SecondOrder

A Whisker client

by Rudolf Cardinal

www.whiskercontrol.com

Distributed by Campden Instruments Ltd (www.campden-inst.com)

SecondOrder

Printed: December 2020 in Cambridge, UK

Creator (Whisker)

Rudolf N. Cardinal

Design and Programming (Whisker)

Rudolf N. Cardinal

Michael R. F. Aitken

Legal Advisor (CUTS)

Adjoa D. Tamakloe

Sales (Campden)

Julie Gill

Contacting the authors:

For information about Whisker, visit http://www.whiskercontrol.com/.

If you have sales enquiries about Whisker, contact Campden Instruments Ltd at http://www.campden-inst.com/.

If you have comments or technical enquiries that cannot be answered by the sales team, contact the authors:

Rudolf Cardinal (rudolf@pobox.com)
Mike Aitken (m.aitken@psychol.cam.ac.uk)

Table of Contents

Foreword

WARNING

Whisker is a system designed for research purposes only, and should never be used to control medical apparatus or other devices that could endanger human life.

DISCLAIMER

The authors, copyright holders, and distributors disclaim all responsibility for any adverse effects that may occur as a result of a user disregarding the above warning.

1 SecondOrder

1.1 About SecondOrder

Purpose

Second-order schedules of reinforcement.

In a second-order schedule of reinforcement, the subject must complete a certain simple schedule (fixed ratio, fixed interval, variable ratio, etc.) to obtain a stimulus. It works on a higher (second-order) schedule for the actual reinforcer; for example, it might obtain a stimulus+drug compound once it has earned five stimuli.

The SecondOrder client is designed to use a {left lever, right lever, nosepoke} response to earn stimuli on an {FR, FI, VR} schedule, and earn reinforcement from a {liquid dipper, infusion pump}on a second-order {FR, FI, VR} schedule. It records all responses, and also records locomotor activity. The schedule, stimulus and reinforcement parameters are all configurable.

(IVSA; intravenous self-administration.)

Software requirements

Requires Whisker v2.0 or greater.

Data storage

- Text-based output to disk.
- ODBC data storage to a database (supplied).

Author

Rudolf Cardinal (rudolf@pobox.com).

Copyright

Sample publication using this form of task

- Arroyo M, Markou A, Robbins TW & Everitt BJ (1998). Acquisition, maintenance and reinstatement of intravenous cocaine self-administration under a second-order schedule of reinforcement in rats: effects of conditioned cues and continuous access to cocaine. Psychopharmacology 140: 331–344. [Original article, with references to earlier work using second-order schedules.]

Version history

- See **www.whiskercontrol.com/version_tracker/SecondOrder.txt**

1.2 Required devices

The program requires to claim devices in groups named **box0, box1, box2...** with device names as listed below in bold:

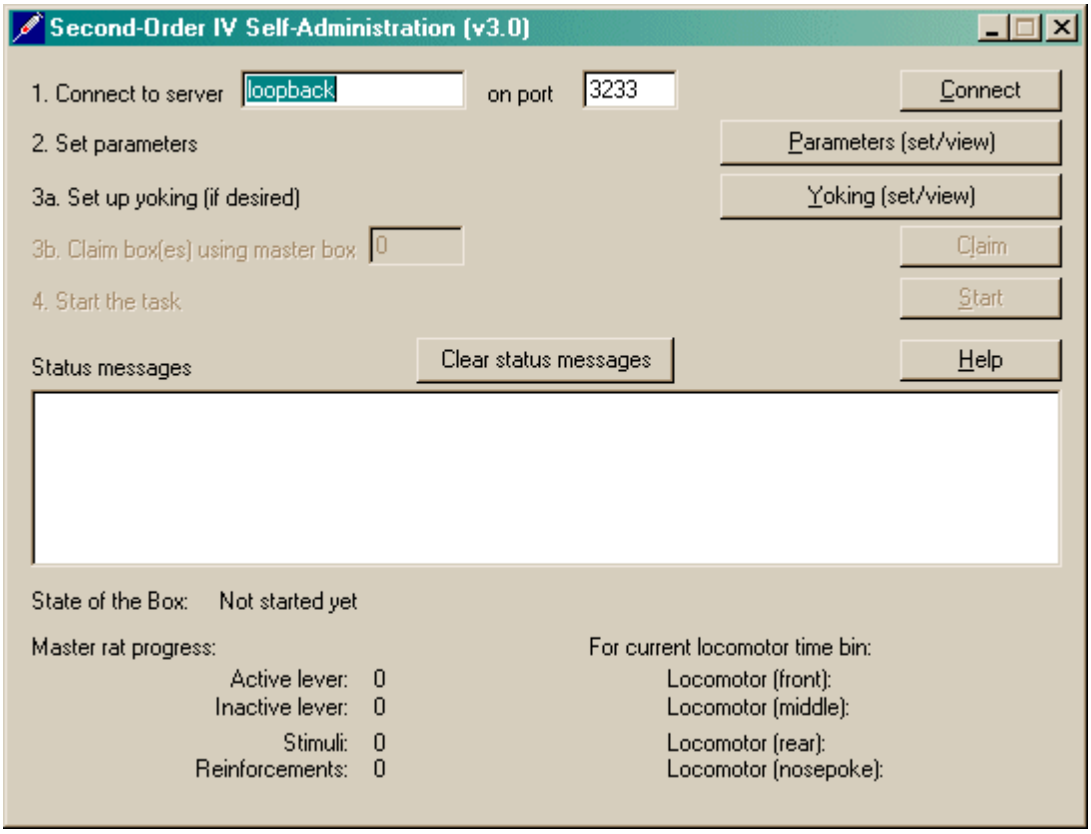
```
# Box 0 definition
```

```
# inputs
line        0           box0    NOSEPOKE
line        1           box0    LEFTLEVER
line        2           box0    RIGHTLEVER
line        3           box0    LOCOBEAM_FRONT
line        4           box0    LOCOBEAM_MIDDLE
line        5           box0    LOCOBEAM_REAR

# outputs
line        24          box0    HOUSELIGHT
line        25          box0    LEFTLIGHT
line        26          box0    RIGHTLIGHT
line        27          box0    TRAYLIGHT
line        28          box0    PUMP
line        29          box0    DIPPER
line        30          box0    LEFTLEVERCONTROL
line        31          box0    RIGHTLEVERCONTROL


# Box 1 definition

# inputs
line        6           box1    NOSEPOKE
line        7           box1    LEFTLEVER
line        8           box1    RIGHTLEVER
line        9           box1    LOCOBEAM_FRONT
line        10          box1    LOCOBEAM_MIDDLE
line        11          box1    LOCOBEAM_REAR

# outputs
line        32          box1    HOUSELIGHT
line        33          box1    LEFTLIGHT
line        34          box1    RIGHTLIGHT
line        35          box1    TRAYLIGHT
line        36          box1    PUMP
line        37          box1    DIPPER
line        38          box1    LEFTLEVERCONTROL
line        39          box1    RIGHTLEVERCONTROL

# ... etc.
```

