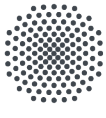

University of Stuttgart
Germany

Institute for Visualization and Interactive Systems

Universitätstraße 38

70569 Stuttgart

Master Thesis

Simulation of EEG Activity based on Sequential Sampling Models

Timo Zaoral

Course of Study:	Master of Science Wirtschaftsinformatik
Examiner:	Jun.-Prof. Dr. Benedikt Ehinger
Supervisor:	Jun.-Prof. Dr. Benedikt Ehinger
Commenced:	15.10.2024
Completed:	15.04.2025

Abstract

Understanding how the brain accumulates evidence to make decisions is a goal in cognitive neuroscience. Decision-making processes are often modeled using Sequential Sampling Models (SSMs), which describe how information is gradually integrated until a decision threshold is reached. Among these, the Drift Diffusion Model (DDM) and Linear Ballistic Accumulator (LBA) have been widely used to explain behavioral and neural data. More recently, the KellyModel has introduced additional parameters to enhance biological plausibility.

However, directly linking such models to electroencephalography (EEG) data remains a challenge. In this work, we address this gap by developing a computational toolbox that enables EEG simulations based on SSMs to systematically investigate how these models can be mapped to neuronal activity. The toolbox was developed for seamless integration into UnfoldSim.jl, a Julia-based package for EEG simulations. The overall package allows researchers to study the decision-making processes and their influence on the event-related potentials (ERPs) through the possibility of creating simulations. The work provides examples of testing deconvolution techniques using the toolbox created. This addresses component overlap in EEG signals, which is a well-known challenge in cognitive neuroscience.

The toolbox was seamlessly integrated into UnfoldSim.jl with a modular design, comprehensive test coverage, and detailed documentation, ensuring ease of use and extensibility. This structured integration into the UnfoldSim.jl package supports the use of the implementation throughout the entire Unfold toolbox.

This work contributes to computational neuroscience by providing a flexible and extensible toolbox for EEG research. Future studies may refine model parameters, validate simulations against real EEG data, and incorporate adaptive modeling approaches to improve accuracy and applicability in decision-making research.

Keywords: Electroencephalography (EEG) Simulation, Sequential Sampling Models (SSMs), Decision-Making Processes. Computational Neuroscience. UnfoldSim.jl Integration

Abstract in German

Ein Ziel der kognitiven Neurowissenschaften ist es zu verstehen, wie das Gehirn Informationen sammelt, um Entscheidungen zu treffen. Entscheidungsprozesse werden häufig mit Hilfe von Sequential-Sampling-Modellen (SSM) modelliert, die beschreiben, wie Informationen schrittweise integriert werden, bis eine Entscheidungsschwelle erreicht ist. Das Drift-Diffusions-Modell (DDM) und der lineare ballistische Akkumulator (LBA) wurden häufig zur Erklärung von Verhaltens- und neuronalen Daten verwendet. In jüngerer Zeit wurden mit dem Kelly-Modell zusätzliche Parameter eingeführt, um die biologische Plausibilität zu erhöhen.

Die direkte Verknüpfung solcher Modelle mit Elektroenzephalographie (EEG)-Daten bleibt jedoch eine Herausforderung. In dieser Arbeit wird diese Lücke durch die Entwicklung einer Toolbox geschlossen, die EEG-Simulationen auf der Grundlage von SSMs ermöglicht, um systematisch zu untersuchen, wie diese Modelle auf die neuronale Aktivität abgebildet werden können. Die Toolbox wurde für eine nahtlose Integration in UnfoldSim.jl, ein Julia-basiertes Paket für EEG-Simulationen, entwickelt. Das Gesamtpaket ermöglicht es Forschern, die Entscheidungsprozesse und deren Einfluss auf die ereigniskorrelierten Potentiale (ERPs) durch die Möglichkeit der Erstellung von Simulationen zu untersuchen. Die Arbeit liefert Beispiele für das Testen von Dekonvolutionstechniken unter Verwendung der erstellten Toolbox. Damit werden Komponentenüberlappungen in EEG-Signalen adressiert, die in den kognitiven Neurowissenschaften eine bekannte Herausforderung darstellen.

Die Toolbox wurde nahtlos in UnfoldSim integriert, mit einem modularen Design, umfassender Testabdeckung und detaillierter Dokumentation, die Benutzerfreundlichkeit und Erweiterbarkeit gewährleistet. Diese strukturierte Integration in das UnfoldSim.jl Paket unterstützt die Nutzung der Implementierung in der gesamten Unfold Toolbox.

Diese Arbeit trägt zur Computational Neuroscience bei, indem sie einen flexiblen und erweiterbaren Werkzeugkasten für die EEG-Forschung bereitstellt. Zukünftige Studien können Modellparameter verfeinern, Simulationen mit realen EEG-Daten validieren und adaptive Modellansätze integrieren, um die Genauigkeit und Anwendbarkeit in der Entscheidungsforschung zu verbessern.

Keywords: Electroencephalography (EEG) Simulation, Sequential Sampling Models (SSMs), Decision-Making Processes. Computational Neuroscience. UnfoldSim.jl Integration

Contents

List of Figures

List of Tables

List of Abbreviations

CPP	centro-parietal positivity
DDM	Drift Diffusion Model
EEG	Electroencephalography
ERP	Event-related potentials
KellyModel	Advanced Drift Diffusion Model from Kelly et. al.
LBA	Linear Ballistic Accumulator
SSM	Sequential Sampling Model

1. Introduction

The human brain continuously processes and integrates sensory information in order to make decisions. This process is typically modelled as evidence accumulation. The complex cognitive function behind this underlies many of our everyday decisions, whether simple or complex. This function is often modelled by Sequential Sampling Models (**SSMs**) (Ratcliff, 1978).

The **SSMs**, such as the Drift Diffusion Model (**DDM**), Linear Ballistic Accumulator (**LBA**) and others, are used to model how information is accumulated over time until a decision threshold is reached. But almost all of their applications are for inference. This means that you have the data and want to access the "hidden" states of the assumed evidence integration model (Brown & Heathcote, 2008). Therefore, **SSMs** can offer a useful framework for investigating the neural mechanisms underlying decision-making. By modeling these processes, researchers can compare model-generated predictions with neural data, gaining deeper insights into the temporal dynamics and cognitive processes involved in decision-making. Furthermore, the different parameters of the **SSMs** are typically interpreted as correlates of underlying cognitive processes and thus contribute to a better understanding of the process. Understanding the neural mechanisms underlying the decision-making process is crucial to improve our understanding of cognition and its relationship to brain activity (Forstmann et al., 2016; Van Vugt et al., 2019).

The **SSMs** like the **DDM** are mostly used to model behaviour. In contrast, Electroencephalography (**EEG**) can be used to record brain activity during decision making, which provides insight into the temporal aspects of evidence accumulation. The **EEG** data can be analysed to identify neural indicators of the decision-making process over time, particularly those that reflect the accumulation of evidence. In many studies, the centro-parietal positivity (**CPP**) a specific **EEG** component has now been identified as an indicator of evidence accumulation during decision-making. This enables the comparison of averaged traces from multiple trials of **SSM** simulations with corresponding **EEG** data (Van Vugt et al., 2019).

SSMs are widely used to explain behavioural aspects of decision making, but their direct link to neural data remains an open challenge. Conventionally, **EEG** data can be integrated into **SSMs** in two ways: as a predictor influencing the model parameters, or as an explanandum where **EEG** features (e.g. **CPP**) are modelled together with behavioural data such as reaction time and choice. To systematically explore these relationships, a specialised computational framework is required. This computational framework should enable the simulation of **EEG** data based on different **SSMs** in decision-making contexts and facilitate the mapping of model parameters to observable neural activity. Mastering this integration remains a central challenge in cognitive neuroscience (Connell & Kelly, 2021; Forstmann et al., 2016).

This thesis aims to provide a computational framework for simulating **EEG** data using different **SSMs** to explore how these models reflect the evidence accumulation process in the brain. The computational framework is to be integrated into the existing Julia package for simulating EEG data *UnfoldSim.jl* (Schepers et al., 2025) and is therefore subsequently referred to as an extension. By systematically manipulating model parameters of the different **SSMs** and and comparing their patterns to patterns observed in the **EEG** literature, researchers can try to find out which model characteristics best reflect the evidence accumulation processes in the brain. Understanding these indicators could lead to more precise interpretations of **EEG** data in the context of cognitive decision-making research.

1.1. Motivation & related work

In the publication "Neurocomputational mechanisms of prior-informed perceptual decision-making in humans" (Kelly et al., 2021) a novel SSM was used to simulate a decision-making process in the brain. This simulation was compared with real EEG results from a decision-making experiment. The real EEG results are shown in the Figure 1 in tile A. The stimulus-locked is shown on the left and the response-locked on the right. Tile B shows the simulated CPP for comparison, also stimulus-locked and response-locked. This comparison shows how the CPP-EEG activity shows markers of an SSM, with the activity increasing from an stimulus onset up to a threshold value when the button is pressed. It should be noted that the increasing activity depends on the actual evidence available. Overall, this shows how EEG activity reflects SSM integration.

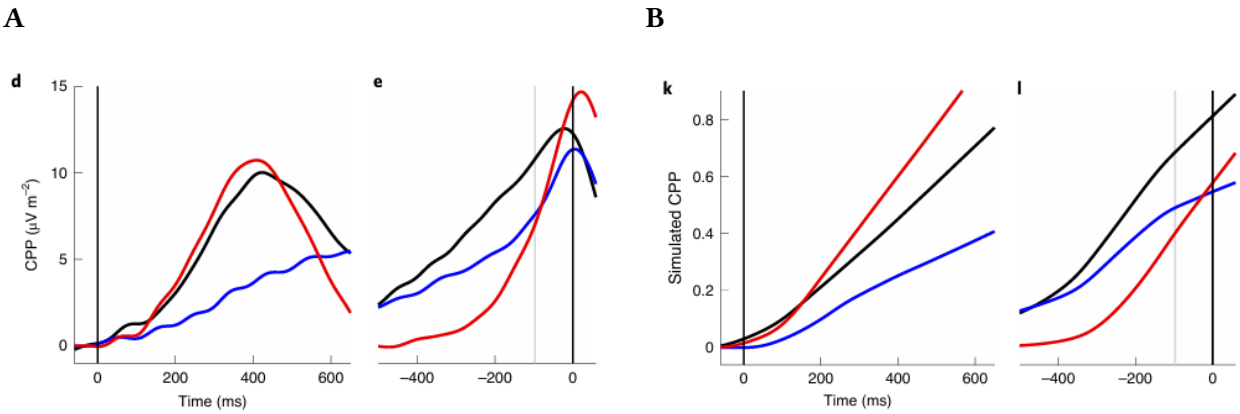

Figure 1: Stimulus-locked and response-locked CPP of a decision-making experiment (Kelly et al., 2021). Legend: conditions of a motion discrimination task experiment black=Easy, blue=LoCoh, red=Deadline. Where Easy: High coherence (20%) and a long response deadline (1.6s). LoCoh (Low Coherence): Lower coherence (8%) and a long deadline (1.6s). Deadline: High coherence (20%) but a short deadline (0.5s). **A)** Event-related potentials (ERP) based on real EEG data of the experiment. d) stimulus-lockend e) response-locked **B)** simulated ERP. k) stimulus-lockend l) response-locked

In another publication by (Frömer et al., 2024), precisely such evidence accumulation EEG signals were investigated. In contrast to the earlier findings by Kelly et al., it has been shown that stimulus-locked activity can "mimic" an evidence accumulation process when it is aligned to the response (Frömer et al., 2024). This means that neural activity, although appearing to reflect gradual evidence accumulation, may instead be structured in a way that systematically increases up to the response moment, without necessarily reflecting an actual accumulation process. The authors show based on four datasets, that signals of evidence accumulation can arise as an artefact of component overlap. Since the reaction time between the stimulus and the response is sometimes very short, the brain response to the stimulus, the response to the evidence accumulation process and the response to the subjects button press overlap in time. This can result in the ramp-up, without any evidence integration signal being present. Furthermore, the four studies were reanalysed using overlap-correction between stimulus and subjects button press. In all four example the response-locked evidence accumulation patterns disappear. That neural signatures of evidence accumulation can arise from simple overlapping components, sparked discussion in the field (Connell et al., 2024).

The research group led by Kelly et. al. responded with a commentary entitled "Regressing Away Common Neural Choice Signals does not make them Artifacts. Comment on Frömer et al (2024, Nature Human Behaviour)"(Connell et al., 2024) to the paper by Frömer et. al.. In this paper, they simulated neural data based on a basic LBA to refute the findings of Frömer et. al. Through these simulations,

the researchers around Kelly et. al. show that the deconvolution analyses used by Frömer et. al. are not necessarily appropriate to correctly capture limited accumulation signals. They also showed that applying the same analyses to real, increasing, accumulation-to-bound signals is consistent with the empirical results of (Frömer et al., 2024) that this can lead to an amplitude reduction effect rather than the alternative scenario in which such signals are not present.(Connell et al., 2024) As a result, they argue that the deconvolution analysis used by (Frömer et al., 2024) actually removes artefactually correct, truthful neural signatures of evidence accumulation in at least some scenarios, possibly making it a misguided analysis tool. Which means that further investigations into this topic are still open.

This discussion on how decision-making is represented in the brain underscores the value of simulations to adress methodological key questions on the validity of an analysis procedure. However, it also reveals that there is currently no simple modelling framework that enables flexible and comprehensive simulations of these specific time series. This can make a contribution to research into cognitive neuroscience. This emphasises the need for a computational framework that enables flexible and comprehensive simulations based on **SSMs** and thus allows a systematic evaluation of analysis methods. Such a tool is essential for the advancement of cognitive neuroscience, as it can remove ambiguities in the interpretation of neural signals and provide a standardised approach to hypothesis testing. Overcoming these challenges is important to improve our understanding of how decision-making is represented in the brain. In order to gain a deeper understanding and clarify the open questions.

2. Fundamentals/Foundations

This chapter serves as an introduction, offering an overview of the terminology and to lay the foundations for the following chapters of the thesis. These foundations consist of a brief explanation of EEG, ERP, Neural Implementation of the Decision-Making Process and the introduction of SSMs and the UnfoldSim.jl package.

