CAL Simulation Guides ISSN 2054-2747

CENTRE FOR COMPUTATIONAL AND ANIMAL LEARNING RESEARCH

CAL Simulation Guides

SOP MODEL SIMULATOR

V. 1 (2017)

John Byers, Esther Mondragón,
and Eduardo Alonso

CAL Simulation Guides

ISSN 2054-2747

SOP model Simulator © Version 1 2017

John Byers, Esther Mondragón & Eduardo Alonso

CENTRE FOR COMPUTATIONAL AND ANIMAL LEARNING RESEARCH

St. Albans, United Kingdom

TABLE OF CONTENTS

1 INSTALLATION

The SOP Simulator v. 1.0 implements Wagner's (1981) SOP Model. It has been built as a Matlab standalone application which means that it runs in a Windows platform without using a Matlab license, i.e., you don't need to have Matlab installed in your machine. Microsoft Excel (version 2007 or later) is however required to write out the experiment results and it is not included in the package.

The SOP Simulator v. 1.0 installer file 'SOP_Installer_mcr.exe' can be downloaded from https://www.cal-r.org/index.php?id=SOP-sim. After download is completed, double click the installer. The first window you'll see is informative. On the second window you will be prompted to change the installation folder, we ***strongly suggest you do so*****.** Please browse to your desired installation folder; make sure that administration rights are not required to write on it because simulation outputs and temporal files will be stored in the folder that contains the application (or its shortcut, if used to run it) by default. We recommend using the following path:

«YOUR-USER-PROFILE»\ Documents\SOP_Simulator [1]

If you wish you can also tick the option "Add a shortcut to the desktop" for easy access.

ATTENTION

WARNING: If you choose not to change the default installation folder, the application will be installed in a hidden folder, in the following path C:\Users\%USERPROFILE%\AppData\Roaming\SOP_Simulator.

Once installed, the SOP_Simulator folder will include the following sub-folders: appdata, application, sys, and uninstall, with the simulator executable file 'SOP_Simulator.exe' located in the application folder. A further folder named 'MATLAB Runtime' will be installed to your 'Program Files' directory[2] (/MATLAB/MATLAB Runtime, if you have Matlab already installed).

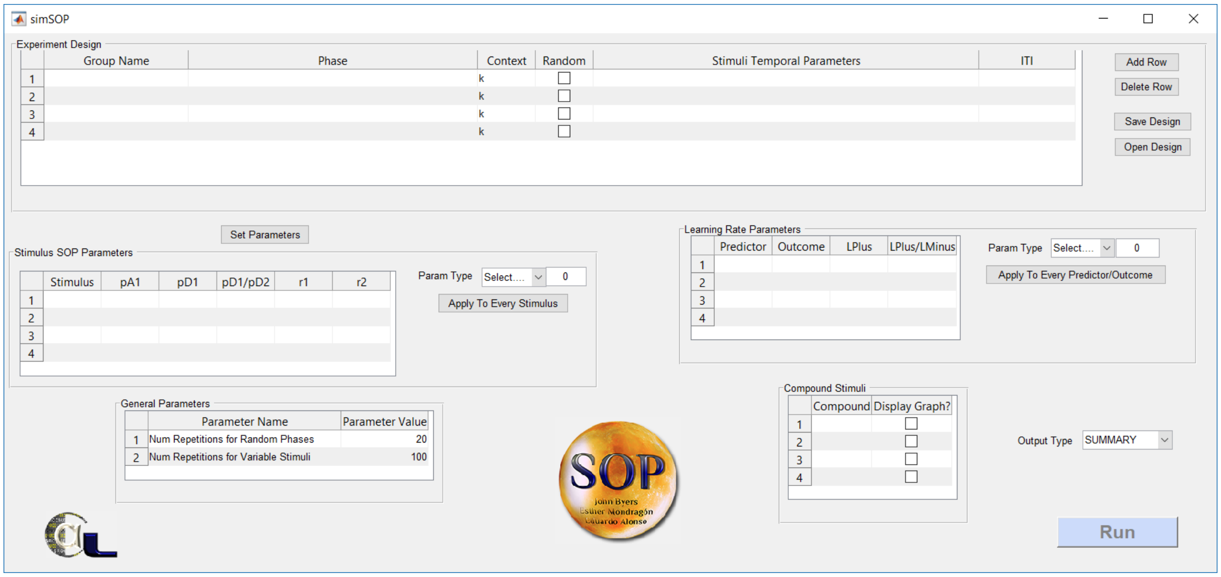
ATTENTION

WARNING: The installation includes a large Matlab set of files and it would take some time to complete. Please be patient.

Double click on ‘SOP_Simulator.exe’ to open the application. This brings up the simulator GUI as shown below (Figure 1).

ATTENTION

WARNING: There might be a delay in displaying the simulator GUI after the splash screen closes. Please, be patient!



Experiment Design

	Group Name	Phase	Context	Random	Stimuli Temporal Parameters	ITI
1			k	□		
2			k	□		
3			k	□		
4			k	□		

Stimulus SOP Parameters

	Stimulus	pA1	pD1	pD1/pD2	r1	r2
1						
2						
3						
4						

Param Type Select.... 0

Learning Rate Parameters

	Predictor	Outcome	LPlus	LPlus/LMinus
1				
2				
3				
4				

Param Type Select.... 0

General Parameters

	Parameter Name	Parameter Value
1	Num Repetitions for Random Phases	20
2	Num Repetitions for Variable Stimuli	100

Compound Stimuli

	Compound	Display Graph?
1		□
2		□
3		□
4		□

Output Type SUMMARY

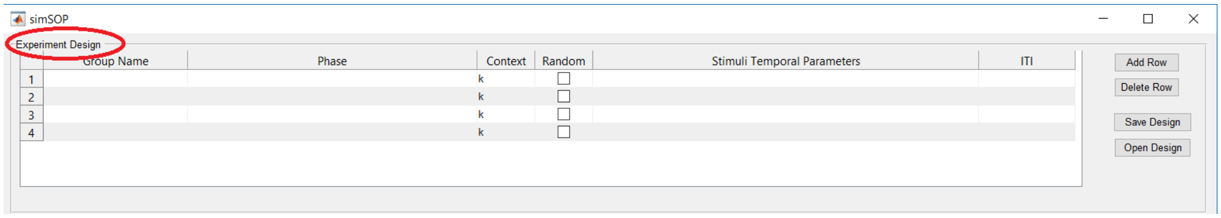
Figure 1*:* SOP Simulator GUI

In some computers you may need to override high DPI scaling to get a clearer image. If so, you need to right-click on the ‘SOP_Simulator.exe’ file, go to ‘Properties’ at the bottom of the window, select the ‘Compatibility’ tab and click “Override high DPI scaling behavior. Scaling performed by ‘Application’”. If at opening you see that the splash screen moves from the center to the left-hand side of your monitor, we recommend you change the scaling as just described.

