

**R&W Simulator: A Java simulator of Rescorla and Wagner's model and  
configural extensions**

**Eduardo Alonso\***

Department of Computing, City University London

London EC1 0HB, UK

**Esther Mondragón**

Centre for Computational and Animal Learning Research

St Albans AL1 1RQ, UK

**Alberto Fernández**

Escuela Técnica Superior de Ingeniería Informática, Universidad Rey Juan Carlos

Madrid 28933, Spain

**\*Corresponding author:**

London EC1 0HB, UK

telephone: +44 20 7040 4049

e-mail: E.Alonso@city.ac.uk

## Abstract

In this paper we present the “R&W Simulator” (version 3.0), a Java simulator of Rescorla and Wagner’s error correction model of classical conditioning (Rescorla & Wagner, 1972). It is able to run whole experimental designs, and compute and display the associative values of elemental and compound stimuli simultaneously, as well as use extra configural cues in generating compound values (Wagner & Rescorla, 1972); it also permits to change the US across phases. The simulator produces both numerical and graphical outputs, and includes a functionality to export the results to a data processor spreadsheet. It is user-friendly, and built with a graphical interface designed to allow associative learning experts to input the data in their own “language”. It is a cross-platform simulator, so it does not require any special equipment, operative system or support program, and does not need installation. The “R&W Simulator” (version 3.0) is available free.

## Rescorla and Wagner Model

One of the basic principles of learning is that repeated pairings of two events will allow an individual to predict the occurrence of one of them upon presentation of the other, as consequence of the formation of an associative link between them. Typically, classical conditioning, a fundamental associative paradigm, involves the presentation of two stimuli, an originally neutral stimulus (e.g., a tone or a light), and an unconditioned stimulus (US), or *reinforcer*, that has biological relevance (e.g., food). Learning is conceptualized as the formation of an association between the mental representations of these two stimuli. Once the association is formed, presentation of the first stimulus (the conditioned stimulus, or CS) will

not only engender activation of its own mental representation, but will also activate the representation of the other stimulus, the US, by means of the link between them. This produces behavior, termed a conditioned response (CR), indicating that the US is anticipated, and hence predicted by the CS. This simple idea is at the basis of many learning phenomena. Indeed, associative learning has proved to be relevant to human learning both theoretically (judgment of causality and categorization, e.g., Shanks, 1995) and in applications to clinical pathology, and health and addiction (for a review, Schachtman & Reilly, 2011).

Rescorla and Wagner's model of classical conditioning (Rescorla & Wagner, 1972) assumes that learning occurs only if a US is surprising or, more precisely, unpredicted. The amount of growth in associative strength ( $V$ ), a concept that represents the weight of the CS-US link on a particular CS-US pairing (or *trial*), is proportional to the degree to which the US is unexpected. With each CS-US pairing there is an increase in associative strength, until the CS fully predicts the US, at which point the US is no longer surprising. Thus early conditioning trials produce larger increases in associative strength, but these changes decrease in size as learning progresses, and the ability of the CS to predict the US increases, until the learning curve approaches its upper limit or asymptote when the CS fully predicts the US.

Formally, learning on trial  $n$  is defined as  $\Delta V^n = \alpha\beta (\lambda - V^{n-1}(total))$ , where  $\alpha$  and  $\beta$  represent the salience of the CS and of the US respectively,  $\lambda$  is the maximum amount of learning that can occur for that given US, and  $V^{n-1}(total)$  the cumulative amount of learning up to trial  $n-1$  –in other words the sum of associative strengths of all CSs that are present on trial  $n$ . The associative strength of each of the CSs is determined on the last trial on which each CS occurred, ordinarily trial  $(n-1)$ . This delta rule is also known as the *error correction rule*: the change in associative strength, learning, is proportional to the prediction error –the difference

between the predicted and the actual reinforcement –and the resultant change in strength reduces the error. Once the increase in associative strength has been computed, it is then used to calculate the new associative strength of the CS using the *update rule*  $V^n = V^{n-1} + \Delta V^n$ . Obviously, as the prediction improves, the prediction error is reduced until there is nothing left to be learned.

