

GenWM: Generic Working Memory task modelling environment.

User's Guide.

Index

1. Introduction	3
2. Installing and starting GenWM	3
3. How does it work?	4
3.1. Main user interface window	4
3.2. Configuring experiment parameters	5
3.2.1. Parameter set management	7
3.3. Configuring model parameters	9
3.3.1. Parameter set management	11
4. Running a simulation	13
4.1. Load experiment parameter set.....	13
4.2. Load a model parameter set.....	13
4.3. Set number of simulations	14
4.4. Run simulation.....	14
4.5. Clear figures	15
4.6. Running further simulations.....	15
4.7. Testing new simulations	15
5. Problems and Troubleshooting	16
Appendix A. Available experiment paradigms	17
Appendix B. Available model configurations	21
Appendix C. Analysis of data files	22
References.....	23

1. Introduction

GenWM is a software package that provides a generic modelling environment for simulating a range of working memory paradigms. GenWM allows specific properties associated with simple and complex span tasks to be tested against computational implementations of current theories of working memory, such as those of the SOB model, as represented by the current instantiation for complex span, SOB-CS.

GenWM includes models for a number of variations of the complex span task. These include tests that manipulate the number of distractors per item, the processing duration per distractor and the amount of free time between distractors. Other variants of the complex span paradigm include simulations that examine the effects of the similarity of representation of items and distractors and a simulation examining cross-domain interference in the encoding of domain specific representations (such as might be anticipated in encoding information across visual and verbal modalities).

These paradigms are pre-defined as experiment parameter sets in the simulation software. The default values for each paradigm can be modified, allowing the user to test alternative parameter values without compromising the default settings. They thus provide a safe environment within which to begin testing out variations of these models. However, the environment also allows new experiment parameter sets to be created, allowing the user to define their own models.

Similarly, parameters associated with the theoretical model can be changed, allowing the effects of these parameters on recall performance to be examined. Again, fixed sets are defined within the environment that cannot be permanently changed but it is also possible to modify these or to create further model parameter configurations.

This manual describes the basic features of the environment in order to configure and run simulations, describing those parameters available to be configured within the simulations. The manual describes how to configure and execute simulations and includes sections describing how to configure, manage and load the parameter sets.

2. Installing and starting GenWM

GenWM is installed by starting the program `genwm_pkg.exe`. This program self-installs the Matlab environment Matlab Compiler Runtime 7.10 (MCR) and the program `genwm.exe` in the current directory. Following completion of the installation process the installation program should be deleted.

GenWM is started by starting the program `genwm.exe`. The program will automatically start the MCR environment and consequently initial start up of the program is quite slow. Once MCR is loaded it remains loaded for the remaining session and so subsequent use of GenWM should be faster to load. On starting GenWM, a second MSDOS window will open in the background. This window is not required by the user but should not be closed down. The window displays runtime messages that may be of help to the user during some simulations but it is not necessary to monitor these.

GenWM is shut down by closing all windows. The main user interface window should be shut down by clicking on the red Close button in the top right-hand corner of the window (as shown in *Figure 1*).

3. How does it work?

Simulations are configured from the main user interface window. From the main interface two further windows can be opened in order to configure and load experiment and model parameter sets. When the experiment and model configurations have been selected and the number of simulations defined, the simulation can be started. Output appears in the form of individual figures that are displayed on the screen. In addition, some paradigms produce data files that can be used for further analysis. In the following pages we will explain the features of each of the three main windows.

3.1. Main user interface window

The main user interface window is shown in *Figure 1*. There are two buttons for the configuration of parameters. The *Configure experiment* button opens a separate window allowing configuration of experiment parameter values. Similarly, the *Configure model* button opens a separate window allowing configuration of model parameters. Below the *Configure model* button, the number of simulations to be performed is specified. This has a default value of 500 simulations but can be edited as required. Further buttons on the screen, the *Run simulation* and *Clear figures* buttons, are disabled until experiment and model parameter sets have been loaded. A status bar at the foot of the screen indicates the current status of the simulation.

When an experiment parameter set is loaded from the experiment configuration window, the name of the selected set is displayed in the box in the top-right of the window. The box below this similarly displays the name of the model configuration set when this is loaded. When both experiment and model configurations are loaded the status bar will indicate: *Status: Parameter sets loaded* and the *Run simulation* button becomes active. Clicking on this will start the simulation.

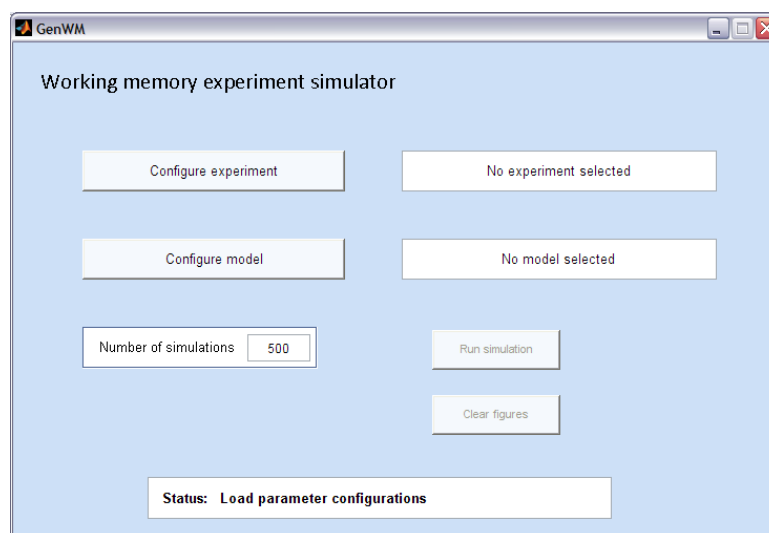


Figure 1. Main user interface window on starting GenWM.

The configuration buttons are disabled while a simulation is in progress. The status bar will indicate: *Status: Simulation running* while the simulation is in progress. On completion of the simulation this is updated to display: *Status: Simulation complete*.