Please ensure that these devices are available and listed in the device definition file in use by the server. (The snippet above shows an extract from a typical definition file.)

1.3 Using the task

When you run the task, the main screen looks as follows:

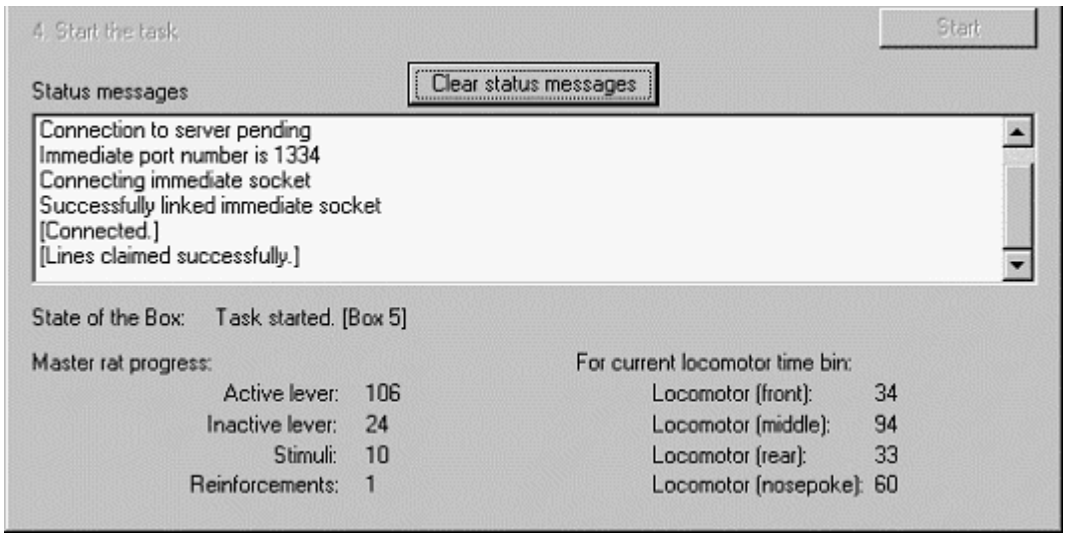


Connect to a Whisker server (which must be running!).

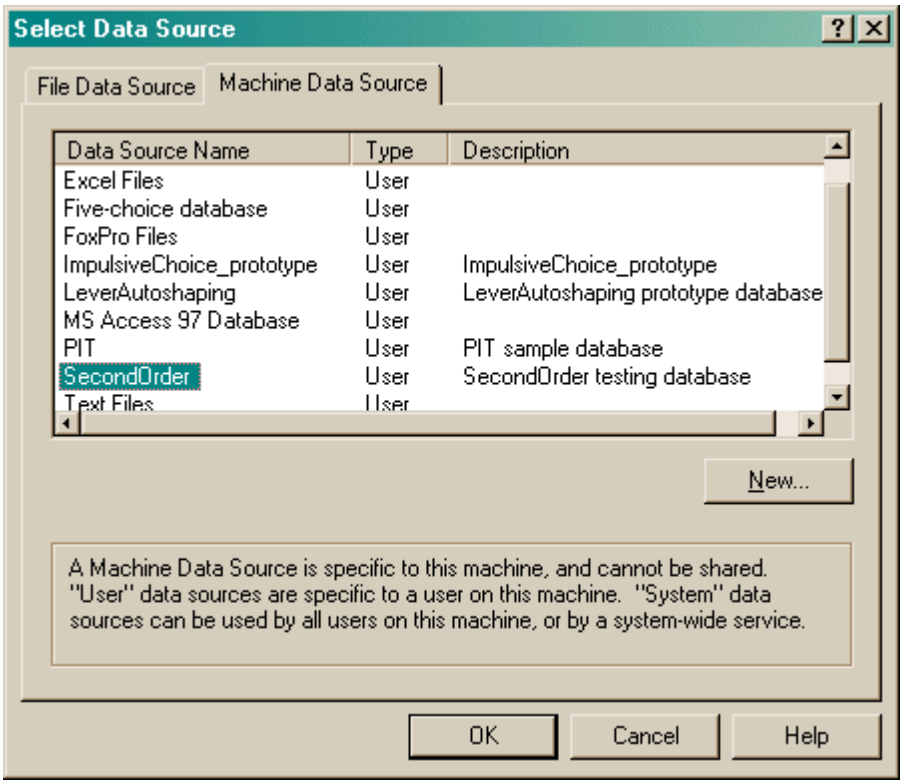
Choose the box you wish to use and claim it. Boxes are numbered from ZERO, not one.

