Insights into cognitive processing of the go/nogo Discrete Sequence Production task: A replication study

Hilde W. Althof
s1790617

Bachelor Thesis

Faculty of Behavioural, Management and Social sciences
Human Factors and Engineering Psychology
University of Twente

First supervisor: Dr. Russell W. Chan
Second supervisor: Prof. Dr. Ing. Willem B. Verwey

July 2, 2021

Abstract

Motor sequence learning (MSL) is important to successfully perform various daily activities. One way to measure motor sequence learning is by using the Discrete Sequence Production (DSP) task to investigate various behavioural phenomena. However, the original DSP task involves action preparation and execution which occur temporarily close to each other, leading to difficulty in detangling these processes. This could lead to confounding issues when using cortical measurements during MSL (e.g. electroencephalogram; EEG). To overcome this issue, a go/nogo DSP task can be used to separate cognitive and motor-related processes. The present study aimed at replicating previous findings and further uncover other associated learning phenomena in the go/nogo DSP task compared to the original DSP task. Thirty-one participants completed five training blocks, each containing 48 trials of various six-key sequences. The focus was to first replicate results of general learning ability during a go/nogo DSP task, i.e. accuracy levels and response times. Secondly, the concatenation phenomenon was investigated in the go/nogo DSP task. Lastly, it was sought to understand functional post-error slowing to further elucidate cognitive processing differences compared to the original DSP task. All outcomes were compared using linear mixed-effects models between the subjects. The results showed similar general learning effects (i.e. improved accuracy and shorter response times with extended practice) as was found in previous studies with the go/nogo DSP task. The concatenation phenomenon also appeared to be evident and comparable to previous results with the go/nogo DSP task. In contrast, post-error slowing appeared to support a non-functional account for the go/nogo DSP task, whereas it was found of a functional account in the original DSP task. In conclusion, the replication of general learning phenomenon shows that the go/nogo DSP task is robust, but new insight to error-related cognitive processes should be further investigated.

Contents

1. Introduction

Human behaviour is mostly composed of sequences of actions which relate to each other to facilitate daily activities and motor skill learning. Motor sequence learning (MSL) is an essential skill across the lifespan that underscore activities such as learning languages and playing musical instruments (Janacsek & Nemeth, 2012). It is commonly known that practicing motor sequences, such as learning to play the piano, may be difficult at first but could be performed effortlessly after enough practice and experience. Gaining proficiency in a movement, often leads to a reduction of conscious attention usage and mental effort when performing that movement (Schmidt, 1987). The storing and recalling of motor sequences in short- and long-term memory are also important aspects that affect MSL. It is believed that the successful storage and recall of these sequences is necessary for efficient and accurate motor performance (Hawkins, George, & Niemasik, 2009). In a broader sense, being good at MSL is also associated with an ability to anticipate and predict future situations (Friston et al., 2016), which makes MSL an important skill for carrying out evolutionary adaptive behaviours in humans (Janacsek & Nemeth, 2012).

One of the most widely used paradigms to study the foundations of MSL is the Discrete Sequence Production (DSP) task (Verwey, 1999). Learners respond to a series of spatially mapped stimuli over a series of training blocks in the original DSP task as quickly as possible. Overtime, presented motor sequences are typically performed with increasing temporal and spatial accuracy, with different cognitive and motor processes involved in developing this sequential representation (Verwey, Shea, & Wright, 2015).

Several theoretical models that outline the MSL phenomenon are covered in the next section. The original DSP task design requires immediate responding when stimuli are presented and leads to a situation where the two processes of action preparation and action execution occur temporally close to each other. This is possibly a limitation when performing cortical measurements during MSL, for example when using electroencephalogram (EEG), as it raises issues of confounds between cognitive and motor-related processes. To detangle stimuli and response processes, a go/nogo version of the DSP task was previously successfully implemented (De Kleine & Van der Lubbe, 2011). The present study aimed to replicate findings from the go/nogo DSP task in comparison to usual learning patterns from the original DSP task. In addition, it would be important to also highlight any differences in cognitive processing and execution with relation to accuracy in the go/nogo DSP task, since the temporal design is changed and might affect how errors may be utilized.

1.1 The Discrete Sequence Production task

The original DSP task is a robust way to investigate MSL and is used to study the building blocks of complex action patterns (Abrahamse, Ruitenberg, De Kleine, & Verwey, 2013). The DSP task involves participants placing eight fingers on a keyboard and responding to short series of stimuli by rapidly pressing the spatially comparable key. Typically, it contains two fixed and equally long key press sequences of three to seven stimuli (Abrahamse et al., 2013). Because of the rapidity in which single sequence movements can be executed, response times (RTs) are sensitive and important indicators of cognitive control and motor-related processes (Rhodes, Bullock, Verwey, Averbeck, & Page, 2004).

Verwey (1999) suggested two processing stages that occur during the DSP task. The first stage known as the response selection stage involves a decision to select which sequence should be performed. The second stage known as the sequence execution stage involves the translation of the code retrieved from the response selection stage into a certain order of individual movements that the sequence entails. With practice, performing these sequences are speeded and executed as one integrated unit (Verwey, 1999). This unit is called a motor chunk, which is a memory code for selection, initiation, and execution of multiple individual movements as one (Verwey, 1994). Motor chunking will be covered more in depth in a later section.

Typically, the two fixed sequences that are used in a DSP task start with a different stimulus. This allows the participants to gradually learn to execute an entire sequence prompted only by the display of the first stimulus. The first key press response is generally slower than all subsequent responses. This slower first element arises because individual movements are searched and loaded into the motor buffer before sequence initiation takes place (Verwey, 1999; Abrahamse et al., 2013) and it takes time during the response selection stage to decide which movement sequence should be executed. Due to limitations in working memory (Miller, 1956; Cowan, 2000), only four to five elements are believed to be loaded into the motor buffer. Because of this, a slower RT typically arises around the fourth key press of a six-key sequence (Abrahamse et al., 2013). It suggests that this slower response may be indicative for the loading of the next motor chunk halfway through the sequence (Verwey, 2001). This slower response is also referred to as the concatenation point and facilitates smooth performance of a longer sequence. With shorter sequences (≤ five key presses) the concatenation point is generally not observed (Abrahamse et al., 2013). The concatenation phenomenon was also found during the go/nogo DSP task (De Kleine & Van der Lubbe, 2011).

1.2 Cognitive control of motor sequence learning

In MSL, there is a distinction between three types of execution modes that support sequence development: the reaction mode, the associative mode, and the chunking mode (Verwey & Abrahamse, 2012). When a sequence is relatively new, sequence information is consciously loaded into the motor buffer for execution, which is often referred to as reaction mode (Abrahamse et al., 2013). In this mode, the keying sequence is executed by converting each key-specific stimulus into the correct response. The reaction mode is typically used for unfamiliar sequences and makes use of encoding at a low level by composing stimulus-response (S-R) bindings of individual movements (Abrahamse et al., 2013). As learning ensues, relationships between correct responses develop and the learner becomes more familiar with the sequences, and therefore starts to create sequence representations leading to the utilization of the associative mode. This mode appears to develop automatically when the responses are repeatedly given in similar order (Verwey & Abrahamse, 2012).