When the simulation is complete, the *Clear figures* button is activated. Any figures displayed on the window may be closed by clicking on the *Clear figures* button.

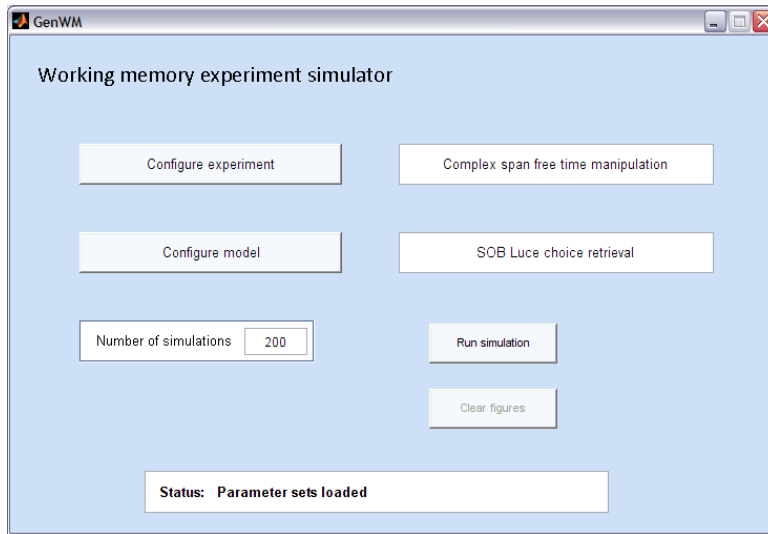


Figure 2. Main display window for user interface following selection of parameter sets.

Figure 2 shows the configuration screen following the selection of an experiment parameter set and a model configuration set. The number of simulations has been set to 200 and the simulation is now ready to be performed. Clicking on *Run simulation* will now start the simulation.

In some simulations the output will comprise up to 30 figures. The *Clear figures* button works on the basis of clearing plots starting from Figure 1 and clearing consecutive figure numbers, e.g., 1, 2, 3, 4, 5. If a figure is closed manually, this will prevent the feature closing any further figures. Hence if Figure 3 is closed manually, Figures 4 and 5 will not be closed by the *Clear figures* button and will need to be cleared manually.

3.2. Configuring experiment parameters

Editing of experiment properties associated with a simulation takes place within this window. The configuration window for experiment parameters is shown in Figure 3.

When the configuration window is opened, the last saved parameter set is displayed. This may not be the last parameter set opened, since closing the window without saving does not automatically update stored parameter settings. Saved parameter set information is updated when the *Save* button is selected or when the *Load* button is selected to load the current parameter set into the simulation.

Within the defined paradigms, some parameters may not be required or may be fixed within the simulation. The parameter will be disabled and the display greyed out to prevent user values from being entered. The properties and parameters associated with the experiment configuration window are explained below:

Experiment parameter set indicates the name of the displayed parameter set. The name can be edited in the text box, allowing a maximum of 40 characters.

Experiment paradigm indicates the paradigm to be simulated. Details of these can be found in *Appendix A*. The selected paradigm can be changed by selecting the *Change paradigm* button.

Number of list items specifies the number of list items that are to be presented for recall. When a fixed list length is used in the simulation, the list length will be configurable up to a maximum of 9 items. Where a span procedure is being simulated, this value will be disabled, as shown in *Figure 3*. In such cases this value is defined within the code. In complex span simulations, the number of list items is increased from 1 to 9 items.

Number of distractor conditions specifies the number of conditions for which the *Number of distractors* is to be experimentally manipulated, activating the required number of displayed boxes for the simulation. The *Number of distractors* indicates the number of distractors following each item for each condition. In *Figure 3*, four distractor conditions are specified. The number of distractors per item is defined for each condition: 0, 1, 4, 8. (Note: 0 distractors is equivalent to simple span). There is a maximum of five distractor conditions and the maximum number of distractors per item is fixed at twelve distractors.

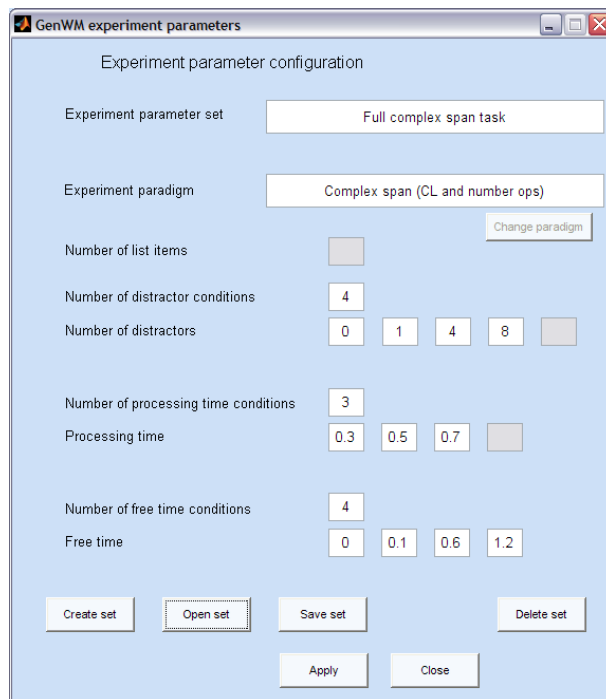


Figure 3. Experiment parameter configuration window.

Number of processing time conditions specifies the number of conditions in which the processing time is experimentally manipulated. Processing time specifies the time assumed for processing of each distractor. In *Figure 3*, three processing time conditions are specified. The time for each condition is defined in the active boxes specifying *Processing time*, immediately below this parameter. The maximum number of processing time conditions is four. There is no limitation on the processing time per distractor.

Number of free time conditions specifies the number of conditions in which free time is experimentally manipulated. Free time specifies the amount of available time

following processing of the current distractor prior to presentation of the next distractor or item. In *Figure 3*, four free time conditions are specified. The time for each condition is defined in the active boxes specifying *Free time*, immediately below this parameter. The maximum number of free time conditions is four. There is no limitation on the free time per distractor.