This deceptively simple model (it represents a linear discrete system of the 1<sup>st</sup> order) is, nevertheless, the most influential theory of classical conditioning up to date. Among other phenomena, it successfully predicts extinction and acquisition curves, blocking, overshadowing, conditioned inhibition and discrimination in most experimental procedures. Moreover, as an *error correction* model it lies at the core of integrative models of schizophrenia (e.g., Kapur, 2003) and of real-time models of conditioning that accurately predict the activity of dopamine neurons (Schultz, Dayan & Montague, 1997). It is well known, though, that the original version of Rescorla and Wagner’s model assumes that the associative strength of any compound stimulus is equivalent to the sum of the associative strengths of each component of that compound –the summation assumption. Thus it does not account for learning phenomena that are insoluble without discarding this assumption, such as negative patterning discrimination (Rescorla, 1972) and biconditional discrimination (Rescorla, Grau & Durlach, 1985; Saavedra, 1975). In negative patterning procedures, for example, two stimuli A and B are reinforced when presented alone, but nonreinforced when presented in compound (i.e., A+, B+, AB–); solution of this discrimination requires the organism to withhold responding to the compound of A and B while responding to A and B alone. This result cannot be predicted by models such as Rescorla and Wagner (1972) because of the summation assumption: if A and B predict the US individually, a compound of A and B *must* predict the US *even more* –the opposite of what is found. To account for the fact that

such discriminations can, nonetheless, be solved, one useful adjustment of Rescorla and Wagner's model is to assume that when two stimuli are presented together they create a stimulus compound that consists of the individual elements plus an additional configural cue, a stimulus which is unique to the combination of the elements (Wagner & Rescorla, 1972). This allows the negative patterning discrimination to be represented as A+, B+, ABX-, where X represents this configural cue. This assumption results in X becoming *inhibitory*, as opposite to excitatory. It can therefore counteract the effect of A and B on compound trials, allowing the discrimination to be solved.

In this paper we present a Java simulator of Rescorla and Wagner's model that incorporates configural cues.

### **Why a Java Simulator**

A number of simulators of Rescorla and Wagner's model can be found in the literature or on-line. Most of them have become obsolete, because of their system requirements or because the programming languages with which they were developed are outdated, or inaccessible: For instance, (Blankenship, Schorie, Shaw & Tumlinson, 1994; Lachnit, Scheinder, Lipp & Kimmel, 1988; Lipp, Stevens & Smith, 1999; Mercier, 1996). Others are only of limited use as they were designed to fulfill specific tasks (Pineño, 2004).

Renner (2004) presents a simulator in Excel, a useful tool for teaching undergraduate students the basics of the model; however, Excel is just a spreadsheet application: Even simple programming routines require the definition of VBA macros that are not always intuitive. We have kept Excel for what is effective –to facilitate data processing for statistical analysis, as

detailed below.

More recently, Rescorla and Wagner's model has been simulated using MATLAB (Thorwart, Schultheis, König & Lachnit, 2009); indeed, MATLAB is becoming a widespread tool in building simulators of associative learning models (Glautier, 2007; Schultheis, Thorwart & Lachnit, 2008a, 2008b).

From the point of view of a programmer, both Java and MATLAB are relatively easy to learn and to use (at least, for simple applications). Speed-wise they are also rather similar, no matter whether they compile or interpret. One of Java's advantages is that it is an object oriented programming language (OOP), allowing the user to create modular programs and reusable code. However, the comparison may seem unfair –after all, MATLAB is just a computing environment, and of course lacks the power and flexibility of a full-bodied OOP. Nonetheless, the development of simulations of models such as Rescorla and Wagner's does not require heavy-weight programming.

Nevertheless, there are two issues that may make a Java simulator more attractive. First, MATLAB is a proprietary software and a proprietary language. MATLAB works only with MathWork's MATLAB software – meaning that if you have created programs in MATLAB, you will generally only be able to use those programs in MATLAB, and would need to do extensive porting to move to a different platform. Second, MATLAB is rather expensive; and although it is always possible to use MATLAB-compatible free software like Octave or Scilab, the fact remains that its cost is a serious impediment to universal usage. By comparison, Java is not only open source but also free.

Having said that, we believe that the choice between MATLAB and Java is a matter of preference: At the end of the day, having two simulators of Rescorla and Wagner's model at our disposal, one in Java another in MATLAB, can only benefit the associative learning community.

However, as a user, the "R&W Simulator" has a clear advantage over Thorwart's simulator (Thorwart *et al.*, 2009). The "R&W Simulator" is already compiled to work in different platforms and does not require any other program to run. Thus the user only needs to download and save the executable file ("R&W Simulator.exe" or the "R&W Simulator.app") for their computer platform, PC or Mac respectively. In addition, the Java executable file ("R&W Simulator.jar") is also available and can be run in Linux or any other platform provided that the Java Runtime Environment is installed in the computer; Java is a *truly* platform-independent language.

