Ludwig-Maximilians-Universität
München
Faculty of Biology, Computational Neuroscience

Graduate School of
Systemic Neurosciences
LMU Munich

Report

Computational Simulation of Time Perception: Model Description and Implementation

Katharina Bracher

Supervision: Dr. Kay Thurley

Contents

1 Behavioral Effects in Magnitude Estimation

Magnitude estimation is subject to noise that arises from external sources i.e. the statistics of the environment and internal sources i.e. neural representation of the input and the behavior. Across sensory modalities, characteristic behavioral effects are identified (Petzschner et al. 2015). The most prominent observation is a regression to the mean of the stimulus range, i.e. small stimuli are overestimated whereas large stimuli are underestimated (*regression effect*). This effect intensifies for ranges with larger stimuli (*range effect*). For larger stimuli the standard deviation of estimates increases monotonically (*scalar variability*). Finally, the recent history of stimuli presentations influences the current stimuli estimation (*sequential effects*). All effects mentioned above are displayed in Fig. 1.

Modality-independence of these effects suggests the existence of a common underlying principle or processing mechanisms, that would explain e.g. an optimal strategy for unreliable judgments due to noise (in stimuli and estimates).

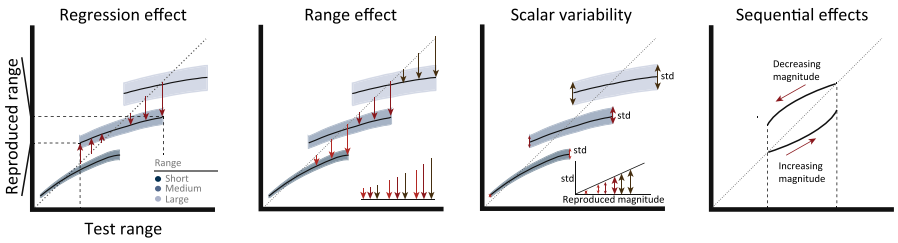

Figure 1: **Behavioral Effects** *Regression effect*: in a range of stimuli large stimuli are underestimated, and small stimuli are overestimated which results in a regression to the mean of the range. *Range effect*: the regression to the mean gets more pronounced for ranges that comprise larger stimuli. *Scalar variability*: the standard deviation of the reproduced magnitude grows linearly with larger rages. *Sequential effects*: the history presented stimuli (e.g. ascending or descending order) has an influence on the reproduced magnitude. Figure from Petzschner et al. 2015.

During time perception and time reproduction experiments, neural activity displays characteristic trajectories in a low-dimensional space (Wang et al. 2018, Henke et al. 2021, Meirhaeghe et al. 2021). The neural trajectories are consistently influenced by prior beliefs. Flexible motor timing can be achieved by controlling the speed of neural dynamics (Sohn et al. 2019, Wang et al. 2018). Further, it has been found that neural activity in anticipation of a delayed response reaches a fixed threshold with a rate inversely proportional to delay

period (Murakami et al. 2014, Mita et al. 2009). Wang et al. 2018 proposed a potential neural mechanism for speed control. Based on that mechanism Egger et al. 2020 developed an extended circuit model for sensorimotor timing.

2 Model Description

2.1 Basic Circuit

Flexible speed control can be achieved by a simple model consisting of three units, u, v, y that represent population activity. The dynamics of u, v, and y are defined as follows:

$$\begin{aligned} \tau \frac{\mathrm{d}u}{\mathrm{d}t} &= -u + \theta(W_{uI} I - W_{uv} v + \eta_u) \ , \\ \tau \frac{\mathrm{d}v}{\mathrm{d}t} &= -v + \theta(W_{vI} I - W_{vu} v + \eta_v) \ , \\ \tau \frac{\mathrm{d}y}{\mathrm{d}t} &= -y + W_{yu} u - W_{yv} v + \eta_y \ . \end{aligned} \tag{1}$$