Set up the parameters for your task. Once that's done, the traffic lights will turn amber.

When you are ready, press *Start* to begin the task. It looks like this when it's running:



When the task finishes, it saves data to disk and pops up a new dialogue box for you to select a database to store the data to. (The data sources are configured under *Control Panel* → *ODBC*.) If you previously specified an ODBC data source in the parameters, that data source is used automatically and you will only see a dialogue box if something goes wrong and the program needs your input.



That's it. The program returns to the main window and displays **'FINISHED'** in its title bar. You can close it.

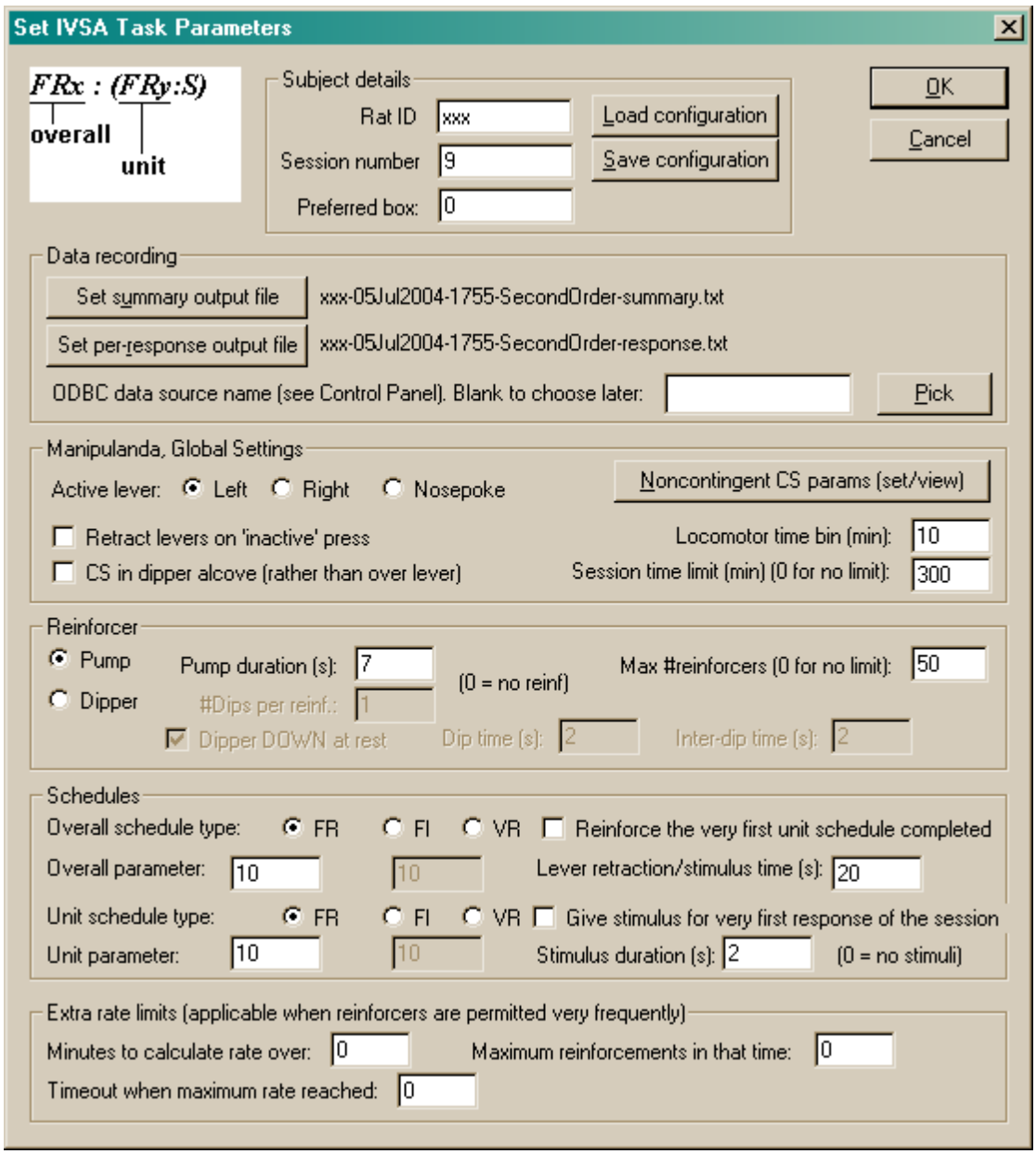
If the program reports an error, another client may be using the same database (see the Technical Note a few pages back) or there may be a problem with the ODBC data source. If the former, then wait a few seconds and click *Retry*. Otherwise, reconfigure the ODBC source - in an emergency, you can always create a new one on the spot using the New button (visible in the screenshot above).

Note that the moment at which the SecondOrder client stores data in a database is quite CPU-intensive.

Note that the program goes to considerable lengths to ensure you never lose data. When-ever you abort the program, it tries to store all the data it has collected in as many places as possible (i.e. on disk; in a database). You should find it hard to abort it accidentally, as it complains loudly when you try.

1.4 Parameters

The parameters dialogue box looks like this:



I hope most of it is obvious.