In short, we have developed our Rescorla and Wagner simulator in Java because it meets the following requirements best: generality, user-friendliness, scalability, fully integrated GUI, Excel export, professional graphical display, free, and platform-independence.

### **The R&W Simulator**

The "R&W Simulator" has been built to allow the user to simulate a wide range of procedures, and can compute whole experimental designs at once. It is thus versatile and general. It is not only capable of simulating well-known tasks such as acquisition, extinction, blocking, overshadowing, etc., but it also permits the computation of associative values for elemental stimuli, compound stimuli and configural cue compounds in a single design and

display. The user can, moreover, run designs with different US across phases and simulate phenomena such as unblocking (Holland, 1984). In addition, the simulator allows to set negative values to non-US  $\lambda$  that may be useful in simulating categorization experiments in humans involving symmetrical outcomes, that is, for experiments in which outcomes are rated as positive, neutral or negative (e.g., Murphy, Schmeer, Vallée-Tourangeau, Mondragón & Hilton, 2011).

The “R&W Simulator” generates both numerical and graphical outputs with a single click. In addition, the user can export the results to a data processor spreadsheet for better manipulation and analysis of data. Its design includes a graphical interface in which the experimental procedure can be entered in a way that resembles standard associative learning designs, so that learning experts can write the program in their own “language”.

### **Installation**

The “R&W Simulator” is available to download at “<http://www.cal-r.org/index.php?id=R-Wsim>”.

Those who *just want to use* the simulator would need to select "PC" or "MAC", depending on which platform they use, and the program will download. Once the file is saved, it is ready to run –it does not require any additional installation. Users of other platforms should select the “JAVA” button to download the “RW\_Simulator.jar” file, which will run on any platform provided that the Java Runtime Environment (JRE) is installed in their computer. Most popular Linux distributions such as Fedora, Debian, Ubuntu, Arch, and CentOS already include JRE.



Users who wish to *access the code* should also download the “RW\_Simulator.jar” file, and uncompress it. A folder named "R&W Simulator.jar" containing the “.class” files will appear. The content of these files can be accessed using a Java editor such as Eclipse or NetBeans, and the Java Development Kit (JDK).

### **Starting the simulator and creating a new experiment<sup>1</sup>**

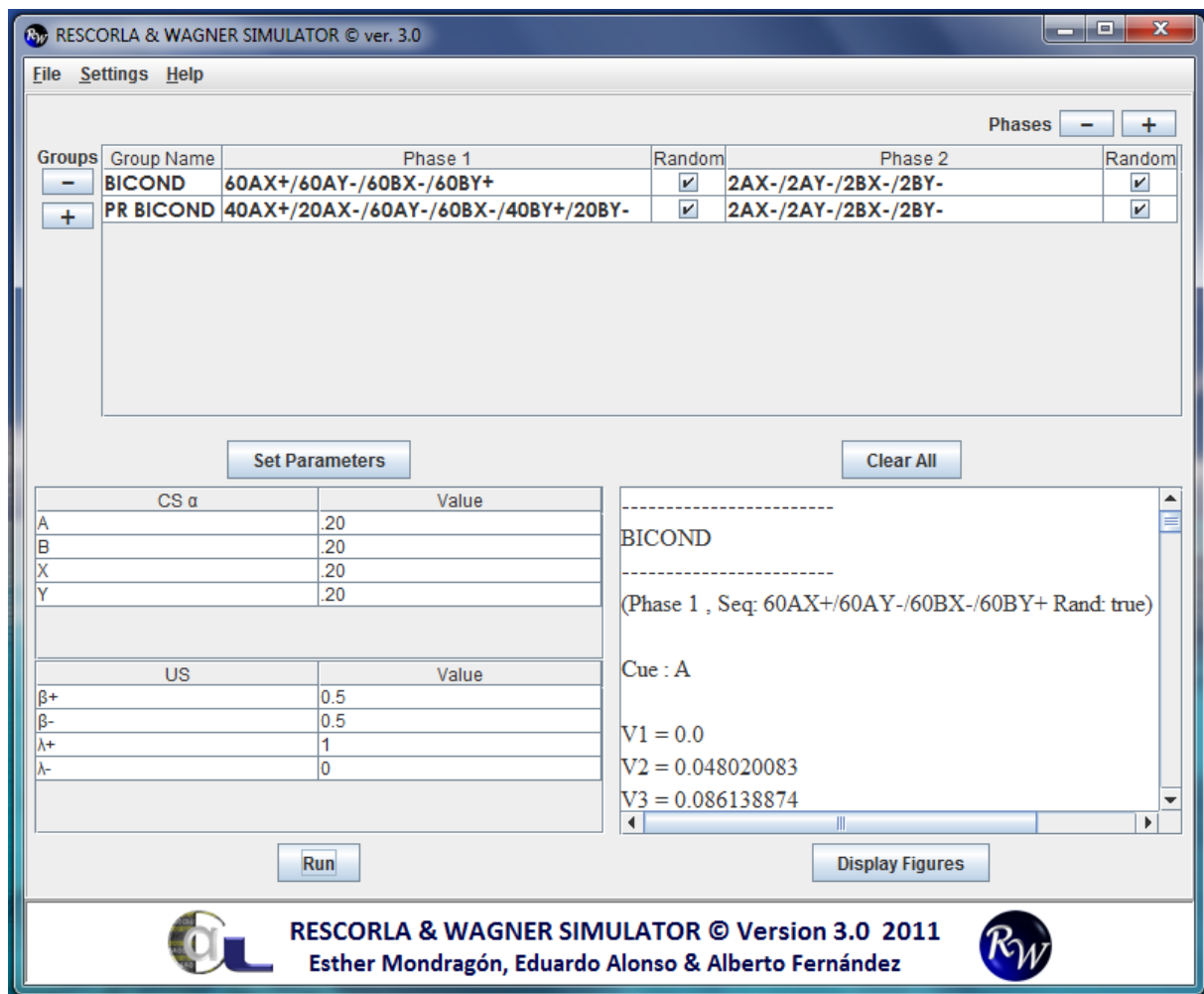
To start the simulator the user needs to navigate to the directory in which the file was stored and double click on the file's icon. The opening screen should look like in Figure 1 (PC version, Mac's GUI differs slightly).