2 ENTERING AND RUNNING EXPERIMENTS

2.1 EXPERIMENTAL DESIGN TABLE

The Experimental Design table (Figure 2) is the topmost table of the GUI window where experimental designs are entered. This table allows to input universal experimental designs as implemented in the RW model Simulator © V.4 (Mondragón, Alonso & Fernández, 2012; Alonso, Mondragón & Fernández, 2012). The 'add row' and 'delete row' buttons to its right apply to this table. 'Add row' adds rows to the end of the table; 'delete row' deletes the row related to the currently selected table cell.



	Group Name	Phase	Context	Random	Stimuli Temporal Parameters	ITI
1			k	☐		
2			k	☐		
3			k	☐		
4			k	☐		

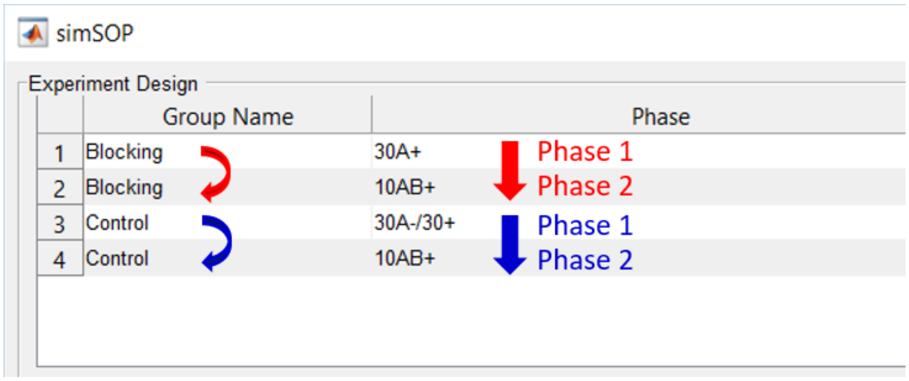
Figure 2: Experimental Design table

Blank table rows are permitted at the end of the table and are just ignored. Blank rows elsewhere are not permitted and will result in errors.

2.1.1 Group Name/Phase

To insert a design, you will need first to set a Group Name. This is just a string and hence any name can be entered. Adding a new group is simple, just enter a different Group Name value in a new row.

A Group can consist of multiple phases. Values from an earlier phase are maintained through subsequent ones. All phases for a group **must be entered as consecutive rows** in the table (see Figure 3) with the row order dictating the phase order.



simSOP

Experiment Design

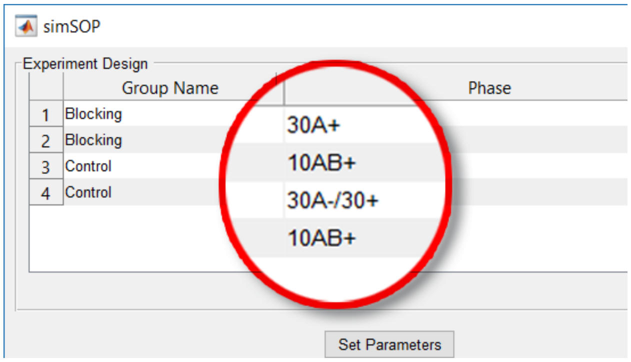
	Group Name	Phase
1	Blocking	30A+
2	Blocking	10AB+
3	Control	30A-/30+
4	Control	10AB+

Figure 3: Group and Phase input

Phase strings (Figure 4) must be entered as a sequence of trial types with each trial type separated by the forward slash character '/'. A conditioned stimulus (CS) must use the English capital letters i.e. in the range A-Z. Each trial type must end with either a '+' character indicating that an Unconditioned Stimulus (US) is presented or a '-' character indicating that there is no US.

The format for a trial type is '<n><cs_list><reinforcement_indicator>' where <n> is the number of trials of that type to be carried out, <cs_list> is a list of CS without any characters between each one and <reinforcement_indicator> is either '+' or '-', as explained.

If <reinforcement_indicator> is '+' and the US is required for a trial type without any CS then <cs_list> should be omitted e.g. 30+.



simSOP

Experiment Design

	Group Name	Phase
1	Blocking	30A+
2	Blocking	10AB+
3	Control	30A-/30+
4	Control	10AB+

Figure 4: Design input

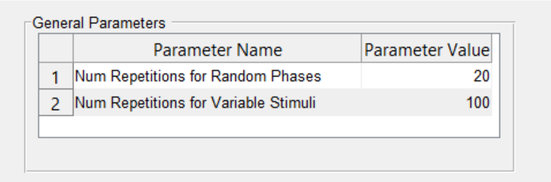
2.1.2 Context

The context stimulus in which training per group and phase occurs is entered in the 'Context' column (see Figure 1) as a single English lowercase letter, i.e. one in the range a-z. This is populated by default with the 'k' context.

2.1.3 Random

If the 'Random' checkbox at each group/phase row is left unselected the trials within the phase are carried out as per the trial type order supplied in the design. For instance, if '10A+/2A-/3B+' is input, then a sequence of ten consecutive A+ trials will occur first, followed by two A-, and, finally, by a sequence of three B+ trials.

If the checkbox is selected, this means the sequence of trials of all types within the phase is randomised. For example, with random selected for 20A+/30AB-, the 50 trials are carried out in random order so that, for example, the first trial may be of type AB- and the next of type A+. Additionally, if the random checkbox is selected, the value set in the 'General Parameters' table for parameter 'Num Repetitions for Random Phases' will indicate the number of different sequences that will be run repeatedly and averaged to give a final value per trial and trial type. This value is initialised at 20 by default (see Figure 5).



