition similar to the within-subjects approach. Consequently, despite significant contribution of the SR processor in all four test conditions, the three-processor model would have still expected an interaction of tone counting and using an unpracticed hand configuration in Study 1. This, of course, begs several questions that will be discussed in the following sections.

10. General discussion

The present study was based on the cognitive framework for sequential motor behavior (C-SMB) and its underlying assumption of multiple processors that race in parallel to trigger the next response of a sequence of movements (Verwey et al., 2015). We investigated to what extent responses are slowed after extensive practice when two processors are presumably eliminated from the race of processors (Study 1). Additionally, we tried to determine whether two or three processors were likely participating in the race by comparing two models that differed in terms of whether stimulus-response translation contributed significantly to sequence execution or not (Study 2).

Study 1 used the DSP task where, after extensive practice with a particular hand configuration, participants had to count tones, use an unpracticed hand configuration, do both of those things at the same time, or execute sequences as practiced leading to four different conditions in the test phase. We assumed that tone counting would eliminate a processor related to central processing and found that tone counting did slow responses. The use of an unpracticed hand configuration was assumed to eliminate the motor processor from the race, and we observed that responses were indeed slowed relative to using the practiced hand configuration. However, the main result of Study 1 was that the increase in RTs was additive rather than interactive when tone counting and the use of an unpracticed hand configuration were combined.

As this contradicted the predictions of the two-processor model including the motor and central processor, we assumed that the contribution of reacting to key-specific stimuli through the stimulus-response translation (SR) processor was still relevant for sequence execution and might have been responsible for the missing interaction of tone counting and using an unpracticed hand configuration in the test phase of Study 1. Study 2 compared the two-processor model from Study 1 (Model 2P) with a three-processor model (Model 3P) including the central-symbolic (CS), motor, and SR processors. We simulated the processors of both models as well as races between the processors to model the (standardized) RT distributions of the four conditions in Study 1. The results suggested that a three-processor model is more likely than the two-processor model assumed in Study 1, as, in contrast to Model 2P, Model 3P could replicate the mean standardized RTs of one half of participants where individual and biomechanical differences were reduced (Figure 5). However, the three-processor model still produced a pattern of results for the four conditions that suggested an interaction while aiming to model an additive effect (Figure 7). Hence, the contribution of the SR processor alone could not explain the additive effect observed in the test phase of Study 1.

10.1 Implications

Overall, the results of both studies partially support the race assumption of the C-SMB that eliminating a processor from the race to trigger the next response slows responses in the DSP task. Even though no model could fully explain the additive effect observed in Study 1 on its own, it is more likely that three processors contributed to sequence execution in the test phase rather than two processors. Study 2 showed that an increased slowing of responses should still be expected given that two processors are eliminated from the race compared to just one processor. However, it seems like this theoretical assumption should be tested in an experimental setting where the effects of still unknown factors unrelated to the cognitive model are reduced.

One such factor was the biomechanical slowing due to the use of fingers of one hand as the unpracticed hand configuration, whose effect we reduced in Study 2. The biomechanical slowing might have been one cause for the additive effect in Study 1 by possibly slowing responses so much that tone counting did not further slow responses as much as expected (Appendix B). On the other hand, as Verwey et al. (2016) have stated, it could also be the case that adjusting to the biomechanical properties of an unpracticed hand configuration was the actual main cause of slowed responses in Study 1 instead of the use of motor chunks being inhibited by using an unpracticed hand configuration. This would imply that the use of an unpracticed hand configuration never eliminated the motor processor from the race. In that case, an alternative manipulation would be required, so the effect of eliminating the motor processor from the race can be studied in future research. As the findings of Sobierajewicz et al. (2017) also raise doubts as to whether motor learning is effector-dependent and hence, affected by a change of effectors, this should be further investigated.

Next, reducing individual differences through standardization for one half of the participants led to a pattern of mean standardized RTs showing an interaction. This interaction was replicable by Model 3P (Figure 5). On the other hand, the RTs for purely two-handed sequence execution could not be standardized without indirectly reflecting the biomechanical slowing. Hence, individual differences were still high which can also be seen in the RT distributions in Appendix D. As the non-standardized RTs for purely two-handed sequence execution showed a pattern of mean RTs suggesting an additive effect while an interaction was observed for the standardized mean RTs, we assumed that individual differences between participants might have been partly responsible for the missing interaction in Study 1. A logical conclusion would be that with greatly reduced biomechanical and individual differences, an interaction of counting tones and using an unpracticed hand configuration should be expected rather than the observed additive effect in Study 1. However, this assumption requires further testing to confirm, so possible alternative explanations that might also account for the results of Study 1 can be excluded.

10.2 Future research

To test whether the theoretical findings of Study 2 hold true in an experimental setting, a future study could investigate whether a repetition of Study 1 produces the expected interaction with the following adjustments. Verwey et al. (2020) found that key-specific stimuli cannot be ignored when the luminance of the stimuli changes, so a future study could repeat Study 1 using isoluminant stimuli which can be ignored and hence, might allow a test of the two-processor model (Verwey, 2021). Additionally, it is recommended that if two different hand configurations are being used with the purpose of eliminating the motor processor from the race, both hand configurations should include fingers from two hands to reduce the effect of the biomechanical slowing. Furthermore, if a tone counting task is being used again as the secondary task, the tones should occur at differing positions within a keying sequence to reduce the predictability of tones. In contrast to previous studies (Verwey et al., 2010; Verwey et al., 2014), Study 1 had tones occur at the same position within a keying sequence which might have decreased the slowing effect of tone counting due to the predictability of tones.

Lastly, while Study 2 incorporated the effect of the SR processor, another aspect of movement sequence learning, namely associative learning, was still largely ignored. Associative learning is assumed to affect sequence execution at all processing levels and is independent from motor chunk learning (Verwey & Wright, 2014). Although Model 3P theoretically incorporated the possible effect of associative learning despite not differentiating it from the other processors, it could be that the addition of an associative processor to the model would change the simulated results in a significant way. [7] Hence, it is recommended to also consider the effect of associative learning on sequence execution in the DSP task.

[7] We assume that the Associative Processor (AP) influences sequence execution at all processing levels, so it would contribute to sequence execution in each test condition from Study 1 similar to the SR processor. If that assumption is not true, the addition of an AP to the model could change the simulated results in a significant way. Otherwise, Model 3P theoretically incorporated the possible effect of associative learning because the relation of processors racing in each condition does not change with a four-processor model. There may be more processors, but the outcome of those processors racing will be the same compared to the processors of Model 3P. The reason behind this is that we assume the simulated results to match the observed RT mean in each condition. Hence, adding processors means that the processors have to be slower on their own to reproduce the same RT mean that a model with less processors can produce. A simple example that might help to understand this logic would be look at Table 1 and remove the SR processor of Model 3P from the TC condition. Now, the MP of both models would be identical. Although there might still be three processors racing in the Control condition, Model 3P and 2P would be almost identical in terms of their simulated results (excluding the HS/TC condition).

10.3 Limitations

The previous two sections have indirectly stated limitations of Study 1 that were in part revealed by Study 2, so the focus in this section will be on Study 2 itself. Study 2 was a post-hoc simulation study which modeled theoretical processor distributions and races between processors based on (standardized) RTs of the conditions from the test phase of Study 1. As we fitted the processors and races of processors to three of the four conditions by adjusting the processor distributions until mean (standard) simRT and mean (standardized) RT of the respective condition were equal, there was only one condition left, namely the control condition, where differences between mean (standard) simRT and mean (standardized) RT were actually possible. This limited the power of the modeling approach, because, for example, it is generally possible that a model including more than three processors produces the same result as a three-processor model when the modeling approach of Study 2 is used and there is only one condition allowing differences between simulated and observed results (Footnote 7). Therefore, we cannot eliminate the possibility that even more processors like the associative processor were involved in sequence execution.

A further issue might have been that we manually cut off the right tails of the RT distributions to transform ex-Gaussian distributions into Gaussian distributions. There was no reason for us to believe that this was problematic for the within-subjects approach where standardization of RTs without cutting off the tails seemed to produce a relatively similar pattern of results compared to the results shown in Figure 5. However, the pattern of mean RTs of the four conditions for the between-subjects approach did relatively change due to cutting off the long right tails. Hence, differently chosen cut-off points might have produced a different pattern of results in favor of one or the other model limiting the between-subjects approach in general.

Lastly, we do not truly know whether it was unproblematic to mix between-subjects data and treat it like within-subjects data. It could be that one half of participants was generally faster than the other half regardless of the used hand configuration which would mean that between-subjects differences in addition to individual differences skewed the modeling approach.

10.4 Conclusion

The present study affirmed the slowing effects of tone counting and using an unpracticed hand configuration for sequence execution in the DSP task after extensive practice. However, new evidence supporting the race assumptions of the C-SMB that parallel processors race to trigger each next movement could not be found in Study 1. While Study 1 initially assumed that the contribution of stimulus-response translation to sequence execution would be small, Study 2 found that stimulus-response translation most likely still contributed to sequence execution in Study 1 in a significant way. This is also in line with recent evidence stating that key-specific stimuli cannot be ignored when a luminance change occurs. Our modeling approach in Study 2 has shown that the elimination of two processors from the race to trigger the next response should theoretically still lead to an increased slowing of responses relative to the slowing when one processor is eliminated. This was not observed in Study 1 when the use of tone counting and an unpracticed hand configuration were combined and each manipulation was assumed to eliminate a processor, namely the motor processor or a processor related to central processing, from the race. Our findings have also shown that researchers should be careful regarding biomechanical and individual differences when testing the cognitive assumptions of the race model in the future, as these differences might have skewed our results to not show the expected results. Hence, future research should aim to find the expected results of the model given that these external factors are not affecting reaction times, so the race assumptions of the C-SMB including the motor, central-symbolic, stimulus-response translation, and possibly also associative processors can be affirmed.

References

Abrahamse, E. L., Ruitenberg, M. F. L., de Kleine, E., & Verwey, W. B. (2013). Control of automated behavior: Insights from the discrete sequence production task. *Frontiers in Human Neuroscience*, *7*(82). https://doi.org/10.3389/fnhum.2013.00082

Brown, S. D., & Heathcote, A. (2008). The simplest complete model of choice response time: Linear ballistic accumulation. *Cognitive Psychology*, *57*(3), 153–178. https://doi.org/10.1016/j.cogpsych.2007.12.002

De Kleine, E., & Verwey, W. B. (2009). Representations underlying skill in the discrete sequence production task: Effect of hand used and hand position. *Psychological Research, 73*(5), 685–694. https://doi.org/10.1007/s00426-008-0174-2

E-Prime® Legacy Versions. (2018). Psychology Software Tools | Solutions for Research, Assessment, and Education. Retrieved February 22, 2022, from https://pstnet.com/products/e-prime-legacy-versions/

Galloway-Long, H. & Huang-Pollock, C. (2018). Using inspection time and ex-Gaussian parameters of reaction time to predict executive functions in children with ADHD. *Intelligence*, *69*, 186–194. https://doi.org/10.1016/j.intell.2018.06.005

Marozzi, M. (2009). Some notes on the location–scale Cucconi test. *Journal of Nonparametric Statistics, 21*(5), 629–647. https://doi.org/10.1080/10485250902952435

Marozzi, M. (2013). Nonparametric Simultaneous Tests for Location and Scale Testing: A Comparison of Several Methods. *Communications in Statistics - Simulation and Computation*, *42*(6), 1298–1317. https://doi.org/10.1080/03610918.2012.665546

Smirnov, N. V. (1939). Estimate of deviation between empirical distribution functions in two independent samples. *Bulletin Moscow University*, *2*(2), 3-16.