Finally, when there is a sufficiently strong sequence representation, minimal stimuli information may be enough to directly trigger the motor buffer content and execute the movement sequences in an autonomous and rapid manner (Verwey, 1999). This is known as the chunking mode, whereby response-to-response bindings (R-R) that make up motor chunks are preloaded into the motor buffer without the need for stimulus feedback, and accurate anticipation of future responses is possible (Schmidt, 1987). This process is also described in the Dual Processor Model (DPM). The DPM assumes that a cognitive processor and a motor processor are intimately connected for responding (Verwey, 2001). The cognitive processor is engaged in preparing a sequence segment, i.e. individual movements one by one or a motor chunk, and loads this segment into the motor buffer. Next, the motor processor takes over and executes the content of the motor buffer (Verwey, 2001). Since the cognitive and motor processor work alongside each other, the anticipation and preparing of the next motor segment may already start while still performing the previous response (Eimer, Goschke, Schlaghecken, & Sturmer, 1996). It is assumed that loading the motor buffer with a motor chunk requires less time compared to individual movements that are selected and loaded one by one (Verwey, 1996). Hence, as the chunking mode develops with extended practice, it can be assumed that the execution rate of sequences will also increase and that RTs become shorter. Behavioural results of both the original DSP task (e.g. Verwey, 1996; Verwey, 1999) and the go/nogo DSP task (De Kleine & Van der Lubbe, 2011) support this assumption that the performance of sequences becomes faster with practice.

Another important hallmark of MSL are the changing accuracy levels in relation to shorter or longer RTs. In some situations, the participants' RT becomes shorter at the cost of decreasing accuracy levels, whereas in other cases the participants' RT becomes longer in order to improve their accuracy level (Proctor & Vu, 2003). In other words, sometimes participants favour accuracy over going faster. This phenomenon is also called the *speed-accuracy tradeoff*. Verwey (2010) showed that elderly had relative longer execution times than younger adults, but improved more in terms of accuracy. This indicates amongst others that elderly focus more on accuracy rather than execution speed during the DSP task. In contrast to these findings, results of a go/nogo DSP task showed an increase of correct responses alongside shorter RT with extended practice (Sobierajewicz, Przekoracka-Krawczyk, Jaśkowski, Verwey, & Van der Lubbe, 2017). Since different results were found between the two versions of the DSP task, the cognitive control mechanisms that are supporting this kind of behaviour need to be further investigated.

1.3 Post-error slowing and accuracy

Another phenomenon that occurs during MSL is post-error slowing, which is the tendency for people to respond slower after an erroneous trial than after a correct trial (Rabbitt, 1966; Notebaert et al., 2009; Danielmeier & Ullsperger, 2011; Ruitenberg, Abrahamse, De Kleine, & Verwey, 2014). Houtman and Notebaert (2013) made a distinction between functional and non-functional theories for understanding post-error slowing. Functional theories hypothesize that error processing and consecutive adjustments are intended to improve performance on the following trials. Post-error slowing is functional from this perspective since it increases the attention on response accuracy in order to avoid future errors. Such results were found in the study of Ruitenberg et al. (2014) using the original DSP task, where accuracy increased after making an error in six-key sequences. In contrast, non-functional theories suggest a decrease in post-error slowing and post-error accuracy due to reduced cognitive processing after erroneous trials. The bottleneck error-monitoring account postulates that error monitoring requires time and resources from a central information processor with limited capacity (Jentzsch & Dudschig, 2009). This leads to less accurate and slowed performance when a trial shortly follows an error. Another non-functional theory is the orienting account (Notebaert et al., 2009). This theory claims that post-error slowing occurs as an orienting response to errors. Errors are infrequent events and therefore steer attention away from the task, which has the consequence of disturbed performance on the next trial. Typically, there is a decreased accuracy dip after making an error due to the attentional orientation towards that error (Houtman & Notebaert, 2013).

1.3 The present study

This study is aimed at replicating previous findings of behavioural phenomena in the go/nogo DSP task and further understand key differences from the original DSP task. The first goal was to show that the go/nogo DSP task is robust and replicates that learners acquire general learning ability and accuracy improvements. Secondly, the aim was to further ascertain that the concatenation phenomena was also robust in the go/nogo DSP task compared to the original DSP task. Lastly, it was sought to understand if post-error slowing during a go/nogo DSP task was different in comparison with the original DSP task, because of the separation of stimuli and response processes which might affect cognitive control during MSL. It is predicted that response times will become shorter as learning ensues and that accuracy improves with practice, similar to the results of De Kleine and Van der Lubbe (2011), and Sobierajewicz et al. (2017). It is also predicted that concatenation phenomenon will be evident in the go/nogo DSP task (De Kleine & Van der Lubbe, 2011). With regards to post-error slowing, it is predicted that post-error trials will be slower than post-correct trials, and that post-error accuracy will increase after making an error, providing support for the functional account for error-slowing in MSL (Ruitenberg et al., 2014).

2. Methods and material

2.1 Participants

Thirty-one participants (19 females, 21.5 ± 2.4 years) were recruited from a pool of test subjects through Sona Systems from the BMS department at the University of Twente. The criterion for participation were: participants had to be naturally right-handed; no professional training in musical instruments or proficiency in gaming; no neurological, psychological, or psychiatric disorders; no depression or anxiety disorders; no sleep problems; no substance addictions; no cognitive impairment (e.g. mild attentional disorders); and no physical injuries or impairments.

The research protocol was approved by the BMS Ethics Committee/Domain Humanities and Social Sciences of the University of Twente with number BCE200776. Prior to the participation in the study, the participants signed a written informed consent form. Participants received four SONA credits to compensate for their time performing the 3.5 hour experiment.

2.2 The go/nogo Discrete Sequence Production task

In the go/nogo DSP task the screen displayed in total eight 30 x 25 mm squares with a black background. In the middle of the screen a fixation cross was shown together with eight squares that were horizontally aligned, four on the left side of the fixation cross and four on the right side. At the beginning of each sequence, the squares were filled with a black background colour with a white colour outline considered blank squares. After 1000 ms of trial onset, the squares lit up in yellow colour successively in a specific sequence order with a time interval of 750 ms in between, either on the right or on the left side. Next, there was a preparation interval of 1500 ms where the initial blank squares were shown. Afterward, the fixation cross lit up either blue (92%) or red (8%). The blue fixation cross was a cue for the participants to press the associated key with their index-, middle-, ring-, or little finger on their right or their left hand (go-signal). If the fixation cross lit up red (nogo-signal), they had to wait for the next sequence to start without responding. The keys used for the task were ‘A’, ‘S’, ‘D’, and ‘F’ for the left hand and ‘J’, ‘K’, ‘L’, and ‘;’ for the right hand. All steps explained above are shown in Figure 1.