Two units, u and v, receive a tonic symmetric input I ($W_{uI} = W_{vI} = 6$) and are mutually inhibiting each other ($W_{uv} = W_{vu} = 6$). The inputs to u and v are governed by a sigmoidal activation function $\theta(x) = \frac{1}{1+exp(-x)}$. The output unit y receives excitatory input from u and inhibitory input from v ($W_{yu} = W_{yv} = 1$) which results in a ramp-like activity of y. All three units have a time constant $\tau = 100$ ms. Stochastic synaptic inputs are modeled as independent white noise η_u, η_v, η_y with standard deviation σ. Initial conditions of u, v and y have been optimized for in Egger et al. 2020 and are set to $u_0 = 0.7, v_0 = 0.2, y_0 = 0.5$ (Fig. 2a)

Depending on the input I, the system shows different dynamics. For low levels of I (0<I<0.5) the system has three fixed points (two stable, one unstable at $u = v$) and y ramps up faster the higher input I. For intermediate values of I (0.5<I<1) the system still shows three fixed points of the same sort as in the low input regime but y ramps up with a slope that is inversely proportional to the input I (y ramps up slower the higher input I, see Fig. 2b). For high I (1<I) the system has one stable fixed point (at $u = v$) and y ramps down faster for higher I. Thus, the speed at which the output y evolves can be controlled by the input I (Fig. 2b) and determines the time interval after which y reaches a fixed threshold y_{th}. In interval reproduction experiments, reaching a threshold y_{th} can be understood as movement initiation time which can be controlled for

by adjusting I. In this report, the intermediate input regime is explored: higher inputs result in a shallower slope of y, such that the threshold y_{th} is reached after a longer time interval.

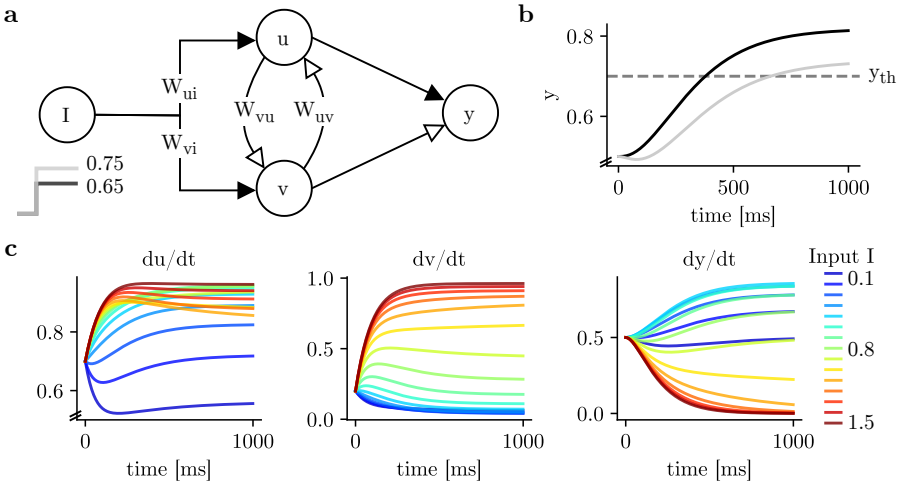

Figure 2: **Basic Circuit and Input Regimes (a)** u and v share a common input I. The input is governed by weights W_{uI} and W_{vI}. The two units have reciprocal inhibitory connections with weights W_{uv} and W_{vu} that determine the inhibitory strength. Both project to the output unit y with an excitatory connection from u and an inhibitory connection from v. Excitatory and inhibitory connections are shown by filled and open arrows, respectively. **(b)** Dynamics of y for intermediate regime with input $I = 0.75$ in gray and $I = 0.65$ black. There is an inverse relation of input strength and slope. With higher input, the threshold at 0.7 (dashed line) is reached after a longer time interval. **(c)** Dynamics of u, v, y for inputs from $0.1 \leq I \leq 1.5$ are shown. Initial conditions are set to $u_0 = 0.7, v_0 = 0.2, y_0 = 0.5$. With these initial conditions and values of $I >= 0.5$, the relation of steady state activity of y (and slope to reach the steady state) is inverse to I (intermediate and high I regime). For values $I <= 1$ the activity of y ramps down (yellow corresponds to $I = 1$). For $I < 0.5$ the steady state (slope) is smaller the smaller I (low I regime, dark blue).