Sobierajewicz, J., Przekoracka-Krawczyk, A., Jaśkowski, W., & van der Lubbe, R. H. J. (2017). How effector-specific is the effect of sequence learning by motor execution and motor imagery? *Experimental Brain Research*, *235*(12), 3757–3769. https://doi.org/10.1007/s00221-017-5096-z

Verwey, W. B. (2021). Isoluminant stimuli in a familiar discrete keying sequence task can be ignored. *Psychological Research*, *85*(2), 793–807. https://doi.org/10.1007/s00426-019-01277-0

Verwey, W. B., Abrahamse, E. L., & de Kleine, E. (2010). Cognitive processing in new and practiced discrete keying sequences. *Frontiers in Psychology*, *1*(32).

https://doi.org/10.3389/fpsyg.2010.00032

Verwey, W. B., Abrahamse, E. L., de Kleine, E., & Ruitenberg, M. F. L. (2014). Evidence for graded central processing resources in a sequential movement task. *Psychological Research*, *78*, 70–83. https://doi.org/10.1007/s00426-013-0484-x

Verwey, W. B., & Clegg, B. A. (2005). Effector dependent sequence learning in the serial RT task. *Psychological Research, 69*(4), 242–251. https://doi.org/10.1007/s00426-004-0181-x

Verwey, W. B., Groen, E. C., & Wright, D. L. (2016). The stuff that motor chunks are made of: Spatial instead of motor representations? *Experimental Brain Research*, *234*, 353–366. https://doi.org/10.1007/s00221-015-4457-8

Verwey, W. B., Shea, C. H., & Wright, D. L. (2015). A cognitive framework for explaining serial processing and sequence execution strategies. *Psychonomic Bulletin & Review*, *22*(1), 54–77. https://doi.org/10.3758/s13423-014-0773-4

Verwey, W. B., & Wright, D. L. (2004). Effector-independent and effector-dependent learning in the discrete sequence production task. *Psychological Research*, *68*, 64–70. https://doi.org/10.1007/s00426-003-0144-7

Verwey, W. B., & Wright, D. L. (2014). Learning a keying sequence you never executed: Evidence for independent associative and motor chunk learning. *Acta Psychologica*, *151*, 24–31. https://doi.org/10.1016/j.actpsy.2014.05.017

Verwey, W. B., Wright, D. L. & van der Lubbe, R. H. (2020). The Simon effect in a discrete sequence production task: Key-specific stimuli cannot be ignored due to attentional capture. *Acta Psychologica*, *205*(103044). https://doi.org/10.1016/j.actpsy.2020.103044

Appendix A

Script SPSS for Study 1

#The first two Anovas can be used for both the reaction times and error proportions

Repeated measures Anova for the practice phase

```
GLM B1.1 B1.2 B1.3 B1.4 B1.5 B1.6 B1.7 B2.1 B2.2 B2.3 B2.4 B2.5 B2.6 B2.7 B3.1 B3.2
B3.3 B3.4 B3.5
  B3.6 B3.7 B4.1 B4.2 B4.3 B4.4 B4.5 B4.6 B4.7 B5.1 B5.2 B5.3 B5.4 B5.5 B5.6 B5.7 B6.1
B6.2 B6.3 B6.4
  B6.5 B6.6 B6.7 BY HandsUsedPract
 /WSFACTOR=Block 6 Polynomial Key 7 Polynomial
 /MEASURE=Reaction_Time
 /METHOD=SSTYPE(3)
 /PLOT=PROFILE(Block Key Block*Key Block*Key*HandsUsedPract
Block*HandsUsedPract Key*HandsUsedPract)
 /EMMEANS=TABLES(Block) COMPARE ADJ(BONFERRONI)
 /EMMEANS=TABLES(Key) COMPARE ADJ(BONFERRONI)
 /EMMEANS=TABLES(Block*Key)
 /PRINT=DESCRIPTIVE ETASQ
 /CRITERIA=ALPHA(.05)
 /WSDESIGN=Block Key Block*Key
 /DESIGN=HandsUsedPract.
```

Repeated measures Anova for the test phase

```
GLM DCD1 DCD2 DCD3 DCD4 DCD5 DCD6 DCD7 DCT1 DCT2 DCT3 DCT4 DCT5
DCT6 DCT7 DNC1 DNC2 DNC3 DNC4 DNC5
    DNC6 DNC7 SCD1 SCD2 SCD3 SCD4 SCD5 SCD6 SCD7 SCT1 SCT2 SCT3 SCT4
SCT5 SCT6 SCT7 SNC1 SNC2 SNC3 SNC4
    SNC5 SNC6 SNC7 BY HandsUsedPract
  /WSFACTOR=DiffSame 2 Polynomial Count 3 Polynomial Key 7 Polynomial
  /MEASURE=Reaction_Time
  /METHOD=SSTYPE(3)
  /PLOT=PROFILE(DiffSame*Key Key*DiffSame DiffSame*Key*HandsUsedPract Count
Key DiffSame HandsUsedPract Count*HandsUsedPract Key*HandsUsedPract
DiffSame*HandsUsedPract
    Count*DiffSame*HandsUsedPract DiffSame*Count DiffSame*Count*HandsUsedPract
Count*DiffSame Key*HandsUsedPract*DiffSame DiffSame*Count*HandsUsedPract
Count*Key)
  /EMMEANS=TABLES(Count) COMPARE ADJ(BONFERRONI)
  /EMMEANS=TABLES(Key) COMPARE ADJ(BONFERRONI)
  /EMMEANS=TABLES(DiffSame) COMPARE ADJ(BONFERRONI)
  /EMMEANS=TABLES(DiffSame*Count)
  /EMMEANS=TABLES(HandsUsedPract*DiffSame*Count)
  /PRINT=DESCRIPTIVE ETASQ
  /CRITERIA=ALPHA(.05)
  /WSDESIGN=DiffSame Count Key DiffSame*Count DiffSame*Key Count*Key
DiffSame*Count*Key
  /DESIGN=HandsUsedPract.
```

Repeated measures Anova for the test phase including only responses 3 to 5 (can also be used for the same Repeated measures Anova including only participants who practiced with one hand, but the between-subjects variable 'HandsUsedPract' has to be removed for that)

```
GLM DCD3 DCD4 DCD5 DCT3 DCT4 DCT5 DNC3 DNC4 DNC5 SCD3 SCD4 SCD5
SCT3 SCT4 SCT5
    SNC5 SNC3 SNC4 BY HandsUsedPract
  /WSFACTOR=DiffSame 2 Polynomial Count 3 Polynomial Key 3 Polynomial
  /METHOD=SSTYPE(3)
 /PLOT=PROFILE(DiffSame*Key Key*DiffSame DiffSame*Key*HandsUsedPract Count
Key DiffSame HandsUsedPract Count*HandsUsedPract Key*HandsUsedPract
DiffSame*HandsUsedPract
    Count*DiffSame*HandsUsedPract DiffSame*Count DiffSame*Count*HandsUsedPract
Count*DiffSame Key*HandsUsedPract*DiffSame DiffSame*Count*HandsUsedPract
Count*Key)
  /EMMEANS=TABLES(Count) COMPARE ADJ(BONFERRONI)
  /EMMEANS=TABLES(Key) COMPARE ADJ(BONFERRONI)
  /EMMEANS=TABLES(DiffSame) COMPARE ADJ(BONFERRONI)
  /EMMEANS=TABLES(DiffSame*Count)
  /EMMEANS=TABLES(HandsUsedPract*DiffSame*Count)
  /PRINT=DESCRIPTIVE ETASQ
  /CRITERIA=ALPHA(.05)
  /WSDESIGN=DiffSame Count Key DiffSame*Count DiffSame*Key Count*Key
DiffSame*Count*Key
  /DESIGN=HandsUsedPract.
```

Repeated measures Anova used for planned comparison separating keys before (3-5) vs keys after a tone (2,6,7)

```
GLM DCDB DCTB DNCB DCDA DCTA DNCA SCDB SCTB SNCB SCDA SCTA SNCA BY HandsUsedPract
  /WSFACTOR=DiffSame 2 Polynomial BefAft 2 Polynomial Tone 3 Polynomial
  /METHOD=SSTYPE(3)
  /PLOT=PROFILE(BefAft Tone BefAft*Tone Tone*BefAft)
  /EMMEANS=TABLES(BefAft) COMPARE ADJ(BONFERRONI)
  /EMMEANS=TABLES(Tone) COMPARE ADJ(BONFERRONI)
  /EMMEANS=TABLES(BefAft*Tone)
  /PRINT=ETASQ
  /CRITERIA=ALPHA(.05)
  /WSDESIGN=DiffSame BefAft Tone DiffSame*BefAft DiffSame*Tone BefAft*Tone DiffSame*BefAft*Tone
  /DESIGN=HandsUsedPract.
```

Repeated measures ANOVA for the mean standardized RTs of Study 2 including responses 3-5 of participants who practiced with one hand with no distinction between distractor and target tones (mean of both values was computed; see Footnote 2)

```
GLM DT3 DT4 DT5 DNC3 DNC4 DNC5 ST3 ST4 ST5 SNC3 SNC4 SNC5
  /WSFACTOR=DiffSame 2 Polynomial Count 2 Polynomial Key 3 Polynomial
  /METHOD=SSTYPE(3)
  /PLOT=PROFILE(DiffSame*Key Key*DiffSame Count Key DiffSame Key
```

```
   DiffSame*Count Count*DiffSame Count*Key)
  /EMMEANS=TABLES(Count) COMPARE ADJ(BONFERRONI)
  /EMMEANS=TABLES(Key) COMPARE ADJ(BONFERRONI)
  /EMMEANS=TABLES(DiffSame) COMPARE ADJ(BONFERRONI)
  /EMMEANS=TABLES(DiffSame*Count)
 /PRINT=DESCRIPTIVE ETASQ
  /CRITERIA=ALPHA(.05)
  /WSDESIGN=DiffSame Count Key DiffSame*Count DiffSame*Key Count*Key
DiffSame*Count*Key.
```

Appendix B

Modeling approach based on RTs of all participants across all conditions including responses three to five

The methodology did not change relative to Study 2 except for what was specific to the RTs of all participants across all conditions which is described below.

Data preparation

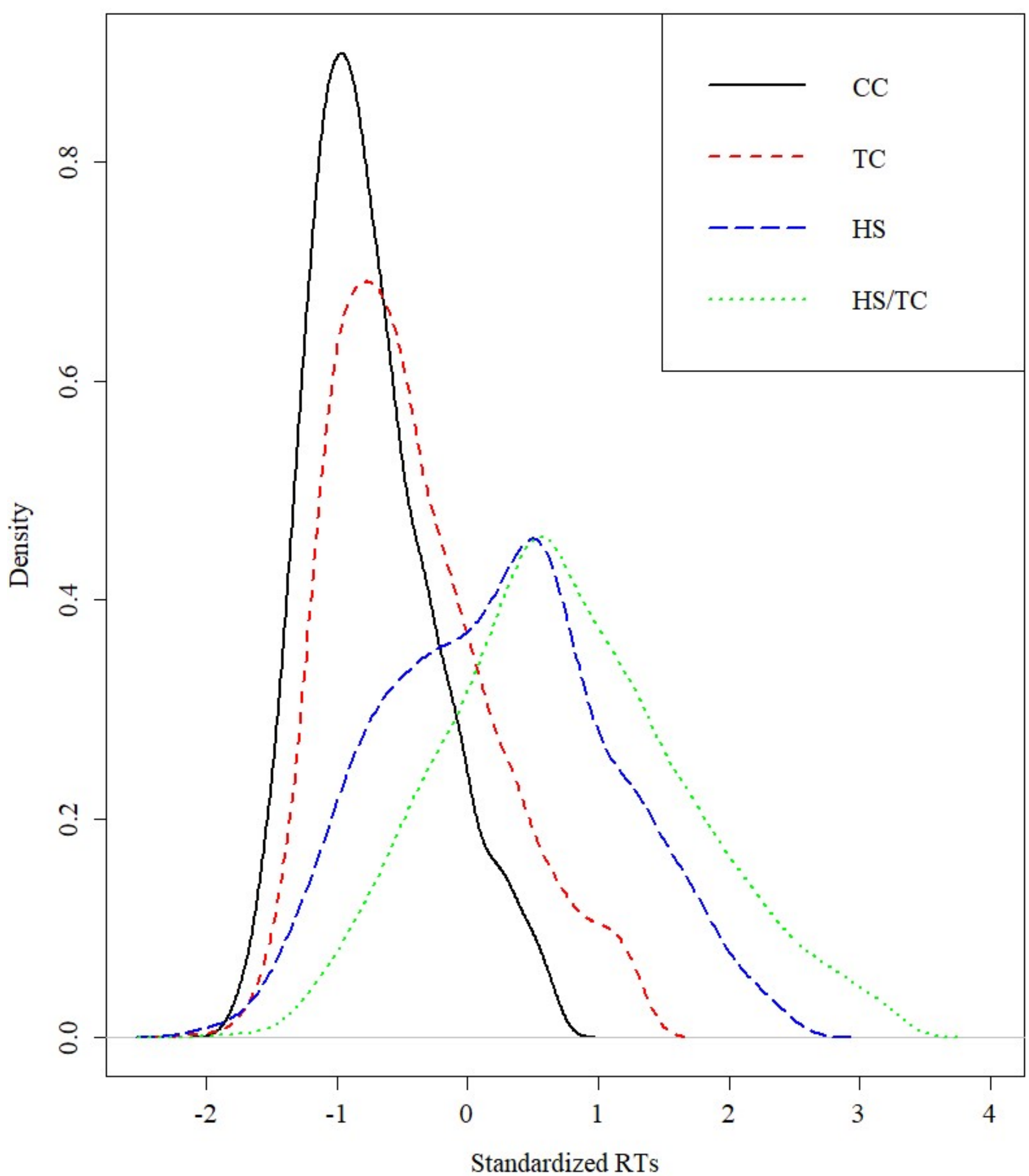
From the initial dataset of all participants across all conditions including responses three to five, a total of 10.6 percent of data points were removed. Excluding all RTs below the threshold of 50 ms led to the removal of 1.1 percent of data points across all conditions. The first cut-off operation increased this percentage of removed data points across all conditions to 7.4 percent and the second cut-off operation (using the exclusion criterion of data points that exceed the mean of a participant's condition by 2.5 standard deviations) increased the percentage to 9.1 percent. The remaining data was standardized in the same way as the RTs of the within-subjects approach except that all participants were included. After standardization, a third cut-off operation was used to cut off the remaining right tails of the distributions which were still quite long. This increased the percentage of removed data points across all conditions to 10.6 percent.