This window is headed by the main menu (“File”, “Settings”, and “Help”), and consists of two input panels and one output panel. The experimental design is specified in a matrix of groups and phases in the top panel; in the bottom left panel the values of the parameters are entered; the output data is displayed on the right.

The user can choose to create a new experiment, or to load one that they may have previously saved. Assuming that this is the first time one uses the simulator, we are creating a new experiment: We are using a biconditional discrimination (AX+,AY-,BX-,BY+) as a guiding example to make it interesting; this discrimination cannot be solved without the addition of configural cues, because otherwise every component of the discrimination (A, B, X, Y) is followed by reinforcement (+) and nonreinforcement (-) on an equal number of occasions, with the result that the associative strength on each of the four trial types will be identical.

---

<sup>1</sup> Step-by-step instructions to use the simulator are available as a “Guide” in the “Help” menu.



**Figure 1.** Main GUI of the “R&W Simulator” showing a biconditional discrimination design with two groups and two phases.

The experimental design is entered describing each trial type as follows: Number of trials followed by stimuli followed by reinforcer. Different trials should be separated by a slash symbol without spaces between the characters. Thus the biconditional discrimination depicted above would read “60AX+/60AY-/60BX-/60BY+” in “Group BICOND”. The order of the trials is defined by the order in which they are entered in each phase; alternatively, if the

design requires that the different types of trial occur in a random order, the “Random” box should be ticked.

To overcome order bias, the simulator runs a number of different random combinations and generates a mean value per stimulus and trial. By default this number is set to 1,000, but it can be changed in the “Settings/Number of Random Combinations” menu.

The values of the fixed parameters,  $\alpha$ ,  $\beta$  and  $\lambda$ , are entered by first pressing the “Set Parameters” button. CS  $\alpha$  values are entered at the top. The bottom area contains a set of default values given to the US, which the user can modify at will. In addition the user can set different US values per phase using the “Settings/Set Different US per Phase” menu.

Pressing the “Run” button will produce a text output on the right hand side. Mean stimulus V values per trial will be displayed for each group in each phase.

### **File menu**

Experimental designs can be saved and opened using the “File” menu. These files will have a “.res” extension. The simulator includes the option to export the results in “.xlsx” type spreadsheets, like the ones used by Excel; a workbook is created with a different sheet for every group. For the sake of clarity, each sheet contains the name of the file followed by the CS and US parameters. Each phase is presented on a different table, and the phase tables are preceded by the experiment design. This functionality allows the user to prepare the data as required for analysis.