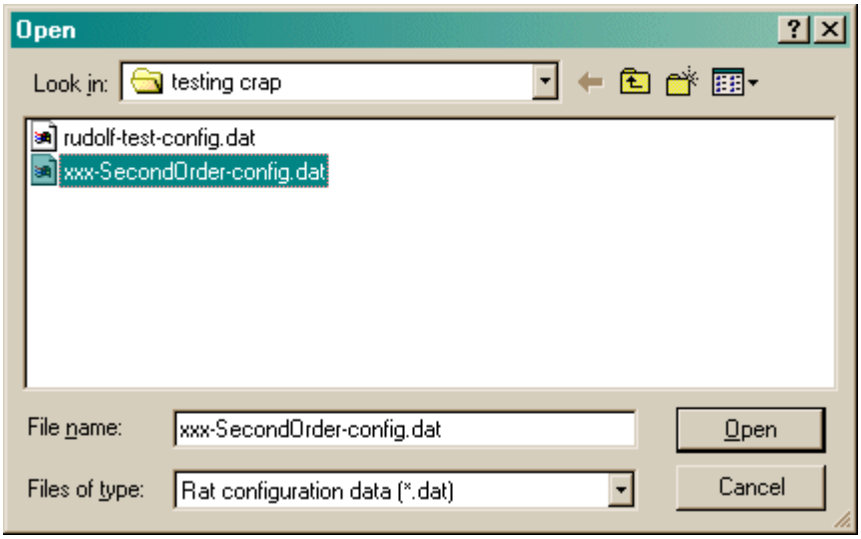
- You should name your subject and assign it a session number. You can save the whole configuration in a configuration file. When you load that configuration file the next time, the program increments the session number by one (and auto-saves any changes you make to the configuration). This makes it easy to take subjects through a training programme – the configuration file remembers where they are at any given moment.
- When you load a configuration file, the program guesses suitable summary-output (human-readable) and per-response-output (database format) filenames. By default, it stores these files in the same directory as the configuration file. **These filenames must be specified before the task will start.** If you try to start two copies of the program using the same filenames, the task will not start.
- If you use a dipper, you can either have the dipper down at rest, and come up one or several

times to deliver liquid so there's no liquid available when it finishes, or you can have the dipper up and dip down to collect liquid so the liquid remains available until it's collected. You can specify the timing of the dips: for example, you can specify that the dipper be down at rest, come up for 2 s (*dip time*), go down for 0.5 s (*inter-dip time*) and then repeat this cycle a total of three times (*#dips per reinforcement*).

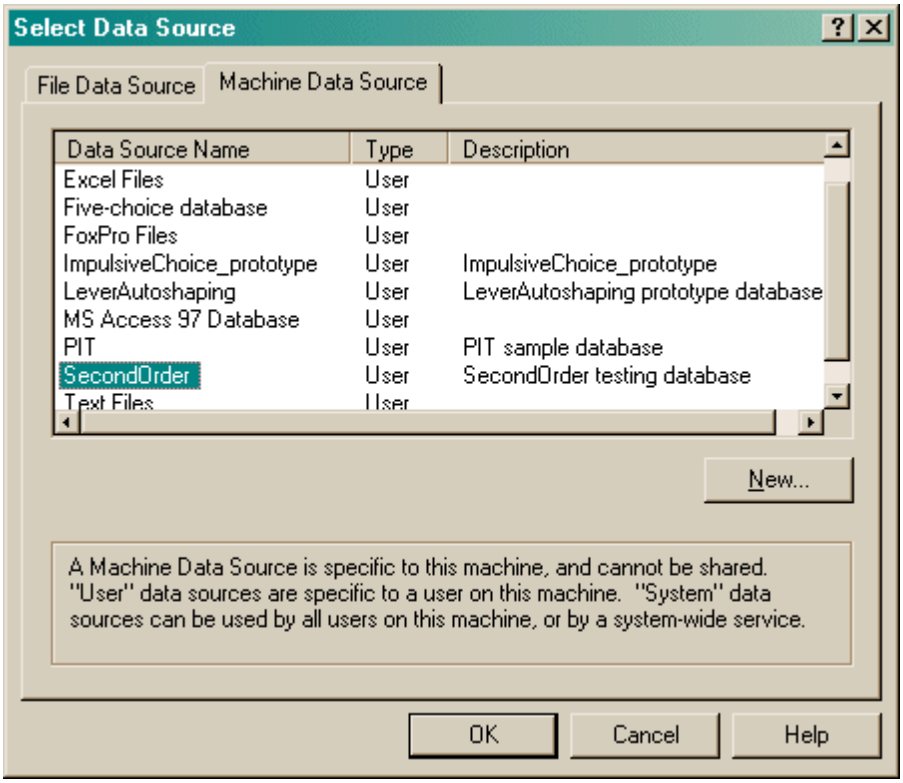
The example above (overall schedule FR10, unit schedule FR2) means that the subject will get a stimulus for every 2 responses, and reinforcement after every 10 stimuli.

Click OK when you're done.

To load a previous subject configuration file, click *Load*.