Next, all four measures combined had the following impacts on each individual condition. The reported upper thresholds are those used for the first and third cut-off operation with the removed RTs referring to the first operation and the removed standardized RTs referring to the third operation. For the Hand Switch/Tone Counting condition, all RTs above the threshold of 680 ms and all standardized RTs above the threshold of 3.3 were removed with 9.7 percent of data points excluded in total for this condition. For the Hand Switch condition, all RTs above the threshold of 620 ms and all standardized RTs above the threshold of 2.5 were removed with 9.6 percent of data points excluded in total for this condition. For the Tone Counting condition, all RTs above the threshold of 450 ms and all standardized RTs above the threshold of 1.4 were removed with 12.5 percent of data points excluded in total for this condition. For the Control condition, all RTs above the threshold of 360 ms and all standardized RTs above the threshold of 0.7 were removed with 10.8 percent of data points excluded in total for this condition.

Standardized RT distributions

Figure B1

Density plot showing the standardized distributions of the obtained RTs for each of the four conditions of the test phase in Study 1 from all participants across all conditions including responses three to five. (CC = Control condition, TC = Tone Counting condition, HS = Hand Switch condition, HS/TC = Hand Switch/Tone Counting condition).

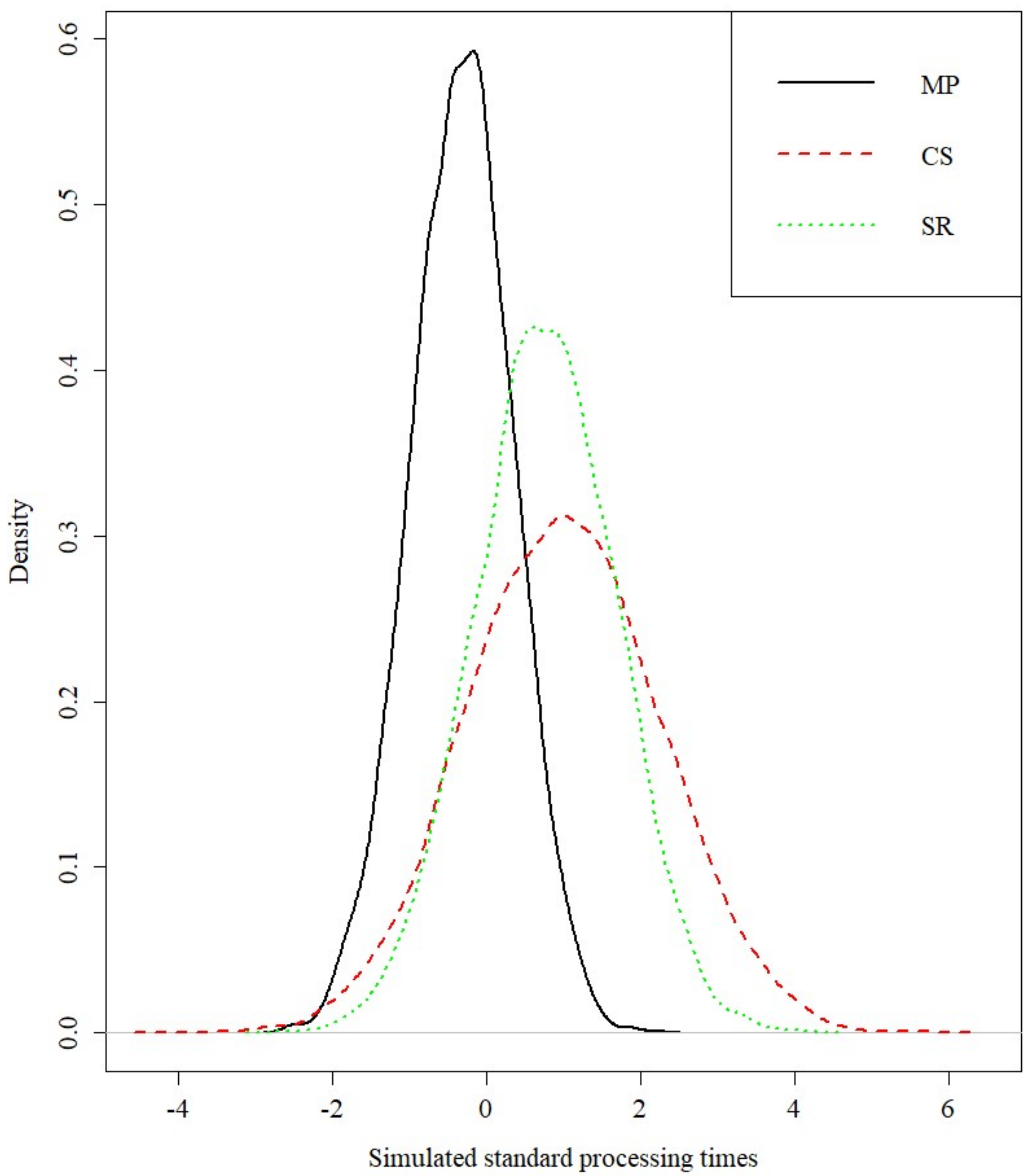


Simulation of processor distributions

SR processor distribution (M = 0.8, SD = 0.9), CS processor distribution (M = 1.0, SD = 1.3), and MP distribution (M = -0.3, SD = 0.7) had been determined through the same basic way as for the two other approaches by using the standardized RT distributions of each condition as basis.

Figure B2

Density plot showing the three processor distributions for Model 3P based (in)directly on the standardized RT distributions from Figure B1 (MP = Motor processor, CS = Central-symbolic processor, SR = Stimulus-response translation processor).

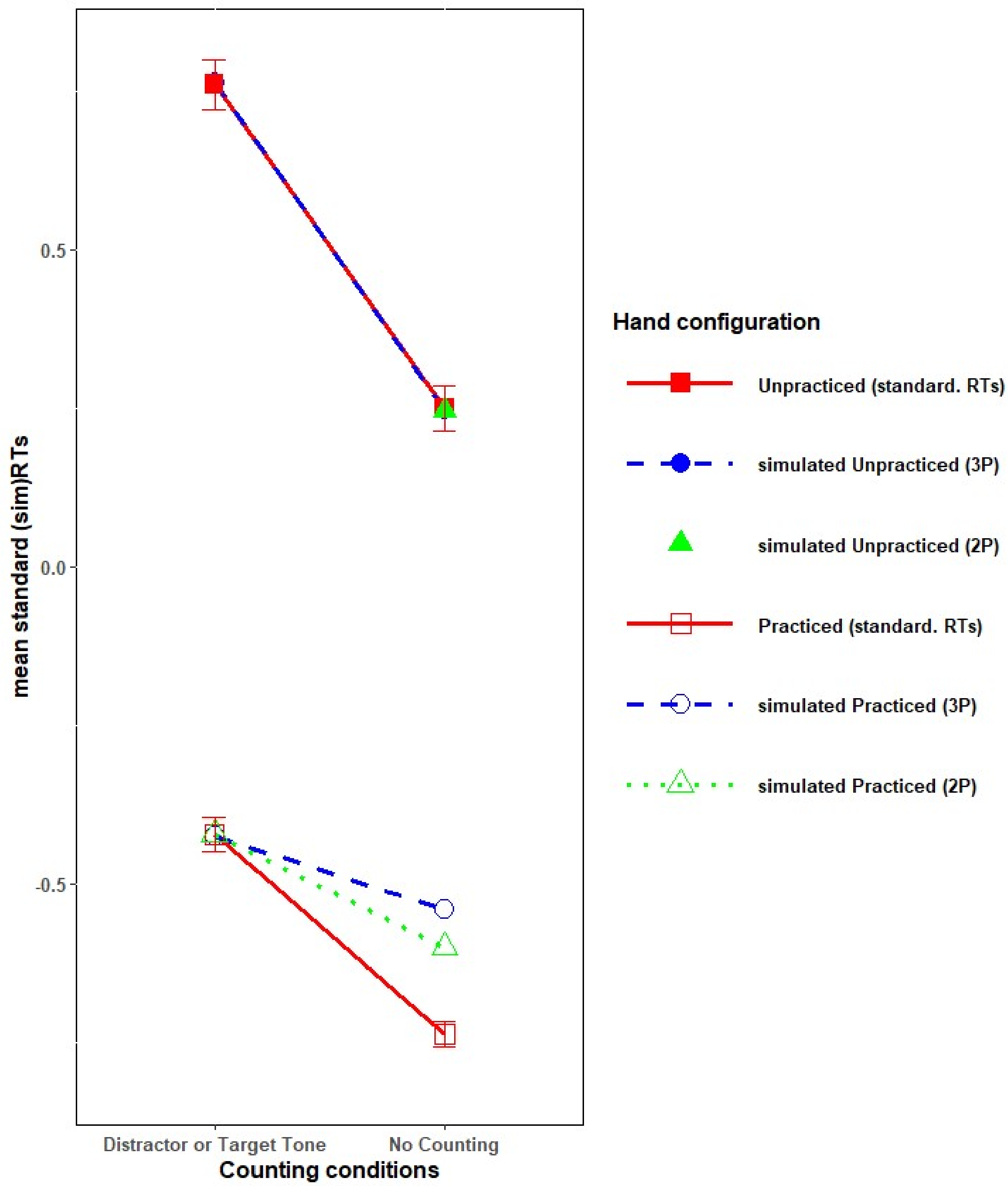


Comparison of means

No model was able to replicate the pattern of results in terms of mean standardized RTs for this approach. The result of the simulated race of motor and central processors for Model 2P (M = -0.6) was closer to the standardized RT mean of the Control condition (M = -0.7) than the result of the simulated race of MP, CS, and SR processors for Model 3P (M = -0.54). Although Model 2P got closer to the mean of the Control condition than Model 3P, this should not be overvalued, as the data of all participants across all conditions also included the biomechanical slowing of using one hand for sequence execution as the unpracticed hand configuration. Model 2P was generally always faster in the Control condition than Model 3P, so it would always have an advantage if both models produced a mean that is higher than the observed one. In this case, the biomechanical slowing increased the simulated means of both models and was probably the main reason why this modeling approach failed.