## **Figures display**

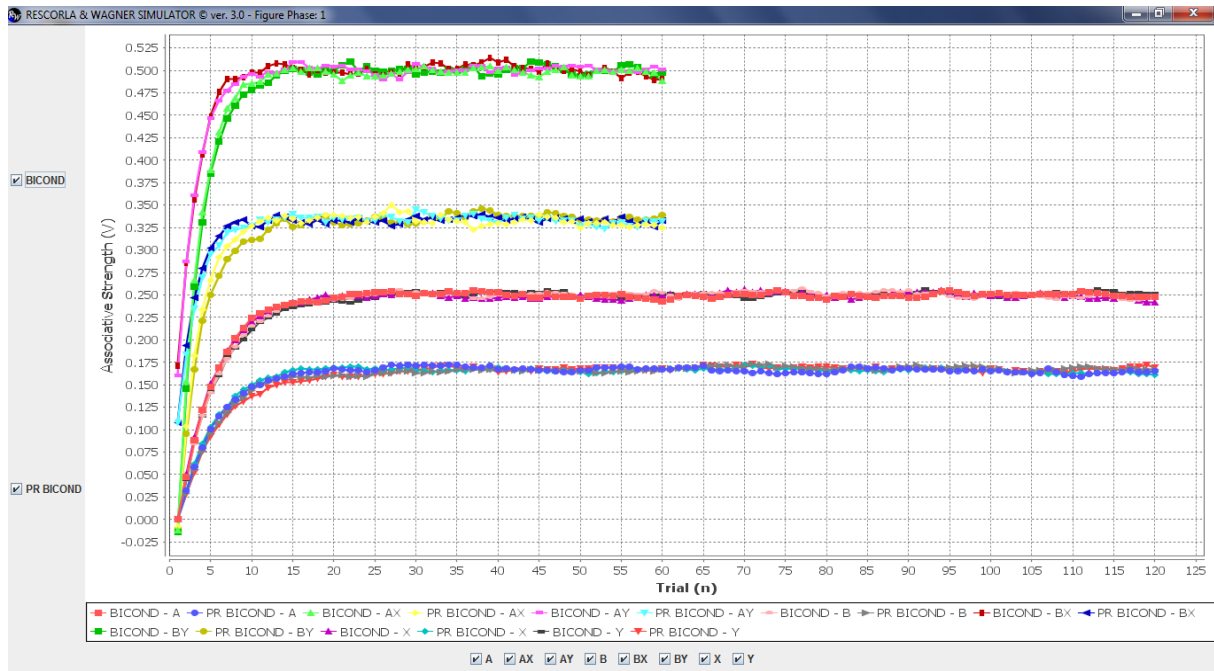
A graphical representation of the output is obtained by pressing the “Display Figures” button. A number of figures will pop up, one per phase. Each figure shows the stimulus mean  $V$  values per trial. The stimulus and group to be displayed can be enabled or disabled as required. Figures can also be saved, copied, printed, zoomed and modified by right-clicking the graphic. The window can remain open while the user chooses to run a new experiment so they can compare the figures.

For example, a simulation of two biconditional discriminations, one with continuously reinforced positive trials (Group BICOND) another with partially reinforced (66.67%) positive trials (Group PR BICOND), was run using the following parameters: 240 trials, 60 of each compound,  $\alpha = 0.2$  for each stimulus,  $\lambda (+) = 1$ , and  $\lambda (-) = 0$ . This simulation produced the graphical output shown in Figure 2.