It should thus be noted that for any one condition the interval between distractors is the sum of the processing time and the free time and the interval between items is the sum of the processing time and free time multiplied by the number of distractor operations to be performed between items.

Change paradigm. Opens a drop-down box as shown in *Figure 4*. This allows selection of the current paradigm from a set of pre-defined experiment paradigms. Descriptions for each paradigm are provided in *Appendix A*. Note: when a default parameter set is loaded the *Change paradigm* button is disabled.

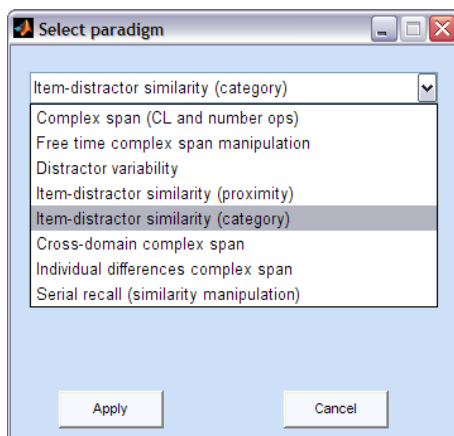


Figure 4. Experiment paradigm selection window.

On selecting an appropriate paradigm the paradigm name will be displayed in the experiment configuration window. Selection of a specific paradigm may lead to some parameters being disabled, as is shown in *Figure 3* for *Number of list items*. These parameters will be assigned fixed values within the model.

3.2.1. Parameter set management

The buttons located along the bottom of the configuration window allow management of the experiment parameter sets. These allow new parameter sets to be created, existing parameter sets to be opened, parameter sets to be saved or deleted and the current parameter set to be loaded into the simulation. These features are explained below:

Create set. Creates a new parameter set. This is assigned the name *Default parameter set* and uses the default parameter values shown in *Figure 5*. The new parameter set can be edited directly from the configuration window, as described previously. On creating a new parameter set, changes to the previous parameter set will be automatically saved.

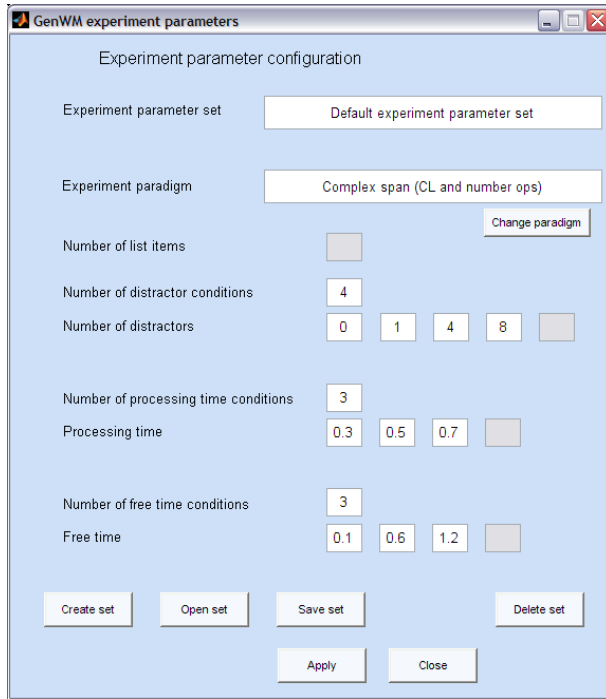


Figure 5. Default parameter set configuration.

Open set. Allows an existing parameter set to be loaded into the configuration window. This opens a new window listing the available parameter sets, allowing the required parameter set to be selected and loaded, as shown in *Figure 6*.

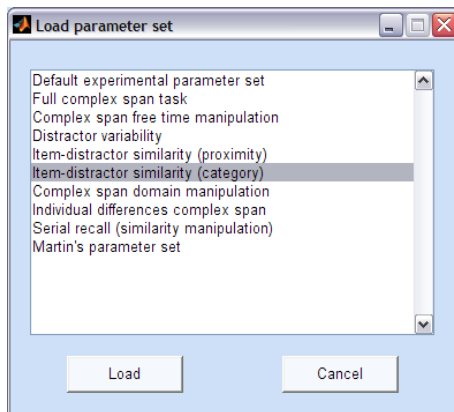


Figure 6. Loading a current parameter set.

Save set. Stores the current parameter settings. When the current parameter set is one of the default parameter sets, the action is disabled and the current settings will not be stored. To save a specific configuration, a new parameter set must be created and the appropriate paradigm and values assigned to this. The defined configuration can then be saved.

Delete set. Deletes the parameter set currently displayed in the experiment configuration window. Confirmation is required to avoid accidental deletion of the current set. Deletion of default parameter sets is blocked.

Apply. Loads the current parameter set into the model and closes the experiment configuration window. On applying a parameter set, any changes to the set are automatically saved.

Close. Closes the configuration window. Closing does not automatically save any unsaved changes to the current parameter set. On re-opening the configuration window the last saved parameter set will be displayed.

3.3. Configuring model parameters

The configuration window for model parameters is shown in *Figure 7*.

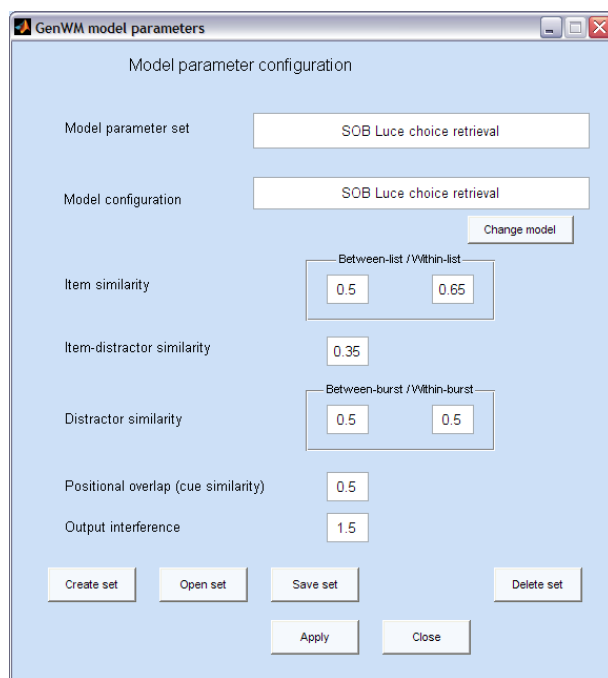


Figure 7. Model parameter configuration window.