General Parameters

	Parameter Name	Parameter Value
1	Num Repetitions for Random Phases	20
2	Num Repetitions for Variable Stimuli	100

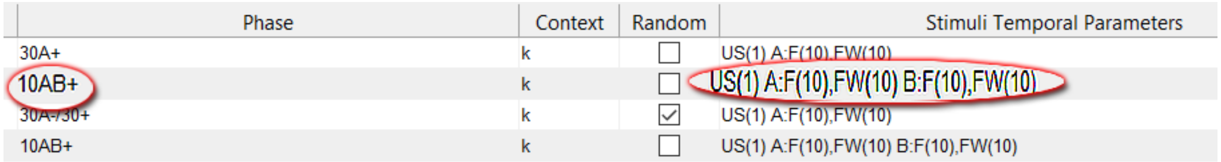
Figure 5: General Parameters table

ATTENTION

WARNING: System memory is used up at a rate of about 10-15 MB/second which means that 100 random repetitions may take a while to run and cause system memory to run out. Closing the simulator does not release this memory – only restarting Matlab clears it.

2.1.4 Stimuli Temporal Parameters

Stimulus durations and order of CS-US presentation (forward, FW or Backward, BW) must be entered in the 'Stimulus Temporal Parameters' column. If any of the trial types contains a US, ('+') then its duration must be entered first as follows 'US(duration)'. Subsequent stimulus information needs to be **separated by a single space** character ' '. The CS temporal parameters should be prefixed by the stimulus name and a colon ':' followed by 'F' or 'V' (for fixed and variable durations, respectively) and the corresponding duration in parentheses (please see below for more detailed information about the stimulus length). Next, a comma ',' should be used to separate the succeeding parameters. FW or BW would indicate whether the CS precedes or follows the US and a digit in parentheses would indicate the inter-stimulus interval (from onset to onset). For instance, if the design contains AB+ trials with both A and B simultaneously presented, lasting 10s each, and the US is presented at the stimuli offset, then the corresponding Stimulus Temporal Parameters (please see Figure 6, below) should be input as follows: **US(1) A:F(10),FW(10) B:F(10),FW(10)**.



Phase	Context	Random	Stimuli Temporal Parameters
30A+	k	□	US(1) A:F(10),FW(10)
10AB+	k	□	US(1) A:F(10),FW(10) B:F(10),FW(10)
30A-/30+	k	☑	US(1) A:F(10),FW(10)
10AB+	k	□	US(1) A:F(10),FW(10) B:F(10),FW(10)

Figure 6: Stimuli Temporal Parameters

ATTENTION

WARNING: Please notice that ***a space is used to separate each stimulus*** and that there must be ***no spaces within a stimulus*** description; that is, in the string specifying the stimulus temporal data, ***commas*** are used instead. CSs are followed by a colon, USs are not.

Stimulus Length: this is mandatory for all stimuli. For CS it is specified using the syntax 'F(<n>)' for fixed-duration or 'V(<n>,<stdev>)' for variable-duration. <n> represents the number of timesteps the stimulus is present for. For a variable-duration stimulus this is only a mean average duration: <stdev> controls the variance according to a Normal (Gaussian) distribution with <stdev> as the standard deviation.

If any stimulus listed for the phase is of variable-length duration, the trials will be subject to the number of repetitions in the General Parameters table for parameter 'Num Repetitions for Variable Stimuli' (Figure 5). For each such stimulus, a single repetition uses the same duration within each trial. Hence the variation occurs over the repetitions rather than individual trials.

Duration for the US is always a fixed length and is the only temporal parameter. Hence the syntax is '(<n>)' where <n> is the number of timesteps it is present for.

ATTENTION

WARNING: The Variable length stimulus functionality has not been extensively tested for this version. As repetitions are used, the same caveat regarding Matlab's memory problem also applies here.

Forward or Backward stimulus presentations: this information must be input for all CSs, but not for the US. It is mandatory even if there is no US because it is used as an 'anchor' to define the relative start timesteps for the other CS involved. Forward or backward presentations lengths are always defined from the CS onset to the US onset. Forward relations are specified with the syntax ‘FW(<n>)’ and backwards with ‘BW(<n>)’ where <n> denotes number of timesteps difference.

For example, US(1) A:F(5),**FW**(5) B:F(4),**BW**(4) will result in a 13 timesteps trial with the stimuli being present (filled) or not present (empty) according to the diagram in Figure 7 below:

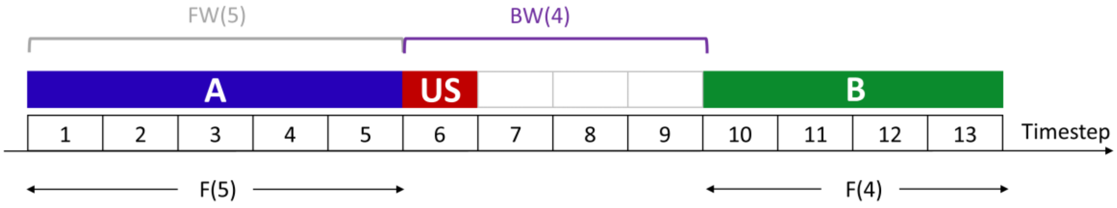


Figure 7: Stimulus distribution in a US(1) A:F(5),FW(5) B:F(4),BW(4) trial

2.1.5 ITI

This specifies the ‘Inter-Trial Interval’ in the format ‘<n>,<stdev>’ where <n> is the interval duration. If <stdev> is zero, the ITI is always the same length <n>; otherwise <n> is the mean duration and the ITI varies according to a Normal (Gaussian) distribution with standard deviation <stdev>. That is, when setting the ITI you must enter a string in the form of ***numeral comma numeral***, e.g., 100,0.

There are no phase repetitions involved here – i.e. the ITI varies per trial rather than per phase repetition (i.e. unlike how variable-length stimuli are handled).

The ITI is also applied as an ‘inter-phase interval’ when switching from one phase to the next within a group.

The ITI is not applied after the final trial of the final phase.

2.1.6 Saving and opening designs

At any time, an experiment can be saved to a file, by clicking the ‘Save Design’ button. The saved file will have the extension ‘.sop’. Once a file is saved it can be retrieved pressing the ‘Open Design’ button (Figure 8).

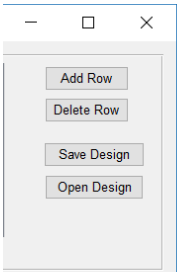
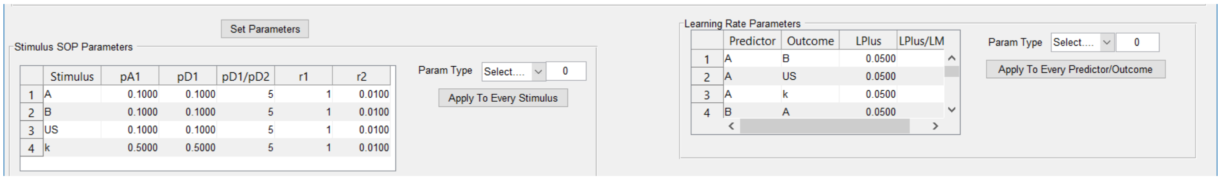


Figure 8: Design operation buttons

2.2 ENTERING SOP PARAMETERS FOR AN EXPERIMENT

After entering the experiment design in the topmost table, the parameters need to be specified. These parameters are dependent on the distinct stimuli involved in the experiment and are specified in two tables: ‘Stimulus SOP Parameters’ and ‘Learning Rate Parameters’ (Figure 9).