## **Compounds and configural cue compounds**

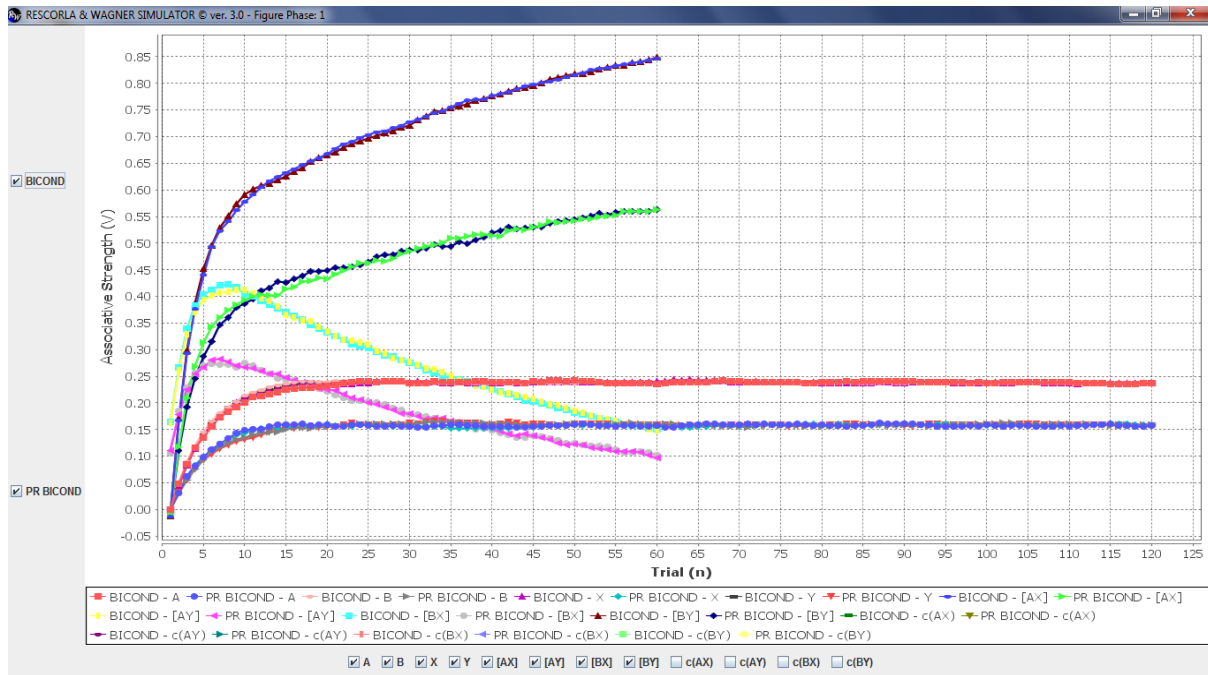
The “R&W Simulator” includes the possibility of computing both standard stimulus compounds (Figure 2) and compound stimuli with additional configural cues as proposed by Wagner and Rescorla (1972). Figure 3 shows the results of a simulation of the biconditional discriminations, with the same parameters and design as described above, using configural cues.

To calculate standard compound associative values “Show Compound Results” must be selected in the “Settings” menu. Running the simulator will produce individual trial values for each stimulus compound (e.g., AX) described in the design, as well as for each of the elemental stimuli. Likewise, compound data will be shown in the figures and in the exported spreadsheets.



**Figure 2.** Stimulus and compound associative strength values during a biconditional discrimination training for Group BICOND and Group PR BICOND.

In order to compute configural cues, “Use Configural Cues” must be chosen and the parameters redefined to set  $\alpha$  values for the configural cues –by default, the product of the  $\alpha$  values of the component elements of the corresponding stimulus compound. Configural cues are represented as “c(AX), c(AY), ...”. Running with these settings will produce associative values for each configural cue compound ([AX]), which will also be displayed in the output and in the figures.

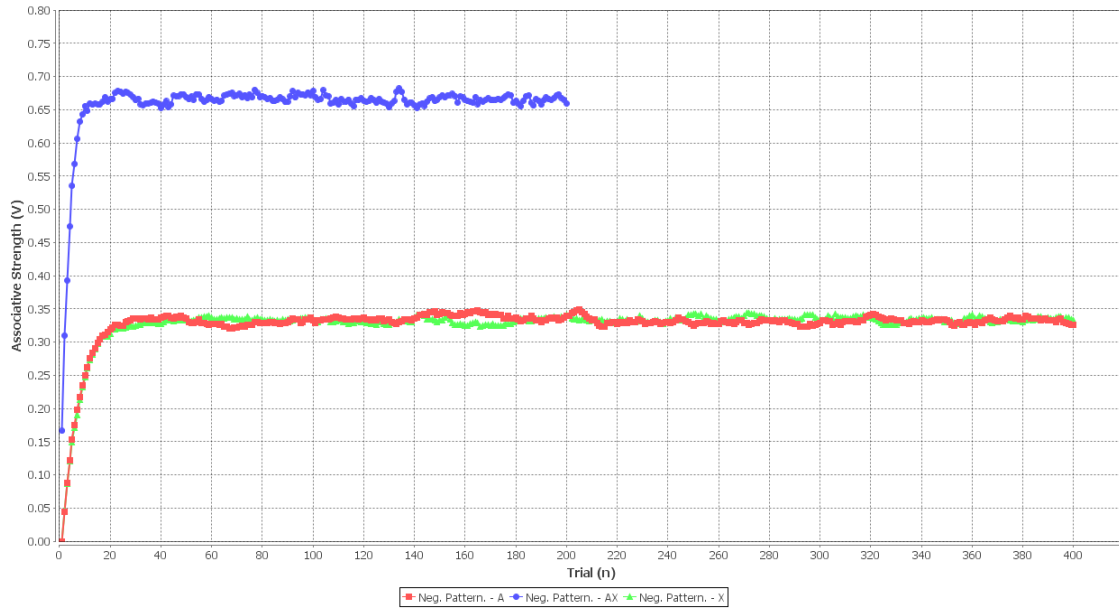


**Figure 3.** Stimulus and configural cue compound associative strength during a biconditional discrimination training for Group BICOND and Group PR BICOND.