2.2 Update Mechanism and Experiment Procedure

Basic interval reproduction experiments can be designed with only two epochs: a measurement epoch that has the duration of the stimulus interval and a reproduction epoch that starts immediately after the measurement epoch (Fig. 3b). The basic circuit described above is modified to perform interval reproduction (see Figure 3a for schematic of modified circuit). The relation of input I with the slope of the ramping activity of y is used in combination with a fixed threshold y_{th}. By adding an update mechanism that flexibly adjusts I, the threshold

crossing of y can be delayed or moved to earlier times. This way, measuring and reproducing an interval is done predictively, by adjusting the slope of the ramp such that the output reaches the threshold after the intended time. In the model the measurement epoch is fixed to the stimulus interval t_s, and the reproduction epoch ends, when y reaches the fixed threshold y_{th} from below. The time from the end of the measurement epoch until the threshold-crossing of y yields the reproduced time interval t_r that is aimed to equal the stimulus interval t_s (Fig. 3c).

The following update mechanism of I is based on the intermediate input regime, that shows an inverse relation of I to the slope of y (Fig. 2b). The error signal is determined at the end of the measurement epoch and is composed of the difference of y to the threshold y_{th}. If the threshold y_{th} is not reached during the measurement epoch, the slope has to be adjusted, such that y ramps up faster to reach the threshold at exactly the time of the stimulus interval. For a steeper slope, I is reduced. If y crossed the threshold before the measurement epoch ends, so is above y_{th} by the end of the stimulus interval, the slope needs to be reduced in order to reach the threshold at a later time in the reproduction. For a shallower slope, I is increased (Fig. 3c). I is adjusted according to the error $(y - y_{\mathrm{th}})$, weighted by a memory parameter K right at the end of the measurement epoch

$$\tau \frac{\mathrm{d}I}{\mathrm{d}t} = sK(y - y_{\mathrm{th}}) \; . \tag{2}$$

The update of I is only active for a pulse between the measurement and reproduction epoch ($s = 1$) and inactive for all other times ($s = 0$). Further, u and v receive a transient input pulse I_r to reset the dynamics for the subsequent epoch (Fig. 3c)

$$\begin{aligned} \tau \frac{\mathrm{d}u}{\mathrm{d}t} &= -u + \theta(W_{uI} I - W_{uv} v + \eta_u - I_r) \; , \\ \tau \frac{\mathrm{d}v}{\mathrm{d}t} &= -v + \theta(W_{vI} I - W_{vu} v + \eta_v + I_r) \; . \end{aligned} \tag{3}$$

Resetting the dynamics after the reproduction epoch enables the model to simulate an arbitrary number of stimulus intervals. Before a new stimulus presentation there is a delay period: In the beginning of an experiment with a sequence of stimuli there is an initial interval with the initial values of u, v, y and an initial