When the window is opened, the last saved model configuration set is displayed. Consistent with the experiment configuration window, the displayed configuration may not represent the last set opened, since closing the window without saving does not automatically update stored parameter settings. Saved parameter set information is updated when the *Save* button is selected or when the *Load* button is selected to load the current parameter set into the simulation.

Within the defined model configurations, some parameters may be fixed within the simulation. The parameter will be disabled and the display greyed out to prevent user values from being entered. The properties and parameters associated with the model parameter configuration window are explained below:

Model parameter set indicates the name of the displayed parameter set. The name can be edited in the text box, allowing a maximum of 40 characters.

Model configuration indicates which of the currently available configurations has been selected. This can be changed by selecting the *Change model* button.

Item similarity defines the similarity coefficient between items. The *between-list* value defines the similarity of items across trials, while the *within-list* value defines the similarity of items within a single list. In both cases the range of similarity values increases from 0 to 1; 0 being dissimilar and 1 being identical.

Item-distractor similarity defines the similarity coefficient of distractors with respect to list items. The similarity value increases from 0 to 1; 0 being dissimilar, 1 being identical.

Distractor similarity defines the similarity of distractors within and between bursts within a trial. The *between-burst* value defines the similarity of distractor bursts between list items. The *within-burst* value defines the similarity of distractors within a single distractor burst. For example, when the within-burst similarity is set to 1, identical distractors are generated within each burst. The similarity coefficients operate over the range 0 to 1; 0 being dissimilar, 1 being identical.

Positional overlap specifies the similarity coefficient between context representations. This defines the extent to which features are shared between a position representation and neighbouring representations. When retrieval is cued by position, increasing similarity between position representations increases the likelihood of retrieving an item from a neighbouring position, resulting in transpositions between neighbouring items. The coefficient operates over the range 0 to 1; 0 indicating no overlap and 1 indicating identical position codes.

Output interference. This defines the amount of noise introduced into the model during item retrieval. Increasing the level of noise increases the probability of retrieval errors. Such an effect may lead to, omissions, intra-list intrusions and transpositions during recall. Noise is typically within the range 0.0 – 2.0 but there is no limitation on this value.

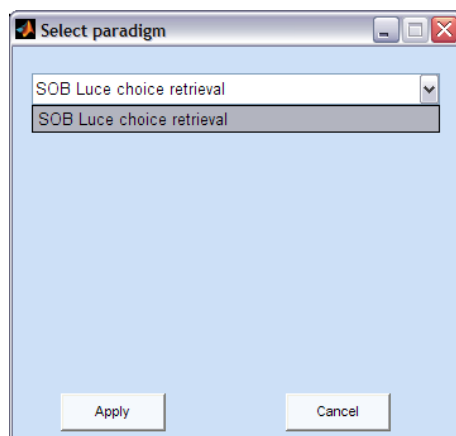


Figure 8. Loading a model configuration.

Change model. Opens a drop-down box to select from possible model configurations, as shown in *Figure 8*. Descriptions for each model are provided in

Appendix B. Note: when a default model configuration set is loaded the *Change model* button is disabled.

3.3.1. Parameter set management

The buttons along the bottom of the configuration window allow management of the model parameter sets. These have the same functions as the buttons for the experiment configuration window, as described in section 3.2.1. The buttons allow new parameter sets to be created, existing parameter sets to be opened, parameter sets to be saved or deleted and the current parameter set to be loaded into the simulation. These features are explained below.

Create set. Creates a new parameter set. This is assigned the name *Default model parameter set* and uses the default parameter values shown in *Figure 9* below. The parameter set can be edited directly in the configuration window. On creating a new parameter set, any changes to the previous set are automatically saved.

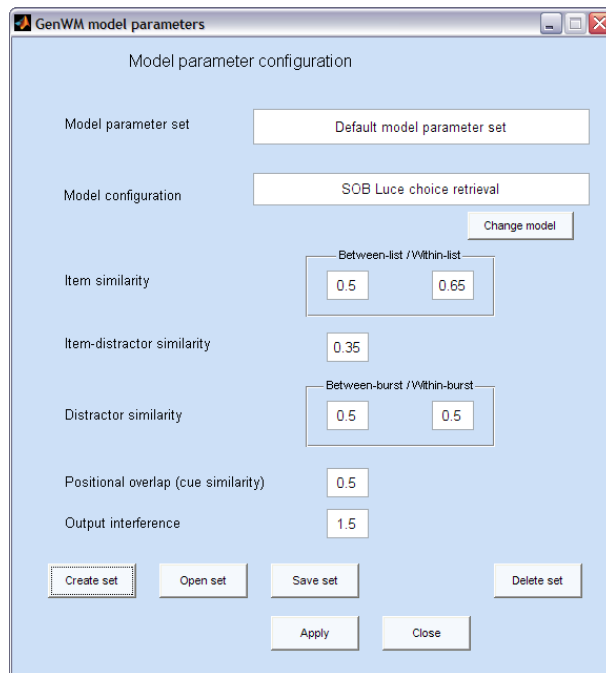


Figure 9. Default model parameter set configuration.

Open set. Allows an existing parameter set to be loaded into the configuration window. This opens a new window listing the available parameter sets, allowing the required parameter set to be selected and loaded, as shown in *Figure 10*.