Figure B3

Mean standard simRTs of Models 2P and 3P as well as mean standardized RTs from Study 1 including RTs from all participants across all conditions with error bars representing 95 percent confidence intervals. Separated by the use of unpracticed and practiced hand configuration in addition to whether tones had to be counted or not. (Conditions: Tone/Unpracticed: HS/TC, No Counting/Unpracticed: HS, Tone/Practiced: TC, No Counting/Practiced: CC; Models 2P and 3P: Processors and races simulated for each condition according to Table 1 with the exclusion of the HS/TC condition for Model 2P)



Goodness-of-fit

The Cucconi test showed that location and scale of the resulting standard simRT distributions for Model 3P were equal to the RT distribution of the Hand Switch/Tone Counting condition, $C = 1.8$, $p = 0.4$, the RT distribution of the Hand Switch condition, $C = 2.7$, $p = 0.9$, and the RT distribution of the Tone Counting condition, $C = 7.0$, $p = 0.6$. However, location and scale of the resulting standard simRT distribution from a race of three processors were different than location and scale of the RT distribution of the Control condition, $C = 214.9$, $p < 0.001$.

We used quantile-quantile (Q-Q) plots to compare the resulting distributions of Model 3P with simulated normal distributions based directly on the mean and standard deviation of the standardized RT distribution of each test condition (Figure B5). Hence, the Q-Q plots were more relevant for conditions that involved a race as, for example, the SR processor was already a simulated normal distribution that was based directly on the parameters of the RT distribution of the Hand Switch/Tone Counting condition. The normal distributions for comparison were also simulated using the rnorm function with 10000 random samples based on the RT mean and standard deviation of the respective condition.

Figure B4

Density plots of the four test conditions for both standardized reaction times of Study 1 including all participants (black line) and the standard simRT distributions of Model 3P (red/dashed line) with the included processors as well as races between those processors ('vs') indicated in the legend.

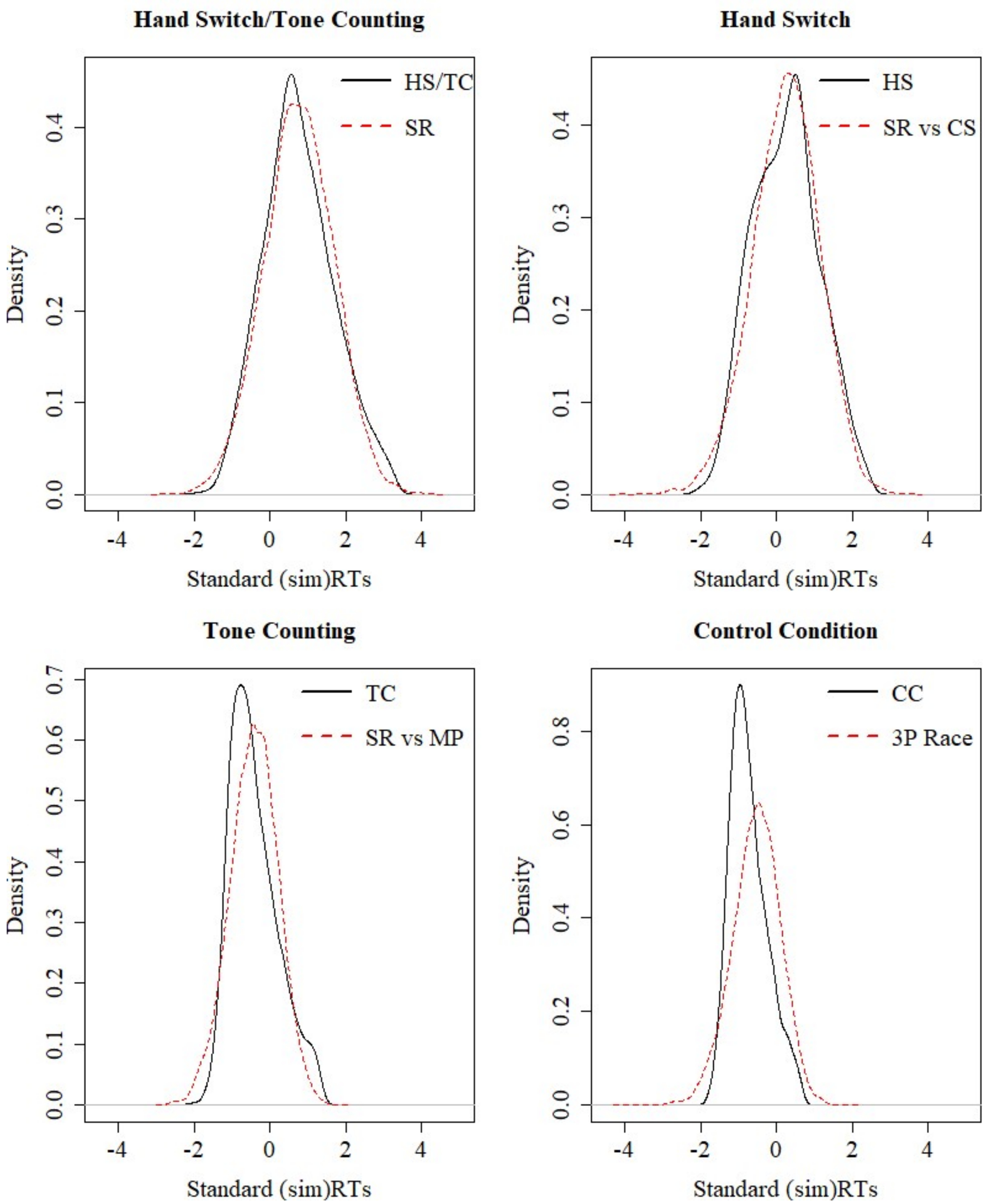
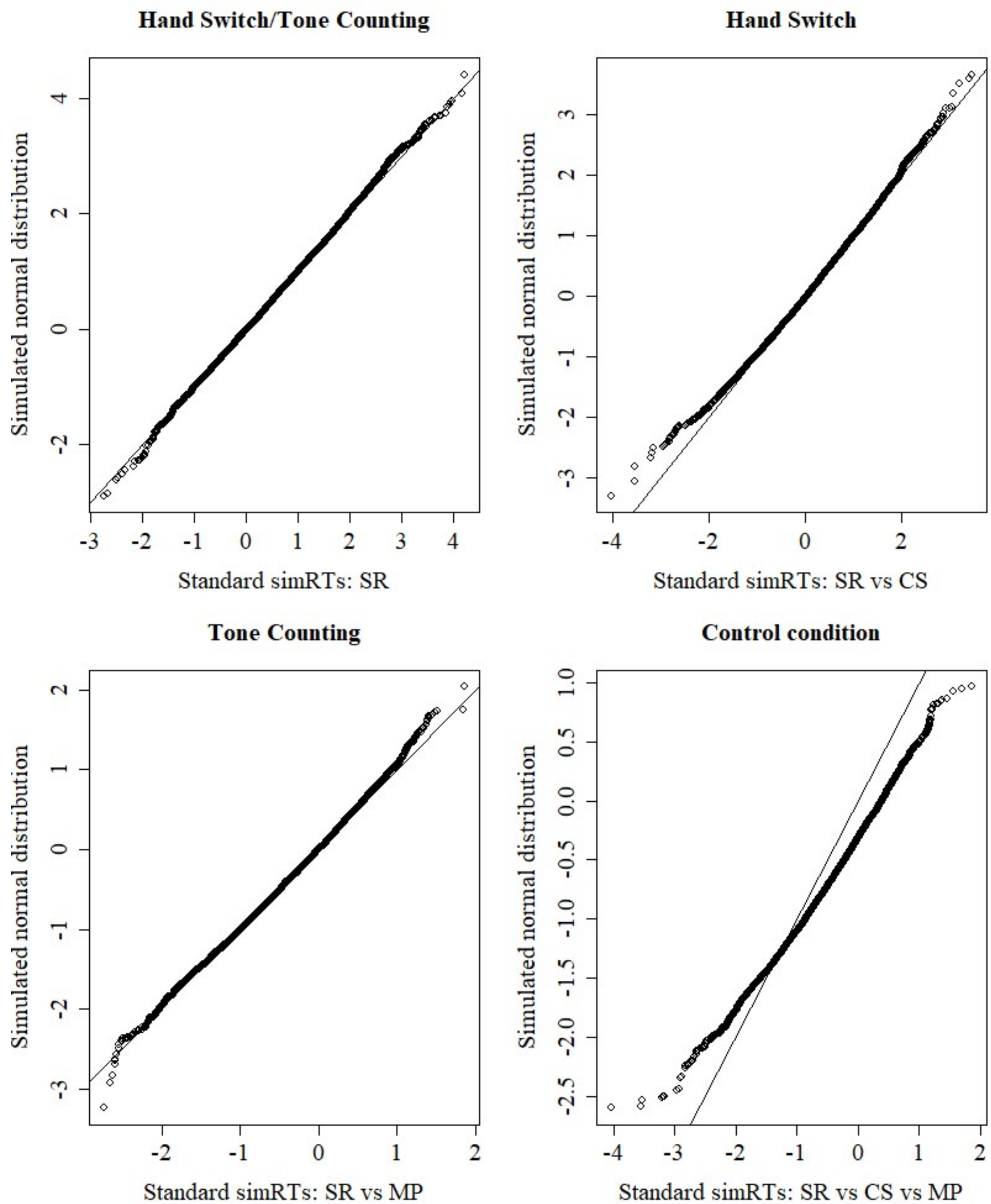


Figure B5

Q-Q plots of the four test conditions with simulated processors or results of race of processors of Model 3P on the x-axis and simulated normal distribution based on mean and standard deviation of standardized RTs of each condition from Study 1 including all participants across all conditions on the y-axis. The line represents x=y.



Appendix C

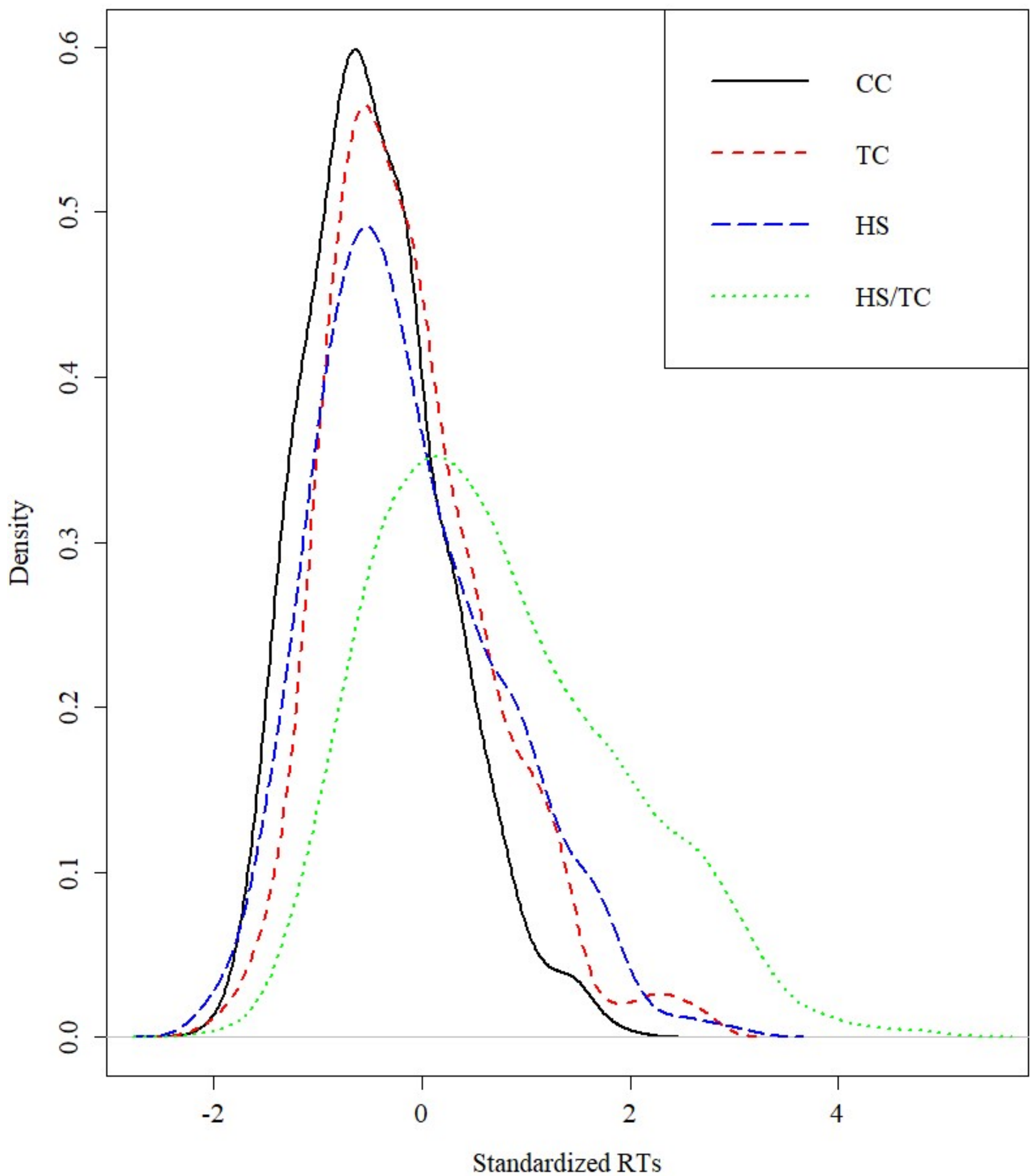
Extended results for the within-subjects approach of participants who practiced with one hand showing an interaction in the Hand Switch/Tone Counting condition

Standardized RT distributions

A clear right skew, that we could not remove without the exclusion of a significant number of further data points, was still present for the standardized RT distributions of the Hand Switch/Tone Counting and Hand Switch conditions where two hands were used as an unpracticed hand configuration.