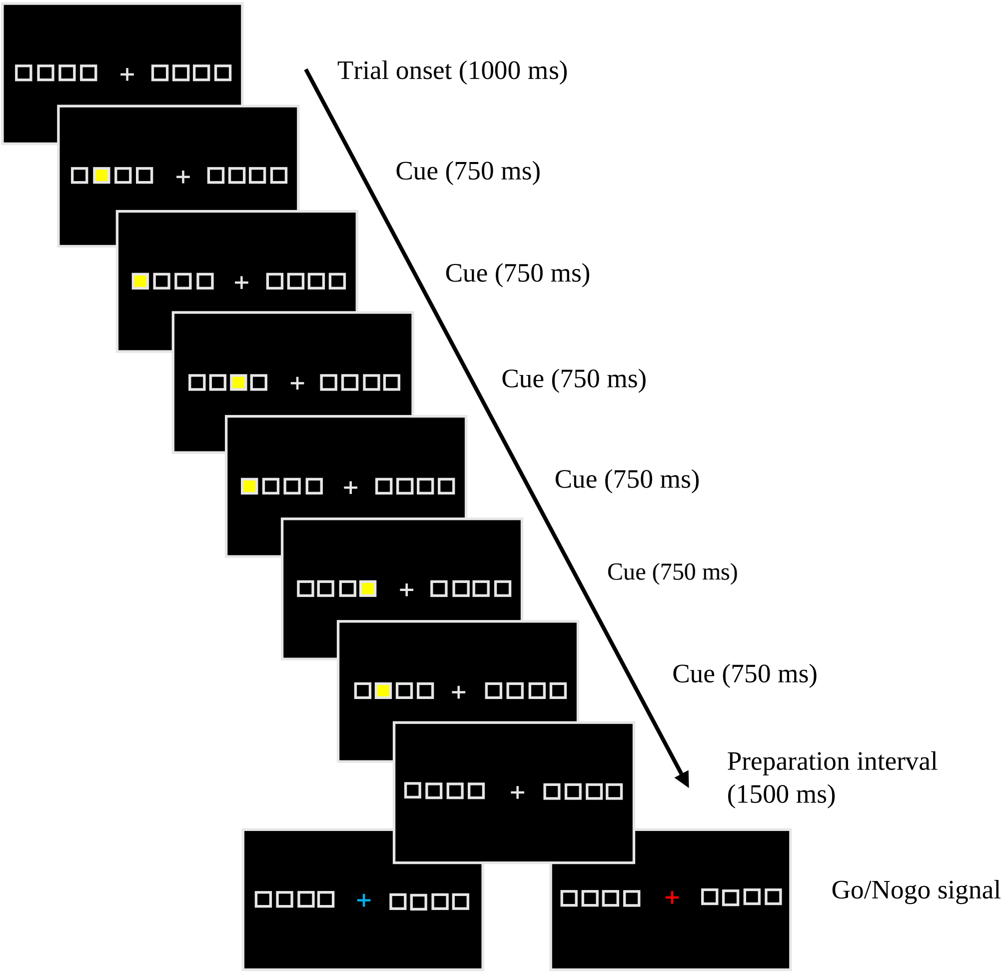

Figure 1. An example of a six-key sequence from the trial onset until the go/nogo signal, with the duration of each stimulus on the time axis.

During the blocks, a break of 20 seconds was given after 24 sequence presentations. When the participant responded prior to the go-signal, a ‘Too early’ message appeared on the screen. After each sequence the participant received either the message ‘Good!’ if the whole sequence was performed correctly, or feedback on which key presses were incorrect. At the end of each block, total number of errors and the average key press RT was presented on the screen.

The key presses in the sequences were counterbalanced across positions for all participants, with the purpose of avoiding any finger-specific effects on RTs. For instance, one participant performed the sequence ADFSDA, whereas another participant performed FSASDA (see Appendix A for full counterbalancing). Each participant executed five blocks of six-key sequences, each block consisting of 48 trials. In total 1440 key presses were performed across both hands. One block was composed of two different fixed sequences which were displayed in either their original order (e.g. ADFSDA and SADAFS) or as their mirror image (e.g. ;KJLK; and L;K;JL). The original sequences were executed randomly across both hands, but shown equally twelve times each.

In addition, electrophysiological recordings were also recorded during the experiment. However, the current project is focused on the behavioural analysis and the electrophysiological findings are the subject of another study.

2.3 Procedure

At the beginning of the experiment, the participant received an information sheet, provided informed consent, and filled out an EEG questionnaire and a handedness questionnaire to indicate their hand preferences. Once consent was given, the participants were equipped with an EEG cap from EasyCap to record cortical activity. The experiment was explained to the participants and they also received written instructions on the computer monitor about the go/nogo DSP task. Once the task was clear for the participant, the go/nogo DSP task began by entering the participant number and block number in the computer. In between blocks the participants had a short break of approximately one minute. The entire experiment took generally 3.5 hours for each participant to complete.

2.4 Apparatus

The go/nogo DSP task was conducted using E-Prime® Version 2.0.10.356. For the experiment two Dell UltraSharp U2518D monitors were used, one as the presenter PC and the other as the recorder PC (for the EEG recordings) and both ran Windows 7. A generic QWERTY keyboard was used for responding to the stimuli.

2.5 Data analysis

2.5.1 Response parameters

Response time (RT) was defined as the time between the go-signal and pushing down all six keys in a sequence. The Sequence RT was therefore calculated as the sum of all the individual key press RTs. The Sequence accuracy was computed by coding sequences with only all accurate key presses as ‘correct’, and each trial with one or more inaccurate key presses was coded as ‘incorrect’. The Accuracy percentage was determined by taking the proportion of all ‘correct’ trials within the block for each participant.

Post-error slowing was analysed using the methodology specified in Ruitenberg et al. (2014). All trials were coded into post-error trials (i.e. the single accurate sequence that directly followed a sequence with an incorrect key press) and post-correct trials (i.e. the single accurate sequence that directly followed another correct sequence). Subsequently, to investigate if accuracy was increasing or decreasing after making an error, the variable Accuracy (Normalized Units; N.U.) was computed with arcsine square root transformations to acquire normally distributed accuracy proportions, which is often recommended for binomial data (Winer, Brown & Michels, 1991; Lin & Xu, 2020).

2.5.2 Linear mixed-effects models

Data analysis was performed in the RStudio environment Ver. 1.3.959 with the lme4 package Ver. 1.1-27. Linear mixed-effects models (MEMs) were utilized as they account for subject-level interclass variance. For each model type II Wald chi-square tests are given in order to compute the significance level of effects for each interaction. When a significant effect was evident in the MEMs, a post-hoc Tukey test was performed to determine the cause of the effect. In total five models were analysed.

The first model aimed to understand how accuracy changes during training. The outcome variable was Accuracy percentage with predictor variable: Block (1 to 5). The second model aimed to assess the influence of Sequence accuracy during training. The outcome variable was Sequence RT (ms) with two predictor variables: (1) Block (1 to 5), and (2) Sequence accuracy (correct vs. incorrect trials). The third model assessed concatenation and whether the RT for each key position was changing across blocks. For this model, the outcome variable was Key press RT (ms) with two predictor variables: (1) Block (1 to 5), and (2) Key position (1 to 6). The fourth model was used to determine the RT difference between post-error trials and post-correct trials across training blocks. The outcome variable was Sequence RT (ms) with two predictor variables: (1) Block (1 to 5), and (2) Trial type (post-error vs. post-correct trials). The fifth model assessed if the slowing that occurred was of a functional or non-functional nature by understanding accuracy changes following an error. The outcome variable was the arcsine transformed Accuracy (N.U.) with two predictor variables: (1) Block (1 to 5), and (2) Trial type (post-error vs. post-correct trials).

3. Results

3.1 Response time and accuracy performance

MEM analysis of the first model on Accuracy percentage revealed a significant interaction of Accuracy percentage x Block, $\chi^2(4, N = 31) = 211.8$, $p < .001$, demonstrating an increasing proportion of correct trials across the training blocks. The post-hoc Tukey test showed that this significant interaction resulted mostly from differences of Block 1 with Blocks 2 to 5 ($p < .001$). Differences between Block 2 and the remainder of the blocks were not evident ($p > .05$), which showed similar accuracy proportions (see Figure 2).