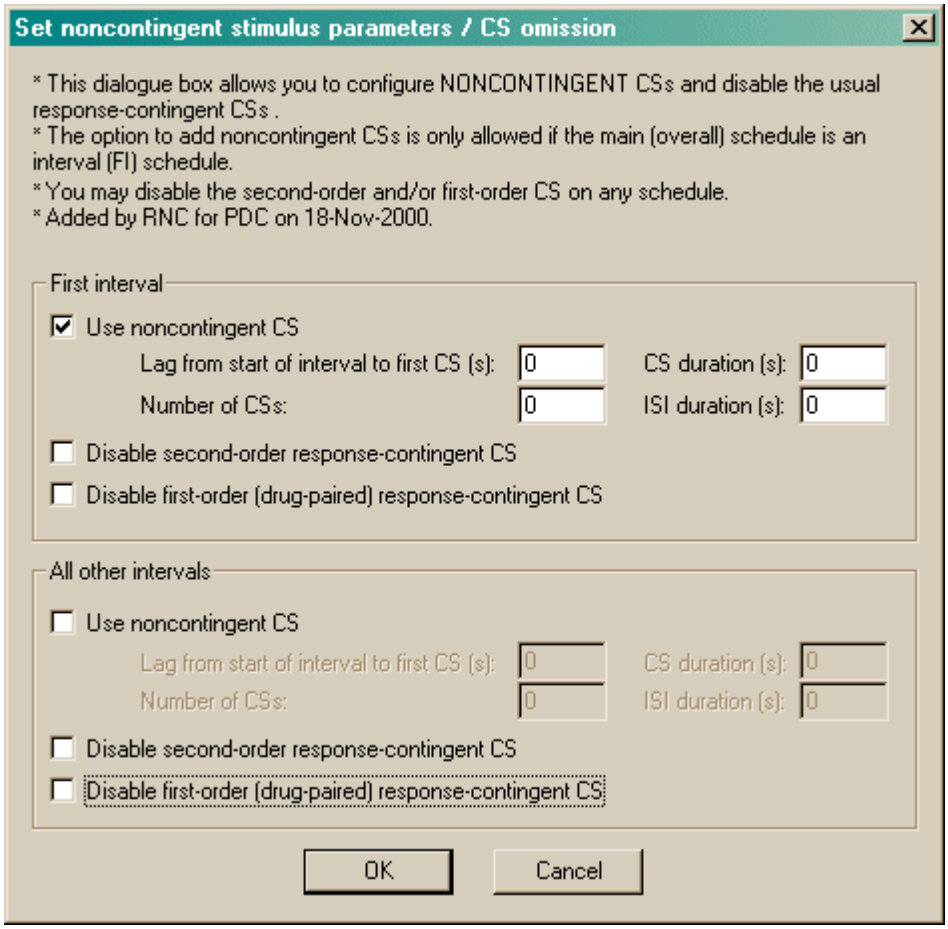


To pick an ODBC database **in advance** of finishing, click *Pick* and you will be offered the ODBC Data Source picker (below). Your choice will be recorded and will apply to this subject from now on (or until you specify a different source).



If you don't specify an ODBC data source now, or you delete the value in the "ODBC data source name" box, you'll be asked to choose when the task ends (and that choice will only apply to the session in progress).

To set up **noncontingent CS presentation** on top of a normal second-order schedule, click *Noncontingent CS params (set/view)*:



1.5 Specifications

Program

- First version finished on 14 Jan 2000, pending examination of a few other quirks of the printout from the Paul Fray second-order program (i.e. do we need these features?).
- Implemented as a dialogue-based Windows program using the Whisker C++ client library. Provides status information to the server to enable its progress to be monitored from a different location (using the WhiskerStatus tool)
- Maintains rat configuration data files.
- Produces a summary file containing the most useful information about the session, and a response file in comma-delimited format containing a time-stamped record of every response made, with information about whether that response produced a stimulus or a primary reinforcer.
- Stores a session's complete data set directly in a database when the run is finished. This feature uses ODBC (Open Database Connectivity, a world standard for communicating with databases). An appropriately-configured database in Microsoft Access 97 format is supplied with SecondOrder.
- Adapted for Whisker v2 server-side device definition files.

Schedules

- Supports FI/FR/VR for both unit and overall schedules. VR schedules are specified with a flat probability distribution function (i.e. you specify the minimum and maximum value, rather than the mean).
- For overall-FI schedules, calculates index of curvature and proportions of responding in each quarter of the interval.
- Supports left/right lever and nosepoke manipulanda, with variable location CS.
- Inactive lever can optionally cause both levers to retract for a timeout period.
- Supports pump/dipper reinforcement, with configurable multi-dip reinforcement, and timeout periods at reinforcement.
- All stimulus and reinforcement timing parameters can be configured.
- Records locomotor activity on up to three beams in bins of configurable size.
- Session time and number of reinforcements can be capped.

Schedule and calculation details

Responding during a stimulus is counted towards totals and index-of-curvature plotting, but does not contribute towards the unit schedule in progress.

Responding during a reinforcement timeout (which is only possible for nosepokes and non-retractable levers) is recorded in the response log, and contributes towards the response totals, but does not contribute to progress on the schedule or to index-of-curvature data, as these responses are considered 'outside' the overall FI (the overall FI is considered to start at the end of this timeout period).

Inner workings of the program

I think it would be fair to say that this is not a simple program by experimental psychology standards, though this is due more to the user interface and the communication with the database than the complexity of the task. However, I tried to write it in as comprehensible a manner as I could. The source code is supplied, and I encourage you to explore it, though it may be easier to start with simpler clients (see the *Whisker Programmer's Guide*). The central function, *CSecondOrderTask::IncomingEvent(CString strEvent)*, in SecondOrderTask.cpp, does all the work relating to the task, and it's quite simple.

1.6 Setting up a database

Why?

The SecondOrder client can store its data in a relational database, which is a Very Good Idea and can save you a great deal of time and effort. It needs a database containing tables with particular names and layouts. Rather than describe this in exhaustive detail, I've supplied a sample database with Whisker, for use with **Microsoft Access 97** or later. It's on the menu (shown earlier).

However, I strongly recommend you make a *copy* of this database and do not use the original that I have supplied; firstly, because then you won't lose any data if you uninstall Whisker (this procedure deletes the supplied database), and secondly, because you can make multiple copies for different purposes (perhaps different experimenters).

Configuring an ODBC data source

SecondOrder chooses and finds its database using the Open Database Connectivity (ODBC) system, which allows programs to talk to databases from a wide variety of manufacturer (i.e. not just Access). You must set up your copy of the database as a *data source*.

ODBC must be installed on the computer. (It is part of Windows 2000 and installed by all database software.)

1. Configure a data source name (DSN) to point to the database(s) you want to use. Run *Start → Settings → Control Panel → ODBC Data Sources*.
2. You may add your data source as a "User DSN", which will be visible only to the user you configure it with, or a "System DSN", which will be visible to all users of the system. (You could probably use a File DSN as well, but never mind that.)
3. Choose Add, then choose a database driver (e.g. "Microsoft Access Driver").
4. Then define a data source name (DSN), which should have no spaces in it (e.g. Rudolf_SecondOrder).
5. Click *Select* to browse your directories and choose the database that this DSN will connect to.
6. Click *OK*.