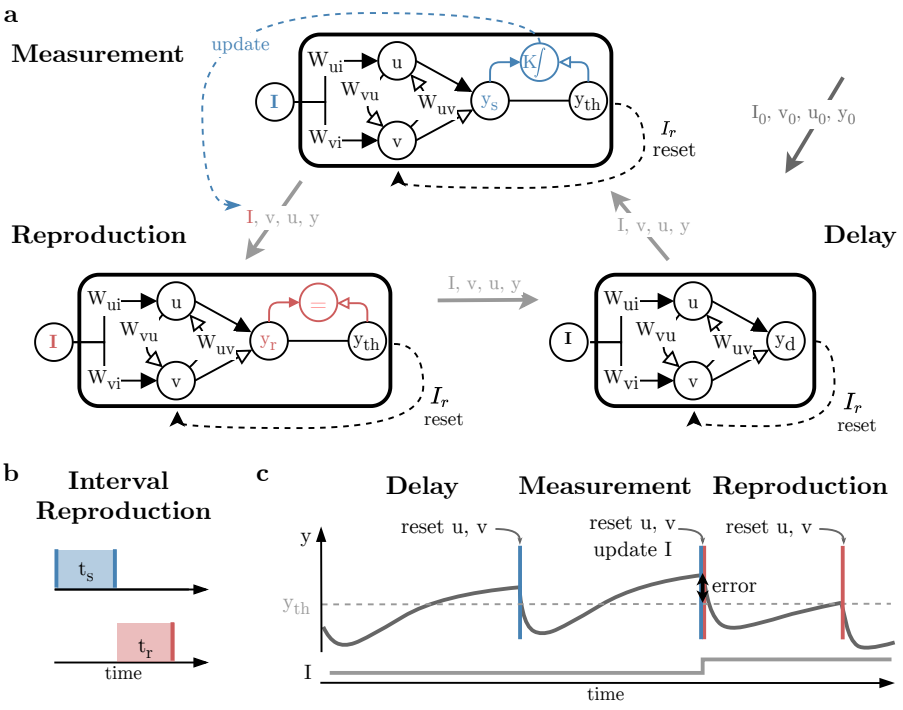

Figure 3: **Extended Circuit for Experiment Simulation (a)** The circuits for measurement, reproduction and delay epoch are displayed. All circuits comprise the same basic structure with different additional elements that are unique for the epoch. Initial values of u, v, y and I are fed into the delay circuit for the duration of the initial interval. u and v are reset with a transient input I_r before end values of u, v, y and I are transferred to the measurement circuit as new initial conditions. After the duration of the stimulus interval the difference between y and the threshold y_{th} is used with the memory parameter K to update I together with another reset of u and v. The values are transferred with all other variables to the reproduction circuit. The reproduction epoch ends when y reaches the threshold y_{th} from below. Before the presentation of another stimulus interval, there is again a delay period with no update of I. The reset mechanism enables the model to simulate an arbitrary number of stimulus intervals. Adapted from Egger et al. 2020. **(b)** Interval reproduction experiment with stimulus interval t_s (blue) and reproduction t_r (red). **(c)** Schematic of one trial. After a delay epoch u and v are reset. The measurement epoch lasts for the duration of the stimulus interval t_s. y should reach the threshold y_{th} (dashed line) at exactly the time the stimulus interval ends. The threshold was crossed well before the end of the interval and at the end of the measurement epoch, the error in y_m to the threshold y_{th} is used to update I. To reach the threshold at a later time, I is increased, which reduces the slope of y in the reproduction epoch. After the reset of u and y and the update of I the reproduction ends when y reaches the threshold. The time after the reset and update until the threshold crossing denotes the reproduced interval t_r.

input I_0 (Fig. 3c). This additional interval in the beginning of the experiment allows the model to get closer to steady-state before the stimulus presentation starts. Reproductions that do not reach the threshold in a certain time span (twice the stimulus interval) are classified as timeout trials. If the threshold is crossed particularly early, the trial is also classified as (early) timeout trial (crossing before 0.2 of stimulus interval).

To simulate variability in the reproductions as it is found in real data, noise is added to u, v and y in the implementation. In data an monotonic increase of the standard deviation of reproductions is found (*scalar variability*). If noise levels in the model are too high ($\sigma > 0.25 * I$) the standard deviation of reproductions is not growing monotonically with σ (Egger et al. 2020). For an example experiment with a sequence of five trials see Figure 4.