Figure C1

Density plot showing the standardized distributions of the obtained RTs for each of the four conditions of the test phase in Study 1 from participants who had practiced with one hand including responses three to five (One hand used: CC = Control condition, TC = Tone Counting condition; two hands used: HS = Hand Switch condition, HS/TC = Hand Switch/Tone Counting condition).

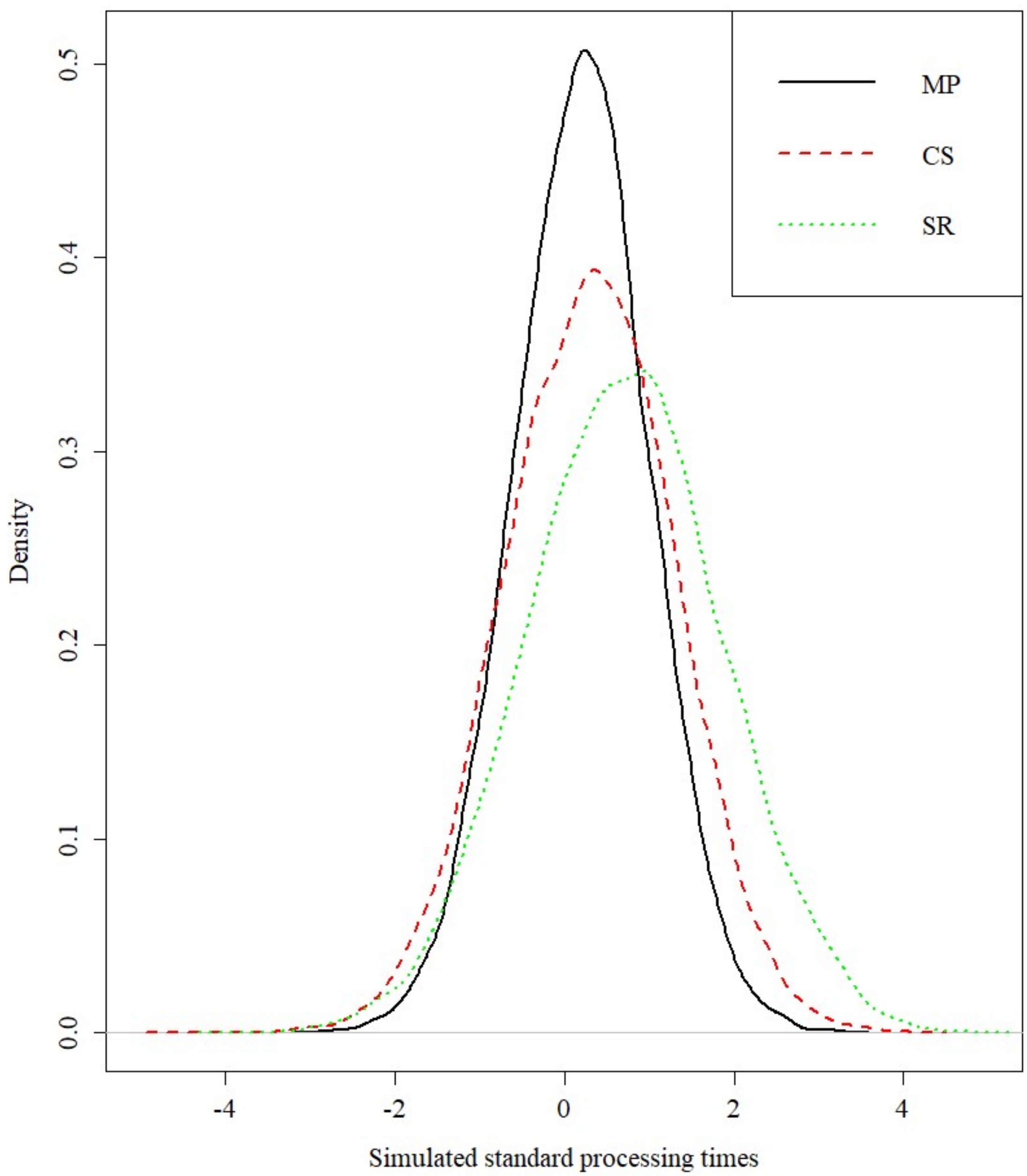


Simulation of processor distributions for Model 3P

Although the MP had the lowest mean of all three, CS and SR processors both had lower minimum values than the MP. Consequently, these minimum values always won the race against the MP and persisted after each race. The minimum values were caused by the high standard deviation of CS and SR processors which was a result of the still skewed standardized RT distributions of Hand Switch/Tone Counting and Hand Switch conditions (Figure C1).

Figure C2

Density plot showing the derived three processor distributions for Model 3P based (in)directly on the standardized RT distributions from Figure C1 (MP = Motor processor, CS = Central-symbolic processor, SR = Stimulus-response translation processor).



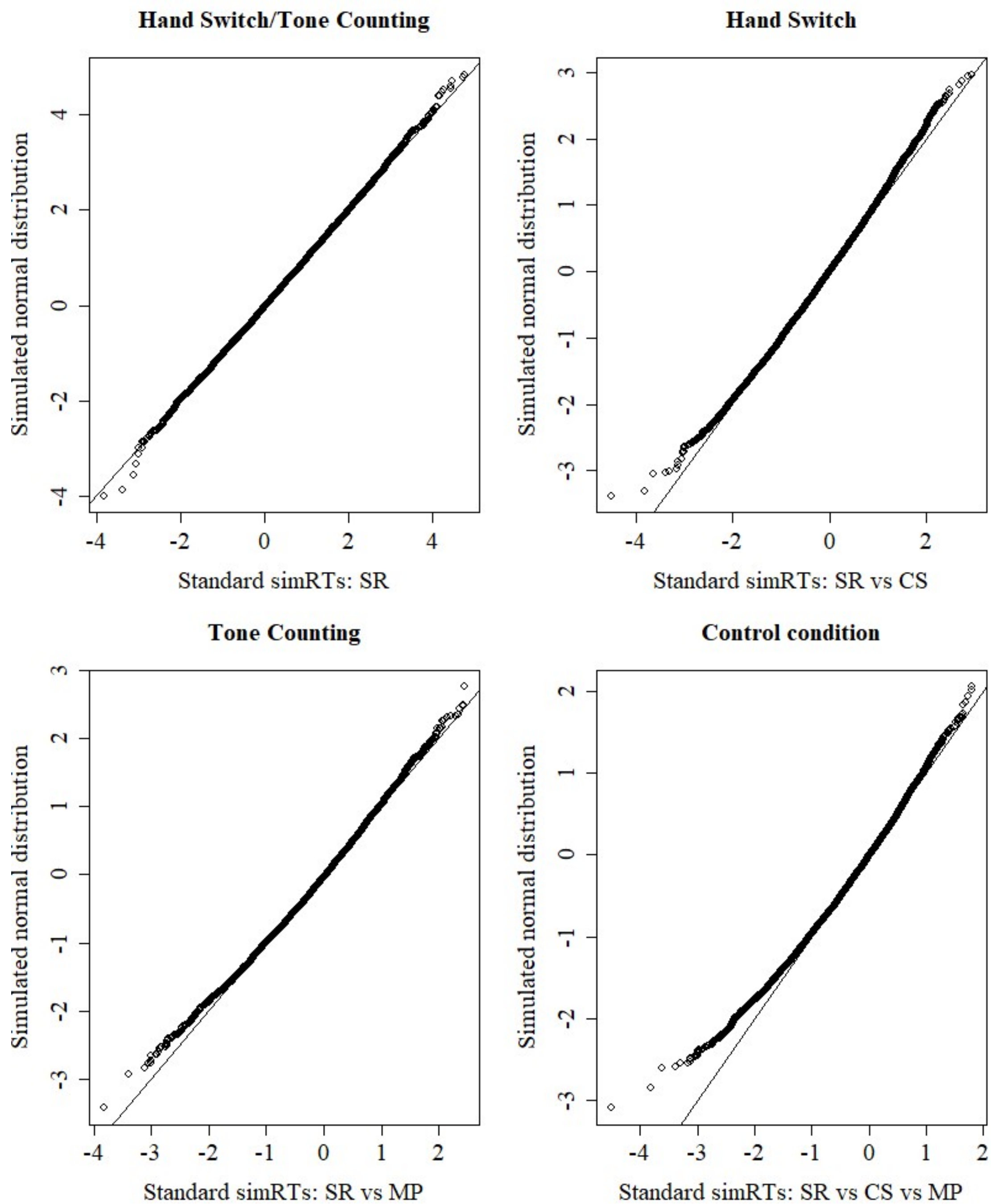
Goodness-of-fit

We used quantile-quantile (Q-Q) plots to compare the resulting distributions of Model 3P with simulated normal distributions based directly on the mean and standard deviation of the standardized RT distribution of each test condition. Hence, the Q-Q plots were more relevant for conditions that involved a race as, for example, the SR processor was already a simulated normal distribution that was based directly on the parameters of the standardized RT distribution of the Hand Switch/Tone Counting condition. The normal distributions for comparison were also simulated using the rnorm function with 10000 random samples based on the standardized RT mean and standard deviation of the respective condition.

Larger deviations can only be seen in the Control condition where the left tail caused by the SR and CS processors persisted while the right tail was mostly eliminated through the race. Hence, the simulated distributions of Model 3P deviated from the simulated normal distributions based directly on the standardized RTs, as the race of processors caused the resulting distribution to become less Gaussian and have a slight left skew. Apart from that, all four distributions of Model 3P seemed to mostly align with the simulated normal distributions based on mean and standard deviation of the standardized RTs of each condition.

Figure C3

Q-Q plots of the four test conditions with simulated processors or results of race of processors of Model 3P on the x-axis and simulated normal distribution based on mean and standard deviation of standardized reaction times of each condition from Study 1 including participants who practiced with one hand on the y-axis. The line represents x=y.