Good. Now when you run SecondOrder, this DSN will be among the list you can choose from when your session is complete.

Technical notes

Record-locking. When several copies of SecondOrder are running, there is a chance that they both try to store their data in the same database at (almost) the same time. This creates a problem: the first copy will lock the tables while it stores new data, during which time the second copy can't gain access. If this happens, the second client will report an error, and the user can (and is recommended to) click Retry to have another go.

Configuring MS Access database parameters. However, we can reduce the likelihood of an error being reported by making the database system try for longer itself. In any Access database that you wish to configure, choose *Tools → Options → Advanced*. You can then specify the *Number of Update Retries* (default 2, you want 10) and the *ODBC Refresh Interval* (default 250 ms, you want 1000 ms). This has already been done for the second-order database.

1.7 Personalized shortcuts

Creating a personalized shortcut to SecondOrder

You can create several shortcuts to the SecondOrder program, and have each shortcut start in a different directory.

1. Find the SecondOrder.exe program and create a shortcut to it, or copy the shortcut that is installed in the Start menu (by right-clicking the Windows Start button and choosing *Open All Users*.
2. Once you have a copy of the shortcut, right-click it and choose *Properties → Shortcut → Start In*, then type in a directory name.

This may be useful for keeping different people's experimental data separate.

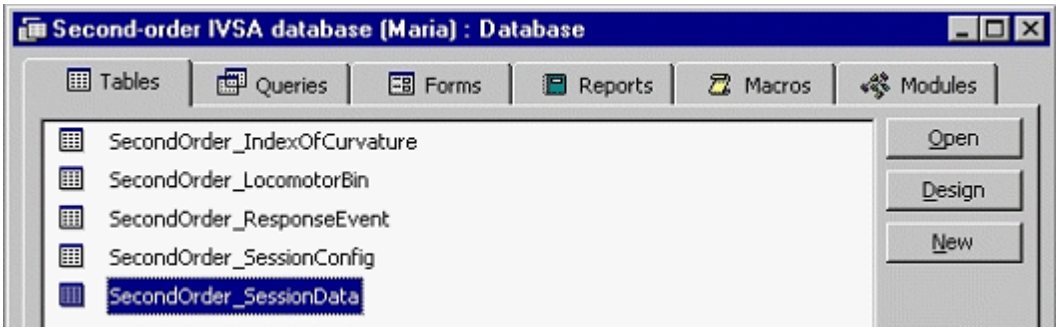
1.8 Using the data

A full data set is stored in the database, including the configuration each session was run under, the exact time of each response and locomotor activity data, together with useful summary information (total number of responses, indices of curvature and quarter-lives for interval schedules).

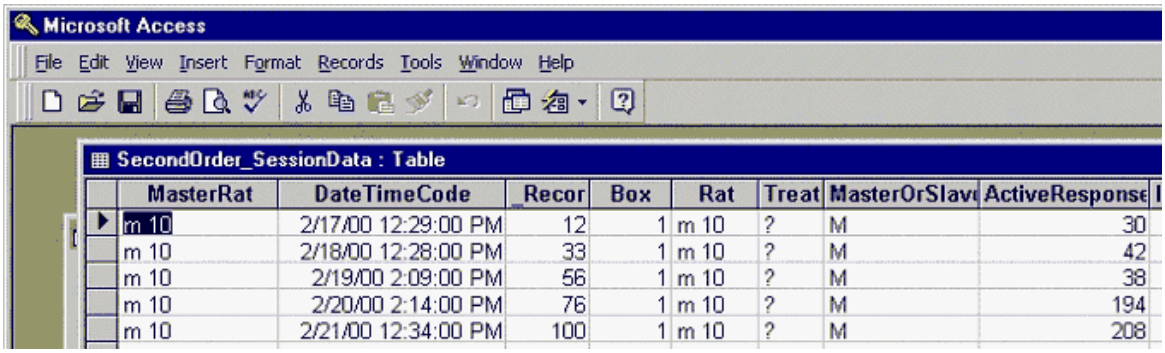
You can design queries to extract the data in any conceivable way (see also Part VII).

For your convenience, there are queries built into the database that will extract data in a suitable form to be copied directly in to the **cumulative record** spreadsheet (in Excel format) that is also supplied with Whisker. This gives you cumulative records and actigrams.

The database looks like this:



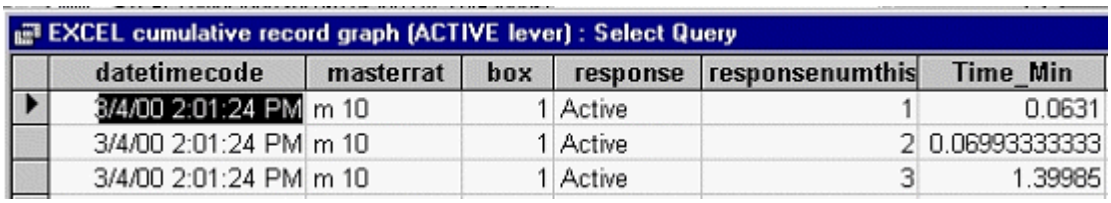
Here's one of the five tables, containing some data:



MasterRat	DateTimeCode	_Recor	Box	Rat	Treat	MasterOrSlav	ActiveResponse
m 10	2/17/00 12:29:00 PM	12	1	m 10	?	M	30
m 10	2/18/00 12:28:00 PM	33	1	m 10	?	M	42
m 10	2/19/00 2:09:00 PM	56	1	m 10	?	M	38
m 10	2/20/00 2:14:00 PM	76	1	m 10	?	M	194
m 10	2/21/00 12:34:00 PM	100	1	m 10	?	M	208

If you want to obtain a cumulative record graph, have a look at the *Queries*. One of them is named **EXCEL cumulative record graph (ACTIVE lever),** and there's another for the inactive lever. Open up the query. You'll be asked for the date/time code of your session of interest (see above; e.g. *3/4/00 2:01:24 PM*), the name of the *master* rat and the box number of the rat you're interested in. **(Note this well, if you're using yoked rats:** to get a record of the yoked rat, enter the name of the master rat and the box number of the yoked rat.)