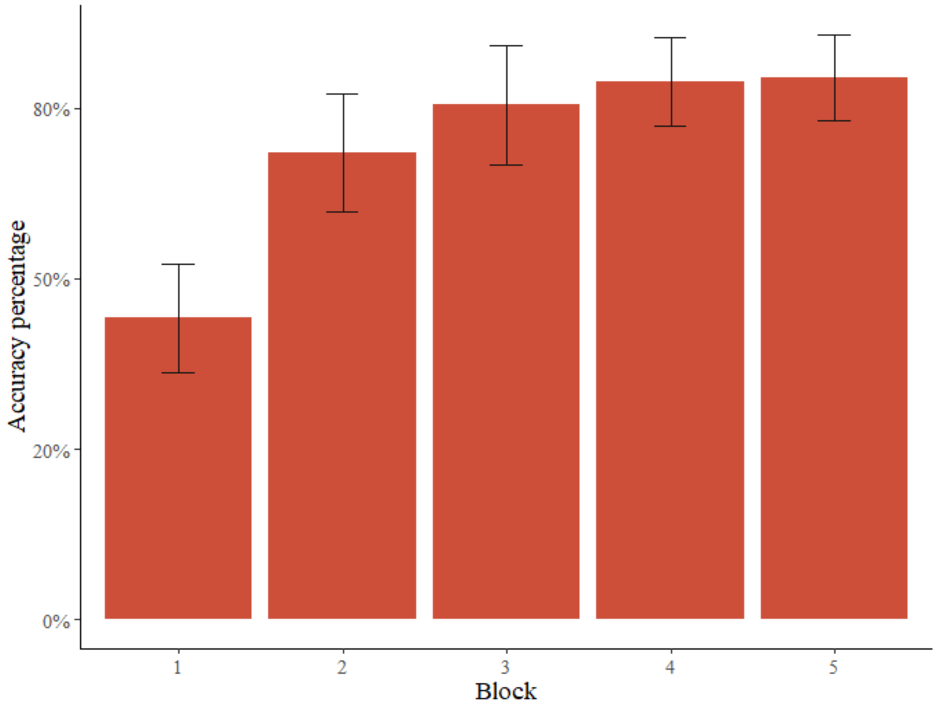

Figure 2. Accuracy percentage as a function per Block. Accuracy proportions increased for correct trials across the training blocks. The error bars represent 95% confidence intervals.

MEM analysis of the second model on Sequence RT performance showed a significant Block x Sequence accuracy interaction for Sequence RT, $\chi^2(4, N = 31) = 24.2$, $p < .001$. Firstly, post-hoc Tukey tests showed that across all training blocks, shorter RT performance was evident for accurate trials compared to inaccurate trails ($p < .001$). For accurate trials, Block 1 had significantly longer Sequence RT compared to Blocks 2 to 5 ($p < .001$); Block 2 had longer Sequence RT than Block 3 to 5 ($p < .05$); no other significant interactions between blocks were found. For inaccurate trials, Block 1 also had significantly longer Sequence RT compared to Blocks 2 to 5 ($p < .001$); Block 2 had longer Sequence RT than Block 3 and 4 ($p < .05$); no other significant interactions between the blocks were found. Figure 3 shows a trend in that inaccurate trials had increasingly longer Sequence RTs in latter training blocks, whilst accurate trials had shorter Sequence RTs in latter training blocks, although these differences were statistically insignificant ($p > .05$) (see Figure 3).

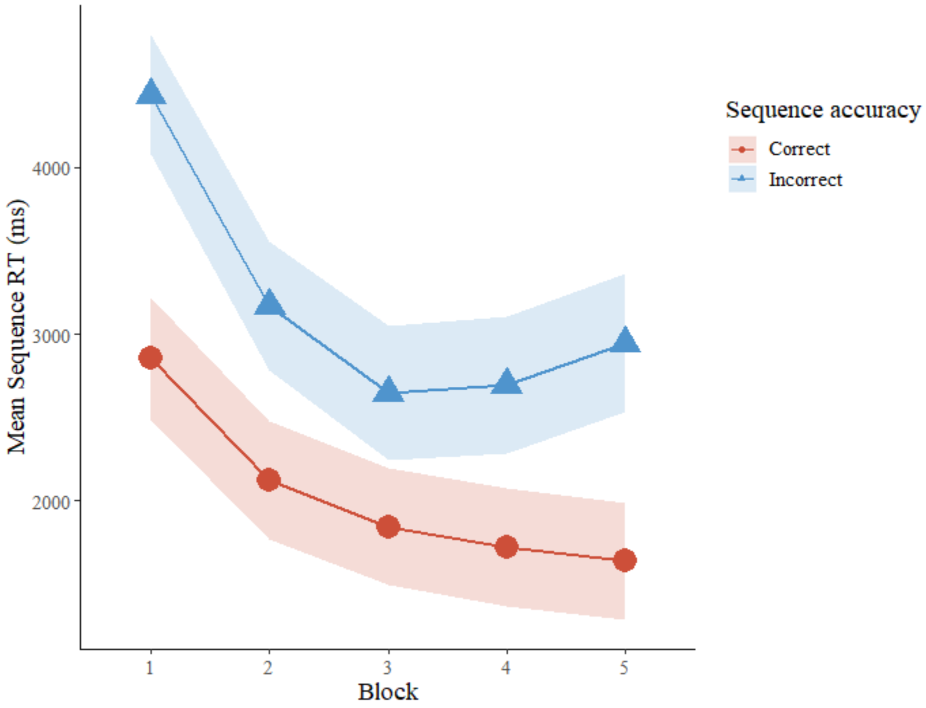

Figure 3. Mean Sequence RT as a function of Block and Sequence accuracy. Correct trials showed shorter RTs than incorrect trials. The error bars display 95% confidence intervals.

3.2 Concatenation phenomenon

MEM analysis of the third model on Key press RT performance revealed a significant Block x Key position interaction on Key press RT, $\chi^2(20, N = 31) = 53.9$, $p < .001$. The post-hoc Tukey tests within Key position revealed that for Key position 1, Block 1 showed longer Key press RT compared to Blocks 2 to 5 ($p < .001$); Block 2 showed longer RTs than Block 4 ($p < .01$); no other significant interactions between blocks for Key position 1 were found. For Key position 2, only Block 1 compared to Blocks 2 to 5 ($p < .001$) showed longer Key press RTs; no other significant interactions were found. Key positions 3, 4 and 5 showed similar interactions, where longer RTs for Block 1 compared to Blocks 2 to 5 ($p < .001$) were found; Block 2 showed longer Key press RTs than Block 3 to 5 ($p < .05$); and no other interactions were found between blocks for Key press positions 3, 4 and 5. Finally, Key position 6 revealed in Block 1 longer Key press RTs than for Blocks 2 to 5 ($p < .001$); Block 2 revealed longer RTs compared to Blocks 5 ($p < .05$); and no other interactions between blocks were found.

The post-hoc Tukey tests within Block, revealed that in the first block Key position 1 had a longer Key press RT than Key positions 2 to 6 ($p < .001$), and no other significant interactions between key positions in Block 1 were found. In Block 2, post-hoc results also showed a longer Key press RT for Key position 1 compared to Key positions 2 to 6 ($p < .001$); Key position 4 had longer RTs than Key position 2 ($p < .05$); Key position 3 had a longer Key press RT than Key position 6 ($p < .01$); Key position 4 had a longer RT than Key position 6 ($p < .001$); Key position 5 showed longer Key press RT than Key position 6 ($p < .05$); no other significant results between key positions in Block 2 were found. Finally, in Blocks 3, 4, and 5 the same interactions between Key positions were repeated. In these blocks, Key position 1 revealed significant longer Key press RTs than Key positions 2 to 6 ($p < .001$); Key position 4 showed longer RTs than Key position 6 ($p < .01$), and no other significant interactions within Blocks 3, 4, and 5 where found (see Figure 4).