To enter the SOP parameters, click on the ‘Set Parameters’ button. This will clear out any data in the 2 parameter tables and automatically populate them with the relevant stimuli as per the experiment design. It also validates the experiment design and will report any syntax errors detected.



Stimulus SOP Parameters

	Stimulus	pA1	pD1	pD1/pD2	r1	r2
1	A	0.1000	0.1000	5	1	0.0100
2	B	0.1000	0.1000	5	1	0.0100
3	US	0.1000	0.1000	5	1	0.0100
4	k	0.5000	0.5000	5	1	0.0100

Param Type Select.... 0

Learning Rate Parameters

	Predictor	Outcome	LPlus	LPlus/LM
1	A	B	0.0500	
2	A	US	0.0500	
3	A	k	0.0500	
4	B	A	0.0500	

Param Type Select.... 0

Figure 9: Parameters settings.

Parameters are initialized to a set of default values, but, of course, you can modify them.

There are buttons to the right of each parameter table that allow for a single value to be applied to all rows for a specified parameter. Alternatively, the tables can be updated directly.

2.2.1 Stimulus SOP Parameters table

Each row is identified by a single stimulus (the first column of the table). The other columns are used to specify the following SOP parameters:

pA1: for timestep t, this specifies the proportion of elements inactive at timestep t-1 that are promoted to active state A1. This only applies for timesteps t when the stimulus is present – at other times, no inactive elements are promoted to A1.

pD1: for timestep t, this specifies the proportion of elements in active state A1 at timestep t-1 that decay to active state A2. This applies whether or not the stimulus is present at timestep t.

pD1/pD2: for timestep t, this indirectly specifies pD2 - the proportion of elements in active state A2 at timestep t-1 that decay to the inactive state. This applies whether or not the stimulus is present at timestep t. The value entered here is the ratio of pD1/pD2 rather than pD2 itself: Wagner (1981) proposes 5 as a ratio and hence the table defaults to this value in the column.

r1: for timestep t, this specifies the weight applied to the proportion of elements in active state A1 at timestep t-1 as part of the 'Retrieval Rule'. Wagner (1981) proposes 1 as a weight and hence the table defaults to this value in the column.

r2: for timestep t, this specifies the weight applied to the proportion of elements in active state A2 at timestep t-1 as part of the 'Retrieval Rule'. Wagner (1981) proposes 0.01 as a weight and hence the table defaults to this value in the column.

2.2.2 Learning Rate Parameters table

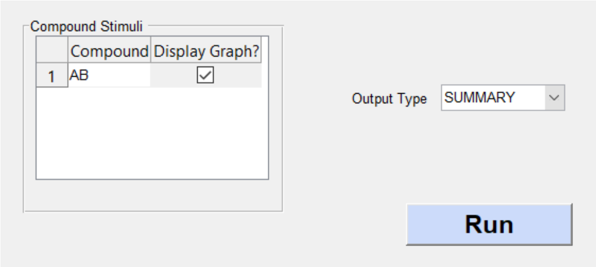
Each row is identified by a combination of two stimuli (the first two columns of the table). The first column represents the predictor stimulus and the second column the outcome stimulus. The other columns are used to specify the following SOP parameters:

LPlus: this specifies the learning rate applied at timestep t to the 'change potential in excitatory associative strength' measured at timestep t-1 between predictor stimulus A and outcome stimulus B. The change potential in excitatory associative strength is the product of the proportion of predictor stimulus elements in active state A1 and the proportion of outcome stimulus elements in active state A1.

LPlus/LMinus: this indirectly specifies the learning rate applied at timestep t to the 'change potential in inhibitory associative strength' measured at timestep t-1 between predictor stimulus A and outcome stimulus B. The 'change potential in inhibitory associative strength' is the product of the proportion of predictor stimulus elements in active state A1 and the proportion of outcome stimulus elements in active state A2. This learning rate is LMinus. Wagner (1981), however, states that the ratio of LPLus/LMinus should be around 5 hence, the value entered in the simulator corresponds to this ratio (LMinus can thus be derived from it). By default the value is set to 5 following Wagner's suggestion.

2.2.3 Compound Stimuli table

The 'Compound Stimuli' table is automatically populated with any compound stimuli in the experiment design after the 'Set Parameters' button is pressed. If running with 'Output Type' of 'SUMMARY' (see Figure 10), any selected compound will be included in the graphical output as predictors for the US.



	Compound	Display Graph?
1	AB	☑

Output Type: SUMMARY

Figure 10: Compound Stimuli table and Output Type

2.3 Running a SOP Experiment

Once the design and parameters have been entered, the experiment is run by pressing the 'Run' button. This firstly runs the experiment and then outputs the results to an Excel file. After the experiment has been run but before the results are output, a 'save as' dialog prompts for the results filename.

While the experiment is running (i.e. before the 'save as' dialog), progress is reported via the Matlab Command Window. Progress specifies completed phases and completed repetitions of phases where applicable.

When all results have been written, an "experiment completed" message dialog will appear as notification, please wait for it.

3 Results output

3.1 Standard Output

The Excel results file consists of multiple worksheets, as shown in Figure 11. There is a sheet called 'Runtime Details' which presents the experiment design and SOP parameters entered from the screen. This sheet also assigns group and phase identifiers automatically.