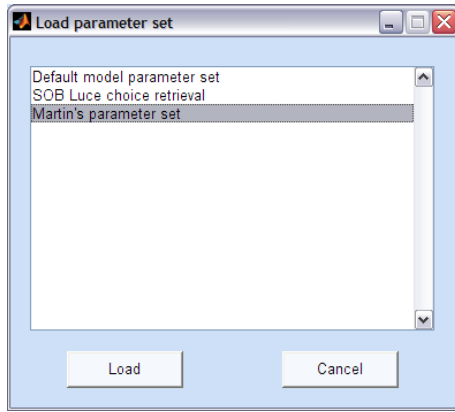


Figure 10. Loading a current model parameter set.

Save set. Stores the current parameter settings. When the current parameter set is one of the default parameter sets, the action is disabled and the current settings will not be stored. To save a specific configuration, a new parameter set must be created and the appropriate paradigm and values assigned to this. The defined configuration can then be saved.

Delete set. Deletes the parameter set currently displayed in the model configuration window. Confirmation is required to avoid accidental deletion of the current set. Deletion of default parameter sets is blocked.

Apply. Loads the current parameter set into the model and closes the model configuration window. On applying a parameter set, any changes to the set are automatically saved.

Close. Closes the configuration window. Closing does not automatically save any unsaved changes to the current parameter set. On re-opening the configuration window the last saved parameter set will be displayed.

4. Running a simulation

The following section explains how to create a model for testing.

GenWM should be started by clicking on the genwm.exe icon. The steps to run a simulation are as follows:

1. Load experiment parameter set.
2. Load model parameter set.
3. Set number of simulations.
4. Run simulation

Each of these steps is discussed further in the following sections.

4.1. Load experiment parameter set

- Click on the *Configure experiment simulation* button.
- The last saved parameter set will be displayed in the experiment configuration window. If the parameter set is correctly configured for immediate use, click on the *Apply* button. The parameter set will be loaded into the model, the window will be closed and the selected parameter set name will be displayed in the main user interface window.
- If you wish to edit the existing values, these can be edited directly in the displayed window. If the parameter set is a default set, it will not be possible to save these changes but you will be able to apply the changes to the model for testing.
- If you wish to load an alternative parameter set, click on the *Open set* button, select a parameter set from the list and click on the *Load* button.
- If you wish to create your own parameter set, click on the *Create set* button. A new window will open. You can change the paradigm and configuration as required. Click on the *Save set* button to save any changes.
- You can change the paradigm by clicking on the *Change paradigm* button, selecting from the list displayed in the new window and then clicking on *Apply*. If you select *Close*, the current paradigm will remain selected.
- Unwanted parameter sets can be deleted by loading these into the configuration window and then clicking on the *Delete set* button.
- When *Apply* is selected the current configuration will be saved and loaded into the model. The experiment configuration window is then closed.
- When *Close* is selected, any changes to the current configuration are not saved and nothing is loaded into the model. This allows users to browse and edit parameter sets without changing the current parameter set loaded into the model.

4.2. Load a model parameter set

- Click on the *Configure model* button.

- The last saved parameter set will be displayed in the configuration window. If the parameter set is correctly configured for immediate use, click on the *Apply* button. The configuration will be loaded into the model, the window will be closed and the selected parameter set name will be displayed on the main window.
- If you wish to edit the existing values, these can be edited directly in the displayed window. If the parameter set is a default set, it will not be possible to save these changes but you will be able to apply the changes to the model for testing.
- If you wish to load an alternative parameter set, click on the *Open set* button, select a parameter set from the list and click on the *Load* button.
- If you wish to create your own parameter set, click on the *Create set* button. A new window will open. You can change the paradigm and configuration as required. Click on the *Save set* button to save any changes.
- You can change the model by clicking on the *Change model* button, selecting from the list displayed in the new window and then clicking on *Apply*. If you select *Close*, the current model will remain selected.
- Unwanted parameter sets can be deleted by loading these into the configuration window and then clicking on the *Delete set* button.
- When *Apply* is selected the current configuration will be saved and loaded into the model. The model configuration window is then closed.
- When *Close* is selected, any changes to the current configuration are not saved and nothing is loaded into the model. This allows users to browse and edit parameter sets without changing the current parameter set loaded into the model.

4.3. Set number of simulations

The value displayed can be edited in the main interface window. Note that increasing the number of simulations will reduce variance in the model but will increase the time taken to complete the simulation. Some paradigms will take longer to complete simply because they involve more conditions, such as when crossing all experimental parameters in the full complex span task.

4.4. Run simulation

Click on the *Run simulation* button and the simulation will start. In most cases, the paradigm will generate one-or-more plots displayed on the screen, as shown in *Figure 11*. The number of figures produced varies dependent on the particular paradigm selected. Further information can be found regarding output in *Appendix A*.

Some paradigms may generate purely data files that require further analysis. *Appendix C* provides information regarding further analysis of model output for specific paradigms.

Running simulations tends to be highly demanding on CPU activity making the computer relatively unresponsive to other activities. This may affect the ability of the computer to re-paint the user interface if this is moved or if you attempt to use the computer for other activities while the simulation is in progress. Simulations appear not to load as heavily on other PC applications but you may experience some delay in these programs.

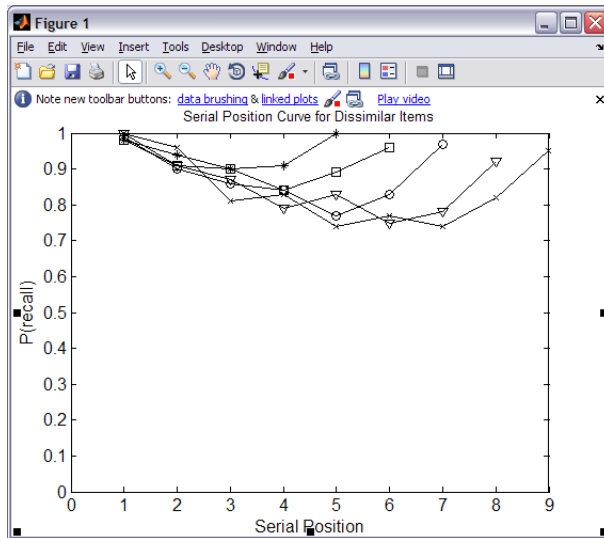


Figure 11. Simulation output plots are displayed as figures on screen.