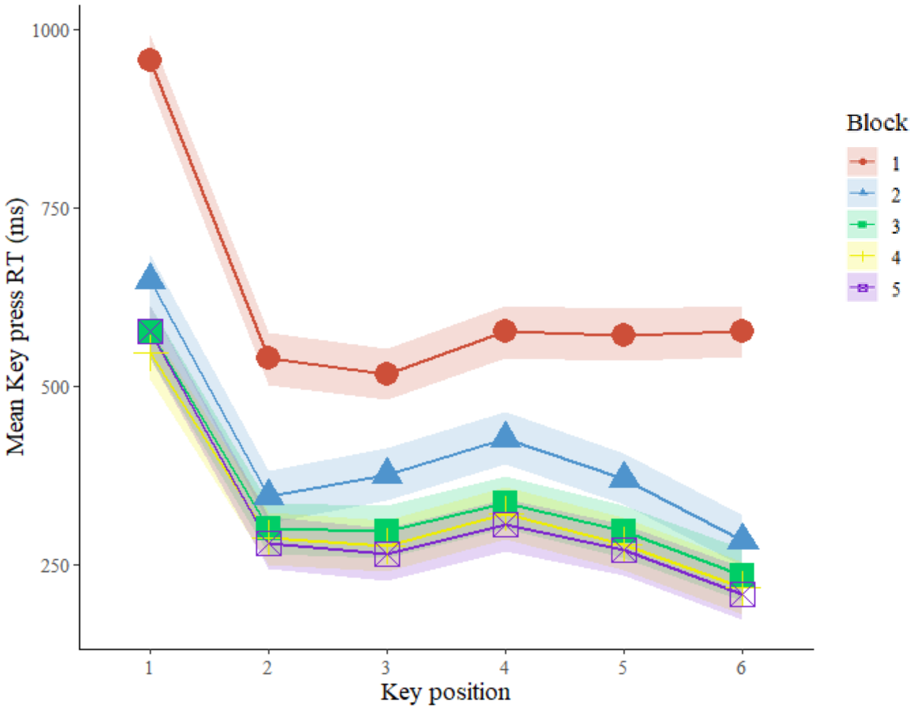

Figure 4. Mean Key press RT as a function of Key position within the sequence and Block. Key position 4 consistently has longer RTs than Key position 6 from Block 2 to 5, signalling that a concatenation phenomenon was evident. The error bars represent the standard error.

3.3 Post-error slowing and accuracy

Results of the MEM analysis of the fourth model on Trial type performance showed a significant Block x Trial type interaction for Sequence RT, $\chi^2(4, N = 31) = 17.1$, $p < .01$. The post-hoc Tukey test between post-error and post-correct trials showed a significant difference between the trial types in Block 1 ($p < .001$), Block 2 ($p < .01$), Block 3 ($p < .01$), and Block 5 ($p < .05$). Block 4 showed no difference ($p > .05$). This revealed a significant interaction of post-error and post-correct sequences on sequence RT per block, where post-error sequences were executed slower than post-correct sequences. Post-hoc results within the blocks for post-error trials revealed that Block 1 showed longer Sequence RT than Blocks 2 to 5 ($p < .001$); Block 2 revealed longer Sequence RT than Blocks 3 to 5 ($p < .05$); Block 3 showed longer Sequence RT compared to Blocks 4 and 5 ($p < .05$); no significant interaction was found between Block 4 and Block 5 ($p > .05$). For post-correct trials, Block 1 revealed longer Sequence RT than Blocks 2 to 5 ($p < .001$); Block 2 showed longer Sequence RT compared to Blocks 3 to 5 ($p < .001$); Block 3 revealed longer Sequence RT than Blocks 4 and 5 ($p < .05$), and no significant interaction was found between Block 4 and Block 5 ($p > .05$).

MEM analysis of the fifth model on arcsine transformed Accuracy (N.U.) performance showed a significant Trial type x Block interaction on arcsine transformed accuracy scores, $\chi^2(4, N = 31) = 75.3$, $p < .001$, revealing a difference in accuracy between the trial types across the blocks. The post-hoc Tukey test showed that this interaction resulted from differences between post-error and post-correct trials between Block 2 to Block 5 ($p < .001$), and that there was no significant difference in Block 1 ($p > .05$). Post-error accuracy continued to decrease throughout learning, whereas post-correct accuracy continued to increase throughout learning (see Figure 5).

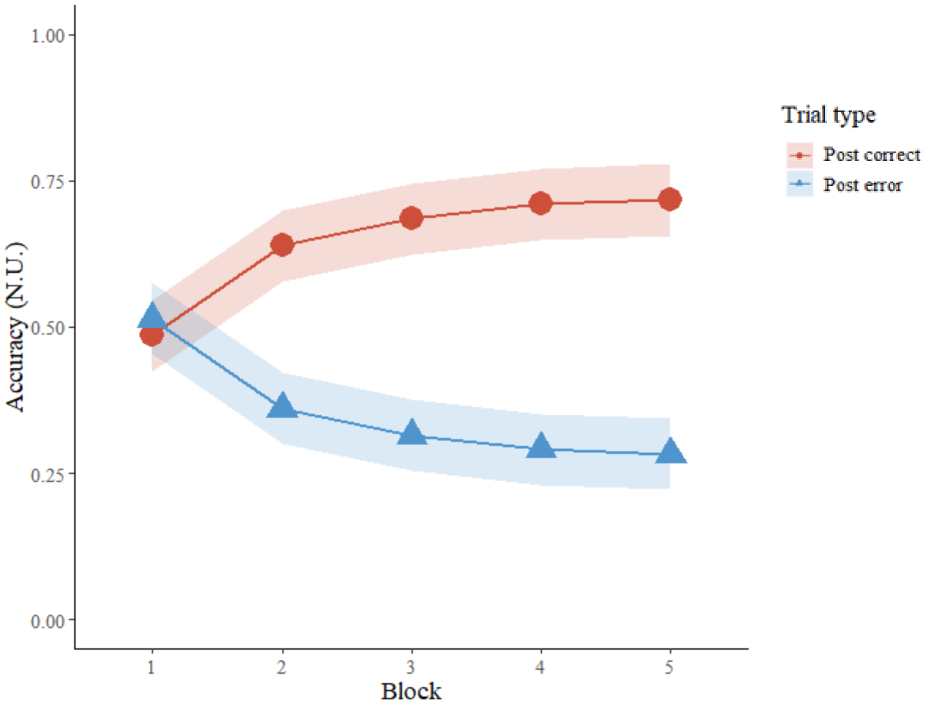

Figure 5. Accuracy (N.U.) as a function of Block. Post-error accuracy decreased, whereas post-correct accuracy increased with more training. Error bars display 95% confidence intervals.

4. Discussion

The purpose of this study was to replicate previous findings of behavioural phenomena (i.e. general learning abilities, concatenation, and error-slowing). The first aim was to replicate findings that participants acquired general learning abilities, i.e. faster responses and accuracy improvements, in the go/nogo DSP task that resulted from studies of De Kleine and Van der Lubbe (2011) and Sobierajewicz et al. (2017). It was predicted that accuracy would improve with extended practice. Current results showed that the overall proportion of correct trials was increasing across blocks (see Figure 2). Even though this result may not be typically found for the original DSP task (Verwey, 2010; Verwey et al., 2011), Sobierajewicz et al. (2017) showed similar results for the go/nogo DSP task. This difference in results between the original DSP task and the go/nogo DSP task may possibly be explained by the fundamental difference in design of both tasks. During an original DSP task, the participants immediately press the corresponding key when a square lights up on the screen. However, during a go/nogo DSP task the participants are presented with the complete sequence and are required to execute the sequence after a go-signal. After this execution they immediately receive feedback on possible errors. Because of this go/nogo design, an extensive time period is created between the execution of two subsequent trials (in this case ± 7000 ms, variable based on the amount of errors). This buffer time is possibly used in a strategic manner for priming one's cognitive control system and reorientate attention on the next subsequent trial (Koechlin & Summerfield, 2007). In doing so, it may be that participants have increased time to better plan and prepare for the next trial with faster responses and increased accuracy.