Appendix D

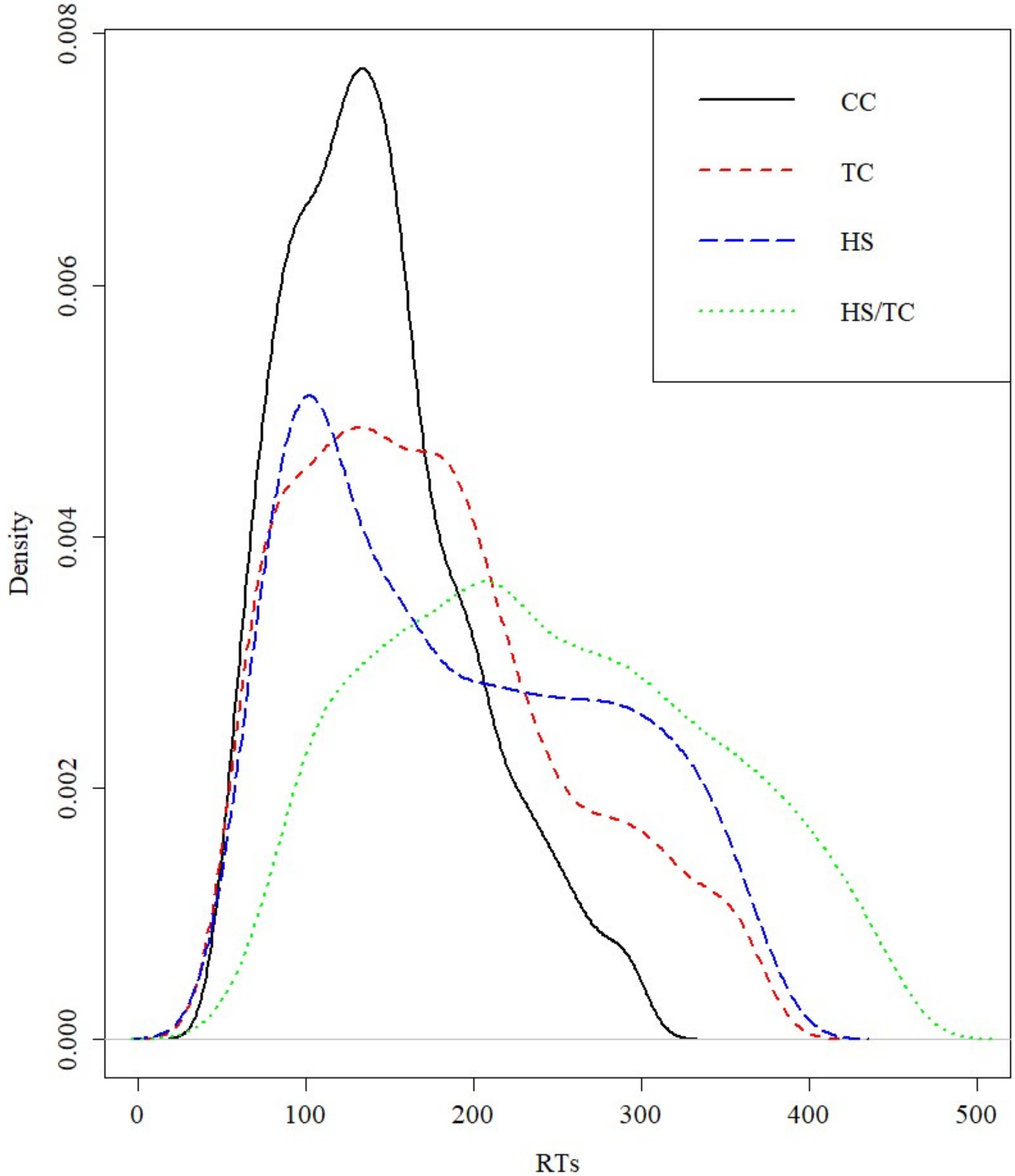
Extended results of the two-handed between-subjects approach showing an additive effect in the Hand Switch/Tone Counting condition

RT distributions

The RT distributions were still heavily skewed by individual differences and did not represent Gaussian distributions.

Figure D1

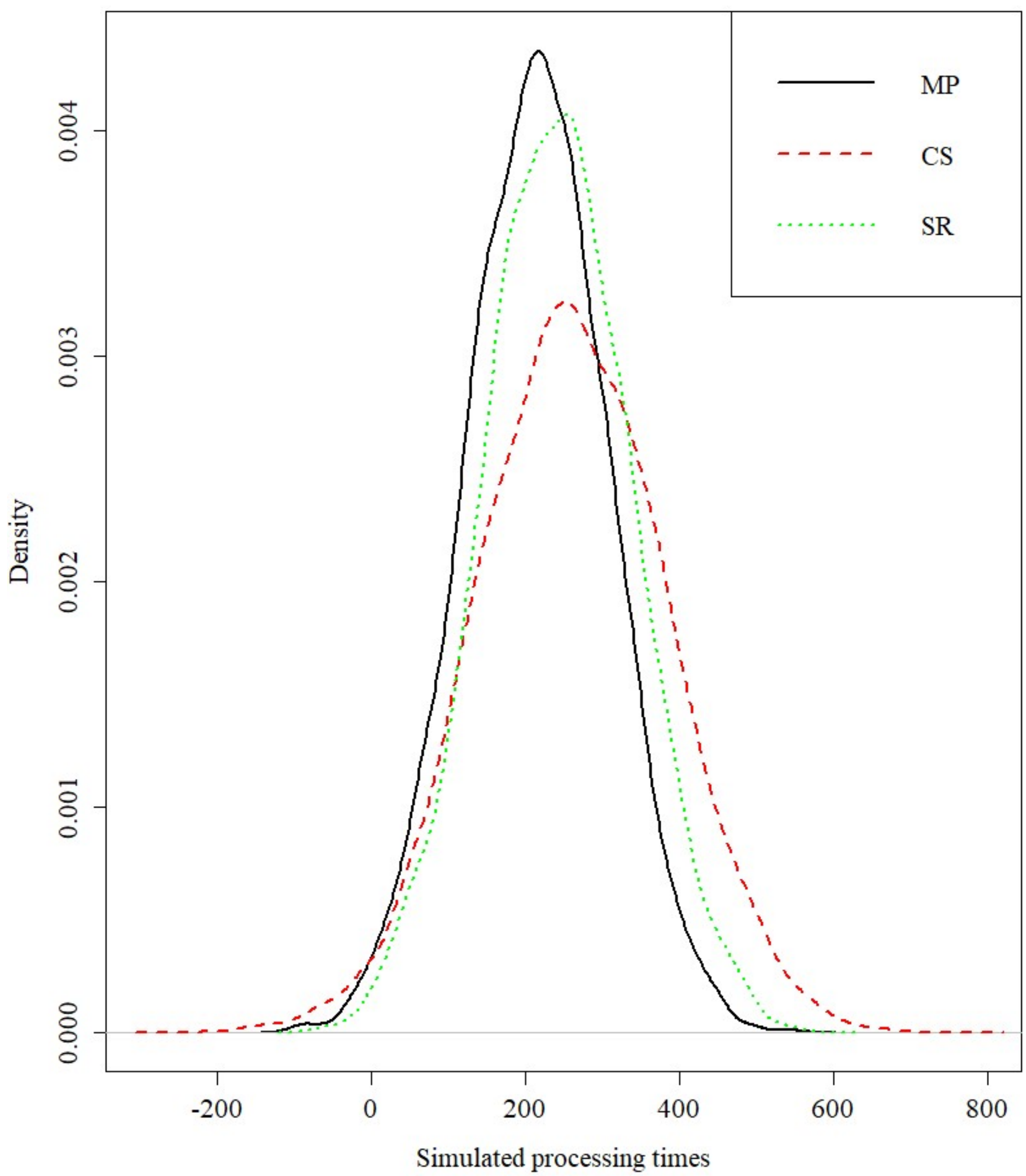
Density plot showing the distributions of the obtained RTs for each of the four conditions of the test phase in Study 1 from two-handed sequence execution including responses three to five. (Participants who practiced with two hands: CC = Control condition, TC = Tone Counting condition; Participants who practiced with one hand: HS = Hand Switch condition, HS/TC = Hand Switch/Tone Counting condition).



Simulation of processor distributions

Figure D2

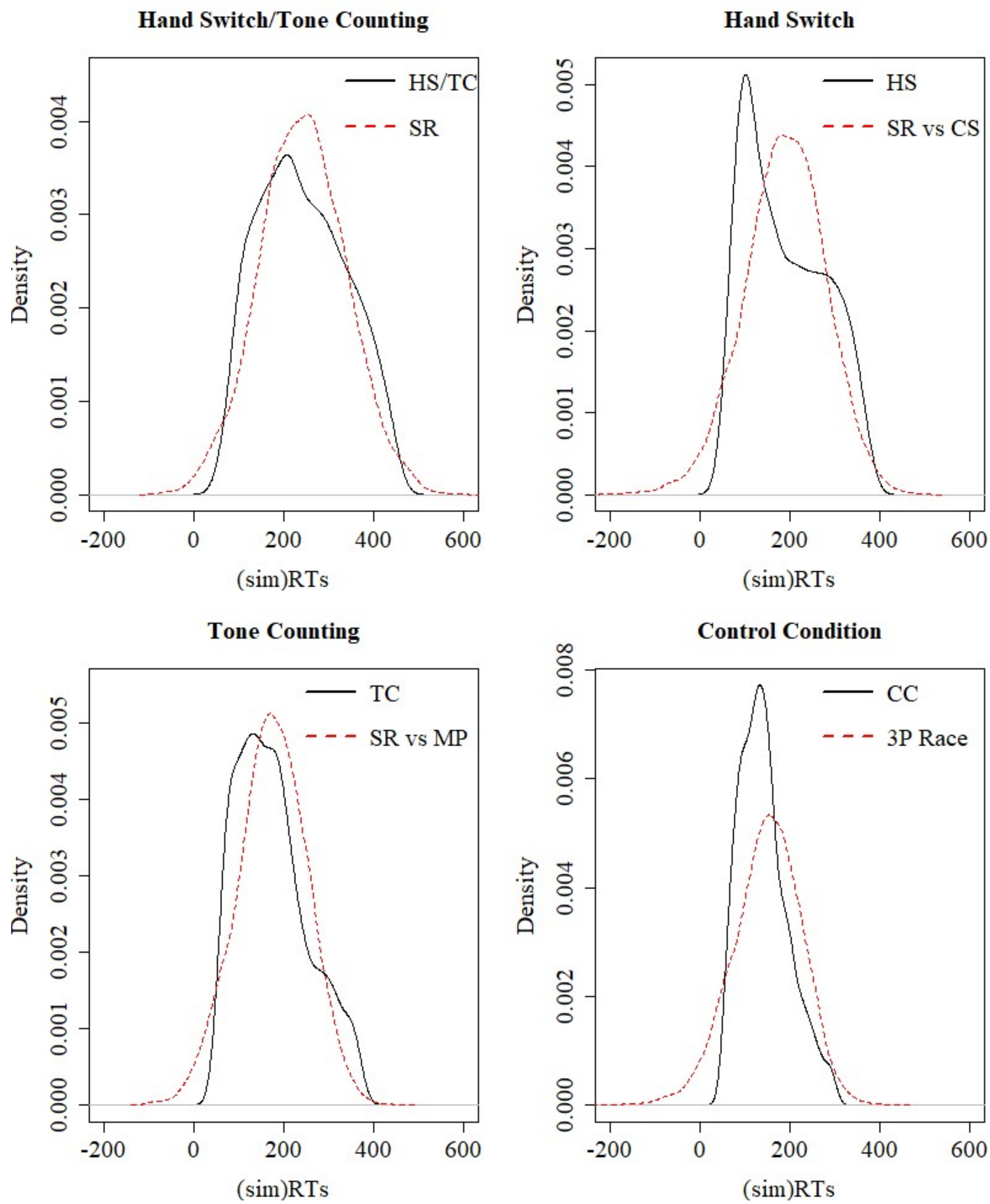
Density plot showing the three processor distributions for Model 3P based (in)directly on the RT distributions from Figure D1 (MP = Motor processor, CS = Central-symbolic processor, SR = Stimulus-response translation processor).