4.5. Clear figures

The figures displayed on the screen can be selected individually and can be closed by using the Close button in the top right-hand corner of each figure. However, in some cases it may be easier to use the *Clear figures* button. This button functions by stepping through each Figure in turn, starting from Figure 1. If a gap in the sequence occurs through closing a Figure manually, this will interrupt and halt the process. In this case, any remaining figures must be closed manually, clicking on the figure's Close button.

4.6. Running further simulations

It is not necessary to restart the user interface on completion of a simulation or when wishing to run a new model. The simulations can be run again by clicking on the *Run simulation* button. Alternatively parameter configurations can be changed and loaded without needing to restart the application.

4.7. Testing new simulations

Parameters can be changed as desired to test new configurations within the existing parameter set. There are no hard and fast rules regarding changing these. The role of individual parameters are defined within this manual and these can be changed as desired where they are enabled. However, within some of the experiment paradigms, parameter values entered are limited within the simulation software to prevent values being entered that conflict with the execution of the experiment paradigm.

5. Problems and Troubleshooting

1. *When I first start GenWM it is very slow to start.*

GenWM needs to load the Matlab Compiler Runtime (MCR) environment under which GenWM runs each time the computer is started. The extended start up time on first use reflects the fact that MCR is being started and this takes a while to complete. Once set up, MCR should remain present in the background, meaning that re-starting GenWM is substantially quicker.

2. *When I changed parameter values the model failed to work.*

Some paradigms require specific numbers of distractor conditions and similarly processing time and free time conditions. If you have reduced these to fewer conditions than allowed for within the paradigm, the particular code that runs this simulation will be disrupted by this specific change. Limitations on parameter values in the code should prevent such circumstances from arising.

Action. Return parameters to previous settings and re-run simulation. It may also be necessary to restart the simulator application. If the simulation fails to work following changing back to previous parameters, it is recommended that the associated default parameter set is loaded and the simulation tested with this. This should work correctly.

3. *When I use the Clear figures button, some figures remain open.*

This situation may arise as a consequence of manually closing some of the figure windows. This can disrupt the code that determines the number of figures to be closed and hence leaves some figures open. The mechanism by which the code determines the number of figures that are open is described in more detail in sections 3.1 and 4.5.

Action. The remaining figures must be closed manually. If you avoid manually closing the windows the Clear figures button will close all figures displayed by the simulation.

4. *When the simulation completed the figures were not informative.*

Typically this is due to values selected for current parameters or the number of simulations to be performed. If you have sufficient simulations –500 is typically a good starting value – then the problem is liable to be the choice of parameter values. In such circumstances it may be advisable to start again, loading the default parameter set. Typically, it is advisable not to change more than one parameter at any one time.

If you experience such any problems that cause the software to crash or that cannot be resolved, please submit a detailed report for further investigation.

Appendix A. Available experiment paradigms

Details of the available default experiment parameter sets and associated paradigms are provided below. These include details of parameter settings and output. The simulations that are represented in the default parameter sets focus on replicating the simulations reported in Oberauer, Lewandowsky, Farrell, Jarrold, and Greaves (2012). Further information may be found in this paper and in additional papers referenced below.

1. Default experiment parameter set/Complex span (CL and number ops)

The default parameter set defaults to full complex span, comprising four levels of distractor operations, 0, 1, 4, 8; three levels of processing time, 0.3, 0.5, 0.7 and three levels of free time, 0.1, 0.6, 1.2. This paradigm is described in more detail in parameter set 2 below.

2. Full complex span task/Complex span (CL and number ops)

The simulation examines performance manipulating all of the major variables surrounding the processing of distractors in the complex span task using a full factorial design. The major variables are: the number of distractors per item, the processing duration for each distractor and the amount of free time between distractors. There are four levels of distractor operations defined, comprising 0, 1, 4, 8 distractors; three levels of processing time, 0.3, 0.5, 0.7 seconds and four levels of free time, 0, 0.1, 0.6, 1.2. The number of list items is increased up to a maximum of 9 items per condition.

A total of 30 plots are generated by this paradigm. These include:

- span and recall probability with cognitive load
- serial position functions plotting recall probability for each of the three main conditions
- span curves for processing duration against free time for each of the distractor conditions
- error curves plotting omissions, intrusions and transpositions against serial position for the three main conditions.

3. Complex span free time manipulation/Free time complex span manipulation

The simulation conducts a more comprehensive examination of factors that influence free time in the complex span task.

The paradigm breaks down distractor processing, defining separate processing times for the first distractor in a burst and subsequent distractors within the same burst, the initial distractor requiring a substantial increase in processing duration compared with subsequent distractors in the burst.

Processing times are not configurable but are defined internally paired across 3 levels representing durations for easy and hard processing activities; as follows:

	Pair 1		Pair 2		Pair 3	
	Easy	Hard	Easy	Hard	Easy	Hard
First distractor:	1.21	1.45	1.39	1.56	1.38	1.56
Additional distractors:	0.89	1.09	1.02	1.18	1.03	1.19

The configuration window defines a single level distractor operation condition, comprising 4 distractors and two levels for free time, a short duration and a long duration: 0.2, 0.8s.

A sensory motor component is subtracted from processing times and added to free time durations such that processing duration represents a relatively pure estimate of the duration

during which the attentional bottleneck is occupied. The sensory-motor component is also defined internally in steps of 0.1s, from 0.1 – 0.8s.

A total of 10 plots are generated. These include:

- recall probability for easy/hard processing durations plotted against short/long free time durations and plotted for each of the 8 sensory-motor durations, 0.1 – 0.8.
- serial position functions for short-long free time conditions (collapsed across paired processing durations)
- serial position functions for short-long processing time conditions (collapsed across free time durations).