Besides accuracy proportions, it was predicted that participants' response times would become shorter with extended training. The results show that participants became faster at executing correct sequences as learning ensued (see Figure 3). The results support the notion that participants started to prioritise usage of the chunking mode from Block 2 since most difference in Sequence RT was found between Block 1 and 2, on which learning appeared to be predicated. Sequence RTs for the remainder of the training blocks were not different for correct trials. It appears that the results of general learning are comparable to both the original DSP task (e.g. Verwey, 1996; Verwey, 1999) and a previous studies with the go/nogo DSP task (De Kleine & Van der Lubbe, 2011). It is worth noting that RTs for inaccurate responses in the final two learning blocks became slower. This hints towards some form of erroneous response slowing and that post-error slowing accounts may be responsible which is covered in a later paragraph.

The second aim of this study was to further ascertain that the concatenation phenomenon was robust in the go/nogo DSP task compared to the original DSP task, and it was predicted that concatenation would be evident in the go/nogo DSP task, similar to the results of De Kleine and Van der Lubbe (2011). The results are suggestive that in the first block the first three key presses were prepared before execution, whereas the latter three key presses were prepared and loaded one key press at a time. It is possible that participants were switching between the reaction mode and the associative mode, since the unfamiliar sequences in the first block still made use of encoding at a lower level through composition of S-R bindings of individual movements (Abrahamse et al., 2013). The predominant use of the reaction mode and associative mode seemed evident only in the first block, since there was a significant improvement in RTs by the second block and not for the remainder of the blocks. This rapid learning phenomenon may be due to the previously mentioned buffer time in the go/nogo DSP task, which helps with sufficient time for reorientating and resetting. It is possible that participants already started to utilise chunking mode in consecutive Blocks 2 to 5 and became increasingly proficient.

In the second block a significant increase in RT is found between the second and the fourth key press, and in Blocks 2 to 5 a significant decrease between the fourth and the sixth key press was shown. The comparisons within Blocks and Key positions show that concatenation stays the same and that key positions become faster across blocks. When comparing this result to previous work (Verwey, 1999; Verwey, 2001; De Kleine & Van der Lubbe, 2011), these results support the same typical phenomenon for concatenation in both the original as well as the go/nogo DSP task. Based on this observation, it can be assumed that due to limitations of working memory capacity (Miller, 1956; Cowan, 2000), the sequences are divided into multiple motor chunks. It appears that the utilization of motor chunks is still clear as evidenced by a longer RT for the fourth key position and a shorter RT for the sixth key position in the latter blocks. In general, the results replicate previous work and show that overall smooth and fast execution of an entire sequence relies on using a slower middle position to preload working memory for the subsequent chunk.

The third aim was to understand if post-error slowing during a go/nogo DSP task was different in comparison with the original DSP task by comparing the results to the findings of Ruitenberg et al. (2014). It was predicted that the results would support the functional account for error-slowing. Current results revealed that post-error accuracy decreased throughout learning, while post-correct accuracy continued to increase (see Figure 4). This is opposed to what Ruitenberg et al. (2014) showed in which they supported functional error-slowing, where post-error accuracy increases at the cost of a slower RT. The current results lend support to a more non-functional error-slowing account, and especially implies influence of the bottleneck error-monitoring account. This non-functional theory assumes less accurate and slower responses because of error monitoring which requires time and additional recourses from a capacity-limited information processor (Jentzsch & Dudschig, 2009). One would expect that non-functional slowing account would not be the dominant mechanism during a go/nogo DSP task, due to the extensive buffer time period between trials where possible erroneous conflicts can be resolved before the next trial starts. However, two other important factors were highlighted in that accuracy rates were increasing and that inaccurate and slower trials were found in the last two training blocks. This would mean that inaccurate trials later in training could be cognitively more costly to re-orientate towards accuracy for subsequent trials (Notebaert et al., 2009). Another possible explanation could be that fatigue played a role in combination with the resetting of the cognitive control system for the slower inaccurate trials (Brandscheidt et al., 2019), which is also linked to the orienting account. It is possible that participants become more tired as the experiment progressed, leading to more required cognitive resources responding correctly.

In short, the results of the current study support the findings for general learning (faster responses (De Kleine & Van der Lubbe, 2011) and improved accuracy (Sobierajewicz et al., 2017)), and concatenation (De Kleine & Van der Lubbe, 2011) for the go/nogo DSP task. However, opposed results were found regarding post-error slowing, which appear to be non-functional for the go/nogo DSP task, i.e. decreased post-error slowing and post-error accuracy, compared to functional error-slowing in the original DSP task (Ruitenberg et al., 2014).

4.1 Limitations and future directions

Some limitations about the present study and future recommendations can be proposed. Regarding the analysis of error-slowing, the results were analysed by using sequence RTs which was computed with the sum of the six key press RTs. The errors are happening within the sequence, so there is no indication where this erroneous response takes place. Therefore, a future study could focus on analysing error-slowing on key press level during a go/nogo DSP task. In addition, considering to perform this experiment with a different population would perhaps give different results. For example, using the population of elder adults may show different learning approaches compared to younger adults, as used in this study (Barnhoorn, Van Asseldonk, & Verwey, 2019). This could be interesting for understanding MSL as presented in the go/nogo DSP task for various individuals. Lastly, using cortical imaging methods to understand the differences between the behavioural patterns used in go/nogo DSP task and the original DSP task, may give more evidence about cognitive control and mechanisms of stimulus-related processes compared to response processes during MSL. These limitations and future suggestions all aim to better understand MSL and the general learning improvement of the behavioural phenomena during a go/nogo DSP task.

4.2 Conclusions

The current study replicated previous findings of behavioural phenomena in the go/nogo DSP task and uncovered insights to differences from the original DSP task. The results were similar regarding general learning effects (i.e. improved accuracy and shorter response times with extended training) and the concatenation phenomenon that were both found in previous studies with the go/nogo DSP task. However, post-error slowing appeared to be of a non-functional account for the go/nogo DSP task, whereas it was found of a functional account in the original DSP task. In conclusion, the replication of general learning phenomena shows that the go/nogo DSP task is robust, but new insight to error-related cognitive processes should be further investigated.

5. References

Abrahamse, E. L., Ruitenberg, M. F. L., De Kleine, E., & Verwey, W. B. (2013). Control of automated behaviour: insights from the discrete sequence production task. *Frontiers in human neuroscience, 7*(82), https://doi.org/10.3389/fnhum.2013.00082

Barnhoorn, J. S., Van Asseldonk, E. H. F., & Verwey, W. B. (2019). Differences in chunking behavior between young and older adults diminish with extended practice. *Psychological Research, 83*(2), 275-285. https://doi.org/10.1007/s00426-017-0963-6

Branscheidt, M., Kassavetis, P., Anaya, M., Rogers, D., Huang, H. D., Lindquist, M. A., & Celnik, P. (2019). Fatigue induces long-lasting detrimental changes in motor-skill learning. *eLife, 9,* 1-25. DOI: 10.7554/eLife.40578

Cowan, N. (2000). The magical number 4 in short-term memory: a reconsideration of mental storage capacity. *Behavioural and Brain Sciences, 24*(1), 87-114. DOI:10.1017/S0140525X01003922

Danielmeier, C., & Ullsperger, M. (2011). Post-error adjustments. *Frontiers in Psychology, 2*, 233-243. https://doi.org/10.3389/fpsyg.2011.00233

De Kleine, E., & Van der Lubbe, R. H. J. (2011). Decreased load on general motor preparation and visual-working memory while preparing familiar as compared to unfamiliar movement sequences. *Brain and Cognition, 75,* 126-134. doi:10.1016/j.bandc.2010.10.013