Apart from 'Runtime Details', the results file consists of sheets with the following name format:

gp<g>ph<ph>-<Pred>(p)-<Out>(o)-<Vtype>

where

<g> is the Group identifier (see Runtime Details sheet)

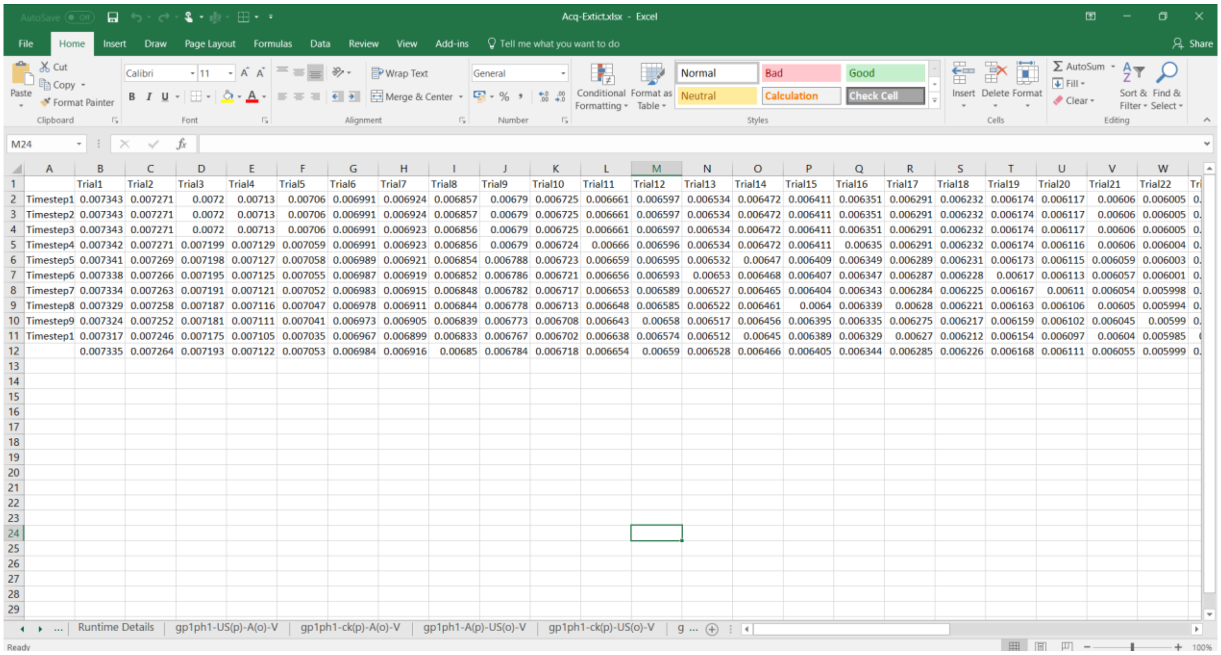
<ph> is the Phase identifier for the group (see Runtime Details sheet)

<Pred> is the name of the predictor stimulus, e.g., A

<Out> is the name of the outcome stimulus, e.g., B

<Vtype> is the associative strength ‘V’

There is a sheet for every possible combination of predictor and outcome stimuli. Context stimuli are represented with a ‘c’ prefix (e.g. ‘ca’) to avoid problems with Excel case-insensitivity (Matlab is case-sensitive).



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	
1		Trial1	Trial2	Trial3	Trial4	Trial5	Trial6	Trial7	Trial8	Trial9	Trial10	Trial11	Trial12	Trial13	Trial14	Trial15	Trial16	Trial17	Trial18	Trial19	Trial20	Trial21	Trial22	Tri
2	Timestep1	0.007343	0.007271	0.0072	0.00713	0.00706	0.006991	0.006924	0.006857	0.00679	0.006725	0.006661	0.006597	0.006534	0.006472	0.006411	0.006351	0.006291	0.006232	0.006174	0.006117	0.00606	0.006005	0.
3	Timestep2	0.007343	0.007271	0.0072	0.00713	0.00706	0.006991	0.006924	0.006857	0.00679	0.006725	0.006661	0.006597	0.006534	0.006472	0.006411	0.006351	0.006291	0.006232	0.006174	0.006117	0.00606	0.006005	0.
4	Timestep3	0.007343	0.007271	0.0072	0.00713	0.00706	0.006991	0.006923	0.006856	0.00679	0.006725	0.006661	0.006597	0.006534	0.006472	0.006411	0.006351	0.006291	0.006232	0.006174	0.006117	0.00606	0.006005	0.
5	Timestep4	0.007342	0.007271	0.007199	0.007129	0.007059	0.006991	0.006923	0.006856	0.00679	0.006724	0.00666	0.006596	0.006534	0.006472	0.006411	0.00635	0.006291	0.006232	0.006174	0.006116	0.00606	0.006004	0.
6	Timestep5	0.007341	0.007269	0.007198	0.007127	0.007058	0.006989	0.006921	0.006854	0.006788	0.006723	0.006659	0.006595	0.006532	0.00647	0.006409	0.006349	0.006289	0.006231	0.006173	0.006115	0.006059	0.006003	0.
7	Timestep6	0.007338	0.007266	0.007195	0.007125	0.007055	0.006987	0.006919	0.006852	0.006786	0.006721	0.006656	0.006593	0.00653	0.006468	0.006407	0.006347	0.006287	0.006228	0.00617	0.006113	0.006057	0.006001	0.
8	Timestep7	0.007334	0.007263	0.007191	0.007121	0.007052	0.006983	0.006915	0.006848	0.006782	0.006717	0.006653	0.006589	0.006527	0.006465	0.006404	0.006343	0.006284	0.006225	0.006167	0.00611	0.006054	0.005998	0.
9	Timestep8	0.007329	0.007258	0.007187	0.007116	0.007047	0.006978	0.006911	0.006844	0.006778	0.006713	0.006648	0.006585	0.006522	0.006461	0.0064	0.006339	0.00628	0.006221	0.006163	0.006106	0.00605	0.005994	0.
10	Timestep9	0.007324	0.007252	0.007181	0.007111	0.007041	0.006973	0.006905	0.006839	0.006773	0.006708	0.006643	0.00658	0.006517	0.006456	0.006395	0.006335	0.006275	0.006217	0.006159	0.006102	0.006045	0.00599	0.
11	Timestep1	0.007317	0.007246	0.007175	0.007105	0.007035	0.006967	0.006899	0.006833	0.006767	0.006702	0.006638	0.006574	0.006512	0.00645	0.006389	0.006329	0.00627	0.006212	0.006154	0.006097	0.00604	0.005985	0
12		0.007335	0.007264	0.007193	0.007122	0.007053	0.006984	0.006916	0.00685	0.006784	0.006718	0.006654	0.00659	0.006528	0.006466	0.006405	0.006344	0.006285	0.006226	0.006168	0.006111	0.006055	0.005999	0.
13																								
14																								
15																								
16																								
17																								
18																								
19																								
20																								
21																								
22																								
23																								
24																								
25																								
26																								
27																								
28																								
29																								

Figure 11: Spreadsheets output

The contents of each such sheet is a matrix of V values. Rows are timesteps and columns are Trial Numbers. Timesteps here only label those when the predictor stimulus is active within the trial starting from 1 and omit those when the predictor is inactive. Hence for fixed-length durations the number of timesteps reported is the same as the stimulus duration specified in the stimuli temporal parameters field.