2.1. Electroencephalography and Event-related potentials

2.1.1. EEG

EEG is a technique for continuously recording electrical voltage fluctuations that reflect brain activity and are measured using electrodes attached to the scalp. However, EEG has inherent limitations due to the brain's natural structure. The human skull, positioned between the brain potentials and the electrodes, attenuates and distorts the signals. As a result, EEG is more sensitive to activity from the outer cortical layers while being less effective at capturing signals from deeper brain structures. The consequence of the limitations is a poorer spatial resolution (Jackson & Bolger, 2014). The poor spatial resolution of EEG can limit our ability to precisely localize the neural sources of decision variables and to distinguish different accumulation mechanisms occurring at various brain levels.

2.1.2. ERP

ERP are neural responses derived from the continuous EEG signal, obtained by averaging time-locked activity across repeated occurrences of the same event. ERP reflect the brain's processing of specific stimuli, such as visual images or motor responses, by isolating stimulus-related electrical activity from ongoing neural fluctuations. Since individual EEG recordings contain significant noise, ERP waveforms are typically averaged across multiple trials and participants to improve the signal-to-noise ratio, making it easier to identify consistent neural responses (Luck, 2014).

2.1.3. Overlapping potential artifact

In EEG analysis, overlapping potential artefacts occur when neuronal signals from different cognitive or motor processes overlap and disturb the measured waveform. For example, activity related to sensory processing can overlap with subsequent motor-related activity, creating a combined signal that does not accurately reflect either process individually. In experiments involving both stimulus presentation and motor responses, this can lead to ambiguities in interpreting ERP. To address this issue, deconvolution techniques are used in EEG analysis to separate overlapping neural signals, improving the accuracy of ERP analysis (Ehinger & Dimigen, 2019).

2.2. Neural Implementation of the Decision-Making Process

A part of cognitive neuroscience is investigating the neural mechanisms underlying decision-making, one research goal is to understand how the brain collects and processes information in order to make a decision. The investigation of electrophysiological signals such as EEG or magnetoencephalography (MEG) play an important role in this. Thanks to their high temporal resolution, these signals make it possible to observe neuronal dynamics during a decision-making task in real time. This makes it possible to identify indicators and influencing factors in the neuronal mechanism of decision-making processes (Connell & Kelly, 2021; Van Vugt et al., 2019).

This has led to an increasing number of scientific papers investigating how brain regions and neuronal circuits contribute to the decision-making process. Further research has identified the **CPP** as a neural marker of decision-making, recorded using EEG electrodes (Cz, CPz, and Pz) over centro-parietal scalp regions. This signal refers to the process of accumulating evidence up to a decision threshold. The decision threshold indicates the point at which enough evidence has been accumulated to make a decision. How exactly the **CPP** signal emerges as a neural correlate of decision-making is still unclear. However, some insights into the **CPP** have been gained. A steeper **CPP** indicated a faster accumulation of evidence, which leads to a faster decision. In addition, the **CPP** was active in both perceptual and memory decisions, reflecting similar dynamic processes but with a delay in perceptual decisions. The **CPP** also showed a graded response depending on the difficulty of the decision. Thus, more difficult tasks have a lower amplitude of the **CPP**. This establishes the **CPP** as a useful marker for investigating the neural basis of decision making (Connell & Kelly, 2021; Kelly & O'Connell, 2013; Van Vugt et al., 2019).

By using **SSMs** to interpret neural data, promising progress has been made in understanding the temporal dynamics of decision-making processes. Publications such as those by (Connell & Kelly, 2021) and (Van Vugt et al., 2012) show that **EEG** can capture neuronal signatures of evidence accumulation in real time by linking specific **SSM** parameters, such as drift rate, to observable neuronal signals. In this way, the **CPP** could again be used to compare neurophysiological data with the decision-making processes generated from **SSM**. Such comparisons can provide deeper insights into how the underlying cognitive mechanism of decision-making takes place in the brain (Forstmann et al., 2016; Mulder et al., 2014). This is also supported by the publication of (Murphy et al., 2016) who demonstrated that decision-related neural signals, including the **CPP**, accurately reflect the dynamics predicted by **SSMs**. Which can demonstrate a direct link between model parameters and brain activity (Murphy et al., 2016).

2.3. Drift Diffusion Model

The **DDM** is a mathematical model used in cognitive neuroscience and psychology to describe decision-making processes. It is often used to explain reaction times and the accuracy of decisions in binary choice processes in which a person has to choose between two alternatives. The **DDM** is based on the idea that information is accumulated over time until a decision criterion is reached, which then triggers the selection of one of the two alternatives (Ratcliff, 1978).

2.3.1. Basic principles of the DDM

The **DDM** depicts the decision-making process as a stochastic (random) drift-diffusion movement. The process begins at a neutral starting point and moves in an 'evidence space' until it reaches one of the two thresholds that represent the decision in favour of one of the two alternatives. The time at which a threshold is reached corresponds to the reaction time for the decision, and the threshold reached indicates the choice made. Figure 2 shows an example of such a model (Ratcliff, 1978).

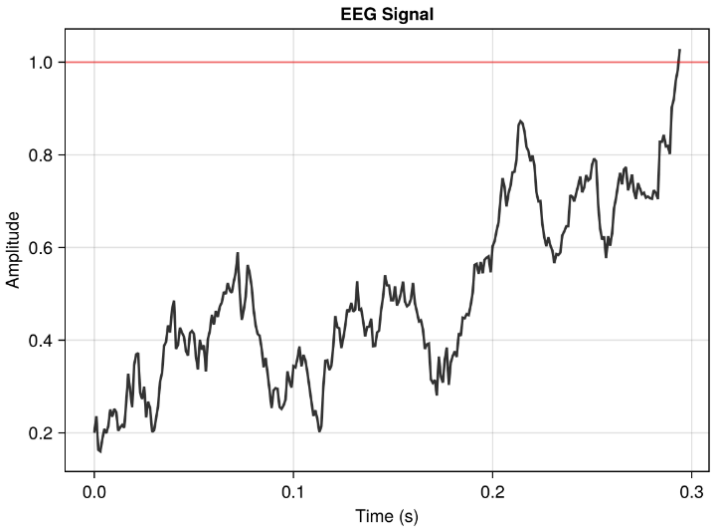

Figure 2: Simulated evidence accumulation signal using the DDM. Model parameters of the figure: Drift rate (v)=2, Threshold distance (a)=1(red), Starting point of accumulation (z)=0.2

2.3.2. Main parameters of the DDM

The DDM comprises several key parameters that quantitatively describe the decision-making process. The parameters are described below with their argument names from the implementation in brackets:

Drift rate (v): The drift rate represents the average rate at which information is accumulated in favour of one of the two alternatives. It is a measure of the conclusiveness or strength of the evidence in favour of a particular choice. A higher drift rate usually leads to faster and more accurate decisions, as the threshold is reached more quickly. The drift rate varies between trials and tasks as it is influenced by the difficulty of the task or the availability of information.

Threshold distance (a): This parameter determines the decision thresholds. A larger distance between the threshold values leads to a longer accumulation time, which results in longer reaction times, but also to a higher level of decision certainty as more evidence has been accumulated. A smaller threshold distance can lead to faster but potentially less accurate decisions. This parameter therefore reflects the preference for speed or accuracy.

Starting point of accumulation (z): The starting point represents the initial position in the accumulation process and indicates a possible bias in favour of one of the two alternatives. If the starting point is closer to one of the two thresholds, there is a bias in favour of this threshold (i.e., this decision), which makes the decision for this alternative more likely and faster. The starting point can vary to reflect biases or preferences in the decision-making process.

Non-decisional time (t): This parameter represents the time required for all processes that are not part of the actual decision accumulation, such as motor response. This time is added to the reaction time and is crucial for modelling the total observed reaction time. (Forstmann et al., 2016; Ratcliff, 1978; Ratcliff et al., 2016; Ratcliff & McKoon, 2008)

2.3.3. Properties and characteristics of the DDM

The DDM has several characteristic properties that make it particularly suitable for modelling decision-making processes in various cognitive tasks:

Stochastic nature of the decision process: the drift-diffusion process involves random fluctuations that reflect the natural uncertainty factor in human decisions. This makes it possible to model the variability of reaction times and accuracy.

Balancing accuracy and speed: The DDM offers an explanation of how people adapt their decisions to the requirements of the tasks, whether by striving for higher accuracy (higher thresholds) or shorter reaction times (lower thresholds). This balance is controlled by the threshold parameter a and reflects real-life decision-making situations well.

Adaptability to different task difficulties: The drift rate v often varies depending on the difficulty of the task or the clarity of the information presented. With complex alternatives that are difficult to distinguish, the model drifts more slowly, resulting in a lower drift rate. As a consequence, decision times are longer, and accuracy may be reduced.

Prediction of reaction times and error rates: The DDM can predict the probability distribution of reaction times for correct and incorrect decisions. By modelling reaction times and errors as the result of the same process, the model can make accurate predictions for both, which contributes to its popularity in cognitive psychology.

Explanation of biased decisions: The starting point parameter z allows the model to take into account preferences or biases that favour a decision in favour of an alternative before accumulation begins. (Forstmann et al., 2016; Ratcliff, 1978; Ratcliff et al., 2016; Ratcliff & McKoon, 2008)

2.3.4. Application and scientific relevance

The DDM has been shown to be useful for modelling a variety of decision-making processes, ranging from simple reaction time tasks to more complex cognitive tasks. In cognitive neuroscience, it is used to understand neural mechanisms of decision making. The accumulation process in the DDM can map specific neural activity patterns in regions such as the prefrontal cortex and parietal cortex. The model can also be used to make comparisons with actual EEG data from a decision-making process. This allows to analyze the entire process in more detail (Forstmann et al., 2016; Myers et al., 2022; Ratcliff, 1978; Ratcliff et al., 2016; Ratcliff & McKoon, 2008).

2.4. Linear Ballistic Accumulator

The LBA model is a mathematical model belonging to the family of SSMs and can also be used to explain decision-making processes. It was developed to provide a simplified and easier to calculate alternative to the DDM. The LBA model is mainly used to model reaction times and decisions in cognitive tasks with two or more choices. Thus it is based on a deterministic, linear accumulation process without the stochastic diffusion that is characteristic of the DDM (Brown & Heathcote, 2008).

2.4.1. Basic principles of the LBA model

The LBA model assumes that a decision process can be described by a series of independent accumulators that collect evidence for each decision option in parallel. Each accumulator represents a choice. In contrast to the DDM, the LBA model is based on a linear, non-stochastic (deterministic) accumulation rate. Therefore it accumulates evidence for each decision option at a constant rate per time step. As soon as an accumulator reaches its threshold value, the corresponding decision is made (Brown & Heathcote, 2008).

2.4.2. Main parameters of the LBA model

The LBA model uses a number of parameters that mathematically describe the decision-making process. These parameters are:

Starting point distribution (A): Each accumulator starts with a random starting point that lies within a fixed distribution (often uniform). This starting point distribution models the initial bias in the collection of evidence. The variability of the starting point allows the model to explain the differences in reaction times and accuracy between different runs. A lower starting point means more time to reach the threshold, while a higher starting point signals less need for evidence to make a decision.

Linear Accumulation rate (v): The accumulation rate, also known as the drift rate, describes the speed at which an accumulator gathers evidence. Each choice option has its own accumulation rate, which is assumed to be normally distributed in order to model fluctuations in decision quality or task difficulty. A higher accumulation rate leads to faster decisions in favour of the corresponding option and reflects the preference or strength of evidence for that choice. Unlike the DDM, the LBA uses a linear accumulation process without noise during the accumulation itself. This makes the accumulation deterministic once the initial parameters are set. (Brown & Heathcote, 2008; Tillman et al., 2020)

Threshold (k): This Parameter defines the amount of evidence needed to accumulate in order to make a decision. Similar to the Threshold distance (a) of the DDM.

Non-decisional time (t): Represents the time required for example the motor response. Similar to the Non-decisional time of the DDM.

The differences and similarities as the threshold or the drift_rate between the LBA and the DDM are also shown in the Figure 3.

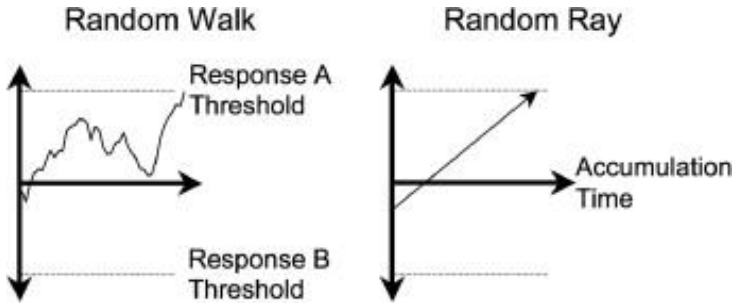

Figure 3: DDM compared to LBA: differences in accumulation over time. Figure from: (Brown & Heathcote, 2008)

2.4.3. Properties and characteristics of the LBA model

The LBA model has some specific features and characteristics that distinguish it from other accumulation models, such as the drift-diffusion model:

Linear and deterministic accumulation: The LBA model is based on linear accumulation, i.e. evidence is accumulated at a constant rate, without the stochastic fluctuations that play a role in the DDM. This simplifies the calculations and makes the model analytically more accessible, which in turn facilitates statistical modelling.

Flexibility in multiple choice decisions: The LBA model is well suited for decisions with more than two alternatives, as it considers each decision process as an independent accumulation channel. The independent accumulators allow an easy extension to multiple choice options, as each accumulator is modelled with a specific accumulation rate and its own threshold value.

Distribution of starting points as an explanation for variability: The initial variability of the accumulators is modelled by the distribution of the starting point. This provides a natural explanation for variations in reaction times between different runs of the same task. As a result, the **LBA** model can predict reaction times more accurately as the **DDM**, especially for fast or slow decisions, without the additional noise component of the **DDM**.

Adaptation to task requirements: The **LBA** model allows the decision thresholds and accumulation rate to be adapted to the requirements of the task. Higher thresholds can be set for tasks that require high accuracy, while lower thresholds can be set for tasks with time pressure. Because the **LBA** model describes accumulation without stochastic drift, it is less sensitive to stochastic noise and is often more robust in situations where fluctuations in evidence accumulation are small. This can allow for more stable modelling, especially in experimental tasks where the evidence is constant or nearly constant. In addition, an accumulation-to-bound process ramping from stimulus to response can be specifically generated (Brown & Heathcote, 2008; Tillman et al., 2020).

2.4.4. Applications and scientific significance

The **LBA** model has gained importance in cognitive psychology and neuroscience, especially because of its simplicity and predictability. It is used to model reaction times and decision accuracy in various cognitive tasks and provides a more efficient analysis of large data sets due to the simplified linear accumulation. Compared to the **DDM**, the **LBA** model is particularly useful in experimental situations where the decision process is less influenced by noise and allows the modelling of decisions with multiple options. The decision in favour of the **LBA** model instead of the **DDM** may be preferred in research situations where the complexity of the **DDM** is unnecessary or difficult to calculate (Brown & Heathcote, 2008; Tillman et al., 2020).