4. Distractor variability

This simulation models the findings reported in Experiment 3 of Lewandowsky, Geiger, Morrell, & Oberauer (2010). The experiment examines recall when the similarity of distractors is manipulated. There are four distractor operation conditions, each condition associated with one similarity condition: a baseline of 0 operations (in effect, simple span); 1 distractor, 4 identical distractors and 3 distractors varying in similarity with a coefficient of 0.5. (The difference in number of operations in conditions 3 and 4 relates to the fact that, experimentally, four identical words and three different words were found to take approximately the same time to articulate).

In the parameter set the within-burst and between-burst similarity is fixed within the code to reflect these four conditions. There are two durations specified for both operation duration and free time. The values defining operation duration specify the time required to process the first distractor and the time required to process any remaining distractors and similarly, for free time duration, the free time following the first distractor and free time following remaining distractors. However, in the default configuration the times specified are not changed between first and subsequent distractors.

Condition:	<u>1</u>	<u>2</u>	<u>3</u>	<u>4</u>
Number of distractors:	0	1	4	3
Within-burst similarity:	1	1	1	0.5
Between-burst similarity:	1	1	1	0.5

Note: Greyed out values indicate values that do not influence model performance for the specified condition.

A single figure is displayed, indicating the serial position curve for each of the four conditions. The figure indicates the influence of different levels of distractor interference on recall in complex span.

5. Item-distractor similarity (proximity)

This simulation examines the similarity of items and distractors drawn from the same broad category, for example, words. These materials are considered as being proximal within the feature space associated with this category of items, sharing key features such as semantic and phonological similarity.

This simulation models findings initially reported in Oberauer, Farrell, Jarrold, Pasiecznik, and Greaves (2012), examining recall of short lists of items when the similarity of pairs of distractors is manipulated with respect to list items. In a high similarity condition, 4-letter, non-word distractors rhymed with associated 4-letter, non-word items. This similarity can be represented in terms of the sequence *Aaa Bbb Ccc Ddd*, in which upper-case letters represent items to be recalled, lower-case letters represent distractors and identical letters (of either case) indicate rhyming non-words. In the low similarity condition, rhyming pairs of distractors were associated with non-rhyming items, as represented by the sequence *Aee Bff Cgg Dhh*. Recall was tested through reconstruction of the list from a set of candidate items. In Experiment 1, the candidate recall set comprised only the original list items, hence only requiring the reconstruction of order [ROO]. In Experiments 2, 3, and 4 the recall set comprised both items, distractors and non-presented lures, hence requiring reconstruction of items and order [ROIO].

Experiment 4 examined similarity when the distractors preceded the associated item. This can be thought of as *Abb Bcc Cdd Daa*. Experiment 3 disrupted the rhyming component of the similarity items, examining similarity when phonological similarity was manipulated about the vowel sound between the first and second pair of letters, for example, *pame, rame, pabe, rabe*. Items and distractors were selected from a pool of these non-word sets. Each item shared the first letter pair with one distractor, e.g., *pa*, and the second letter pair with the other distractor, e.g., *be*. Hence an item might be *pabe*, and the associated distractors would be *pame* and *rabe*. This similarity can be represented by the sequence *Aaa' Bbb' Ccc' Ddd'*.

The simulations examine these reported manipulations in item-distractor similarity and in the candidate sets.

In the simulation, a single level of distractor operations is defined, comprising 2 distractors with a single processing time of 1 second; a single level of free time is set to 0 seconds.

The four experiments are modelled as follows:

(1) High and low similarity distractors immediately follow each item and the candidate set for recall comprises items only [ROO] (Experiment 1 in Oberauer et al., 2012).

(2) High and low similarity distractors immediately follow each item and the candidate set for recall comprises list items and distractors [ROIO] (Experiment 2 in Oberauer et al., 2012).

(3) High and low similarity distractors immediately precede each item and the candidate set for recall comprises list items and distractors [ROIO] (Experiment 4 in Oberauer et al., 2012).

(4) High and low similarity distractors immediately follow each item and the candidate set for recall comprises list items and distractors [ROIO]. Similarity between items and distractors is weakened at the level of constituent phonemes in this simulation, reflected in the use of non-rhyming item-distractor representations (Experiment 3 in Oberauer et al., 2012).

Two figures are produced in this paradigm. The first figure plots serial position curves for the Experiment simulations 1 and 2 and the second figure produces serial position curves for Experiment simulations 3 and 4.

6. Item-distractor similarity (category)

This simulation examines the effects of similarity of items and distractors when they are from the same or from different categories of materials. In this case the categories are words and digits crossed across four conditions in a full factorial design, comprising word or digit items and word or digit distractors.

A single level of distractor operations is defined, comprising 4 distractors; there are four levels of processing time, set to 0.5 seconds for each of the four item-distractor similarity conditions and, similarly, four levels of free time, set to 0.5 seconds in each case.

The simulation generates three plots indicating:

- Serial position curves for proportion correct by position
- Serial position curves for correct item recall (ignoring position)
- Serial position curves for distractor intrusions.

7. Complex span domain manipulation/Cross-domain complex span

This simulation examines the effect of cross-domain interference between items and distractors. The assumption underlying cross-domain interference is that there are seldom domain-pure representations, e.g., representations that involve a purely visual-spatial representation. It is argued that such domain-specific representations include varying degrees of additional representation in other domains such as in the verbal domain.

This simulation examines model predictions when item and distractor representations overlap across domains. The model examines the outcome on recall when varying the proportion to which domain representations overlap.

There is a single level of distractor operations defined, comprising 4 distractors, with three levels of processing time, 0.3, 0.5, 0.7 seconds and four levels of free time, 0, 0.1, 0.6, 1.2 seconds.