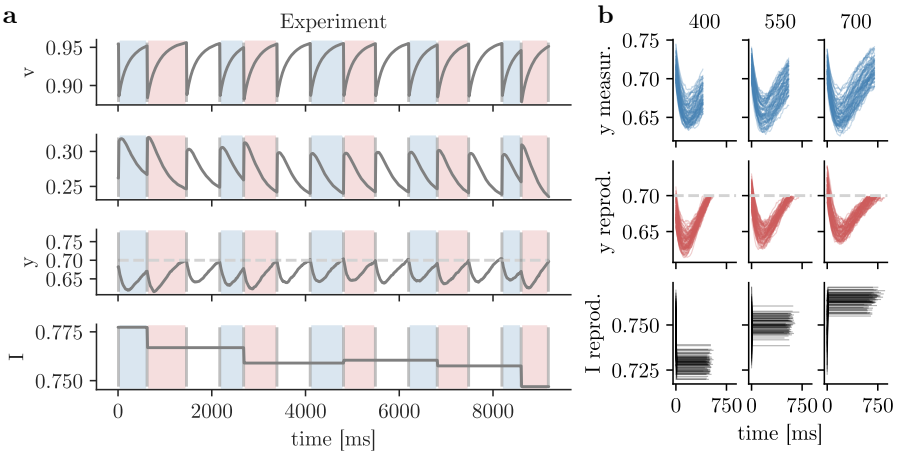

Figure 4: **Experiment Simulation** Example trial with a sequence of five stimulus intervals (600, 700, 400, 500, 700 ms). Initial values are set to $u_0 = 0.7, v_0 = 0.2, y_0 = 0.5. I_0 = 0.8$, other parameters are $\sigma = 0.01$, the initial duration of 750 ms and a 700 ms delay period before new stimuli. The dynamics of u, v, y and I are displayed over time. The threshold is set to $y_{\text{th}} = 0.7$ (dashed line). The first stimulus interval is reproduced poorly as the system needs more trials to adapt the initial I_0 to a suitable range (reproduced times 820, 780, 490, 450, 570 ms).

2.3 Optimal Update Parameter

A crucial parameter is the weight with which the input to the circuit is adjusted to balance the mismatch between the predicted interval and the stimulus du-

ration. Adjusting this weight in accordance with error minimization, leads to putting more weight on the prior experience for longer stimuli, which naturally entail more uncertainty, and is thus biological plausible.

3 Supplements

3.1 Methods

Circuit

For simulating the dynamics of u, v, y and I Euler's method was used with a step size Δt set to 10 ms. The reset mechanism of u and v is switched on for one time step after every epoch and the update mechanism of I is switched on for one time step after the measurement epoch only. Noise is independently sampled for every time step from a Gaussian distribution with standard deviation σ.

Experiment Simulation

Optimality by Minimizing MSE

We minimized the error in the interval reproductions across the experiment, we defined the MSE = bias^2 + var, with the bias specified as the squared difference between the mean reproduction $\bar{t_r}$ for each stimulus interval t_s in the stimulus range S and variance as mean σ^2 for all stimuli intervals in S.

$$\begin{aligned} \text{bias} &= \frac{1}{S}\sum_{i=1}^{S}(\bar{t_{r_i}} - t_{s_i}) \ , \\ \text{bias}^2 &= \frac{1}{S}\sum_{i=1}^{S}(\bar{t_{r_i}} - t_{s_i})^2 \ , \\ \text{var} &= \frac{1}{S}\sum_{i=1}^{S}(\sigma_i^2) \ . \end{aligned} \tag{4}$$

3.2 Intermediate and High Input Regime

In the intermediate input regime higher I result in a shallower slope of y.

3.3 Design

The code is designed in a modular way, such that multiple types of experimental procedures with shared functionality are accessing the same basic circuit, which is implemented in `BaseSimulation` as shown in Figure 5. The implementation of the basic circuit can be reused for all epochs. Different experiments can have different result types, all of which can be found in `result.py`. After each experiment, the results are gathered and stored together, consisting of the parameter set, the simulation time course, a list of reset time points, a list of production times, a list of indices of timeout trials and the stimulus list. Analysis of the results is performed in the same file. Depending on the analysis (e.g. behavioral), different plots are implemented in `plot.py` to visualize the results.