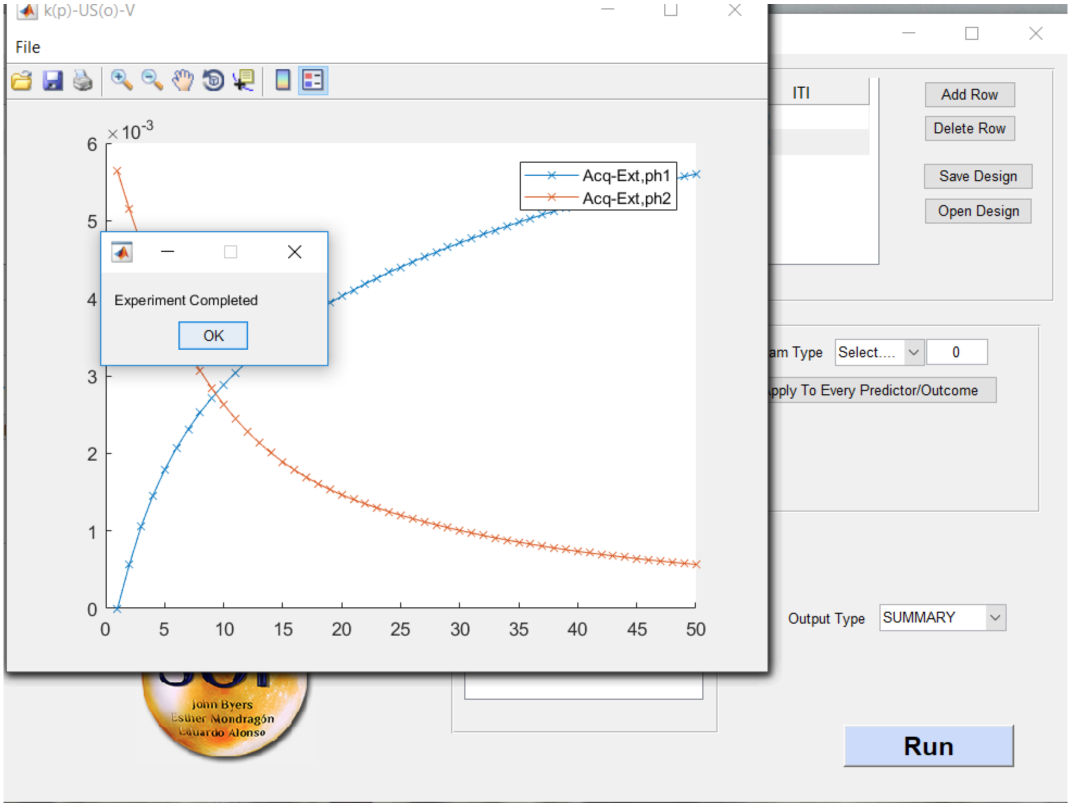
Trial Numbers also only label those where the predictor stimulus is active for the phase repetition. The sequential label ordering corresponds to the order of trials irrespective of whether that ordering is random or not.

Where a phase is set to ‘random’ and hence is repeated several times with randomised trials, the relevant V-value is computed for each repetition. The result reported is the arithmetic mean for that trial number and timestep across all the repetitions. This averaging is also applied for repetitions due to the inclusion of variable-length stimuli.

The final row in every sheet is the average for each trial across the timesteps.

3.1.1 Figure display

The default output ‘SUMMARY’ also provides a Figure of the V-values for each worksheet. These will automatically appear on your screen once the simulation is run (see Figure 12). The final row in the sheet provides the data for the plot – x axis is trial number and y-axis the V-value.



Experiment Completed

Output Type SUMMARY

Figure 12: Figures generated following the simulation

3.2 VERBOSE OUTPUT

If the standard results file output is insufficiently detailed, there is an Output Type dropdown (Figure 13) which can be changed before 'Run Experiment' is pressed. There is no need to re-enter the experiment design or parameters again.

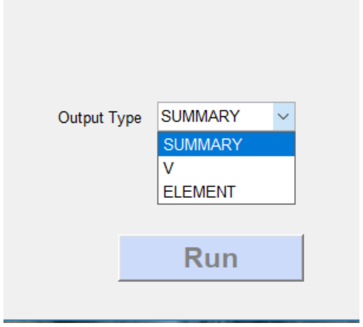
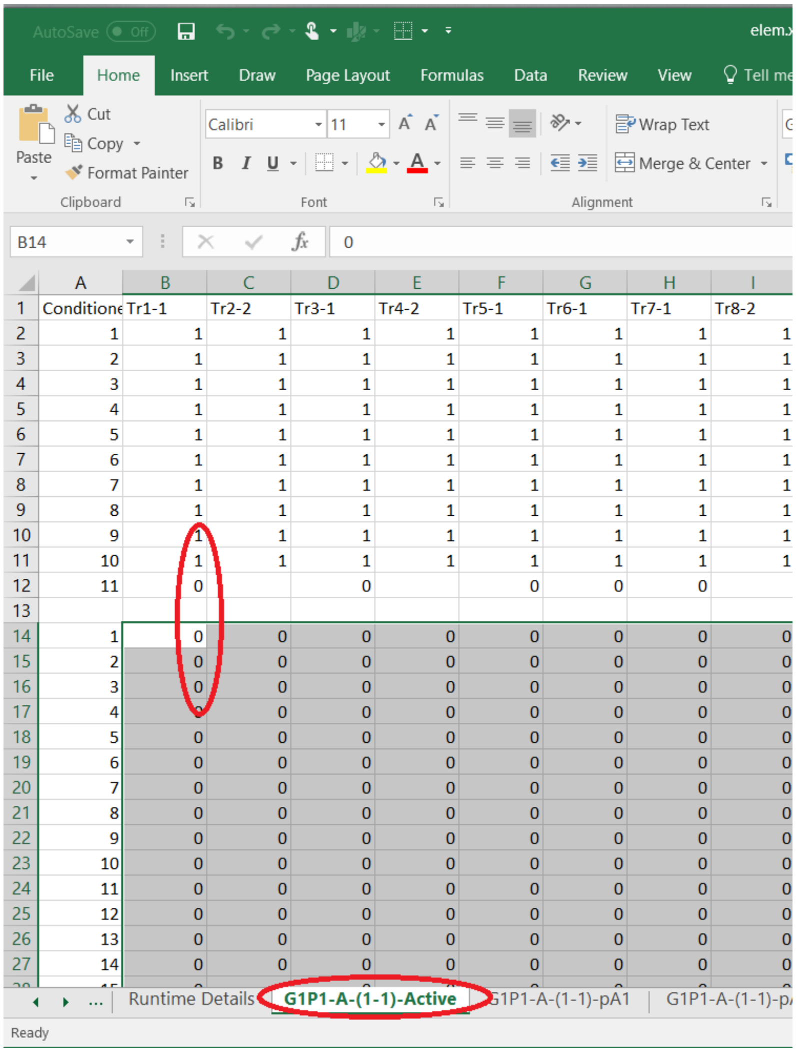