2.5. Advanced Drift Diffusion Model from (Kelly et al., 2021)

In their publication "Neurocomputational mechanisms of prior-informed perceptual decision-making in humans" by (Kelly et al., 2021), the authors construct a decision making model that includes a build-up to threshold process with multiple build-up (evidence accumulation and urgency) and delay (pre- and post-decision) components. The model extends the **DDM** by incorporating additional parameters, resulting in a more complex accumulation process. The original model was implemented in MATLAB by Kelly et al. and later transferred to Julia through an initial conversion by my supervisor, Benedikt Ehinger (Ehinger, 2025). Therefore, the further use and description of the model in this paper is based on the initial Julia version which was refactored by me. In this paper, the Advanced Drift Diffusion Model from Kelly et. al. is abbreviated as **KellyModel**. You can find the original MATLAB code here: OSF Storage .

2.5.1. Basic principles of the Model

As the **DDM** the **KellyModel** depicts the decision-making process as a stochastic (random) drift-diffusion movement. The process begins at a neutral starting point and moves in an "evidence space" until it reaches the threshold that represent the decision. The time at which a threshold is reached corresponds to the reaction time for the decision. Additionally, the process includes a sensor encoding delay, causing a time shift before accumulation begins after stimulus onset. It also features a post-accumulation phase, called a non decision time, during which evidence continues to accumulate even after the threshold is reached and the decision is made. An example of an trace of such a model is shown in the Figure 4.

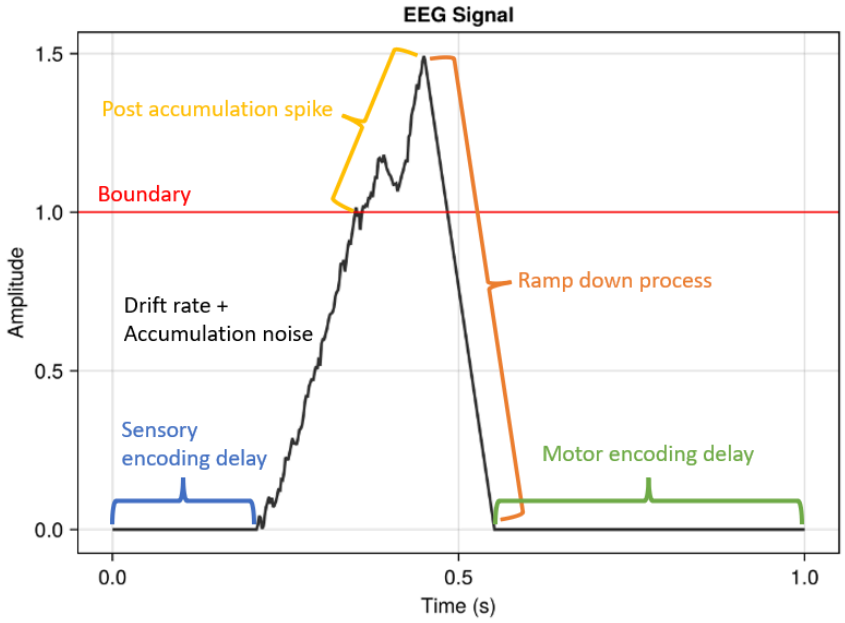

Figure 4: Simulated evidence accumulation signal using the Advanced Drift Diffusion Model from Kelly et. al. (KellyModel) with parameter annotations (Fixed parameters: drift_rate=6.0, boundary=1.0, event_onset=0.2, post_accumulation_duration=0.1, ramp_down_duration=0.1, motor_onset=0.4; Parameters based on distributions: accumulative_level_noise=0.5, sensor_encoding_delay=0.1, motor_delay=0.1, post_accumulation_duration_variability=0.2)

2.5.2. Main parameters of the KellyModel

The KellyModel comprises several key parameters that quantitatively describe the decision-making process. The influence of the model's parameters can be seen in Figure 4 through annotations on a simulated evidence accumulation signal.:

Drift rate (drift_rate): The drift rate represents the average rate at which information is accumulated per time step. A higher drift rate usually leads to faster and more accurate decisions, as the threshold is reached more quickly. The drift rate varies between individuals and tasks as it is influenced by the difficulty of the task or the availability of information.

Accumulation noise (accumulative_level_noise): To introduce variability in the accumulation process over time, the accumulative_level_noise parameter can be used to control the amount of evidence accumulated per time step. This allows for variations between individual traces or, if desired, greater consistency by reducing the variability.

Sensory encoding delay (event_onset, sensor_encoding_delay): This component is divided in two parameters and determines the delay between the stimulus onset and the actual start of the evidence accumulation process as shown in Figure 4. This can be seen as a reaction time where an participant need time to analyze the environment/task. For the Model the event_onset defines a fixed delay in seconds. The sensor_encoding_delay defines a uniform distributed(range[−0.5*x, 0.5*x] where x is the parameters value) delay also given in seconds which is applied on top. With this combination a wide variability in parameterizing the sensory encoding delay between subjects can be achieved. This parameters therefore reflect the preference for speed up or delay in the overall process.

Boundary/Threshold height (boundary): Similar to the DDM. Indicated in Figure 4 as a horizontal red line.

Post accumulation spike (post_accumulation_duration, post_accumulation_duration _variability): Once the boundary has been reached, it might be assumed that the accumulation process will continue for a certain period of time as shown annotated in yellow in Figure 4. This behavior can be influenced by the two parameters post_accumulation_duration/_variability. As with other components, the post_accumulation_duration is a fixed time specification in seconds for the excess and the post_accumulation_duration_variability is a variable influence in the form of a uniform distribution(range[−0.5*x, 0.5*x] where x is the parameters value).

Ramp down process (ramp_down_duration): After the spike of the accumulation process a ramp down component back to zero is implemented. The duration of this fixed ramp down is specified in seconds by the parameter ramp_down_duration and can be seen in Figure 4 with the orange annotation. EEG studies show that neuronal decision signals do not end abruptly after reaching a threshold value, but gradually fade away, which can be realistically simulated by using this ramp down component (Myers et al., 2022).

Motor encoding delay (motor_onset, motor_delay): This component can be seen as the opposite of the sensory encoding delay. As this is the motor delay that is required after a decision has been made to perform the resulting action to communicate the decision. This amount of time starts therefore after the model turns back to zero as shown in Figure 4. A practical example would be, when a person has to press a button or point to something, this needs time to be proceed. The model parameter motor_onset specifies a fixed value in seconds for the delay. A variable value is specified with motor_delay, which also influences the delay as a uniform distribution(range[−0.5*x, 0.5*x] where x is the parameters value). These parameters can therefore delay or accelerate a subsequent event (Kelly et al., 2021) .

2.5.3. Properties and characteristics of the KellyModel

The KellyModel is designed to simulate the neural temporal dynamics of decision-making through an evidence accumulation framework. It incorporates several key characteristics that make it a flexible and biologically plausible model for cognitive processes:

Adaptability to Individual Differences: The model allows for variation in drift rate, sensory encoding, and motor delays, enabling it to reflect differences in cognitive processing speed, task difficulty, and participant-specific response behaviors.

Temporal Structure and Variability: With distinct parameters for sensory and motor delays, as well as post-accumulation effects, the model captures the full decision-making process—from stimulus onset to motor execution—while introducing natural variability through stochastic components.

Noisy Evidence Accumulation: The inclusion of accumulative level noise ensures that the model accounts for fluctuations in information processing, which aligns with empirical findings from neural and behavioral studies. In contrast to the classic DDM, where the drift rate remains constant across trials (and even more so in the LBA), the inclusion of accumulative level noise introduces trial-by-trial variability in the accumulation process.

Gradual Decision Termination: Unlike traditional drift-diffusion models that assume abrupt decision termination, the ramp-down process provides a gradual decline in accumulation signals, making the model more consistent with observed neural data.

Task-Specific Customization: By adjusting parameters such as boundary height, drift rate, and delay distributions, the model can be tailored to simulate different cognitive tasks, from rapid perceptual judgments to more complex decision-making scenarios.

2.5.4. Application and scientific relevance

As previously described, the model was already used by its namesakes in the publication 'Neurocomputational mechanisms of prior-informed perceptual decision-making in humans' (Kelly et al., 2021). The model was used to compare simulated evidence signals with evidence accumulation processes that are reflected in the **CPP** in real **EEG** data.

The model was also used in the publication by (Frömer et al., 2024) mentioned under Section 1.1. In this publication, the model was used to perform a comparison of mass univariate and deconvolution analyses on the basis of simulated evidence accumulation signals. The sensor encoding and motor encoding parameters were varied so that the evidence accumulation peak was closer to the stimulus, closer to the response and neutral. These three scenarios showed that the evidence accumulation signal is most likely to appear in the stimulus-locked deconvolution when it is more closely coupled to the stimulus and vice versa for the response. Thus, it could be shown that under these different conditions no signal is lost depending on the deconvolution.

2.6. UnfoldSim.jl Package (Schepers et al., 2025)

UnfoldSim.jl is a Julia-based framework for simulating multivariate time-series data, particularly for **EEG** and **ERPs**. It provides a controlled environment to test and validate statistical modeling approaches by generating synthetic datasets that resemble real **EEG** data while allowing full control over underlying parameters.

EEG analysies often faces challenges such as overlapping neural responses, complex experimental designs, and measurement noise. UnfoldSim.jl enables the modeling of realistic, continuous **EEG** data by allowing precise control over event timing, response functions, and noise properties. This makes it a valuable tool for testing **EEG** analysis methods, validating statistical approaches, illustrating conceptual issues, evaluating toolbox functionalities, and identifying limitations in traditional analysis workflows. Ultimately, this facilitates addressing the aforementioned challenges.

Key Features:

- Modularity and extensibility: Modular structure in the building blocks (designs, components, onset distributions or noise types) of which different variants can be selected and new ones can be added easily as a contributor.
- Flexible Experimental Designs: Enabling realistic event structures and hierarchical effects, such as creating designs with repeated event sequences.
- Handling Overlapping Events: Generates continuous **EEG** signals with overlapping event-related activity, enabling validation of deconvolution-based approaches.
- Noise Modeling: Incorporates realistic noise sources (e.g., pink noise).

Integration with Other Tools: As part of the overall Unfold package family, UnfoldSim.jl is designed to work seamlessly with the other packages. For example, with Unfold.jl, to create the **ERP** from the simulated **EEG** and further analysis tools like the deconvolution method. Or UnfoldMakie.jl, which provides visualisation tools for **EEG**/**ERP** analysis. This integration ensures a standardised workflow from simulation to analysis.

UnfoldSim.jl plays a key role in testing and improving new analysis methodologies, making it an valuable tool for computational neuroscience. Therefore, it provides the best framework to incorporate the simulation of the evidence accumulation process using the **SSMs** in this package. The following section introduces the new contribution and extension that further enhance the capabilities of UnfoldSim.jl.

3. Implementation

This chapter describes the implementation and integration of the DriftComponent extension in UnfoldSim.jl. The first part covers the basic implementation and the second part addresses the integration and related topics.

3.1. Sequential Sampling Models

This section examines the SequentialSamplingModels.jl package (itsdfish & contributors, 2025) used for the **SSMs** and the additional functions it requires. It outlines the requirements, the final package selection, and also includes a separate discussion on the refactoring of the Advanced Drift Diffusion Model from Kelly et. al. (**KellyModel**).

3.1.1. The SequentialSamplingModels.jl Package

We chose to use the preexisting SequentialSamplingModels.jl package (itsdfish & contributors, 2025) for their well-documented implementations of the **LBA** and the **DDMs**. Furthermore the package is regularly maintained and further developed.

In the SequentialSamplingModels.jl package the two required **SSMs** **LBA** and **DDMs** are split into two functions: defining model parameters and simulating. The simulation function shown below as Listing 1 uses a random object, the associated model with parameters and a sampling rate as parameters. From this, the function then simulates the time_steps, i.e. a time vector in steps of the sampling rate and the associated evidence at the respective point in time.

```
using SequentialSamplingModels as SSM

time_steps, evidence = SSM.simulate(rng,model;Δt=0.001)
```

Historically the function of the **LBA** was adapted to the syntax of the function of the **DDM** in collaboration with the developers of the SequentialSamplingModels.jl package. In addition, a small bug in the simulation was removed. Credits at this point for the quick reaction and co-operation with the developers.

The simulation functions then only had to be encapsulated by the multiple dispatch functionality of Julia. Therefore, the following Listing 2 was defined for each of the two models.

```
SSM_Simulate(rng,model::(LBA/DDM),sfreq,max_length)::Tuple{Float64, Vector{Float64}}
```

Further details about the code on GitHub

In each of these, the model simulate Listing 1 is called and the response time is then determined as *rt* and, if necessary, the evidence length is shortened to *max_length* as there should be a maximum range for a simulated signal. Finally, the response time *rt* and the *evidence* vector are returned.

All in all, this can be used to simulate the progressions of the respective models. Therefore these functions can be integrated into UnfoldSim.jl at a later stage.

3.1.2. Refactoring of the KellyModel

As already described in Section 2.5 the implementation by (Kelly et al., 2021) was in MATLAB and roughly transferred to Julia by Benedikt Ehinger (Ehinger, 2025). Therefore, I first refactored and reimplemented the functionality. The refactoring included the creation of a composite type as **KellyModel** with sensibly described parameters and documentation.

Additionally, the evidence accumulation function of the model required a complete revision, as its previous implementation contained numerous parameters and was difficult to interpret. The simulation process has been simplified, making it more intuitive and accessible. Furthermore, the new simulation function was adapted to align with the existing "SSM_Simulate" paradigm Listing 2. As a result, it now clearly incorporates appropriate model parameters by passing the model class.

The usage of the same "SSM_Simulate" function ensures a general use for the simulation of **SSMs**, as all models are used for simulation via a single function call with the corresponding parameter. In addition the functionality can be easily extended with other models. Under this condition, the previous implementation can be used and integrated into UnfoldSim.jl, as described in the next chapter.

3.2. Integration to UnfoldSim

The groundwork for integration into UnfoldSim.jl has been established. However, further adaptations are needed to align with the package's modular structure and ensure seamless integration. The most important elements of the modularity of UnfoldSim.jl are the design, components, onsets and additional functionalities/models. For each of these elements, structures and functions must be newly created or reused in order to enable successful integration in the end. The following section describes the integration step by step using a modular building block kit and finally illustrates the interaction.