Goodness-of-fit

Figure D3

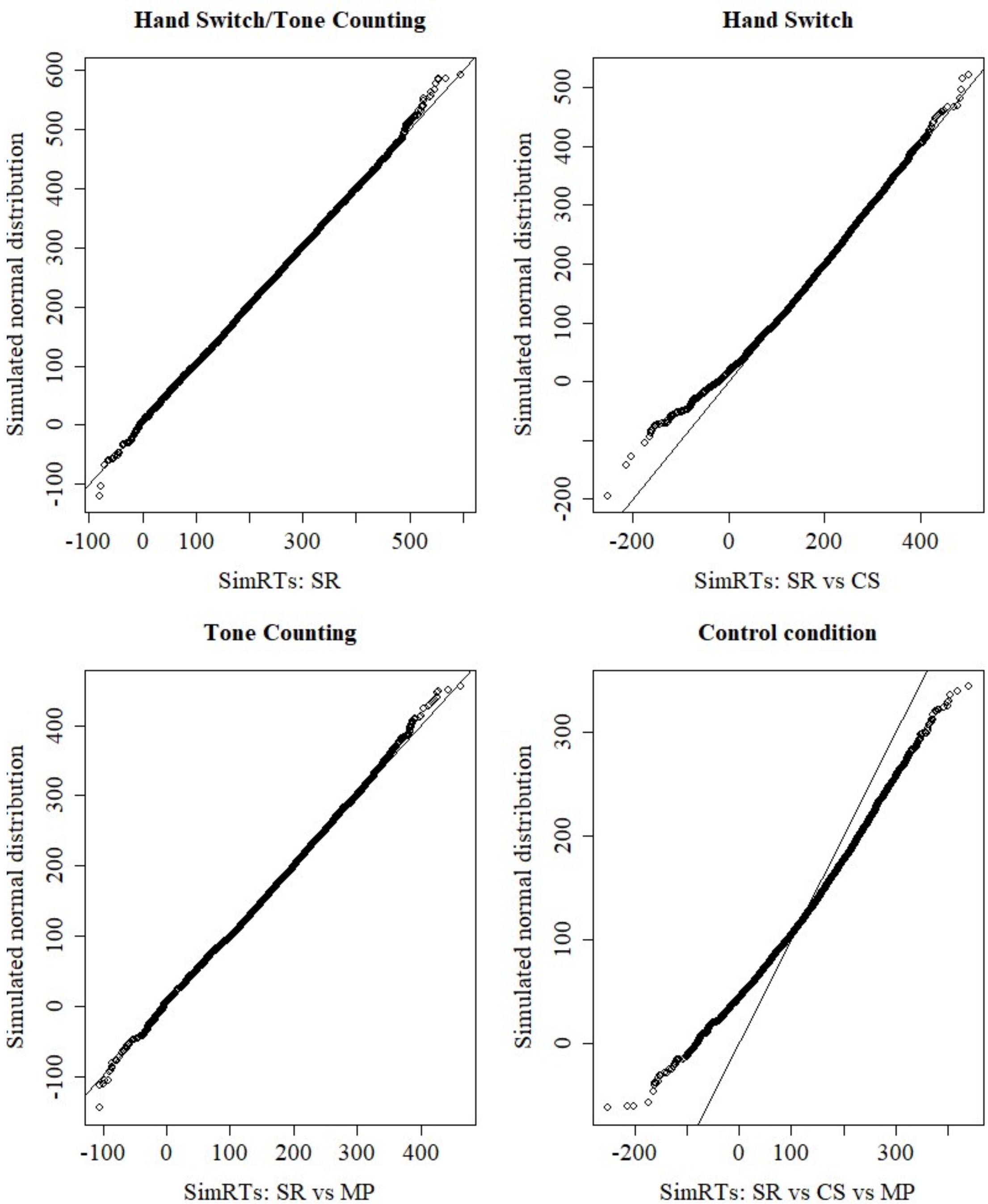
Density plots of the four test conditions for both reaction times of Study 1 from two-handed sequence execution (black line) and the simRT distributions of Model 3P (red/dashed line) with the included processors as well as races between those processors ('vs') indicated in the legend.



We used quantile-quantile (Q-Q) plots to compare the resulting distributions of Model 3P with simulated normal distributions based directly on the mean and standard deviation of the RT distribution of each test condition. Hence, the Q-Q plots were more relevant for conditions that involved a race as, for example, the SR processor was already a simulated normal distribution that was based directly on the parameters of the RT distribution of the Hand Switch/Tone Counting condition. The normal distributions for comparison were also simulated using the rnorm function with 10000 random samples based on the RT mean and standard deviation of the respective condition.

Figure D4

Q-Q plots of the four test conditions with simulated processors or results of race of processors of Model 3P on the x-axis and simulated normal distribution based on mean and standard deviation of RTs of each condition from Study 1 including two-handed sequence execution on the y-axis. The line represents x=y.



Appendix E

Script R for Study 2

```
# libraries (not all might be needed; some might require import of library from Github using devtools)
library(tidyverse)
library(rstanarm)
library(mascutils)
library(brms)
library(GGally)
library(bayr)
library(readxl)
library(BBmisc)
library(Rmisc)
library(reshape2)
library(nonpar)
library(papaja)


#Data import (RTs below 50 ms filtered out; only responses three to five included)

#All participants across all conditions

Bl7 <- read_excel("Bl7bv2.xlsx") %>%

  filter(Present6SqRT.RT > 50 & LogLevel6 > 2 & LogLevel6 < 6)


#Participants who practiced with one hand

Bl71 <- read_excel("Bl7bv2.xlsx") %>%

  filter(Present6SqRT.RT > 50 & LogLevel6 > 2 & LogLevel6 < 6 & NrHandsUsedPract ==
1)


#Participants who practiced with two hands

Bl72 <- read_excel("Bl7bv2.xlsx") %>%
```

```
  filter(Present6SqRT.RT > 50 & LogLevel6 > 2 & LogLevel6 < 6 & NrHandsUsedPract == 2)


#Separation of data into conditions with first cut-off operation already included (order of conditions: TC, HS, HS/TC, Control condition)

#Within-subjects approach for participants who practiced with one hand

Gl7m <- Bl71 %>%

  filter(HandInstruct == 'the SAME' & ToneInstruct == 'count the LOW tones' & Present6SqRT.RT < 400)

Gl7c <- Bl71 %>%

  filter(HandInstruct == 'DIFFERENT' & ToneInstruct == 'IGNORE all tones' & Present6SqRT.RT < 430)

Gl7r <- Bl71 %>%

  filter(HandInstruct == 'DIFFERENT' & ToneInstruct == 'count the LOW tones' & Present6SqRT.RT < 520)

Gl7n <- Bl71 %>%

  filter(HandInstruct == 'the SAME' & ToneInstruct == 'IGNORE all tones' & Present6SqRT.RT < 330)


#Between-subjects approach for two-handed sequence execution

Gl7mns2 <- Bl72 %>%

  filter(HandInstruct == 'the SAME' & ToneInstruct == 'count the LOW tones' & Present6SqRT.RT < 370)

Gl7cns2 <- Bl71 %>%

  filter(HandInstruct == 'DIFFERENT' & ToneInstruct == 'IGNORE all tones' & Present6SqRT.RT < 380)
```

```
Gl7rns2 <- Bl71 %>%
  filter(HandInstruct == 'DIFFERENT' & ToneInstruct == 'count the LOW tones' &
Present6SqRT.RT < 450)
Gl7nns2 <- Bl72 %>%
  filter(HandInstruct == 'the SAME' & ToneInstruct == 'IGNORE all tones' &
Present6SqRT.RT < 300)

#Third modeling approach reported in Appendix B
Gl7m3 <- Bl7 %>%
  filter(HandInstruct == 'the SAME' & ToneInstruct == 'count the LOW tones' &
Present6SqRT.RT < 450)
Gl7c3 <- Bl7 %>%
  filter(HandInstruct == 'DIFFERENT' & ToneInstruct == 'IGNORE all tones' &
Present6SqRT.RT < 620)
Gl7r3 <- Bl7 %>%
  filter(HandInstruct == 'DIFFERENT' & ToneInstruct == 'count the LOW tones' &
Present6SqRT.RT < 680)
Gl7n3 <- Bl7 %>%
  filter(HandInstruct == 'the SAME' & ToneInstruct == 'IGNORE all tones' &
Present6SqRT.RT < 360)

#Example code for second cut-off operation and standardization of within-subjects data
(similar code was used for data of all participants for the third modeling approach)
Bl7G <- rbind(Gl7m,Gl7n,Gl7c,Gl7r)
part1 <- Bl7G %>%
  filter(Subject == 1)
```

```
part4 <- Bl7G %>%
  filter(Subject == 4)
part6 <- Bl7G %>%
  filter(Subject == 6)
part7 <- Bl7G %>%
  filter(Subject == 7)
part9 <- Bl7G %>%
  filter(Subject == 9)
part12 <- Bl7G %>%
  filter(Subject == 12)
part14 <- Bl7G %>%
  filter(Subject == 14)
part15 <- Bl7G %>%
  filter(Subject == 15)
part17 <- Bl7G %>%
  filter(Subject == 17)
part20 <- Bl7G %>%
  filter(Subject == 20)
part22 <- Bl7G %>%
  filter(Subject == 22)
part23 <- Bl7G %>%
  filter(Subject == 23)
```

#Separation into conditions of each participant for 2nd cut-off operation (only example for 1st participant is provided; similar code used for 11 other participants): Code has to be run first like written below to categorize the data. Then, every ‘#’ and the preceding ‘)’ have to be removed and code has to be run again to apply rule used for 2nd cut-off operation.

```
part1m <- part1 %>%
  filter(HandInstruct == 'the SAME' & ToneInstruct == 'count the LOW tones')# & Present6SqRT.RT < mean(part1m$Present6SqRT.RT) + 2.5*sd(part1m$Present6SqRT.RT))
part1c <- part1 %>%
  filter(HandInstruct == 'DIFFERENT' & ToneInstruct == 'IGNORE all tones')# & Present6SqRT.RT < mean(part1c$Present6SqRT.RT) + 2.5*sd(part1c$Present6SqRT.RT))
part1r <- part1 %>%
  filter(HandInstruct == 'DIFFERENT' & ToneInstruct == 'count the LOW tones')# & Present6SqRT.RT < mean(part1r$Present6SqRT.RT) + 2.5*sd(part1r$Present6SqRT.RT))
part1n <- part1 %>%
  filter(HandInstruct == 'the SAME' & ToneInstruct == 'IGNORE all tones')# & Present6SqRT.RT < mean(part1n$Present6SqRT.RT) + 2.5*sd(part1n$Present6SqRT.RT))
```

#Putting data of conditions for each participant back together and standardizing RTs

```
part1 <- rbind(part1n,part1m,part1c,part1r)
part4 <- rbind(part4n,part4m,part4c,part4r)
part6 <- rbind(part6n,part6m,part6c,part6r)
part7 <- rbind(part7n,part7m,part7c,part7r)
part9 <- rbind(part9n,part9m,part9c,part9r)
part12 <- rbind(part12n,part12m,part12c,part12r)
part14 <- rbind(part14n,part14m,part14c,part14r)
part15 <- rbind(part15n,part15m,part15c,part15r)
```

```
part17 <- rbind(part17n,part17m,part17c,part17r)

part20 <- rbind(part20n,part20m,part20c,part20r)

part22 <- rbind(part22n,part22m,part22c,part22r)

part23 <- rbind(part23n,part23m,part23c,part23r)


part1$Present6SqRT.RT <- normalize(part1$Present6SqRT.RT)

part4$Present6SqRT.RT <- normalize(part4$Present6SqRT.RT)

part6$Present6SqRT.RT <- normalize(part6$Present6SqRT.RT)

part7$Present6SqRT.RT <- normalize(part7$Present6SqRT.RT)

part9$Present6SqRT.RT <- normalize(part9$Present6SqRT.RT)

part12$Present6SqRT.RT <- normalize(part12$Present6SqRT.RT)

part14$Present6SqRT.RT <- normalize(part14$Present6SqRT.RT)

part15$Present6SqRT.RT <- normalize(part15$Present6SqRT.RT)

part17$Present6SqRT.RT <- normalize(part17$Present6SqRT.RT)

part20$Present6SqRT.RT <- normalize(part20$Present6SqRT.RT)

part22$Present6SqRT.RT <- normalize(part22$Present6SqRT.RT)

part23$Present6SqRT.RT <- normalize(part23$Present6SqRT.RT)


#Putting data of all participants back into one dataset and then, separation into four conditions (for third modeling approach, a few more data points were removed here for each condition like reported in Appendix B)

Bl7gs <- rbind(part1,part4,part6,part7,part9,part12,part14,part15,part17,part20,part22,part23)

Bl7m <- Bl7gs %>%

  filter(HandInstruct == 'the SAME' & ToneInstruct == 'count the LOW tones')

Bl7c <- Bl7gs %>%
```

```
  filter(HandInstruct == 'DIFFERENT' & ToneInstruct == 'IGNORE all tones')
Bl7r <- Bl7gs %>%
  filter(HandInstruct == 'DIFFERENT' & ToneInstruct == 'count the LOW tones')
Bl7n <- Bl7gs %>%
  filter(HandInstruct == 'the SAME' & ToneInstruct == 'IGNORE all tones')


#Simulation of processors for Models 3P and 2P of within-subjects data
sr1 <- rnorm(10000, mean(Bl7r$Present6SqRT.RT), sd(Bl7r$Present6SqRT.RT))
cs1 <- rnorm(10000, 0.3, 1)
mp1 <- rnorm(10000, 0.2, 0.8)


mp12P <- rnorm(10000, mean(Bl7m$Present6SqRT.RT), sd(Bl7m$Present6SqRT.RT))
cp12P <- rnorm(10000, mean(Bl7c$Present6SqRT.RT), sd(Bl7c$Present6SqRT.RT))


#Races between processors for TC, HS, and Control conditions (as the order of the list of
values for each processor distribution is already random, the race is only being further
randomized for the race of three processors, so the races between SR processors and the other
two processors are not being repeated in the same way as before)
gsm <- matrix(c(sr1,mp1), ncol = 2)
gtc <- apply(gsm,1,min)


gcs <- matrix(c(sr1,cs1), ncol = 2)
ghs <- apply(gcs,1,min)


gmcs <- matrix(c(sample(sr1),sample(cs1), sample(mp1)), ncol = 3)
```

```
gcc <- apply(gmcs,1,min)


#Race for Model 2P

gcm2P <- matrix(c(mp12P,cp12P), ncol = 2)

gcc2P <- apply(gcm2P,1,min)
```