All parameters are set and described in `Params` and can be modified individually or by reading a parameter dictionary. The parameters configure the circuit. To initiate a simulation, the parameter set is handed to one of the implemented experiments. An interval reproduction experiment, as described in this report is implemented in `experiment_simulation.py`. Parallel simulations of one trial (one delay, measurement and reproduction epoch) is implemented in `parallel_simulation.py`. Both `experiment_simulation.py` and `parallel_simulation.py` contain a simulate function that accesses the base simulation (`BaseSimulation`) with the implementation of the basic circuit. For each epoch or update/reset step, the simulate function feeds the according time steps and initial conditions into the network in `BaseSimulation`. Depending on the epoch, the reset and update mechanism are tuned on or off. After each epoch or pulse, the results of the network are joint to the time course of the experiment in trial_update. The simulate function returns a `SimulationResult` or `RangeParallelSimulationResult` object.

For an overview of the code structure see Figure 5 and for its usage see simulations.ipynb at github.com/KatharinaBracher/MScThesis.

3.4 Implementation

All simulations and analysis were performed with Python 3.9.7. The following libraries were used: matplotlib (3.5.1), NumPy (1.22.3), SciPy (1.8.0), scikit-learn (1.0.2). An experiment simulation can not be parallelized, since each step depends on the previous one.