3.2.1. Modeling the Integration

The initial step is to outline the modular building block kit to determine which elements need to be expanded.

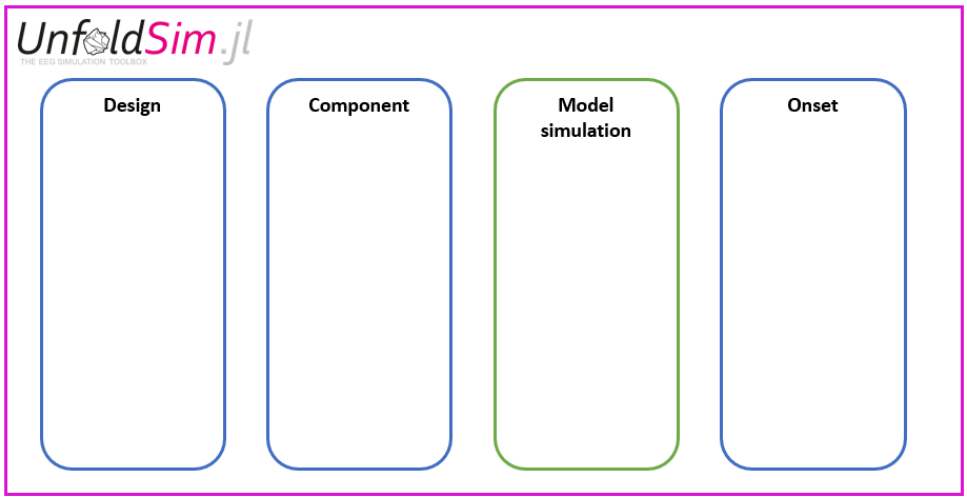

Figure 5: Empty building box of UnfoldSim.jl elements

The three existing elements Design, Component and Onset are shown in the Figure 5. These structures are already in place for other simulations and are utilized accordingly. The **green** Model simulation component is a new functionality that should contain the **SSMs**.

The empty elements in the Figure 5 are now sequentially discussed and filled with implementation parts from left to right. The first element "Design" represents the experimental setup for the simulation, such as one or more participants performing conditions classified as heavy, medium, and light. Existing designs that are important for the DriftComponent extension are the SingleSubjectDesign, SequenceDesign and RepeatDesign. The SingleSubjectDesign is used as the base and defines the conditions of a single participant. The SequenceDesign is the decisive one for the DriftComponent extension, as it defines a sequence of components/events. Therefore it allows to build a complex of a stimulus, evidence accumulation and response signal ('SCR'). Finally, the RepeatDesign is used for the repetition of previously defined designs, so that a sequence is repeated 100 times, for example. The designs are predefined and can therefore be reused without the need for customisation. They are therefore already inserted in Figure 6 as blue rectangles.

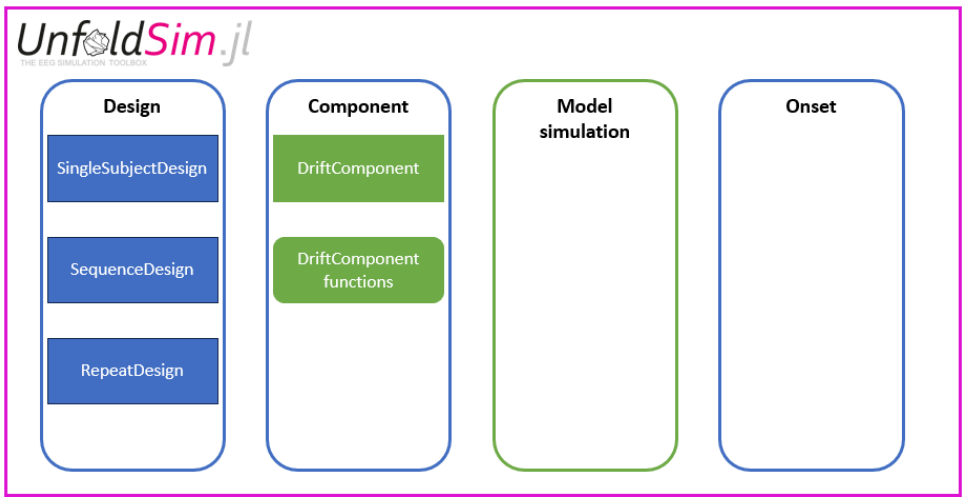

Figure 6: Building box with component adjustments of UnfoldSim.jl package

In the Figure 6, the first new structure DriftComponent and associated functions have also been added to the component group.

$$\text{DriftComponent}(\text{max\_length::Int}, \text{sfreq::Real}, \text{model\_type::Any}, \text{model\_parameters::Dict}) \quad (1)$$

A DriftComponent is defined as described in function 1 by a max_length the maximum length of a simulated component, sfreq the sampling frequency, model_type the SSM type and model_parameters the parameter specifications for the SSM. This means that this component of UnfoldSim.jl can be selected for a simulation. However, a few associated functions are still missing to run the simulation. One of these is the "get_model_parameter" function, which fulfills the purpose of making it possible to specify parameters for an SSM as a condition in the design. A key value pair for the parameter is specified in the design and the key is referenced as a string in the model_parameters dict. The function then unpacks the values from the design for the model as parameters. The advantages of this are shown in Section 4.

The second function is "calculate_response_times_for_ssm", which determines the response times of the simulated traces for the onsets. This will be discussed further in the description of the onsets. Finally, the "simulate_component" function which ultimately performs the simulation of the SSM traces based on the transferred DriftComponent and design. This calls the function "trace_sequential_sampling_model" which is now added to the "Model simulation" functionality in the following extended diagram Figure 7 of the building box.

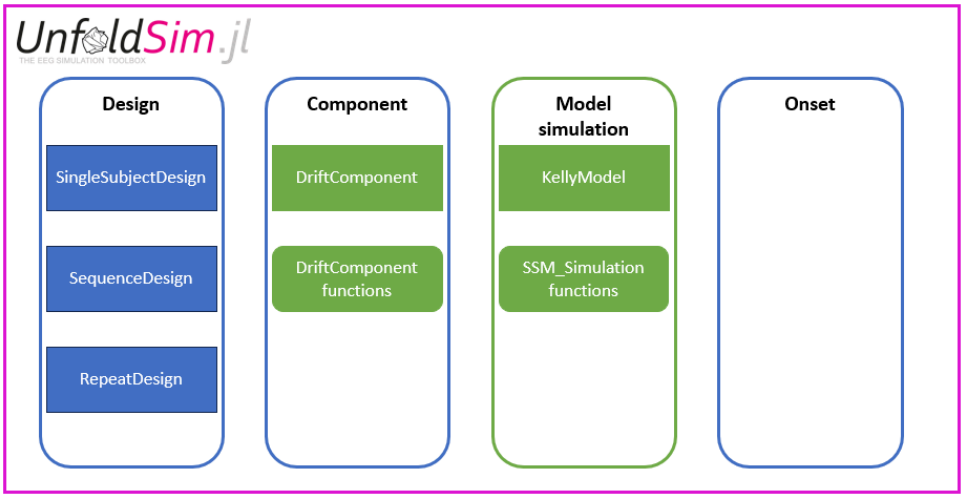

Figure 7: Building box with component adjustments of UnfoldSim.jl package

The structures and functions described in the previous section Section 3.1, such as the **KellyModel**, have now been added to the Figure 7. On the other hand, new functions such as the previously mentioned "trace_sequential_sampling_model" have also been added.

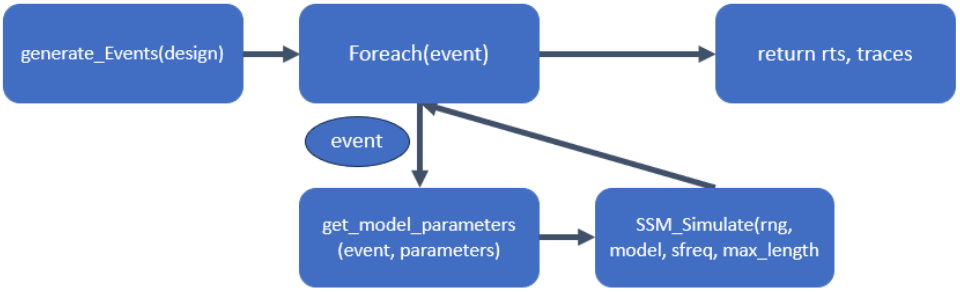

Figure 8: Flow chart of the trace_sequential_sampling_model function

The flow of the function is shown in Figure 8. First, the function creates a DataFrame of events from the given design. The events are iterated over, and the 'get_model_parameters' method is called for each event, allowing for the definition of different parameters per event. The event is represented by an ellipse in the diagram, as it is provided as a parameter in each iteration of the loop. The actual simulation of the response time and the evidence accumulation of the model can then be carried out. This is done by calling the "SSM_Simulate" method, which has also already been described. Once all events have been processed, a list of response times and a matrix of the associated traces of evidence is returned. This function therefore maps the complete simulation scope of **SSMs**.

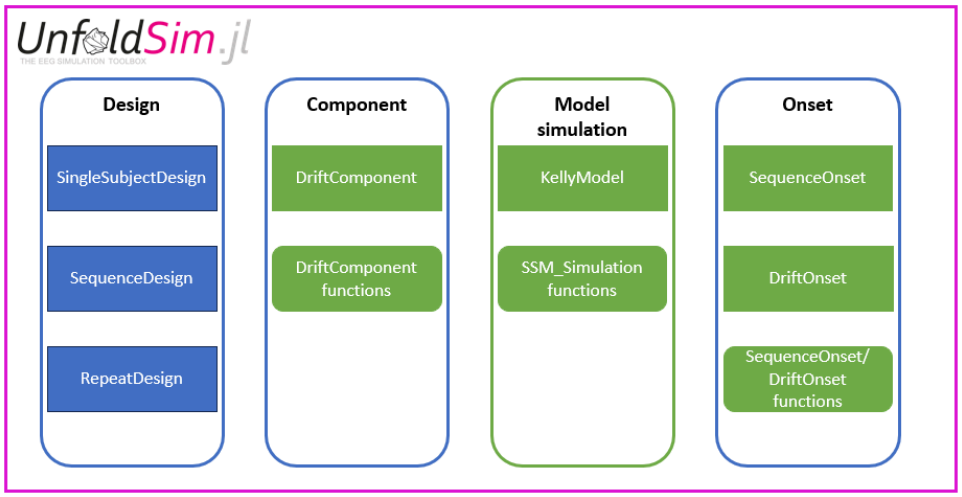

Figure 9: Building box with onset adjustments of UnfoldSim.jl package

The last building block is the onset component. This is used to define a time onset for events in EEG, i.e. when an event has to start. The structures and functions added to the entire building box are therefore entered in the Figure 9. Two structures have been added to the onset group, the SequenceOnset and the DriftOnset. As the name suggests, a SequenceOnset defines a dictionary of event to onset relations for a SequenceDesign. This means that each defined event in the sequence can have its own onset definition.

An onset is therefore also required for the DriftComponent. This was defined with the DriftOnset structure itself. The idea behind the DriftOnset is to use the response time simulated by SSM as the onset for the next event. This is why the aforementioned function "calculate_response_times_for_ssm" was implemented, which returns the pure response times of the simulation and can therefore be used as an onset. To create the DriftOnset, the general function "simulate_interonset_distances" is dispatched and calls "calculate_response_times_for_ssm" and returns the response times. As an additional functionality, the function is dispatched again, but this time it accepts a tuple (DriftOnset, UniformOnset). A UniformOnset can be used to define a fixed value as an onset. This is used here to have a fixed influence on the DriftOnset. Examples of use are shown in Section 4.

Another convenience function is a dispatch of the "simulate_interonset_distances" function with the option of specifying "_" as an onset. This special specification means that the maximum length of the respective component is used as the onset. In sequence design, for example, this can be used as a separation after each sequence.

Last but not least, the all-encompassing dispatch function for the SequenceOnset must be created. This has the special task of generating all onsets for the events in the sequence and adding them up at the end. Since a EEG is a continuous signal and therefore the onsets always increase from event to event. In addition, in this case there is the special feature that in the SequenceOnset the defined onset of a component defines the actual onset for the next component. An example is provided below for better visualisation.

A

Event	Onset
S	UniformOnset(100)
C	DriftOnset()
R	UniformOnset(150)

B

Event	Onset per component	Onset shifted	Onset accumulated
S	100	150	150
C	320	100	250
R	150	320	570
S	100	150	720
C	280	100	820
R	150	280	1100

Table 1: SequenceOnset Example. **Table A:** Event Onset definition for a 'SCR' sequence. **Tabel B:** Onset calculation sample for two repetitions of the sequence.

The onset definitions of the events are given in Table A, which results in an onset for 'S' of 100 time steps for 'C' a flexible one depending on the response time and for 'R' of 150 time steps. This is also reflected in the second column of Table B, as the example sequence of two repetitions is shown here. The third column of B then shows the onsets shifted by one, which results in the start of each subsequent event. Finally, in the final result in column four of table B, the totalised onset times are given, which then also apply to the **EEG**.

This completes the source code technical integration in UnfoldSim.jl and all elements have been created or reused appropriately if possible. The following shows a complete picture of the interaction of the elements.