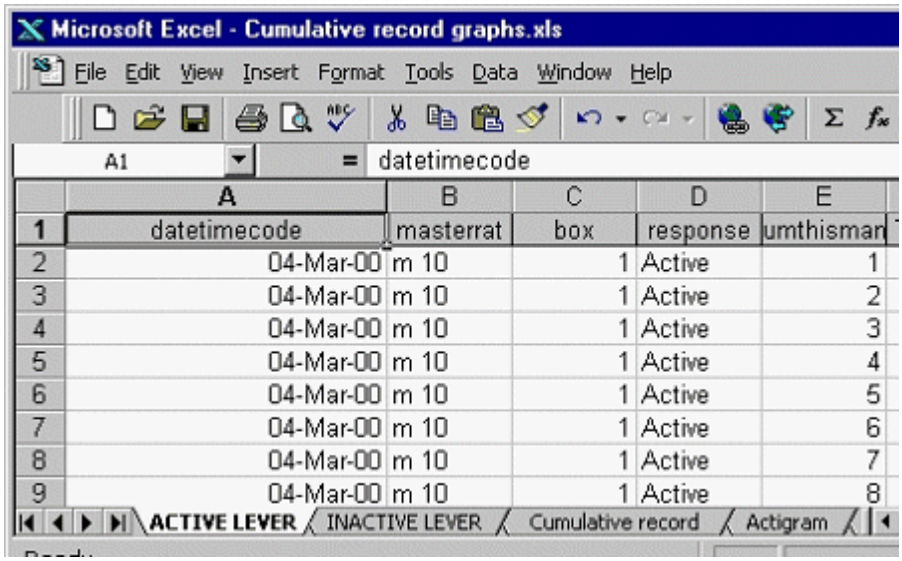
The query then runs and produces something like this.



datetimecode	masterrat	box	response	responsenumthis	Time_Min
3/4/00 2:01:24 PM	m 10	1	Active	1	0.0631
3/4/00 2:01:24 PM	m 10	1	Active	2	0.06993333333
3/4/00 2:01:24 PM	m 10	1	Active	3	1.39985

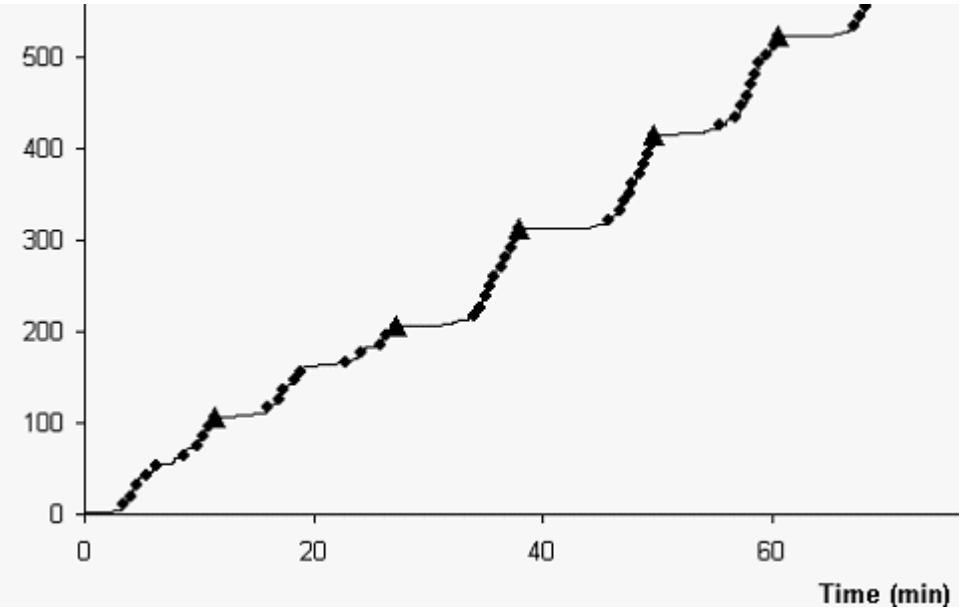
Copy the whole thing (Ctrl-A to select the whole query, Ctrl-C to copy it).

Open up the supplied Excel spreadsheet. Put the cursor in the top-left cell of the ACTIVE LEVER worksheet and press Ctrl-V to paste in your data.