#Basic procedure of simulating processors and races for between-subjects and third modeling approaches was the same. Below are exact parameters used for MP and CS processor of both approaches for Model 3P (other processors always used mean and standard deviation of respective condition as parameters)

```
#Between-subjects approach

csad <- rnorm(10000, 260, 125)

mpad <- rnorm(10000, 214, 92)


#Third modeling approach

cs3 <- rnorm(10000, 1, 1.25)

mp3 <- rnorm(10000, -0.315, 0.68)
```

#Example code for Cucconi test used for Control Condition of within-subjects approach as well as code used to determine mean and standard deviation for simulated and experimental results of this condition

```
cucconi.test(Bl7n$Present6SqRT.RT, gcc, method = "bootstrap")


mean(gcc)

sd(gcc)
```

```
mean(Bl7n$Present6SqRT.RT)

sd(Bl7n$Present6SqRT.RT)


#Example code for creating means plot (Figure 5) of within-subjects approach (similar code used for other approaches)

#Data from Study 1

Bl7gs2 <- rbind(Bl7n,Bl7m,Bl7c,Bl7r)

Bls <- Bl7gs2

Bls <- Bls %>%

  rename(c("Present6SqRT.RT" = "mean_value"))

Bls$Hand_configuration <- NA

Bls$Hand_configuration[grepl("DIFFERENT",  Bls$HandInstruct)] <- "Unpracticed (standard. RTs)"

Bls$Hand_configuration[grepl("the SAME", Bls$HandInstruct)] <- "Practiced (standard. RTs)"

Bls$Hand_configuration <- factor(Bls$Hand_configuration)

Bls$Counting_conditions <- NA

Bls$Counting_conditions[grepl("count the LOW tones",  Bls$ToneInstruct)] <- "Distractor or Target Tone"

Bls$Counting_conditions[grepl("IGNORE all tones", Bls$ToneInstruct)] <- "No Counting"

Bls$Counting_conditions <- factor(Bls$Counting_conditions)


#Data of Model 3P

sim <- matrix(c(sr1,ghs,gtc,gcc), ncol = 4)

colnames(sim) <- c("Different/Count","Different/Ignore","Same/Count","Same/Ignore")
```

```
sim <- melt(data=sim,
           measure.vars=c("Different/Count", "Different/Ignore", "Same/Count",
"Same/Ignore"),
           variable.name="Condition")
sim$Hand_configuration <- NA
sim$Hand_configuration[grepl("^Different",  sim$Var2)] <- "simulated Unpracticed (3P)"
sim$Hand_configuration[grepl("^Same", sim$Var2)] <- "simulated Practiced (3P)"
sim$Hand_configuration <- factor(sim$Hand_configuration)
sim$Counting_conditions <- NA
sim$Counting_conditions[grepl("Count$",  sim$Var2)] <- "Distractor or Target Tone"
sim$Counting_conditions[grepl("Ignore$", sim$Var2)] <- "No Counting"
sim$Counting_conditions <- factor(sim$Counting_conditions
sim <- sim %>%
  rename(c("value" = "mean_value"))


#Data of Model 2P for three of four conditions
sim2P <- matrix(c(cp12P,mp12P,gcc2P), ncol = 3)
colnames(sim2P) <- c("Different/Ignore","Same/Count","Same/Ignore")
sim2P <- melt(data=sim2P,
            measure.vars=c("Different/Ignore", "Same/Count", "Same/Ignore"),
            variable.name="Condition")
sim2P$Hand_configuration <- NA
sim2P$Hand_configuration[grepl("^Different",  sim2P$Var2)] <- "simulated Unpracticed
(2P)"
sim2P$Hand_configuration[grepl("^Same", sim2P$Var2)] <- "simulated Practiced (2P)"
```

```
sim2P$Hand_configuration <- factor(sim2P$Hand_configuration)

sim2P$Counting_conditions <- NA

sim2P$Counting_conditions[grepl("Count$",  sim2P$Var2)] <- "Distractor or Target Tone"

sim2P$Counting_conditions[grepl("Ignore$", sim2P$Var2)] <- "No Counting"

sim2P$Counting_conditions <- factor(sim2P$Counting_conditions)

sim2P <- sim2P %>%

  rename(c("value" = "mean_value"))


#Producing summary data including means for Models 2P and 3P as well as standardized RTs
of within-subjects approach

simsum <- summarySEwithin(sim, measurevar="mean_value",
withinvars=c("Hand_configuration","Counting_conditions"),

                          na.rm=FALSE, conf.interval=.95)

Blsum <- summarySEwithin(Bls, measurevar="mean_value",
withinvars=c("Hand_configuration","Counting_conditions"),

                         na.rm=FALSE, conf.interval=.95)

simpsum <- summarySEwithin(sim2P, measurevar="mean_value",
withinvars=c("Hand_configuration","Counting_conditions"),

                           na.rm=FALSE, conf.interval=.95)


sumsum <- rbind(simsum,Blsum, simpsum)


#Creating the plot and showing it on screen

compare <- ggplot(sumsum, aes(x=Counting_conditions, y=mean_value, colour =
Hand_configuration, group = Hand_configuration, linetype = Hand_configuration)) +
```

```
  geom_line(size = 1) +
  geom_point(aes(shape = Hand_configuration, color = Hand_configuration), size = 3) +
  scale_shape_manual(values=c(15, 15,16, 16, 17, 17)) +
  geom_errorbar(width=.1, aes(ymin=mean_value-ci, ymax=mean_value+ci), data = Blsum) +
  annotate("text", x = "No Counting", y = -0.66, label = "Model 2P", col = 6) +
  theme(legend.key.height= unit(1.5, 'cm'),
        legend.key.width= unit(2.0, 'cm'), legend.key = element_rect(fill = 'white', color =
'white')) +
    theme(text=element_text(family="Times New Roman", face="bold", size=12)) +
  theme(panel.background = element_rect(fill = "white", color = "black"))


compare + ylim(-0.7, 0.8) + labs(x = "Counting conditions", y = "mean standard (sim)RTs") +
labs(linetype = "Hand configuration", color = "Hand configuration", shape = "Hand
configuration")


#Code used to create Figure 6 (was also used to create similar figures in Appendices with
respective data)
par(family = 'serif')
par(mfrow = c(2,2)) #This is needed for all figures where a 2x2 design exists. Otherwise, it is
always a 1x1 design.
par(mar=c(4, 4, 4, 2)) #default: c(5.1, 4.1, 4.1, 2.1) which was used for all figures that did not
have a 2x2 design


plot(density(Bl7r$Present6SqRT.RT), xlab = "Standard (sim)RTs", ylab = "Density",main =
"Hand Switch/Tone Counting", xlim = c(-4.5,5), zero.line = TRUE, cex.lab = 1.5, cex.axis =
1.5, cex.main = 1.5)#, yaxt="n")#, ylim = c(0,0.005))
```

lines(density(sr1), col = 'red', lty = 2)

legend(x = "topright", legend = c("HS/TC", "SR"), lty = c(1, 2), col = c(1, 'red'), lwd = 2, cex = 1.5, inset = c(-0.28, -0.08), bty = "n" , xpd=TRUE, seg.len = 1, x.intersp = 0.3, y.intersp = 0.6)

plot(density(Bl7c$Present6SqRT.RT), xlab = "Standard (sim)RTs", main = "Hand Switch", xlim = c(-4.5,5), zero.line = TRUE, cex.lab = 1.5, cex.axis = 1.5, cex.main = 1.5)#, ylim = c(0,0.5))

lines(density(ghs), col = 'red', lty = 2)

legend(x = "topright", legend = c("HS", "SR vs CS"), lty = c(1, 2), col = c(1, 'red'), lwd = 2, cex = 1.5, inset = c(-0.35, -0.08), bty = "n", xpd=TRUE, seg.len = 1, x.intersp = 0.3, y.intersp = 0.6)

plot(density(Bl7m$Present6SqRT.RT), xlab = "Standard (sim)RTs", main = "Tone Counting", xlim = c(-4.5,5), zero.line = TRUE, ylim = c(0,0.6), cex.lab = 1.5, cex.axis = 1.5, cex.main = 1.5)

lines(density(gtc), col = 'red', lty = 2)

legend(x = "topright", legend = c("TC", "SR vs MP"), lty = c(1, 2), col = c(1, 'red'), lwd = 2, cex = 1.5, inset = c(-0.35, -0.08), bty = "n", xpd=TRUE, seg.len = 1, x.intersp = 0.3, y.intersp = 0.6)

plot(density(Bl7n$Present6SqRT.RT), xlab = "Standard (sim)RTs", main = "Control Condition", xlim = c(-4.5,5), zero.line = TRUE, cex.lab = 1.5, cex.axis = 1.5, cex.main = 1.5)#, ylim = c(0,0.65))

lines(density(gcc), col = 'red', lty = 2)

legend(x = "topright", legend = c("CC", "3P Race"), lty = c(1, 2), col = c(1, 'red'), lwd = 2, cex = 1.5, inset = c(-0.3, -0.08), bty = "n", xpd=TRUE, seg.len = 1, x.intersp = 0.3, y.intersp = 0.6)

#Figures that can only be found in Appendices: Figure C1, C2, and similar figures in the other appendices were also created using plot(density) and lines(density) for the respective (standardized) RTs or (standard) simRTs. Apart from different data being plotted in each figure, differences to the previous figure were only of aesthetic nature

```
#Example code used to simulate normal distributions based directly on parameters of (standardized) RT distributions and Q-Q plots to compare them with distributions of Model 3P for each condition of the within-subjects approach (Figure B3)

xhstc <- rnorm(10000, mean(Bl7r$Present6SqRT.RT), sd(Bl7r$Present6SqRT.RT))

xhs <- rnorm(10000, mean(Bl7c$Present6SqRT.RT), sd(Bl7c$Present6SqRT.RT))

xtc <- rnorm(10000, mean(Bl7m$Present6SqRT.RT), sd(Bl7m$Present6SqRT.RT))

xcc <- rnorm(10000, mean(Bl7n$Present6SqRT.RT), sd(Bl7n$Present6SqRT.RT))


#2x2 design was used again

qqplot(sr1,xhstc, xlab = "Standard simRTs: SR", ylab = "Simulated normal distribution", main = "Hand Switch/Tone Counting", cex.lab = 1.5, cex.axis = 1.5, cex.main = 1.5, font = 1)

abline(0,1)

qqplot(ghs,xhs, xlab = "Standard simRTs: SR vs CS", ylab = "Simulated normal distribution", main = "Hand Switch", cex.lab = 1.5, cex.axis = 1.5, cex.main = 1.5)

abline(0,1)

qqplot(gtc,xtc, xlab = "Standard simRTs: SR vs MP", ylab = "Simulated normal distribution", main = "Tone Counting", cex.lab = 1.5, cex.axis = 1.5, cex.main = 1.5)

abline(0,1)

qqplot(gcc,xcc, xlab = "Standard simRTs: SR vs CS vs MP", ylab = "Simulated normal distribution", main = "Control condition", cex.lab = 1.5, cex.axis = 1.5, cex.main = 1.5)

abline(0,1)
```