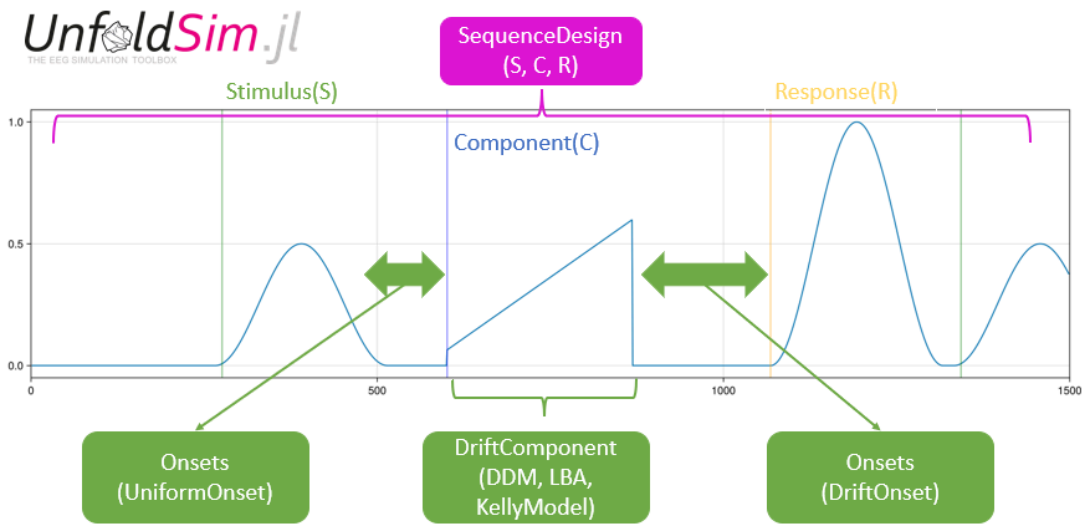

Figure 10: Complete picture of the elements interaction in the UnfoldSim.jl package

This example shows an 'SCR' sequence design. A DriftComponent with the **LBA** was used for the 'C' component. The DriftOnset was therefore also specified for 'C' in the SequenceOnset. By shifting the SequenceOnset, the DriftOnset is not used for 'C' as shown in the figure, but for the subsequent 'R' Event(yellow). Similarly, the UniformOnset of the 'S' component is applied to 'C', blue event. This shows the entire chain of interacting modules and can be applied and extended in a modular way. Concrete examples of application are shown in the Section 4.

3.2.2. Tests

In order to deliver a complete development and integration scope, it is also crucial to provide reasonable tests in the implementation. In the UnfoldSim.jl package, unit tests have already been created for the main elements, design, component and onset. It was therefore necessary to create tests for the added elements as shown in the building box Figure 9.

Unit tests were therefore written for the DriftComponent and associated functions to check the basic functionality of the functions. As the two implemented simulation methods access the functions of the "Model simulation", their integration was also tested. Another important test coverage was the verification and application of a random seed, which can be used to check the reproducibility of the simulation results and their consistency. As this is a very important part of the simulation, because it should deliver the same results for the same random seed applied.

The tests for the "Model simulation" group had to be newly created, just like the implementation itself. On the one hand, the **KellyModel** structure was tested. On the other hand, the basic functions of the **SSMs** simulation were tested. This included unit tests from the lowest level of the individual model simulations to the overarching method "trace_sequential_sampling_model". Its integration into the overall simulation has already been tested on the DriftComponent side.

Finally, the onset structures and their functionalities were tested. This was tested analogue to the resolution of the onsets shown in Table 1. It was particularly important to test the special functionality of the combination of DriftOnset with UniformOnset, as this represents an important core component for later use cases Section 4.

Overall, the newly added functionalities were tested as part of the existing test structure of the UnfoldSim.jl package. This ensures functionality and integration in line with the existing framework.

3.2.3. Documentation

The last equally important part of an implementation and integration into an existing package is the documentation. This includes code comments, DocStrings and also a contribution to the public documentation page of the package.

The development team created a specific DocString template for the UnfoldSim.jl package. This was also used here and a suitable DocStrings were written for each new function and structure. This facilitates the contribution of other interested parties to the package enormously and is also essential for uncomplicated use of the extension. In addition, inline code comments have been added for longer, more complicated functions, such as the **KellyModel** simulation function. This ensures easier maintainability and customisation.

The public documentation page of the UnfoldSim.jl package is structured according to the Diátaxis framework (Ament, n.d.). Technical documentation is divided into the four categories *tutorials, how-to guides, technical reference and explanation.* As part of the added DriftComponent extension, it makes sense to design two *how-to guides.* A somewhat simpler basic guide on how to use the DriftComponent for simulation. A second, somewhat more complex guide, which deals with a more specialised use case.

Therefore, the basic how-to guide explains "How to Simulate an Evidence Accumulation EEG". It explains step by step how the process works. First the setup, then create a design, select the components, define onsets and finally simulate and display in a figure.

In the second how-to guide, "How-to Simulate an Evidence Accumulation Overlap and Deconvolution", the use case of an overlap simulation and subsequent deconvolution described in the following Section 4.2 is explained.

4. Use Cases / Applications

The possible use of the UnfoldSim.jl extension is shown below. The fundamental options are first demonstrated through three examples — one for each model — illustrating the simulation space. This approach provides a clear understanding of the basic functionalities. The second section deals specifically with the case of overlap correction from the comment by (Connell et al., 2024). The overlap correction is shown using the example of two different **SSMs**.

4.1. Simulation Space

In this basic use case, the three **SSMs** **LBA**, **DDM** and the **KellyModel** were used to demonstrate their possible uses. The same setting of basic requirements for design, components and onsets as well as the presence or absence of noise was used for each of the three experiments. The design consists of a SingleSubjectDesign, which has one condition of two different drift_rates (0.45, 0.8). This design is then supplemented by an "SCR" SequenceDesign, which results in 6 events that are shown in Table 2.

	Event	drift_rate
1	S	0.45
2	C	0.45
3	R	0.45
4	S	0.8
5	C	0.8
6	R	0.8

Table 2: 'SCR' SequenceDesign with two different drift_rates

The events S,C,R stand for e.g. Stimulus, Component, Response. This design would thus represent two trials of an experiment, for example of one participant for two drift_rate conditions. In order to ensure a slightly larger amount of data in the experiment, the RepeatDesign is applied to it, whereby the six events from Table 2 are repeated. In this scenario, the SequenceDesign was repeated 100 times, resulting in 600 events. This completes the creation of the design.

For the components an existing LinearModelComponent was used for the events S,R. This was used with an hanning function to create two normally distributed peaks at 0.5 for the stimulus and −0.5 for the response. The 'C' component is then the DriftComponent with the respective **SSM**. The onsets between the events are defined by two UniformOnsets for S,R with a fixed length equivalent to that of the component. The standard DriftOnset is used for the **SSM**, whereby the pure response time of the models is applied.
The sampling frequency has been set to 500 and the associated maximum time to 1.0.

Basic LBA Simulation: The parameters of LBA were specified as shown below in Table 3.

Linear accumulation rate (v)	Starting point distribution (A)	Threshold (k)	Non-decisional time (Ter)
"drift_rate"	0.01	0.6	0.0

Table 3: Model parameters for LBA simulation

The linear accumulation rate is taken from the design. The start point shift is kept to a minimum and no non-decisional time is specified so that the response appears directly after the peak.

In the following Figure 11, the results of the simulation are shown once with PinkNoise(noiselevel=1) (Wikipedia contributors, 2024) and once without. The noiselevel indicates how the noise is scaled. Therefore, a noiselevel=1 indicates the basic PinkNoise, as it is multiplied by 1. Tile A shows the raw EEG signal from two repetitions of the sequence. The vertical lines show the onset of the respective event. As you can see, the events of the 'SCR' sequence occur as defined in the design and there are no overlaps due to the matching onsets. The difference between the two drift_rates defined in the design can also be seen in the 'C' component, with the first accumulating for longer time than the second. Cause the amount of evidence gained per time step is lower.

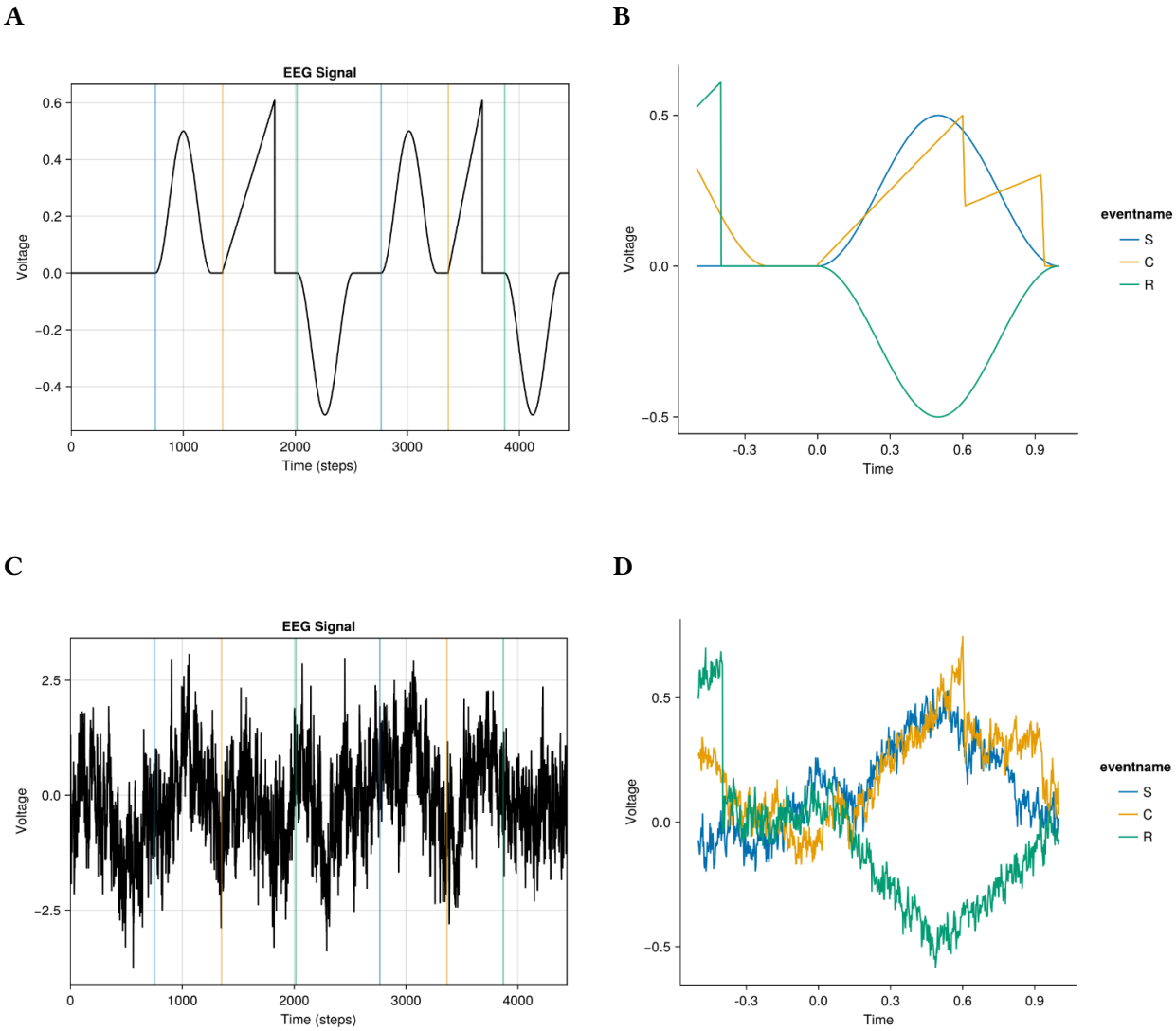

Figure 11: EEG and ERP of a 'SCR' sequence using LBA Model as 'C' component. **A:** Raw EEG signal from two repetitions of the 'SCR' sequence. **B:** ERP created from the raw EEG. **C:** EEG signal with PinkNoise(noiselevel=1). **D:** ERP with PinkNoise(noiselevel=1).

The corresponding ERP of the three events created using the Unfold.jl package is displayed directly next to it in tile B. In it, the components are averaged over all occurring events in the EEG and thus result in the average expression of these. No difference is to be expected for the events 'S' and R, as these are always the same. For event C, variations in the drift_rate parameter used in the simulation are reflected in the ERP. Since the two simulated traces differ in duration, their overall average also changes. As shown in the figure, both peaks are no longer as high as the threshold. This occurs because the lower drift_rate traces contribute to the first peak, reducing its magnitude, as their values remain below the threshold at this point in time. Similarly, the second peak of the lower drift_rate is significantly lower because the higher drift_rate traces have already returned to zero, pulling the overall average down.

The bottom two tiles C and D of the Figure 11 show the same diagrams as in the top two. However, in this case the PinkNoise was used in the simulation. The noiselevel makes the raw EEG signal hard to interpret visually, highlighting the need for modelling-procedures as applied throughout this thesis. By applying epoching to get the ERP, the general shapes of the original components of the three events can again be easily recognised despite the noise.

This is the basic use case for the LBA, which can be extended in various ways and adapted to the desired research objectives. An example of this is shown in Section 4.2 when applying an overlap correction.

Basic DDM Simulation: For the simple simulation of the sequence with the DDM, its parameters are defined as in the following Table 4.

Drift rate (v):	Starting point of accumulation (z)	Threshold distance (a)	Non-decisional time (Ter)
"drift_rate"	0.5	4.0	0.2

Table 4: Model parameters for DDM simulation

The drift_rate is again specified in the simulation design and therefore varies from trial to trial. The starting point is set to a positive value in order to set a bias towards the positive threshold. The threshold value is set significantly higher compared to LBA, as we no longer have linear accumulation and the threshold value may be reached quite early. In addition, this time a non-decisional time of 0.2 is set as an example, which delays the response time and thus the onset of the response.

Once again, the EEG and ERP of the simulation are shown in the Figure 12 once without (A,B) and once with PinkNoise (C,D). In the EEG signal without noise, from tile A, the differentiated diffusion curve that is typical for the DDM can be seen directly. Compared to the LBA, the variability in the accumulation process is more apparent, as the progression is no longer strictly linear. For instance, in the first trial, the accumulation initially moves in a positive direction before ultimately reaching the negative threshold. In contrast, in the second trial, no threshold is reached, and the simulation is terminated after reaching the maximum simulation duration. This highlights the increased unpredictability introduced by drift variability, making it nearly impossible to precisely determine when the threshold will be reached for a given drift_rate.

As expected, the ERP in tile B Figure 12 shows that, on average, the 'C' component tends toward the positive threshold. This is evident from the initial positive bias in the starting value and the overall positive trend in the accumulation process. In decision-making terms, this suggests a predisposition toward one decision over the other. Another notable observation is the negative time of the 'R' component, where the prior ramp-up of the accumulation process is clearly visible.