Figure 13: Output Type dropdown

By default, the Output Type is set to SUMMARY. However other legal values are:

- ELEMENT: this will output all propA1, propA2 and propInact values computed for all groups, phases and stimuli. These outputs are NOT limited to when the stimulus is active – hence there are as many columns as there are trials in the phase and as many rows as there are timesteps in the trials. To help determine when stimuli are active, there are also sheets with the same row and column labels that specify at which combinations of trial and timestep the stimulus is active (Figure 14). Repetitions are not averaged but reported in separate sheets.
- V: this will output all V, VPlus, VMinus and pA2V values computed for all groups, phases and combinations of predictor/outcome stimuli. As for ELEMENT, these are NOT limited to when the stimulus is active and so the 'active' flag sheets are present to clarify at which points the predictor stimulus is active. Repetitions are not averaged but reported in separate sheets.

The 'Runtime Details' sheet is present for these verbose outputs as it is for the standard SUMMARY output.



B14 | 0

	A	B	C	D	E	F	G	H	I
1	Conditione	Tr1-1	Tr2-2	Tr3-1	Tr4-2	Tr5-1	Tr6-1	Tr7-1	Tr8-2
2	1	1	1	1	1	1	1	1	1
3	2	1	1	1	1	1	1	1	1
4	3	1	1	1	1	1	1	1	1
5	4	1	1	1	1	1	1	1	1
6	5	1	1	1	1	1	1	1	1
7	6	1	1	1	1	1	1	1	1
8	7	1	1	1	1	1	1	1	1
9	8	1	1	1	1	1	1	1	1
10	9	1	1	1	1	1	1	1	1
11	10	1	1	1	1	1	1	1	1
12	11	0		0		0	0	0	
13									
14	1	0	0	0	0	0	0	0	0
15	2	0	0	0	0	0	0	0	0
16	3	0	0	0	0	0	0	0	0
17	4	0	0	0	0	0	0	0	0
18	5	0	0	0	0	0	0	0	0
19	6	0	0	0	0	0	0	0	0
20	7	0	0	0	0	0	0	0	0
21	8	0	0	0	0	0	0	0	0
22	9	0	0	0	0	0	0	0	0
23	10	0	0	0	0	0	0	0	0
24	11	0	0	0	0	0	0	0	0
25	12	0	0	0	0	0	0	0	0
26	13	0	0	0	0	0	0	0	0
27	14	0	0	0	0	0	0	0	0

Figure 14: Element output, active/inactive stimulus

4 SOP IMPLEMENTATION DETAILS

Wish you to look inside the code, you can also download it in a zip file at https://www.cal-r.org/index.php?id=SOP-sim. Of course, you will need to have Matlab installed. The simulator has been tested on version 2015b of Matlab so users should ensure their version is no older than this before running the code.

The code consists of 2 files:

- simSOP.m
- simSOP.fig

These should be copied into a local machine folder and should not be renamed.

The original Wagner 1981 paper and subsequent papers describing SOP theoretically do not explicitly provide an algorithm for progression from timestep t-1 to timestep t. For clarity, the algorithm is outlined in pseudo-code below. Where values such as propA1 below are expressed in notation 'value(p1,p2)' then p1 represents a stimulus or pair of stimuli and p2 represents timestep. Where there is no dependence on time, p2 is omitted.

4.1 INITIALISATION

```
for each stimulus A
        % set all elements to be initially inactive
        propA1(A,0) = 0
        propA2(A,0) = 0
        propInact(A,0) = 1
        % obtain SOP parameters
        pD1_param(A) =  "pD1" _value_in_Stimulus_SOP_Parameters_table
        pD2_param(A) = pD1_param(A) /  "pD1/pD2"
        _value_in_Stimulus_SOP_Parameters_table
         r1_param(A) =  "r1" _value_in_Stimulus_SOP_Parameters_table
        r2_param(A) =  "r2" _value_in_Stimulus_SOP_Parameters_table
 end for
 for each predictor->outcome pair A->B:
        % set all V-values to zero
        VPlus = 0
        VMinus = 0
        V = 0
        pA2V = 0
        % obtain SOP parameters
        LPlus_param(A->B) =  "LPlus" _value_in_Learning_Rate_Parameters_table
        LMinus_param(A->B) = LPlus_param(A->B)/ “LPlus/LMinus”
        _value_in_Learning_Rate_Parameters_table

    end for
```

4.2 PROGRESSION

```
    for each predictor->outcome pair A->B:

        % calculate changes in associative strength using element proportions
        at time t-1

        deltaVPlus(A->B,t) = LPlus*propA1(A,t-1)*propA1(B,t-1)

        deltaVMinus(A->B,t) = LMinus*pA1(A,t-1)*pA2(B,t-1)

        deltaV(A->B,t) = deltaVPlus(A->B,t) - deltaVMinus(A->B,t)

        V(A->B,t) = V(A->B,t-1) + deltaV(A->B,t)

        VPlus(A->B,t) = VPlus(A->B,t-1) + deltaVPlus(A->B,t)

        VMinus(A->B,t) = VMinus(A->B,t-1) + deltaVMinus(A->B,t)

        if t is first step of ITI

            % pA2V is value of V used in the retrieval rule. It is only
            updated after the end of each trial

            pA2V(A->B,t) = V(A->B,t)

        else

            pA2V(A->B,t) = pA2V(A->B,t-1)

        end if

    end for

    for each stimulus A:

        % calculate new element proportions at time t

        If A is not present at time t

            pA1_param(A,t) = 0

        else

            pA1_param(A,t) = “pA1”_value_in_Stimulus_SOP_Parameters_table
        end

        % new proportion of A1 is that which hasn't decayed to A2 plus
        elements promoted from Inactive

        propA1(A,t) = propA1(A,t-1)*(1-pD1_param(A)) + propInact(A,t-
        1)*pA1_param(A,t)

        % new proportion of A2 is that which hasn't decayed to Inactive plus
        elements decayed from A1 plus elements promoted from inactive using
        the 'retrieval rule' (the formula for obtaining pA2 below)

        pA2(A,t) = sum$_{S<>A}$ pA2V(S->A,t)*[r1_param(S)*propA1(S,t-1) +
        r2_param(S)*propA2(S,t-1)]

        propA2(A,t) = propA2(A,t-1)*(1-pD2_param(A)) + propA1(A,t-
        1)*pD1_param(A) + propInact(A,t-1)*pA2(A,t)

        % new proportion of Inactive is that which has decayed from A2 plus
        what remains after promotions to A1 and A2

        propInact(A,t) = propInact(A,t-1)*(1-pA1_param(A,t)) + propA2(A,t-
        1)*pD2_param(A) - propInact(A,t-1)*pA2(A,t)

    end for
```