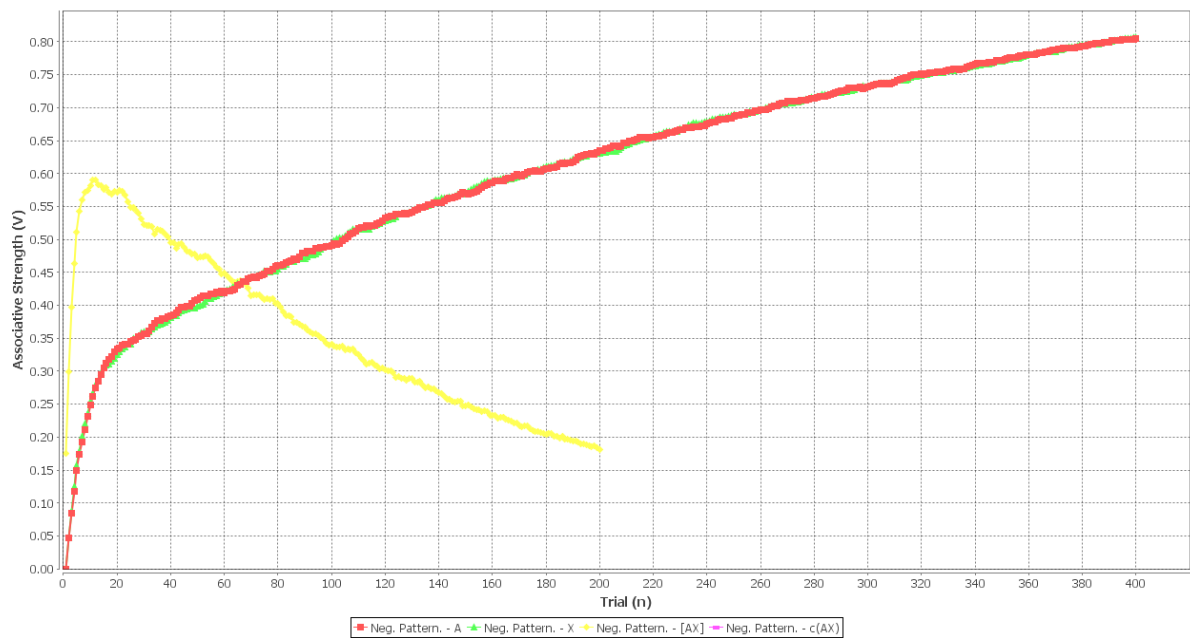
## Test

The simulator has been thoroughly tested against phenomena that the Rescorla and Wagner model successfully accounts for, and also against some it notoriously does not. In this we have followed an exhaustive review of the model by Miller and collaborators (Miller, Barnet & Grahame, 1995). For example, the simulator accurately predicts extinction and acquisition curves, blocking, overshadowing, conditioned inhibition and positive patterning discrimination; and, without choosing the configural cue option, fails to predict biconditional discriminations (Figure 2) –because, as is well-known, the Rescorla and Wagner model can only solve these discriminations by including configural cues (Figure 3). As another example, a negative patterning discrimination ( $AX-/A+/X+$ ), was simulated with a total of 600 trials, 200 each type,  $\alpha = 0.2$  for each stimulus;  $\lambda(+) = 1$ ;  $\lambda(-) = 0$ , and  $\beta = 0.5$ . Figure 4 shows the compound values for this simulation. It is clear that the Rescorla and Wagner model cannot

solve negative patterning without configural cues. Figure 5 displays the configural cue compounds calculated with  $\alpha(AX) = 0.04$ . As it can be seen, the model can now solve the task.



**Figure 4.** Compound stimuli associative strength throughout a negative patterning training discrimination.



**Figure 5.** Associative strength of the compound stimuli with added configural cues throughout a negative patterning training discrimination.

## Conclusions

The “R&W Simulator” (version 3.0) provides an easy-to-use yet specialized, fast and free tool to test the predictions of the original Rescorla and Wagner model as well as modifications involving configural cue compounds. Users will be able to enter whole designs, save figures, and export the data for further analysis and manipulation. The simulator runs in all computer platforms and does not require installation.

## Acknowledgements

We would like to thank Dionysios Skordoulis and Rocío García-Durán for their contribution in developing version 1.0 and version 2.0 of the “R&W Simulator” respectively. Also, this project would have not been possible without insightful feedback from various colleagues.