Eimer, M., Goschke, T., Schlaghecken, F., & Sturmer, B. (1996). Explicit and implicit learning of event sequences: Evidence from event-related brain potentials. *Journal of Experimental Psychology, 22*(4), 970-987. DOI: 10.1037/0278-7393.22.4.970

Friston, K., FitzGerald, T., Rigoli, F., Schwartenbeck, P., O□Doherty, J., & Pezzulo, G. (2016). Active inference and learning. *Neuroscience & Biobehavioral Reviews, 68*, 862-879. https://doi.org/https://doi.org/10.1016/j.neubiorev.2016.06.022

Hawkins, J., George, D., & Niemasik, J. (2009). Sequence memory for prediction, inference and behaviour. *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, *364*(1521), 1203–1209. https://doi.org/10.1098/rstb.2008.0322

Houtman, F., & Notebaert, W. (2013). Blinded by an error. *Cognition, 128*(2). 228-236. https://doi.org/10.1016/j.cognition.2013.04.003

Janacsek, K., & Nemeth, D. (2012). Predicting the future: From implicit learning to consolidation. *International Journal of Psychophysiology, 83*(2), 213-221. https://doi.org/10.1016/j.ijpsycho.2011.11.012

Jentzsch, I., & Dudschig, C. (2009). Why do we slow down after an error? Mechanisms underlying the effects of post-error slowing. *Quarterly Journal of Experimental Pschology, 62*, 209-218. https://doi.org/10.1080/17470210802240655

Koechlin, E., & Summerfield, C. (2007). An information theoretical approach to prefrontal executive function. *Trends in Cognitive Sciences, 11*(6), 229-235. https://doi.org/10.1016/j.tics.2007.04.005

Lin, L., & Xu, C. (2020). Arcsine-based transformations for meta-analysis of proportions: Pros, cons, and alternatives. *Health Science Reports, 3*(3), e178. https://doi.org/10.1002/hsr2.178

Miller, G. A. (1956). The magical number seven, plus or minus two: some limits on our capacity for processing information. *Psychological Review, 63*(2), 81-97. https://doi.org/10.1037/h0043158

Notebaert, W., Houtman, F., Van Opstal, F., Gevers, W., Fias, W., & Verguts, T. (2009). Post-error slowing: an orienting account. *Cognition, 111*(2), 275-279. DOI: 10.1016/j.cognition.2009.02.002

Proctor, R. W., & Vu, K.-P. L. (2003). Action selection. In A. F. Healy & R. W. Proctor (Eds.), *Experimental psychology* (pp. 293-316). New York, NY: Wiley.

Rabbitt, P. M. (1966). Errors and error correction in choice-response tasks. *Journal of Experimental Psychology, 71*(2), 264-272. https://doi.org/10.1037/h0022853

Rhodes, B. J., Bullock, D., Verwey, W. B., Averbeck, B. B., & Page, M. P. A. (2004). Learning and production of movement sequences: Behavioural, neurophysiological, and modelling perspectives. *Human Movement Science, 23*(5), 699-746. https://doi.org/10.1016/j.humov.2004.10.008

Ruitenberg, M. F. L., Abrahamse, E. L., De Kleine, E., & Verwey, W. B. (2014). Post-error slowing in sequential action: an aging study. *Frontiers in Psychology*, *5*, 199. https://doi.org/10.3389/fpsyg.2014.00119

Schmidt, R. A. (1987). The acquisition of skill: Some modifications to the perception-action relationship through practice. In Heuer, H., & Sanders, A. F. (Eds.), *Perspectives on perception and action* (pp. 77-103). Hillsdale, NJ: Lawrence Erlbaum Associates.

Sobierajewicz, J., Przekoracka-Krawczyk, A., Jaśkowski, W., Verwey, W. B., & Van der Lubbe, R. (2017). The influence of motor imagery on the learning of a fine hand motor skill. *Experimental Brain Research, 235*, 305-320. https://doi.org/10.1007/s00221-016-4794-2

Verwey, W. B. (1994). Evidence for the development of concurrent processing in a sequential keypressing task. *Acta Psychologica, 85*(3), 245-262. https://doi.org/10.1016/0001-6918(94)90038-8

Verwey, W. B. (1996). Buffer loading and chunking in sequential keypressing. *Journal of Experimental Psychology: Human perception and performance, 22*(3), 544-562. https://doi.org/10.1037/0096-1523.22.3.544

Verwey, W. B. (1999). Evidence for a multistage model of practice in a sequential movement task. *Journal of experimental psychology: human perception and performance, 25*(6), 1693-1708. https://doi.org/10.1037/0096-1523.25.6.1693

Verwey, W. B. (2001). Concatenating familiar movement sequences: the versatile cognitive processor. *Acta Psychologica, 106*(1-2), 69-95. DOI: 10.1016/s0001-6918(00)00027-5

Verwey, W. B. (2010). Diminished motor skill development in elderly: indications for limited motor chunk use. *Acta Psychologica, 134*(2), 206- 214. DOI: 10.1016/j.actpsy.2010. 02.001

Verwey, W. B., & Abrahamse, E. L. (2012). Distinct modes of executing movement sequences: Reacting, associating, and chunking. *Acta Psychologica, 140*(3), 274-282. https://doi.org/10.1016/j.actpsy.2012.05.007

Verwey, W. B., Abrahamse, E. L., Ruitenberg, M. F. L., Jiménez, L., & De Kleine, E. (2011). Motor skill learning in the middle-aged: limited development of motor chunks and explicit sequence knowledge. *Psychological Research, 75,* 406-422. DOI 10.1007/s00426-011-0320-0

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*, 54-77. https://doi.org/10.3758/s13423-014-0773-4

Winer, B. J., Brown, D. R., & Michels, K. M. (1991). *Statistical principles in experimental design.* New York, NY: McGraw-Hill.

Appendix A

Counterbalancing across key positions

Participant	ID number	Six-key sequences	
		Original	Mirrored
1, 9, 14, 24	1	ADFSDA SADAFS	;KJLK; L;K;JL
2, 10. 18, 25	2	FSASDA ADSDAF	JL;LK; ;KLK;J
3, 11, 19, 28	3	SFADFS DSFSDA	LJ;KJL KLJLK;
4, 12, 26, 29	4	ADSDFS SFDFSA	;KLKJL LJKJL;
5, 15, 20, 27	5	DASFAD FDADFS	K;LJ;K JK;KJL
6, 16, 21, 30	6	SFDFAD DAFADS	LJKJ;K K;J;KL
7, 17, 22, 31	7	FSADAF AFSFAD	JL;K;J ;JLJ;K
8, 13, 23, 32	8	DAFASF FSASFD	K;J;LJ JL;LJK

Appendix B

RStudio script for data analysis

##Packages

```
library(readxl)
library(haven)
library(tidyverse)
library(lme4)
library(effects)
library(lattice)
library(car)
library(ggplot2)
library(knitr)
library(reshape2)
library(dplyr)
library(forcats)
library(DHARMa)
library(Hmisc)
library(phia)
library(lsmeans)
library(emmeans)
library(multcomp)
library(nlme)
```

##Preparing the data

#Importing data sets

```
require("readxl")
d.MSL <- read_xlsx("C:\\Users\\Hilde Althof\\Desktop\\6Key_Final_Hilde_Triallvl_210518.xlsx")
view(d.MSL)
attach(d.MSL)

d.MSL.pos <- read_excel("C:\\Users\\Hilde Althof\\Desktop\\6Key_Final_Hilde_210518_V1.xlsx")
view(d.MSL.pos)
attach(d.MSL.pos)

d.ACC <- read_excel("C:\\Users\\Hilde Althof\\Desktop\\6Key_Final_Hilde_ACC.proportion.xlsx")
view(d.ACC)
attach(d.ACC)

d.Error <- read_excel("C:\\Users\\Hilde Althof\\Desktop\\6Key_Final_Hilde_Error.xlsx")
view(d.Error)
attach(d.Error)

d.Arcsine <- read_excel("C:\\Users\\Hilde Althof\\Desktop\\6Key_Final_Hilde_Arcsine.xlsx")
```

```
view(d.Arcsine)
attach(d.Arcsine)
```