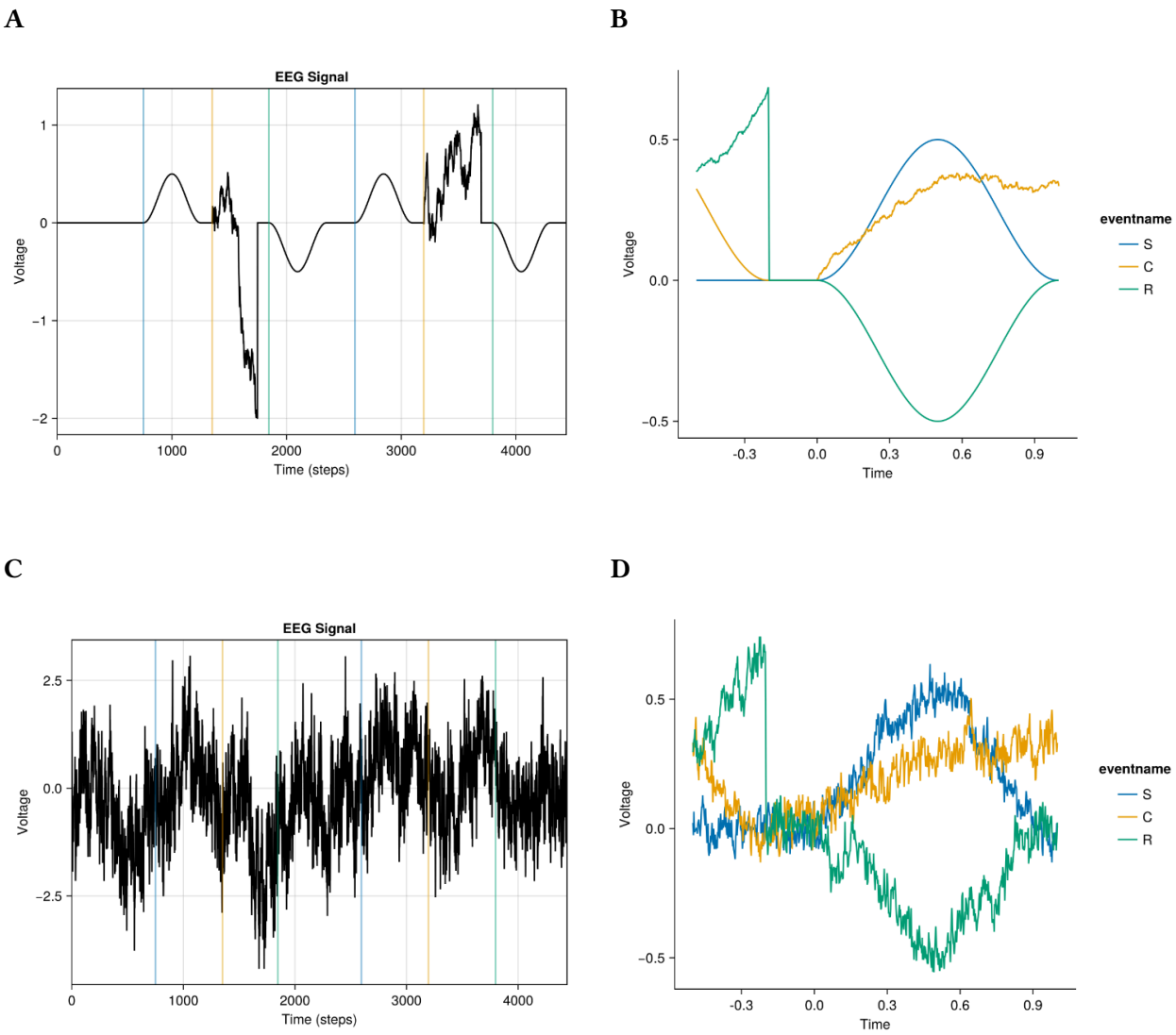

Figure 12: EEG and ERP of a 'SCR' sequence using DDM Model as 'C' component. **A:** Raw EEG signal from two repetitions of the 'SCR' sequence. **B:** ERP created from the raw EEG. **C:** EEG signal with PinkNoise(noiselevel=1). **D:** ERP with PinkNoise(noiselevel=1).

In tiles C and D, the simulation with PinkNoise also behaves in the same way, as expected. This means that the clear trends of the events can be recognised in the ERP in D.

This basic simulation of the DDM already demonstrates, to some extent, how biases can influence the decision-making process. Next, the basic application of the KellyModel is demonstrated, which, as previously mentioned, provides significantly more parameters for adjustment and influence.

Basic KellyModel Simulation: As the KellyModel offers such a large number of parameters, the default parameters are used in this basic example. These were determined through various tests and defined for the model. In this example simulation only the drift_rate parameter is specified. As in the other two use cases the drift_rate originates from the simulation design. But this time the rates are higher (5.45, 7.8) compared to the ones used before, for better visibility of differences.

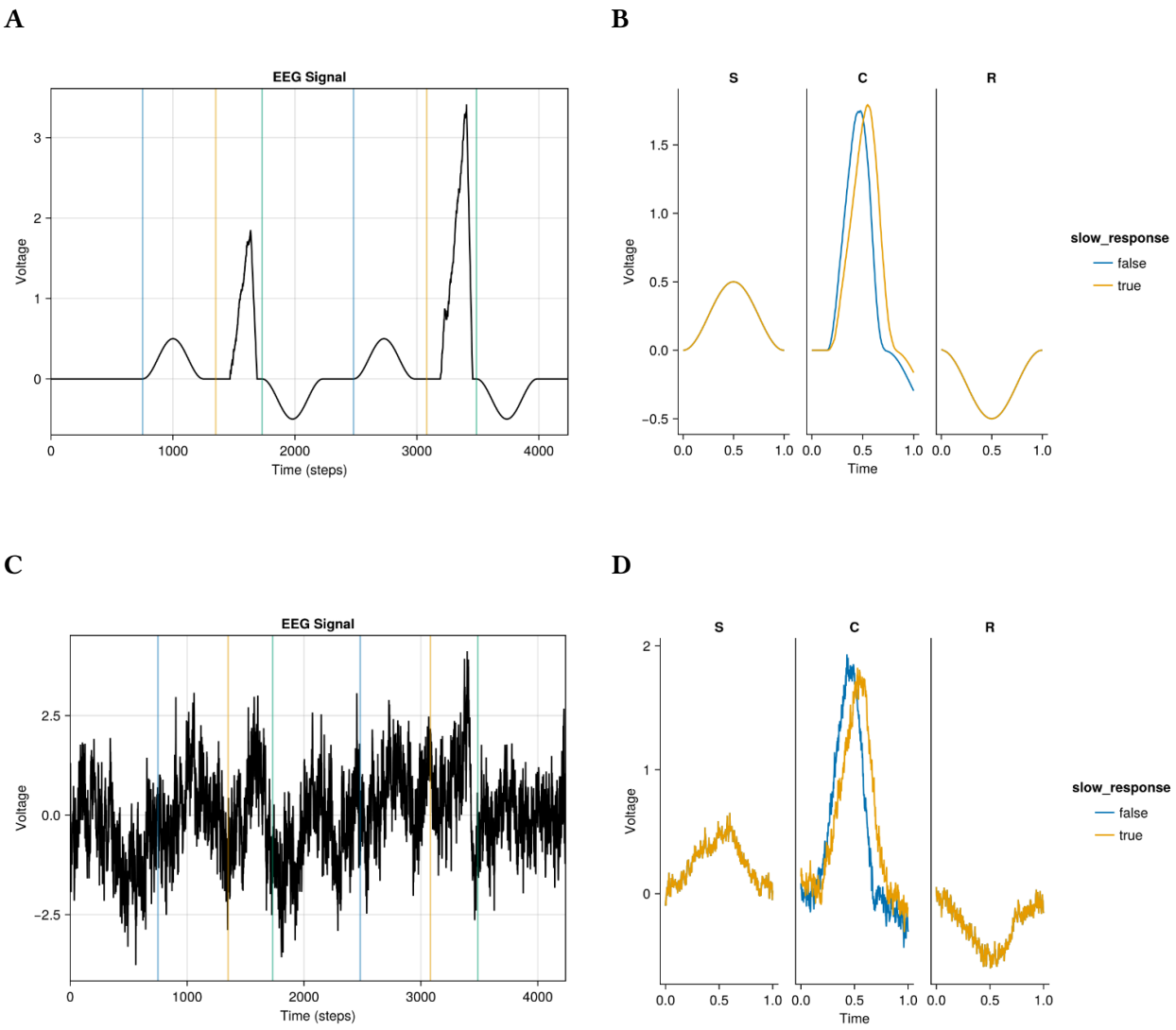

Figure 13: EEG and ERP of a 'SCR' sequence using KellyModel as 'C' component. **A:** Raw EEG signal from two repetitions of the 'SCR' sequence. **B:** ERP with a distinction between fast and slow response time. **C:** EEG signal with PinkNoise(noiselevel=1). **D:** ERP with a distinction between fast and slow response time and PinkNoise(noiselevel=1).

In the diagram of the EEG signal without noise from Figure 13 A, you can directly see how the additional parameters have an influence compared to the other two SSMs. For example, the delay in the start of the accumulation after the end of the stimulus due to the sensory encoding is recognisable. Furthermore, the excess after reaching the threshold at 1.0 and the subsequent slow linear decay can be seen. This is also reflected in the ERP, as it now looks more like a normal distribution. This is due to the variabilities in sensor and motor delay as well as the difference in drift_rates. Additionally, the ERP now captures the distinction between fast and slow response times. To achieve this, individual trials were categorized based on whether the response time of the 'C' component was above or below the median. The effects of this classification are visible in tile B, specifically in the middle ERP diagram for the 'C' component. Notably, the blue trace, representing trials with shorter response times, reaches the threshold and peaks earlier, as expected. Consequently, these traces also return to zero sooner, indicating an earlier onset of the response event. The 'S' and 'R' components, however, do not show any effect of this division, as they are not influenced by it.

Tiles C and D using the PinkNoise reflect the same findings as the previous simulations and the events are relatively easy to recognise in the **ERP**.

With these basic simulations, the possible customisation options of **SSMs** could be demonstrated using the basic application. In addition, the clear differences between the models can be recognised, making it necessary to consider which model could be used for which research objectives. In the next section, two use cases will be discussed in more detail, one with the **KellyModel** and the other with the **LBA**.

4.2. Deconvolution as a method for correcting an overlap between components

As specified in the description of the UnfoldSim.jl package, such a package can be used to analyse tools and test, validate other toolbox functionalities. This is precisely the case in the two use cases that are build up below. In these cases, the deconvolution method is analysed with the help of the DriftComponent extension. In addition, it can also be shown how the extension can support the further investigation of the discussion between (Frömer et al., 2024) and (Connell et al., 2024), described in Section 1.1.

Simulate an Overlap using the KellyModel: The aim of this UseCase is to simulate an overlap between a simple stimulus component and the DriftComponent with the **KellyModel**. This overlap should then be separated back into the individual components by deconvolution.

The basis from Section 4.1 is used again for the setup of the simulation. However, this must now be adapted slightly to create the desired overlap. The drift_rates in the design are changed to 3 and 5. In addition, the negative response component has been replaced by a positive one to enable a positive overlap. There are now two ways to create an overlap from the evidence accumulation to the response component. Firstly, the motor encoding delay parameters (motor_onset, motor_delay) and post accumulation spike parameters (post_accumulation_duration, post_accumulation_duration_variability) of the **KellyModel** could be adjusted so that there is no delay and the spike is extended. This would cause the response component to start earlier and cause an overlap. An example parameter definition for this is shown in the following Table 5.

motor_onset	motor_delay	post_accumulation_duration	post_accumulation_duration_variability
0.1	0.05	0.4	0.1

Table 5: Model parameters for Advanced Drift Diffusion Model from Kelly et. al. (**KellyModel**) overlap simulation

Another somewhat simpler option is to shorten the onset of the subsequent response component and thus have it start earlier. The implementation used here is that a DriftComponent accepts a tuple of onsets. This means that in addition to the DriftOnset, which returns the standard response time of the model, a UniformOnset is also defined. With the UniformOnset, a fixed value can be added/subtracted to the given response time of the DriftOnset. In the simulation, the default parameters of the **KellyModel** were used.

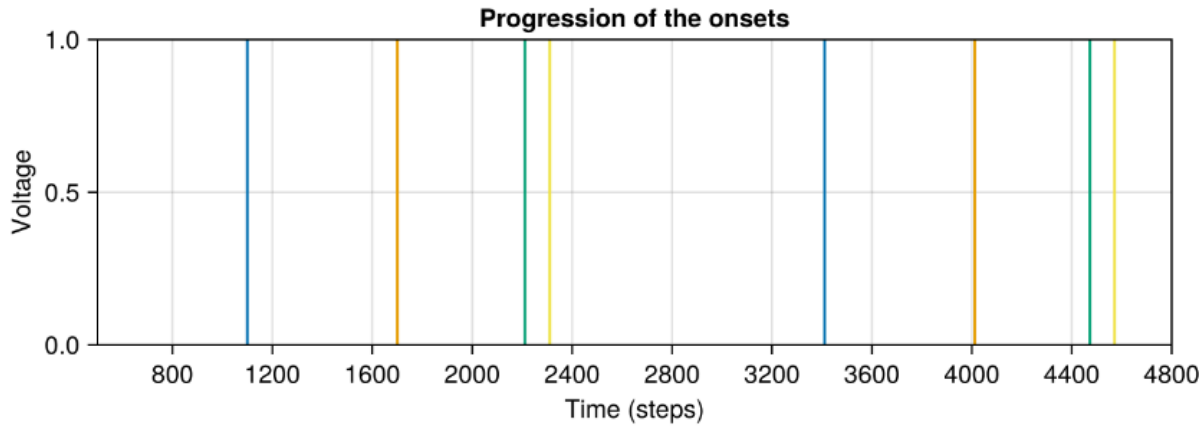

Figure 14: Progression of the onsets for a 'SCR' sequence with the following onsets defined: 'S'=>UniformOnset(width=40,offset=600), 'C'=>(DriftOnset(), UniformOnset(width=0, offset=-100)), 'R'=>UniformOnset(width=100,offset=1000). The onset of 'S' event is marked with the blue line, 'C' with orange, 'R' with green and the DriftOnset response time with yellow.

The onsets for the simulation are defined as follows:
- 'S'=>UniformOnset(width=40,offset=600),
- 'C'=>(DriftOnset(), UniformOnset(width=0, offset=-100)),
- 'R'=>UniformOnset(width=100,offset=1000)

The 'S' and 'R' components use a UniformOnset with a fixed offset and a variable width, which follows a standard deviation. In contrast, the 'C' component is defined as a tuple combining DriftOnset and UniformOnset, ensuring an intentional overlap with the upcoming 'R' event. The offset of −100 in UniformOnset subtracts 100 time steps from the response time provided by the DriftOnset.

An important aspect to note is that the onsets are cyclically linked, meaning:

- The onset definition of 'S' determines the beginning of the 'C' event.
- The onset definition of 'C' determines the start of 'R'.
- The onset definition of 'R' determines the next occurrence of 'S'.