The simulation generates a total of 15 plots. These include:

- Span and recall probability functions against cognitive load for varying proportions of domain overlap.
- Serial position curves for varying proportions of domain overlap.
- Omissions, intrusions and transpositions for varying proportions of domain overlap.

8. Individual differences complex span

This simulation applies the model to the individual differences study of Kane, Hambrick, Tuholski, Wilhelm, Payne, and Engle (2004) using simple and complex span tasks from different domains. There are two levels of distractor operations defined, comprising 0, 4 distractors; processing time and free time are both set to 0.5 seconds.

The simulation produces a data file only. Further details regarding analysis of these data are provided in *Appendix C*.

9. Serial recall (similarity manipulation)

The simulation examines recall performance in a simple span task for lists of rhyming, non-rhyming and mixed-list items. The simulation runs as a span task for lists of 5 – 9 items; distractor processing does not impact upon model performance, hence these parameters are not defined within the model.

The simulation produces a total of 8 plots. These include:

- Serial position curves for lists of 5 – 9 items.
- Serial position curves for the different list types for item by position.
- Serial position curves for the different list types for item memory only.
- Omission and intrusion curves for the different list types.
- Transposition gradients for different list lengths and list types.

Appendix B. Available model configurations

Details of the available model configurations are provided below, indicating parameter settings for the specific models. Further details regarding SOB parameters can be found in Oberauer, Lewandowsky, Farrell, Jarrold, and Greaves (2012).

1. Default model parameter set.

The default model parameter set definition for SOB-CS is currently applied in all defined models. Some parameters are not configurable through the user interface when running specific experiment paradigms. These require fixed values that are necessary for the particular model simulation. Further details on these parameters can be found below and in Section 3.3.

2. SOB Luce choice retrieval.

The default parameter set for the SOB-CS model uses a Luce choice retrieval rule in order to match items for recall. The key parameters within the configuration window define the similarity of items for encoding and, with the exception of output interference, these parameters operate over the range 0 – 1, 0 being dissimilar and 1 being identical. In the following section, the role of each parameter is described. Further information on these parameters is also given in Section 3.3.

1. *Item similarity.* The default item similarity parameters are set to 0.5 for between-list similarity and 0.65 for within-list similarity. These values reflect a greater degree of perceived similarity between within-list items compared with between-list items, leading to a greater probability of within-list errors.

2. *Distractor similarity.* Distractor similarity parameters reflect the degree of similarity between distractors within a single burst, i.e. the distractors following an individual list item, and the similarity across bursts, i.e. the similarity of distractors associated with different list items. Default values are set to 0.5 for both within-burst and between-burst.

3. *Item-distractor similarity.* The default item-distractor similarity value, 0.35, reflects the fact that items will generally be more similar to one another than they are to distractors.

4. *Positional overlap.* Positional overlap determines the extent to which positional representations differ in similarity to one another. This impacts upon the probability of transposition errors arising in recall. The default value is set to 0.5, representing a moderate degree of overlap between representations. When there is greater overlap of representations, cueing of the positional code to retrieve an item increases the activation received by neighbouring items, leading to an increase in the probability of recall of a neighbouring item.

5. *Output interference.* Output interference determines the amount of noise present during the retrieval process. The default value, 1.0, represents a moderate level of noise.

Appendix C. Analysis of data files

Output files are stored in the current working directory for the simulator. Details of further analytical procedures for data files are given below.

1. Individual differences complex span.

The output from the simulation of individual differences, "GenKane04.out", is organized as follows:

Every line is one trial, and the variables are listed in the following order:

Subject ID

Domain (1 = verbal, 2 = visual-spatial)

Task (1 to 3 for the 3 different tasks of each kind)

Operation duration (meaningful for complex-span tasks only)

Free time (meaningful for complex-span tasks only)

Number of operations (0 for simple-span tasks, 4 for complex-span tasks)

List length

Trial number

Zero (irrelevant entry included for technical reasons)

Set of recall candidates (1 = only list items, 2 = whole item vocabulary, 3 = list items and distractors, 4 = whole item vocabulary and distractors)

Threshold for recall

Output interference (N_sub_o)

Distinctiveness of items (c)

Encoding rate (R)

Removal rate (r)

Threshold (e, for logistic function translating energy into encoding strength)

Gain (g, for logistic function translating energy into encoding strength)

Item-distractor similarity (s_sub_c)

Overlap of position markers (s_sub_p)

Time for recall of an item

The next 9 entries are the list items coded by their numbers (for lists shorter than 9, the remaining positions are filled with 0)

The final 9 entries are the recalled list items (for lists shorter than 9, the remaining positions are filled with 0)

These data are first processed in SPSS with the syntax "SOBCS.Kane04.sps". The code for this is included within the downloaded zip folder for GenWM. The resultant output is then analyzed in AMOS with the model file "Kane04.4factors.amw". This file is also included in the zip folder.

References

- Barrouillet, P., Bernardin, S., Portrat, S., Vergauwe, E., & Camos, V. (2007). Time and cognitive load in working memory. *Journal of Experimental Psychology: Learning, Memory & Cognition*, *33*, 570-585.
- Kane, M. J., Hambrick, D. Z., Tuholski, S. W., Wilhelm, O., Payne, T. W., & Engle, R. W. (2004). The generality of working-memory capacity: A latent-variable approach to verbal and visuo-spatial memory span and reasoning. *Journal of Experimental Psychology: General*, *133*, 189-217.
- Lewandowsky, S., Geiger, S. M., Morrell, D., & Oberauer, K. (2010). Turning simple span into complex span: Time for decay or interference from distractors? *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *36*, 958-978.
- Oberauer, K., Farrell, S., Jarrold, C., Pasiiecznik, K., & Greaves, M. (2012). Interference between maintenance and processing in working memory: The effect of item-distractor similarity in complex span. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *38*, 665-685.
- Oberauer, K., Lewandowsky, S., Farrell, S., Jarrold, C., & Greaves, M. (2012). Modeling working memory: An interference model of complex span. *Psychonomic Bulletin & Review*, *19*, 779-819.