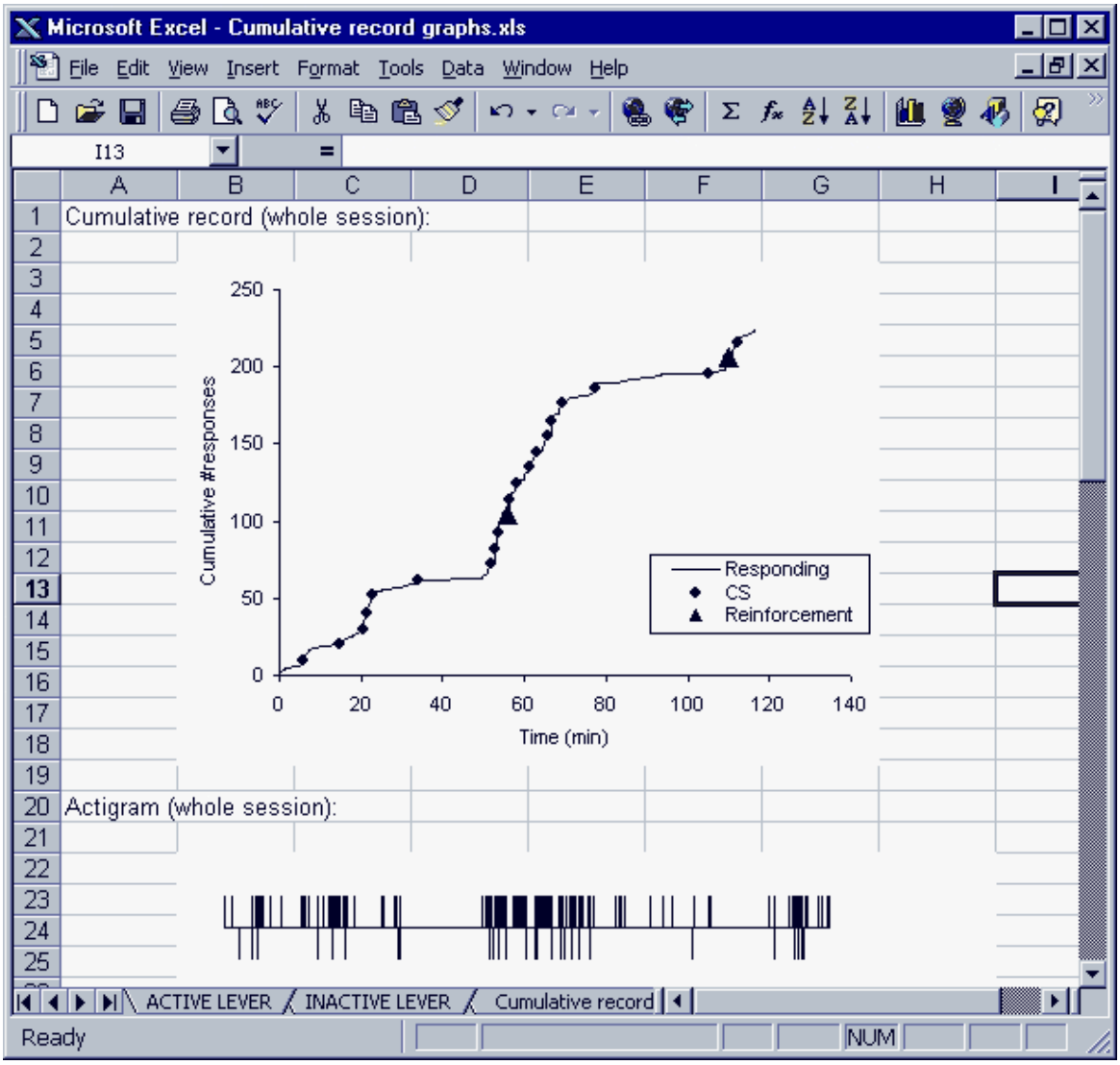


	A	B	C	D	E
1	datetimecode	masterrat	box	response	umthisman
2	04-Mar-00	m 10	1	Active	1
3	04-Mar-00	m 10	1	Active	2
4	04-Mar-00	m 10	1	Active	3
5	04-Mar-00	m 10	1	Active	4
6	04-Mar-00	m 10	1	Active	5
7	04-Mar-00	m 10	1	Active	6
8	04-Mar-00	m 10	1	Active	7
9	04-Mar-00	m 10	1	Active	8

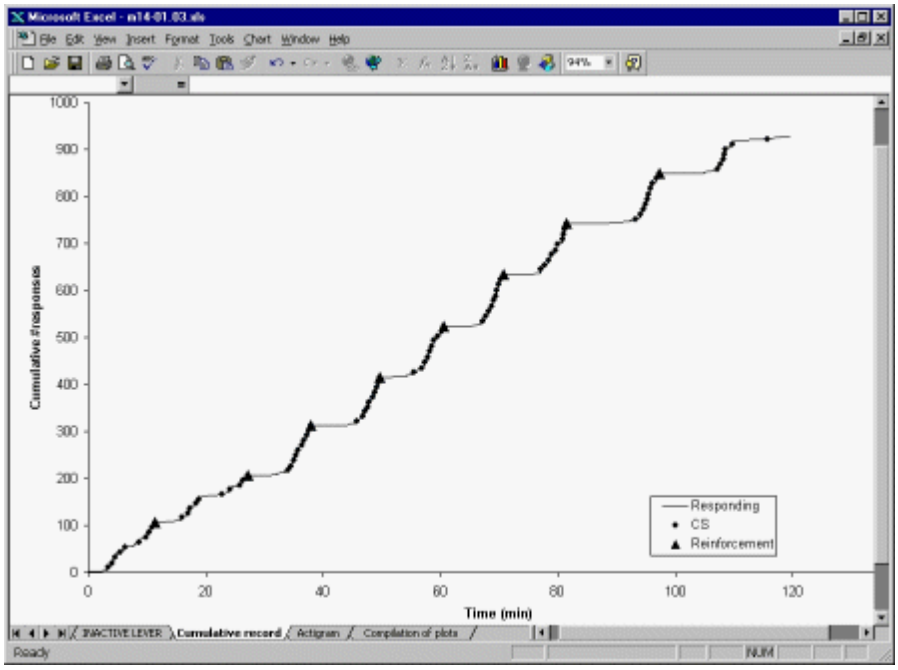
Here's part of the cumulative record worksheet that results:



Actigrams are also generated. If you want to add inactive lever responses to the actigram, just run the other Access query and paste the results into the INACTIVE LEVER worksheet.



Here's a nice example of scalloped responding:



Index

- S -