## References

- Blankenship, V., Schorie, S.S., Shaw, A.J., & Tumlinson, J. (1994). Teaching the Rescorla-Wagner model using STELLA-II. *Behavior Research Methods, Instruments, & Computers*, 26, 128-133.
- Glautier, S. (2007). Simulation of associative learning with the replaced elements model. *Behavior Research Methods*, 39, 993-1000.
- Holland, P. C. (1984). Unblocking in Pavlovian appetitive conditioning. *Journal of Experimental Psychology: Animal Behavior Processes*, 10, 476 – 497.



- Kapur S. (2003). Psychosis as a state of aberrant salience: a framework linking biology, phenomenology, and pharmacology in schizophrenia, *Am. J. Psychiatry*, *160*, 13-23.
- Lachnit, H., Scheinder, R. L., Lipp, O.V., & Kimmel, H.D. (1988). RWMODEL: A program in Turbo Pascal for simulating predictions based on the Rescorla-Wagner model of classical conditioning. *Behavior Research Methods, Instruments, & Computers*, *20*, 413-415.
- Lipp, O.V., Stevens, J. & Smith, T. (1999). RWMODEL II: Computer simulation of the Rescorla-Wagner model of Pavlovian conditioning. *Behavior Research Methods, Instruments, & Computers*, *31*, 735-736.
- Mercier, P. (1996). Computer simulations of the Rescorla-Wagner and Pearce-Hall models in conditioning and contingency judgment. *Behavior Research Methods, Instruments, & Computers*, *28*, 55-60.
- Miller, R. R., Barnet, R. C., & Grahame, N. J. (1995). Assessment of the Rescorla-Wagner model. *Psychological Bulletin*, *117*, 363-386.
- Murphy, R.A., Schmeer, S., Vallée-Tourangeau, F., Mondragón, E. & Hilton, D. (2011). Making the illusory correlation effect appear and then disappear: The effects of increased learning. *Quarterly Journal of Experimental Psychology*, *64*, 24-40.
- Pineño, O. (2004). **Rescorla-Wagner / Van Hamme-Wasserman - Simulation Program.** [Computer software]. New York: Oskar Pineño's website. <http://www.opineno.com/>
- Renner, R. (2004). Learning the Rescorla-Wagner Model of Pavlovian Conditioning: An Interactive Simulation. *Teaching of Psychology*, *31*, 146-148.
- Rescorla, R.A. (1972). "Configural" conditioning in discrete-trial bar pressing. *Journal of Comparative and Physiological Psychology*, *79*, 307-317.
- Rescorla, R. A., Grau, J. W., & Durlach, P. J. (1985). Analysis of the unique cue in configural discriminations. *Journal of Experimental Psychology: Animal Behavior Processes*, *11*, 356-366.

- Rescorla, R.A., & Wagner, A.R. (1972). A theory of Pavlovian conditioning: The effectiveness of reinforcement and non-reinforcement. In A.H. Black & W.F. Prokasy (Eds.), *Classical Conditioning II: Current Research and Theory* (pp. 64-99). New York: Appleton-Century-Crofts.
- Saavedra, M. A. (1975). Pavlovian compound conditioning in the rabbit. *Learning and Motivation*, 6, 314–326
- Schachtman, T.R., & Reilly, S. (2011). *Associative Learning and Conditioning Theory: Human and Non-Human Applications*. Oxford, England: Oxford University Press.
- Schultheis, H., Thorwart, A., & Lachnit, H. (2008a). HMS: A MATLAB simulator of the Harris model of associative learning. *Behavior Research Methods*, 40, 442-449.
- Schultheis, H., Thorwart, A., & Lachnit, H. (2008b). Rapid-REM: A MATLAB simulator of the replaced-elements model. *Behavior Research Methods*, 40, 435-441.
- Schultz, W., Dayan, P., & Montague, P.R. (1997). A neural substrate of prediction and reward, *Science*, 275, 1593-1599.
- Shanks, D.R. (1995). *The Psychology of Associative Learning*. Cambridge, England: Cambridge University Press.
- Thorwart, A., Schultheis, H., König, S., & Lachnit, H. (2009). ALTSim: A MATLAB simulator for current associative learning theories. *Behavior Research Methods*, 41, 29-34.
- Wagner, A.R., & Rescorla, R.A. (1972). Inhibition in Pavlovian conditioning: Application of a theory. In R.A. Boakes & M.S. Halliday (Eds.), *Inhibition and Learning* (pp. 301-336). New York: Academic Press.