#Creating factors

```
d.MSL$Subject <- factor(d.MSL$Subject)
d.MSL$Block <- factor(d.MSL$Block)
d.MSL$Trial.ACC <- factor(d.MSL$Trial.ACC)
d.MSL$Hand.indicator <- factor(d.MSL$Hand.indicator)

d.MSL.pos$Subject <- factor(d.MSL.pos$Subject)
d.MSL.pos$Block <- factor(d.MSL.pos$Block)
d.MSL.pos$Position <- factor(d.MSL.pos$Position)

d.ACC$Block <- factor(d.ACC$Block)
d.ACC$Subject <- factor(d.ACC$Subject)

d.Error$Subject <- factor(d.Error$Subject)
d.Error$Error <- factor(d.Error$Error)
d.Error$Block <- factor(d.Error$Block)

d.Arcsine$Subject <- factor(d.Arcsine$Subject)
d.Arcsine$Block <- factor(d.Arcisne$Block)
d.Arcsine$TrialType <- factor(d.Arcsine$TrialType)
```

##Building the different models

```
#First model
m.MSL.ACCper <- lmer(ACC.per ~ Block + (1|Subject), data = d.ACC)
Anova(m.MSL.ACCper)
summary(m.MSL.ACCper)

#Second model
m.MSL.RTACC <- lmer(Trial.RT ~ Block * Trial.ACC + (1|Subject), data = d.MSL)
Anova(m.MSL.RTACC)
summary(m.MSL.RTACC)

#Third model
m.MSL.position <- lmer(feedback.RT ~ Position * Block + (1|Subject), data = d.MSL.pos)
Anova(m.MSL.position)
summary(m.MSL.position)

#Fourth model
m.MSL.Error <-lmer(Trial.RT ~ Block * Error + (1|Subject), data = d.Error)
Anova(m.MSL.Error)
summary(m.MSL.Error)

#Fifth model
m.MSL.Arcsine <- lmer(ACC.nu ~ TrialType * Block + (1|Subject), data = d.Arcsine)
Anova(m.MSL.Arcsine)
summary(m.MSL.Arcsine)
```

##Effects for each model

```
##Second model
#Need Effects lib
ae.m.MSL.RTACC <- allEffects(m.MSL.RTACC)
ae.m.df.RTACC <- as.data.frame(ae.m.MSL.RTACC[[1]])

#The main plot.
ae.m.MSL.RTACC <- allEffects(m.MSL.RTACC)
ae.m.df.RTACC <- as.data.frame(ae.m.MSL.RTACC[[1]])

#The main plot.
ae.RTACC<-ggplot(ae.m.df.RTACC, aes(x=Block,y=fit, group=Trial.ACC))+
  geom_ribbon(aes(ymin=lower, ymax=upper, fill=Trial.ACC), alpha=0.2) +
  geom_line(aes(size=1, color=Trial.ACC)) +
  geom_point(aes(color=Trial.ACC, shape=Trial.ACC, size=2))+
  ylab("Mean Sequence RT (ms)")+
  xlab("Block")+
  theme_classic()

#Printing Session effects facet
print(ae.RTACC) + scale_fill_manual(values = c(“tomato3”, “steelblue3”)) +
scale_color_manual(values = c(“tomato3”, “steelblue3”)) +
theme(text = element_text(size = 14, family = "serif"))

##Third model
#Need Effects lib
ae.m.MSL.position<-allEffects(m.MSL.position)
ae.m.df.position<-as.data.frame(ae.m.MSL.position[[1]])

#The main plot.
ae.position<-ggplot(ae.m.df.position, aes(x=Position,y=fit, group=Block))+
  geom_ribbon(aes(ymin=lower, ymax=upper, fill=Block), alpha=0.2) +
  geom_line(aes(size=1, color=Block)) +
  geom_point(aes(color=Block, shape=Block, size=2))+
  ylab("Mean Key press RT (ms)")+
  xlab("Key position")+
  theme_classic()

#Printing Session effects facet
print(ae.position) + scale_fill_manual(values = c(“tomato3”, “steelblue3”,springgreen3”,
“yellow2”, “purple3”)) +
scale_color_manual(values = c(“tomato3”, “steelblue3” + “springgreen3”,   “yellow2”,
“purple3”)) +
theme(text = element_text(size = 14, family = "serif"))

##Fifth model
#Need Effects lib
ae.m.MSL.Arcsine <- allEffects(m.MSL.Arcsine)
ae.m.df.Arcsine <- as.data.frame(ae.m.MSL.Arcsine[[1]])
```

```
#The main plot.
ae.Arcsine<-ggplot(ae.m.df.Arcsine, aes(x=Block,y=fit, group=TrialType))+
  geom_ribbon(aes(ymin=lower, ymax=upper, fill=TrialType), alpha=0.2) +
  geom_line(aes(size=1, color=TrialType)) +
  geom_point(aes(color=TrialType, shape=TrialType, size=2))+
  ylab("Accuracy (N.U.)")+
  xlab("Block")+
  theme_classic()

#Printing Session effects facet
print(ae.RTACC) + scale_fill_manual(values = c(“tomato3”, “steelblue3”)) +
scale_color_manual(values = c(“tomato3”, “steelblue3”) + ylim(0,1) +
theme(text = element_text(size = 14, family = "serif"))
```

##Post-hocs

```
#Summary of posthocs
summary(glht(m.MSL.ACCper, linfct = mcp(Trial.ACC = "Tukey")), test = adjusted("holm
"))
summary(glht(m.MSL.RTACC, linfct = mcp(Block = "Tukey")), test = adjusted("holm"))
summary(glht(m.MSL.position, linfct = mcp(Block = "Tukey")), test = adjusted("holm"))
summary(glht(m.MSL.Error, linfct = mcp(Block = "Tukey")), test = adjusted("holm"))
summary(glht(m.MSL.Arcsine, linfct = mcp(TrialType = "Tukey")), test = adjusted("holm"
))

#Interaction post-hocs
lsmeans(m.MSL.RTACC, pairwise ~ Trial.ACC | Block)
lsmeans(m.MSL.RTACC, pairwise ~ Block | Trial.ACC)
lsmeans(m.MSL.position, pairwise ~ Position | Block)
lsmeans(m.MSL.position, pairwise ~ Block | Position)
lsmeans(m.MSL.Error, pairwise ~ Error | Block)
lsmeans(m.MSL.Error, pairwise ~ Block | Error)
lsmeans(m.MSL.Arcsine, pairwise ~ TrialType | Block)
```

##Barplot accuracy proportion

```
#Barplot with accuracy per block (first model)
theme_set(theme_classic())

ggplot(d.ACC, aes(x=Block)) +
  stat_summary(aes(y = ACC.per),
               fun = mean, na.rm = TRUE,
               geom = "bar",
               size = 1),
               fill = “tomato3” +
  stat_summary(aes(y = ACC.per),
               fun.data = mean_cl_normal, na.rm = TRUE,
               geom = "errorbar",
               color = “gray8”
               width = 0.2) +
  scale_y_continuous(labels = scales::percent_format(scale = 1)) +
  ylab("Accuracy percentage")+
```

```
xlab("Block")+
theme(legend.position = "none") +
theme(text = element_text(size = 14, family = "serif"))
```