To better understand this concept, refer to Section 3.2.1, where the implementation details are described. Additionally, Figure 14 illustrates an example of two trials of onsets. In this figure:

- The first event 'S' (marked in blue) begins after 1100 time steps, as dictated by the 'R' onset definition.
- Then, the 'S' onset definition (with an offset of 600) triggers the 'C' event (marked in orange).
- The 'C' component onset has two marks:
  - The response time determined by the DriftOnset (yellow mark).
  - The final 'C' onset, which is the sum of the DriftOnset and the UniformOnset (with a −100 offset), resulting in the green mark.
- The green mark then determines the start of the 'R' event.

This process is repeated across all trials, ensuring consistent onset relationships between the components.

The results of the simulation are shown below in Figure 15.

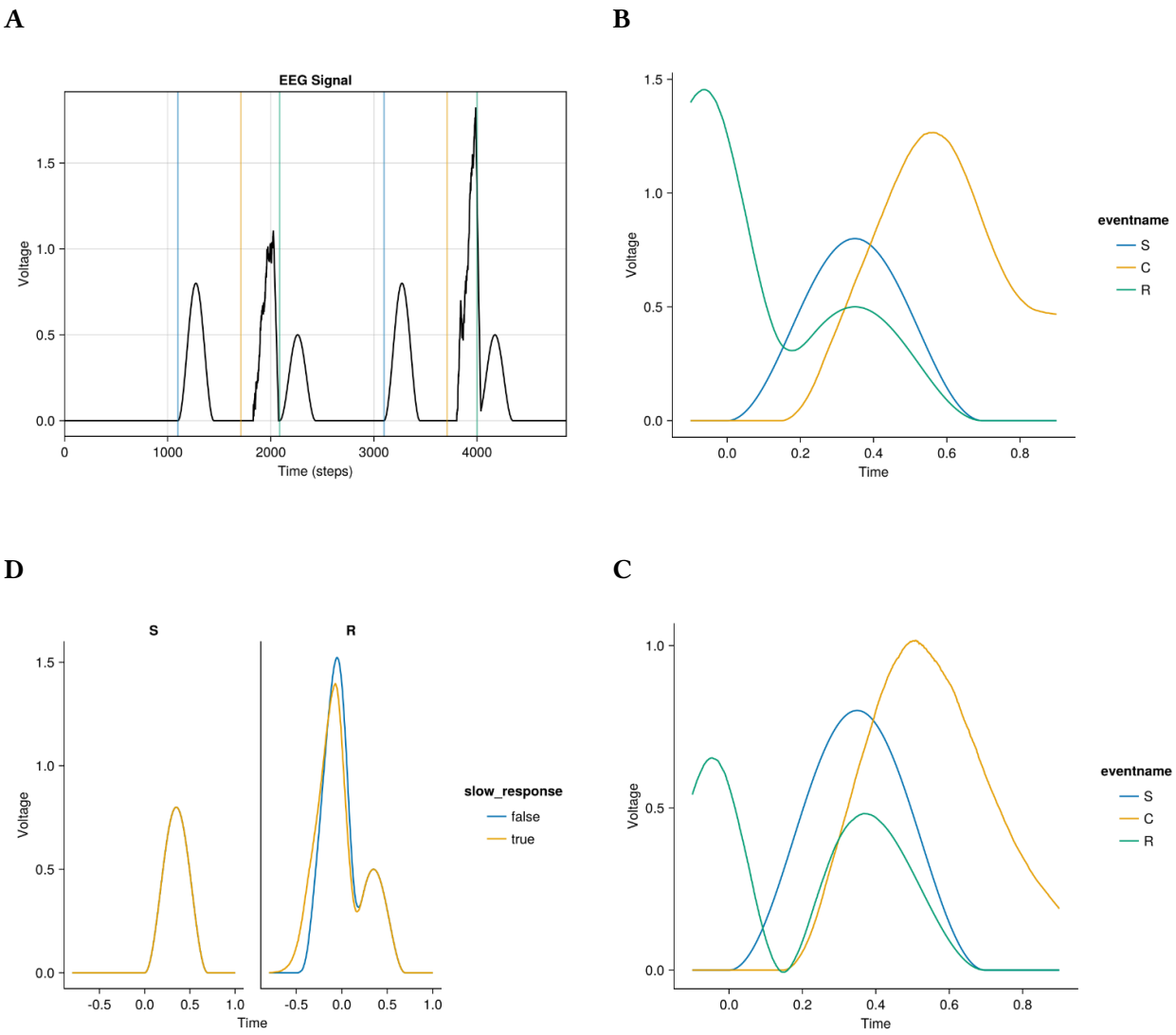

Figure 15: EEG and ERP of a 'SCR' sequence using KellyModel as 'C' component with an overlap between 'C' and R

Tile A shows the EEG signal from two trials. A minimal overlap can be seen in the first and a somewhat clearer one in the second. This results from the fact that a variability in motor delay is integrated in the KellyModel parameters. This overlap is directly reflected in the ERP from tile B. In this you can see how the response component never starts at 0 as it should and thus includes the previous course of the 'C' component. The same can also be seen at the end of the 'C' component, which does not run towards 0 but starts to reverse again at 0.5 and merges into the 'R' component.

Such an overlap is not unusual in EEG research and occurs relatively frequently, as shown by (Frömer et al., 2024) and (Connell & Kelly, 2021). The Unfold.jl package offers a possibility of deconvolution for a renewed equalisation into the individual components. The resulting ERP is shown in tile C directly below the "normal" ERP. In tile C you can see in comparison how the deconvolution method worked and the 'R' component starts again at 0 and assumes the normal course except for a small time shift. The same can also be seen for the 'C' component at the end, where it now also runs in the direction of 0.

This showed how an overlap using **KellyModel** and two basic components can be simulated and corrected at the end. It can therefore be concluded that the deconvolution method of Unfold.jl works for such a use case.

As an additional small investigation, the influence of the choice of two different drift_rates can be seen in tile D. In this, the total DriftComponent events were again divided into fast and slow responses according to their mean response time. This shows that the processes labelled as slow reactions have a lower and slower peak value. It is mainly due to the lower drift_rate of 3 in contrast to the fast reactions with 5. Thus showing how such variation in experimental design can be utilised for different experiments.

No component activity after deconvolution: In this second use case, the first is extended again and an attempt is made to adapt it in the direction of the findings from (Frömer et al., 2024). In their publication, as already described in Section 1.1, they had shown that in some studies investigating decision-making, the ramp-up component in **CPP** disappears after deconvolution. This needs to be replicated in a simulation, where two overlapping components create the illusion of a third component in the **ERP**. However, after applying deconvolution, this apparent component disappears, revealing that it was merely an artifact of signal overlap. This extends the previous use case, where a real component contributed to the overlap, and all genuinely present components remained after deconvolution. In contrast, this simulation demonstrates how an artificially induced activity —one that never existed as a distinct component— vanishes upon proper signal separation, highlighting the effectiveness of deconvolution in disentangling overlapping signals.

A sequence design consisting of two “SR” events was created for the simulation setup. Like the previous designs, this was supplemented with a list of drift_rates. This time, the list of drift_rates contains 500 normally distributed values with an average of 1.2 and a standard distribution of 0.5 and therefore a result as shown in Figure 16.

1.39	2.00
1.98	1.30
1.11	0.32
0.94	0.32
2.13	0.72
...	...
1.65	2.03
1.26	1.43

Figure 16: Drift_rate distribution for simulation

Due to the resulting 500 conditions, no further repetition of the design was carried out, as there are therefore 1000 events. The Kelly model with the parameters defined as follows Table 6 was used as components for ‘S’.

drift_rate	motor_onset	post_accumulation_duration	ramp_down_duration	boundary
5.5	0.6	0.0	0.6	0.5

Table 6: Model parameters for **KellyModel** deconv use case simulation

This parameter setting ensures that the **KellyModel** reaches the boundary early and pulls the ramp down into the next ‘R’ component with a long delay. The motor delay and ramp down are therefore extended and the post accumulation spike is prevented.

The component for 'R' was defined as a DriftComponent with an LBA. This uses the drift_rate specified in the design and has the following parameter definition.

Linear accumulation rate (v)	Starting point distribution (A)	Threshold (k)	Non-decisional time (Ter)
"drift_rate"	0.01	1.8	0.0

Table 7: Model parameters for LBA deconv use case simulation

This defines the components to simulate an activity between these two, using an overlap. As an onset for the KellyModel, the option of specifying a tuple as *(DriftOnset(), UniformOnset(width=0,offset=-10))* was used. With this notation the response time can be reduced slightly by 10. As the UniformOnset has an offset of −10. For the LBA, the maximum length of the possible simulation of the component (sample_frequency*time_end) was selected as onset. The simulation produces the results shown below in Figure 17.

A **B**

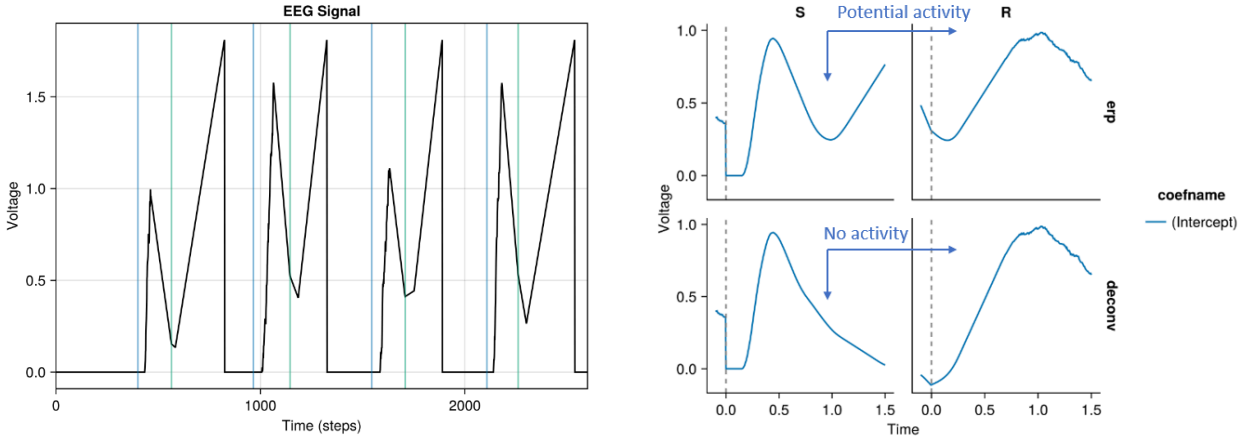

Figure 17: EEG and ERP of a 'SR' sequence with an overlap between 'S' and 'R'. **A:** Raw EEG signal for four repetitions of the sequence, showing the overlap between the two components. **B:** ERP and deconvolution of the 'S' and 'R' component.

Tile A shows the EEG of the trials. This clearly shows how the two components overlap. During the ramp down of the KellyModel, the LBA already begins, which means that it starts to increase. In addition, the variations can be seen in the drift_rate and the delay as well as the resulting intensity of the overlap.

In the "normal" ERP in tile B in the above figure, the events described show an activity between 'S' and 'R'. In the figure, this could suggest that there is another active component between the two components 'S' and 'R'. This component would therefore not be an artefact as an overlap but a real brain activity. However, this cannot be answered in the ERP and can only be guessed whether it is a real activity or just an artefact coming from a overlap. As a component overlap can look like evidence accumulation. Therefore a deconvolution would be advisable.

The lower part of tile B from Figure 17 shows the ERP with deconvolution. In this, the 'S' and 'R' components have been separated from each other, causing the supposed activity to disappear. As a result, there is no longer any effect between the two components, which means that they once again correspond to their original separate form. This shows that if there really is a component between the two, it should also be visible in the deconvolution. As it was shown in the previous use case. But now

in this one similar to the findings of (Frömer et al., 2024) a activity coming from an overlap can be removed by using the deconvolution method.

(Connell et al., 2024) comment on (Frömer et al., 2024) rebuild: In this last use case, a replica of a study from the comment by (Connell et al., 2024) is created. This is intended to demonstrate the further application possibilities of the extension in research. In the study by (Connell et al., 2024), the methodology used to analyse the EEG studies by (Frömer et al., 2024) was investigated by using ramping evidence accumulation signals. The focus was primarily on the "normal" mass-univariate ERP and the deconvolution with Unfold.jl. In the comment, ramping evidence accumulation signals (similar to LBA) were simulated from a stimulus to the peak and thus response with different drift_rates. These were then displayed in the mass-univariate ERP and the deconvolution.

For the reconstruction with this Unfold.jl extension, an 'SR' design is therefore used, where 'S' represents a drift component with LBA and 'R' is an empty component which only has the event onset at the peak of the LBA. For the variability in the drift_rates, the list of distributed drift_rates was added to the design as in the previous use case (see Figure 16). The following parameters listed in Table 8 were selected for the LBA in order to obtain a replica that is as true to the original as possible.

Linear accumulation rate (v)	Starting point distribution (A)	Threshold (k)	Non-decisional time (Ter)
"drift_rate"	0.01	0.8	0.0

Table 8: Model parameters for LBA comment rebuild use case simulation

In the comment, the signals also started at 0 and had a threshold of 0.8. The non-decisional time is set to 0, as this means that the response is directly at the peak of LBA. The DriftOnset was also selected for this component and the maximum simulation length of 1 was simply selected for the empty component.The simulation could then be carried out and the results shown in Figure 18 were obtained.

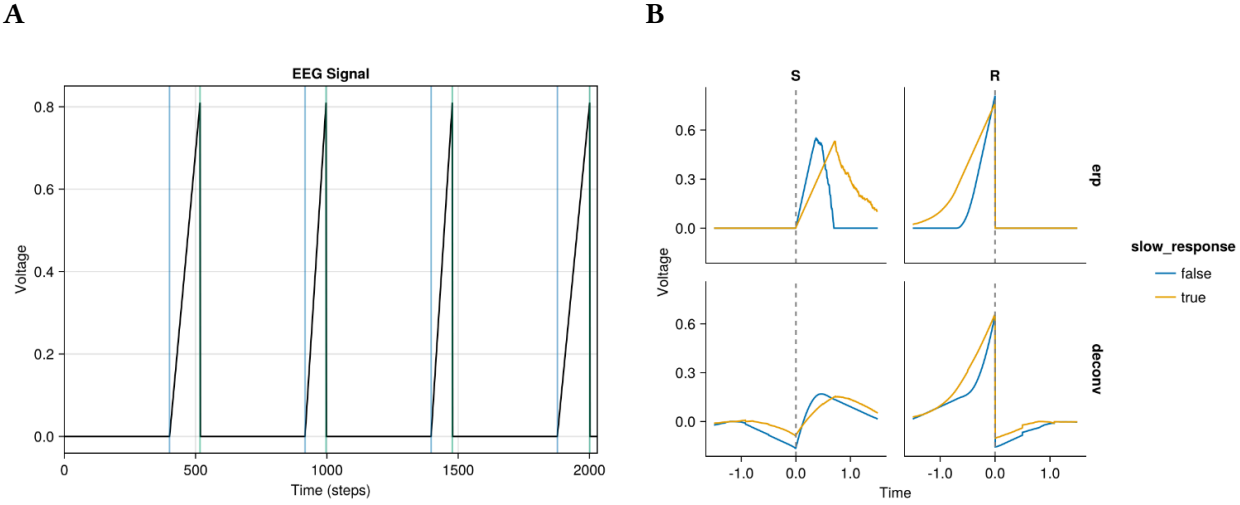

Figure 18: EEG and ERP of the rebuild from (Connell et al., 2024). **A:** Raw EEG signal of four repetitions. **B:** ERP and deconvolution of the 'S' and 'R' locked events.