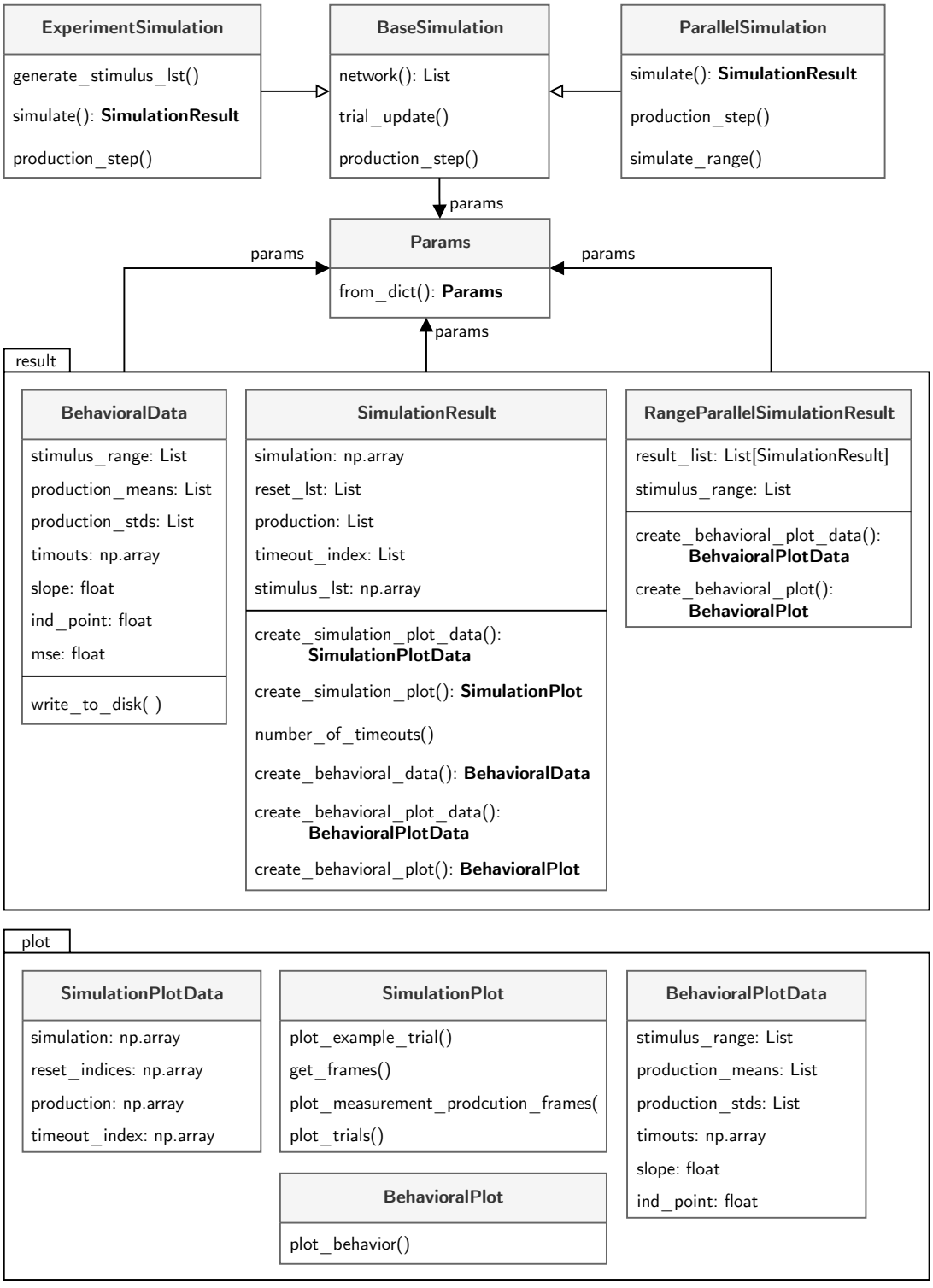

Figure 5: **Design** The base simulation and different procedures (experiment, parallel) are all implemented in separate files. All results and analysis are collected in `result.py`, all plot for different result types are collected in `plot.py`

References

Egger, Seth W., Nhat M. Le, and Mehrdad Jazayeri (2020). "A neural circuit model for human sensorimotor timing". *Nature Communications* 11.1, pp. 1–14. ISSN: 20411723. DOI: 10.1038/s41467-020-16999-8.

Henke, Josephine, David Bunk, Dina von Werder, Stefan Häusler, Virginia L. Flanagin, and Kay Thurley (2021). "Distributed coding of duration in rodent prefrontal cortex during time reproduction". *eLife* 10, pp. 1–24. ISSN: 2050084X. DOI: 10.7554/eLife.71612.

Meirhaeghe, Nicolas, Hansem Sohn, and Mehrdad Jazayeri (2021). "A precise and adaptive neural mechanism for predictive temporal processing in the frontal cortex". *Neuron* 109.18, 2995–3011.e5. ISSN: 10974199. DOI: 10.1016/j.neuron.2021.08.025.

Mita, Akihisa, Hajime Mushiake, Keisetsu Shima, Yoshiya Matsuzaka, and Jun Tanji (2009). "Interval time coding by neurons in the presupplementary and supplementary motor areas". *Nature Neuroscience* 12.4, pp. 502–507. ISSN: 10976256. DOI: 10.1038/nn.2272.

Murakami, Masayoshi, M. Inês Vicente, Gil M. Costa, and Zachary F. Mainen (2014). "Neural antecedents of self-initiated actions in secondary motor cortex". *Nature Neuroscience* 17.11, pp. 1574–1582. ISSN: 15461726. DOI: 10.1038/nn.3826.

Petzschner, Frederike H., Stefan Glasauer, and Klaas E. Stephan (2015). "A Bayesian perspective on magnitude estimation". *Trends in Cognitive Sciences* 19.5, pp. 285–293. ISSN: 1879307X. DOI: 10.1016/j.tics.2015.03.002.

Sohn, Hansem, Devika Narain, Nicolas Meirhaeghe, and Mehrdad Jazayeri (2019). "Bayesian Computation through Cortical Latent Dynamics". *Neuron* 103.5, 934–947.e5. ISSN: 10974199. DOI: 10.1016/j.neuron.2019.06.012.

Wang, Jing, Devika Narain, Eghbal A. Hosseini, and Mehrdad Jazayeri (2018). "Flexible timing by temporal scaling of cortical responses". *Nature Neuroscience* 21.1, pp. 102–112. ISSN: 15461726. DOI: 10.1038/s41593-017-0028-6.