4.3 Deviations from original model

The algorithm as described above deviates from the model as described in Wagner's 1981 paper in 3 ways as described below:

1) **Distractor Rules**: the original paper contained rules for modifying decay parameters. We have decided that the rules as they were originally presented would need to be researched further before incorporating them.
2) **Response Generation Rule**: the function for response generation is left as a generic theoretical function.

5 References

Alonso, E., Mondragón, E. & Fernández, A. (2012). A Java simulator of Rescorla and Wagner's prediction error model and configural cue extensions. *Computer Methods and Programs in Biomedicine vol. 108, 1*, 346-355.

Mondragón, E., Alonso, Fernández, A & Gray, J. (2012). Rescorla & Wagner Simulator © V.4 [Computer software]. St. Albans, UK: CAL-R. https://www.cal-r.org/index.php?id=R-Wsim.

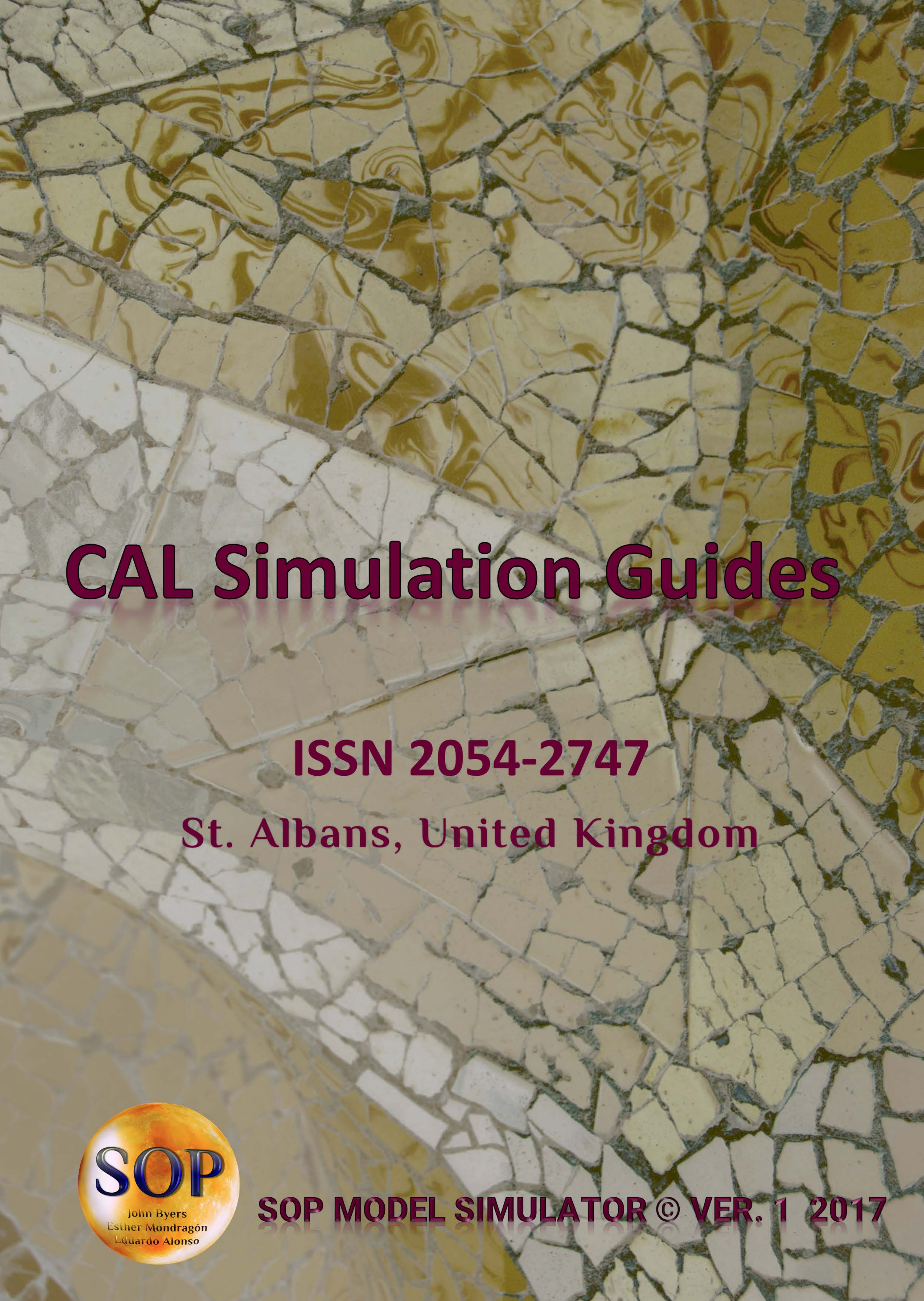
Wagner, A. R. (1981). SOP: A model of automatic memory processing in animal behavior. In N. E. Spear & R. R. Miller (Eds.), *Information processing in animals: Memory mechanisms* (pp. 5-47). Hillsdale, NJ: Erlbaum.

6 Acknowledgements

We would like to thank Edgar Vogel, Guillem Esber, and Omar D. Perez for useful discussions on the interpretation of the original SOP model.

[1] **«YOUR-USER-PROFILE»**/Documents/SOP_Simulator, e.g., **John Smith**/Documents/SOP_Simulator

[2] Or Program Files (x86) if you are using a 32bit platform or have MATLAB 32b installed.



CAL Simulation Guides

ISSN 2054-2747

St. Albans, United Kingdom

SOP
John Byers
Esther Mondragón
Eduardo Alonso