On the left-hand side in tile A, the raw EEG signal is shown again. This shows how the ramping evidence accumulation signals from the stimulus onset in blue to the response onset in green show a linear accumulation. The variation of the distributed dirft_rate can also be seen. This looks very comparable to the upper part of Figure 1b from the comment by (Connell et al., 2024). The more interesting part, however, is tile B, which contains the summary of Figure 1b and 1f from the comment by (Connell et al., 2024). In tile B, the mass-univariate ERP is shown in the part above, once stimulus-

locked and once response-locked. This representation is very close to that of (Connell et al., 2024) from Figure 1e, but the overall distortion of the x-axis should be noted. The lower part of tile B also shows the deconvolution of the stimulus-locked and response-locked component. This comes very close to Figure 1f from (Connell et al., 2024) and thus also fulfils the expectations. Overall, this final use case demonstrates how the Unfold.jl extension can be applied in research. For instance, the simulations conducted by (Connell et al., 2024) could have been performed using this extension. Moreover, this represents just one specific example where **SSM**-based simulations were used for a targeted investigation. In the future, there may be even greater potential for utilizing such simulations to explore neural processes underlying decision-making.

5. Discussion

The results of this study demonstrate the feasibility of using **SSMs** to simulate **EEG** activity and examine the neural correlates of decision-making processes. The use cases illustrated how the implemented UnfoldSim.jl extension can simulate event overlap in **EEG** data and how deconvolution can effectively remove such overlaps. This highlights its potential as a validation tool for **EEG** analysis methods. Additionally, one use case demonstrated the usability of the extension by kind of replicating aspects of the comment from (Connell et al., 2024), further showcasing its applicability. This highlights how the toolbox extension could be used in research in the future. Therefore the extension serves as a valuable resource for cognitive neuroscience research, allowing researchers to investigate decision-making processes in a modular and flexible manner.

A key discussion point is the extent to which these models can represent aspects of real evidence accumulation processes in **EEG** data. This topic was not examined in detail in this thesis, as the primary focus was on general simulation capabilities and use cases related to event overlap. Consequently, the default model parameters were chosen for broad applicability rather than for biological accuracy. Among the implemented models, the **KellyModel**, with its additional parameters — such as sensory encoding delay and post-accumulation dynamics — offers the highest potential for biological plausibility. Future studies could optimise its parameters to better mimic real evidence accumulation processes. For instance, model parameters could be systematically adjusted, and simulation outputs iteratively compared with empirical **EEG** data until a high level of similarity is achieved. Successfully replicating real evidence accumulation in **EEG** could pave the way for more realistic analyses of human decision-making processes.

The choice of **SSMs** used in this work was primarily based on the degree of familiarity from the literature and to enable a certain degree of differentiation. In addition to the two selected models **LBA** and **DDM**, there are several other **SSMs** in the literature. However, the extent to which the models also fit well with the topic of evidence accumulation and whether there is a better selection of **SSMs** was not investigated. In this case, it would therefore make sense to consider other models in a future study. The inclusion of alternative models, such as the racing diffusion model (Tillman et al., 2020) or the attentional drift diffusion model (Krajbich et al., 2010), could offer additional perspectives on the decision dynamics and further improve the variation of the **EEG** simulations.

A key methodological contribution of this thesis is the use of **EEG** simulations to validate deconvolution techniques. This aligns with an ongoing debate in cognitive neuroscience, highlighted by the conflicting interpretations of (Frömer et al., 2024) and (Connell et al., 2024) regarding the application of deconvolution in analysing evidence accumulation processes. The use cases presented in this thesis demonstrate how component overlaps in **EEG** data can be simulated and subsequently separated through deconvolution. This approach allows for the systematic investigation of potential misinterpretations, where evidence accumulation signatures may artifactually emerge from stimulus-response overlap.
Through deconvolution, (Frömer et al., 2024)‘s findings suggest that response-locked evidence accumulation patterns — often referred to as the **CPP** effect — previously thought to reflect true accumulation may, in fact, be artifacts of overlapping neural activity. The first two use cases in Section 4.2 kind of replicated these conditions using simplified simulations. The first case confirmed that if a component contributing to an overlap is genuinely present, it remains after deconvolution. In contrast, the second case showed that when an apparent component is merely the result of overlap and does not truly exist, it disappears after deconvolution. This supports (Frömer et al., 2024)’s findings that if an evidence accumulation process is genuinely present, it should remain detectable after deconvolution.

The ability of the Unfold.jl extension to replicate these conditions in part supports the researchers to systematically investigate such artefacts and continue the debate.

Nevertheless, only Unfold's previous deconvolution method was analysed in the use cases. Another possibility for investigation would be the application of the RIDE algorithm. This is a method for the decomposition, reconstruction, and single trial analysis of **ERP** (Ouyang et al., 2015). RIDE allows for the separation of **EEG** components even if they have different latency variability across trials. This is particularly relevant for evidence accumulation processes, where response-locked neural activity may vary in timing due to differences in decision latencies. Therefore RIDE could provide a clearer separation of decision-related **EEG** activity from unrelated neural processes. However, as discussed above, it remains an open question whether the evidence accumulation pattern — **CPP** effect — truly reflects an accumulation process or merely arises from overlapping signals. Applying RIDE in future research could help determine if the pattern persists as an independent signal after decomposition or whether it is an artifact of neural overlap. Resolving this question would provide stronger evidence for or against interpreting the evidence accumulation pattern as a genuine marker of decision-making processes.

Overall, this study underscores the importance of computer-aided modeling in neuroscience and shows how **SSM**-based **EEG** simulations could be used to contribute to a deeper understanding of decision-making processes in the brain. The developed Unfold.jl extension not only facilitates theoretical investigations, but also offers practical implications for the design and interpretation of **EEG** experiments in cognitive neuroscience. Furthermore, the modularity and neat documentation make the implementation easy to expand.

6. Conclusion

This thesis has demonstrated the feasibility and value of using computational modeling to simulate **EEG** activity in the context of decision-making. By integrating **SSMs** such as **DDM**, **LBA**, and the **KellyModel** into the UnfoldSim.jl framework, it provides a structured approach to investigating the neural mechanisms underlying evidence accumulation. The ability to replicate key **EEG** dynamics through these models reinforces their relevance in cognitive neuroscience.

A key insight from this work is that **EEG** simulations can help clarify methodological challenges, particularly in distinguishing true decision-related signals from artifacts caused by overlapping neural activity. The results support the notion that some observed **EEG** correlates of evidence accumulation may be influenced by component overlap rather than reflecting actual decision-making processes. This underscores the necessity of simulation-based validation in **EEG** research.

Beyond its immediate contributions, the Unfold.jl extension developed in this thesis establishes a foundation for future studies. Its modular design allows for further expansion, including the refinement of model parameters and the integration of additional decision-making models. However, challenges such as model assumptions and parameter optimization remain, requiring future improvements to enhance biological plausibility.

In conclusion, this work highlights the critical role of computational simulations in advancing **EEG** research on decision-making. By providing a UnfoldSim.jl extension which enables rigorous testing and refinement of theories and methods, it contributes to the broader effort of unraveling the complexities of human cognition and neural processing.

Bibliography

Ament, D. *Diátaxis Documentation Framework*. https://diataxis.fr/

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/https://doi.org/10.1016/j.cogpsych.2007.12.002

Connell, R. G., & Kelly, S. P. (2021). Neurophysiology of Human Perceptual Decision-Making. *Annual Review of Neuroscience*, *44*(Volume44, 2021), 495–516. https://doi.org/https://doi.org/10.1146/annurev-neuro-092019-100200

Connell, R. G., Corbett, E. A., Parés-Pujolràs, E., Feuerriegel, D., & Kelly, S. P. (2024). Regressing Away Common Neural Choice Signals does not make them Artifacts. Comment on Frömer et al (2024, Nature Human Behaviour). *Biorxiv*. https://doi.org/10.1101/2024.09.26.614447

Ehinger, B. (2025,). *Science*. https://benediktehinger.de/blog/science/

Ehinger, B. V., & Dimigen, O. (2019). Unfold: an integrated toolbox for overlap correction, non-linear modeling, and regression-based EEG analysis. *Peerj*, *7*, e7838. https://doi.org/10.7717/peerj.7838

Forstmann, B. U., Ratcliff, R., & Wagenmakers, E.-J. (2016). Sequential sampling models in cognitive neuroscience: Advantages, applications, and extensions. *Annual Review of Psychology*, *67*(1), 641–666.

Frömer, R., Nassar, M. R., Ehinger, B. V., & Shenhav, A. (2024). Common neural choice signals can emerge artefactually amid multiple distinct value signals. *Nature Human Behaviour*, *8*(11), 2194–2208. https://doi.org/10.1038/s41562-024-01971-z

itsdfish, & contributors. (2025,). *SequentialSamplingModels.jl: A Julia package for sequential sampling models*. https://github.com/itsdfish/SequentialSamplingModels.jl

Jackson, A. F., & Bolger, D. J. (2014). The neurophysiological bases of EEG and EEG measurement: A review for the rest of us. *Psychophysiology*, *51*(11), 1061–1071. https://doi.org/https://doi.org/10.1111/psyp.12283

Kelly, S. P., & O'Connell, R. G. (2013). Internal and external influences on the rate of sensory evidence accumulation in the human brain. *The Journal of Neuroscience*, *33*(50), 19434–19441.

Kelly, S. P., Corbett, E. A., & O'Connell, R. G. (2021). Neurocomputational mechanisms of prior-informed perceptual decision-making in humans. *Nature Human Behaviour*, *5*(4), 467–481. https://doi.org/10.1038/s41562-020-00967-9

Krajbich, I., Armel, C., & Rangel, A. (2010). Visual fixations and the computation and comparison of value in simple choice. *Nature Neuroscience*, *13*(10), 1292–1298. https://doi.org/10.1038/nn.2635

Luck, S. (2014). *An Introduction to the Event-Related Potential Technique, second edition*. MIT Press. https://books.google.de/books?id=SzavAwAAQBAJ

Mulder, M. J., Van Maanen, L., & Forstmann, B. U. (2014). Perceptual decision neurosciences–a model-based review. *Neuroscience*, *277*, 872–884.

Murphy, P. R., Robertson, I. H., Harty, S., & O'Connell, R. G. (2016). Neural evidence accumulation persists after choice to inform metacognitive judgments. *Elife*, *5*, e11946.

Myers, C. E., Interian, A., & Moustafa, A. A. (2022). A practical introduction to using the drift diffusion model of decision-making in cognitive psychology, neuroscience, and health sciences. *Frontiers in Psychology*, *13*. https://doi.org/10.3389/fpsyg.2022.1039172

Ouyang, G., Sommer, W., & Zhou, C. (2015). A toolbox for residue iteration decomposition (RIDE)–A method for the decomposition, reconstruction, and single trial analysis of event related potentials. *Journal of Neuroscience Methods*, *250*, 7–21. https://doi.org/10.1016/j.jneumeth.2014.10.009

Ratcliff, R. (1978). A theory of memory retrieval. *Psychological Review*, *85*(2), 59–108.

Ratcliff, R., & McKoon, G. (2008). The diffusion decision model: theory and data for two-choice decision tasks. *Neural Computation*, *20*(4), 873–922.

Ratcliff, R., Smith, P. L., Brown, S. D., & McKoon, G. (2016). Diffusion Decision Model: Current Issues and History. *Trends in Cognitive Sciences*, *20*(4), 260–281. https://doi.org/https://doi.org/10.1016/j.tics.2016.01.007

Schepers, J., Lips, L., Marathe, M., & Ehinger, B. V. (2025,). *UnfoldSim.jl: Simulating continuous event-based time series data for EEG and beyond*. The Open Journal. https://doi.org/10.21105/joss.06641

Tillman, G., Van Zandt, T., & Logan, G. D. (2020). Sequential sampling models without random between-trial variability: The racing diffusion model of speeded decision making. *Psychonomic Bulletin & Review*, *27*(5), 911–936.

Van Vugt, M. K., Beulen, M. A., & Taatgen, N. A. (2019). Relation between centro-parietal positivity and diffusion model parameters in both perceptual and memory-based decision making. *Brain Research*, *1715*, 1–12. https://doi.org/https://doi.org/10.1016/j.brainres.2019.03.008

Van Vugt, M. K., Simen, P., Nystrom, L. E., Holmes, P., & Cohen, J. D. (2012). EEG oscillations reveal neural correlates of evidence accumulation. *Frontiers in Neuroscience*, *6*, 106.

Wikipedia contributors. (2024,). *Colors of Noise*. https://en.wikipedia.org/wiki/Colors_of_noise

Declaration:

I hereby declare that the work presented in this thesis is entirely my own. I did not use any other sources and references that the listed ones. I have marked all direct or indirect statements from other sources contained therein as quotations. Neither this work nor significant parts of it were part of another examination procedure. I have not published this work in whole or in part before. The electronic copy is consistent with all submitted hard copies.

Date and Signature:

Erklärung:

Ich versichere, diese Arbeit selbstständig verfasst zu haben. Ich habe keine anderen als die angegebenen Quellen benutzt und alle wörtlich oder sinngemäß aus anderen Werken übernommene Aussagen als solche gekennzeichnet. Weder diese Arbeit noch wesentliche Teile daraus waren bisher Gegenstand eines anderen Prüfungsverfahrens. Ich habe diese Arbeit bisher weder teilweise noch vollständig veröffentlicht. Das elektronische Exemplar stimmt mit allen eingereichten Exemplaren überein.

Datum und Unterschrift: