How to make the protocols

Table of contents

1 How to make the protocols

P1 protocols are located within the repository within the `protocols` folder. However, a user may wish to make a new modified version of P1, so I will outline the steps to create a version of P1 from scratch below.

1.1 Making P1 from scratch

In order to run P1, there are a few steps that need to be done manually. Briefly, the user needs to:

- Run `src.stimulus_generation.generate_protocol1_stimuli`. This creates all of the pattern files and position functions that are needed.
- Open the `G4_experiment_designer` and compile the experiment.
- Save the experiment with the folder `protocols/LHS` or `protocols/RHS` etc.

- Make a copy of the file `protocols/processing_settings.mat` and put it within the experiment folder.

First, you need to make the patterns and functions that will be used in the protocol. This is done by running the function `src/stimulus_generation/generate_protocol1_stimuli.m`. Within this function you can modify different parameters to suit your needs.

- Display settings: The range of pixels within which to present the stimuli is defined within this function by `disp_x1`, `disp_x2`, `disp_y1` and `disp_y2`. For the LHS version of the protocol, the pixel limits start with `disp_x1` = 17. This is because we skip the leftmost column (from the fly's perspective, right most column when looking at the panels from the front). The LED arena is also 3 rows high, so the y range is from 1 to 48, since there are 16 pixels per panel.
- Stimulus parameters:

A number of metrics are assigned to a structure called `params` which is the input to the function `generate_stimulus` to create the patterns and position functions.

- `params.px_intensity` - The pixel intensity values to be used for the stimuli. This parameter should be a [1 x 3] array corresponding to [bkg_color, off_color, on_color] and should be three integer values between 0 and 15. 0 means the LED is off and 15 is the maximum intensity.
- `params.flash_sz` - The size of the flash (`flash_sz` x `flash_sz`) in pixels.
- `params.root_dir` - Directory to create the `patterns` and `functions` folders for saving the files.
- `params.px_rng = [disp_y1, disp_y2, disp_x1, disp_x2]` - Pixel range of screen within which to present the flashes - [row_start, row_end, col_start, col_end]
- `params.overlap` - Percentage of overlap of the grid within which the flashes are presented. If `overlap` = 0, then there is no overlap between flash positions.
- `params.bkg_frame` - The background frame = frame 1 in the pattern.
- `params.interval_dur` - The duration of interval background screen between flashes (seconds).
- `params.flash_dur` - The duration of each flash stimulus (seconds).
- `params.on_off` - Whether to present bright 'on', dark 'off' or both 'both' flashes. For P1 this should be set to 'both'.

This `params` structure is then fed into the function `generate_stimulus` which then creates the patterns and postion functions required for P1.

The static position function is also made within `src/stimulus_generation/generate_protocol1_stimuli.m`. There is an additional parameter that can be modified: `t_bkg_s`, which is the duration for which the background frame is shown at the beginning of each repetition. This duration is in seconds. This position function will be saved within the same 'Functions' folder as the other position functions that are generated within `generate_stimulus`.

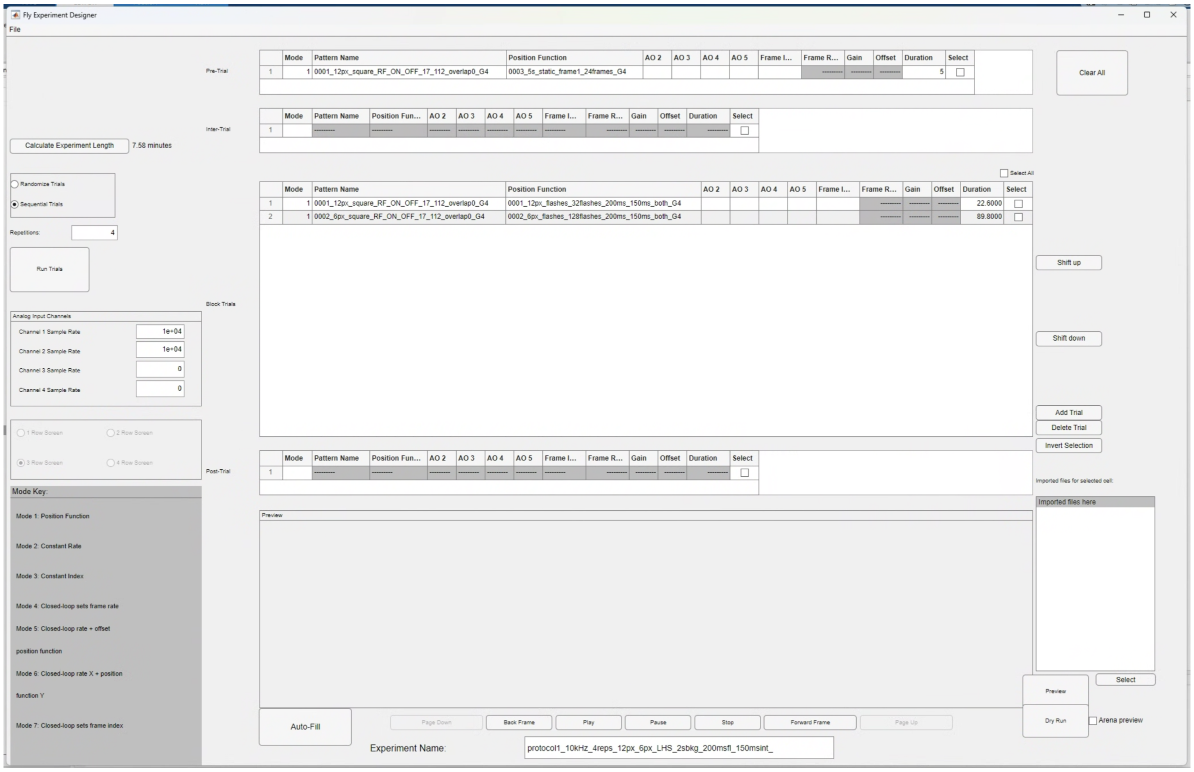
- Creating the protocol in the G4_experiment_designer

After running src/stimulus_generation/generate_protocol1_stimuli.m you should now have two pattern files within protocols/LHS/patterns etc. and three position functions within protocols/LHS/functions etc. You are now ready to assemble these components into a protocol. To do this:

- Run the code: G4_experiment_designer in the MATLAB Command Window and the GUI window will appear.
- Load the patterns and position functions into the experiment designer

File-Import-Folder - navigate to the folder where you saved the patterns and position functions that you just made. All patterns and position functions that are saved in this folder will be imported into the GUI. You will have to do this twice, once for the patterns and once for the position functions.

- Set the "pre" stimulus with pattern 0001 and the static position function (0003).
- Set condition 1 as pattern 0001 and position function 0001.
- Set condition 2 as pattern 0002 and position function 0002.
- Set the number of repetitions to 4.
- Set the acquisition rate for the first two channels (voltage and frame position) as 10,000 Hz.


Fly Experiment Designer
File
Clear All
Calculate Experiment Length
Run Trials
Shift up
Shift down
Add Trial
Delete Trial
Invert Selection
Imported files for selected cell:
Imported files here
Select
Preview
Dry Run
Auto-Fill
Page Down
Back Frame
Play
Pause
Stop
Forward Frame
Page Up
Mode Key:
Mode 1: Position Function
Mode 2: Constant Rate
Mode 3: Constant Index
Mode 4: Closed-loop sets frame rate
Mode 5: Closed-loop rate + offset position function
Mode 6: Closed-loop rate X + position function Y
Mode 7: Closed-loop sets frame index

7.58 minutes

○ Randomize Trials
● Sequential Trials

Repetitions: 4

Analog Input Channels

Channel 1 Sample Rate: 1e+04
Channel 2 Sample Rate: 1e+04
Channel 3 Sample Rate: 0
Channel 4 Sample Rate: 0

○ 1 Row Screen ○ 2 Row Screen
● 3 Row Screen ○ 4 Row Screen

Pre-Trial

	Mode	Pattern Name	Position Function	AO 2	AO 3	AO 4	AO 5	Frame I...	Frame R...	Gain	Offset	Duration	Select
1	1	0001_12px_square_RF_ON_OFF_17_112_overlap0_G4	0003_5s_static_frame1_24frames_G4						---------	---------	---------	5	□

Inter-Trial

	Mode	Pattern Name	Position Fun...	AO 2	AO 3	AO 4	AO 5	Frame I...	Frame R...	Gain	Offset	Duration	Select
1		---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	□

□ Select All

Block Trials

	Mode	Pattern Name	Position Function	AO 2	AO 3	AO 4	AO 5	Frame I...	Frame R...	Gain	Offset	Duration	Select
1	1	0001_12px_square_RF_ON_OFF_17_112_overlap0_G4	0001_12px_flashes_32flashes_200ms_150ms_both_G4						---------	---------	---------	22.6000	□
2	1	0002_6px_square_RF_ON_OFF_17_112_overlap0_G4	0002_6px_flashes_128flashes_200ms_150ms_both_G4						---------	---------	---------	89.8000	□

Post-Trial

	Mode	Pattern Name	Position Fun...	AO 2	AO 3	AO 4	AO 5	Frame I...	Frame R...	Gain	Offset	Duration	Select
1		---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	□

Preview

□ Arena preview

Experiment Name: protocol1_10kHz_4reps_12px_6px_LHS_2sbkg_200msfl_150msint_

Figure 1: Screenshot of the G4_Experiment_Designer used to make a version of protocol 1 with 6 and 12 pixel square flashes

- Set the name of the protocol to something logical. For instance, one of the protocols within the repository is named: `protocol1_4reps_12px_6px_LHS_2sbkg_200msfl_50msint`.
- Save the protocol.

You should then see that a folder has been made within the `protocols/LHS` folder with the name of the protocol and within that folder there should be a `.g4p` file with the same name as the folder.

- Make a copy of the file `protocols/processing_settings.mat` and paste it within the newly created experiment folder `protocols/LHS/protocol1_....`

Within the `processing_settings.mat` file, whenever a new project is created the `settings.path_to_protocol` variable must be updated to the path of the `.g4p` file within the new experiment folder. For P1 protocols, it is impoprtant to make sure that the correct 'ephys processing' variables are set too.

`settings.path_to_protocol` - must be set to the path to the protocol, including the full name of the `.g4p` file (with the extension).

The following variables should be set for a P1 protocol:

- `settings.is_ephys_grid = 1;`
- `settings.neutral_frame = 1;`
- `settings.grid_columns = [8, 16];`
- `settings.grid_rows = [4, 8];`

If a variation of P1 is made that only uses one pixel size, then the values for `settings.grid_columns` and `settings.grid_rows` will be integers and not tuples.

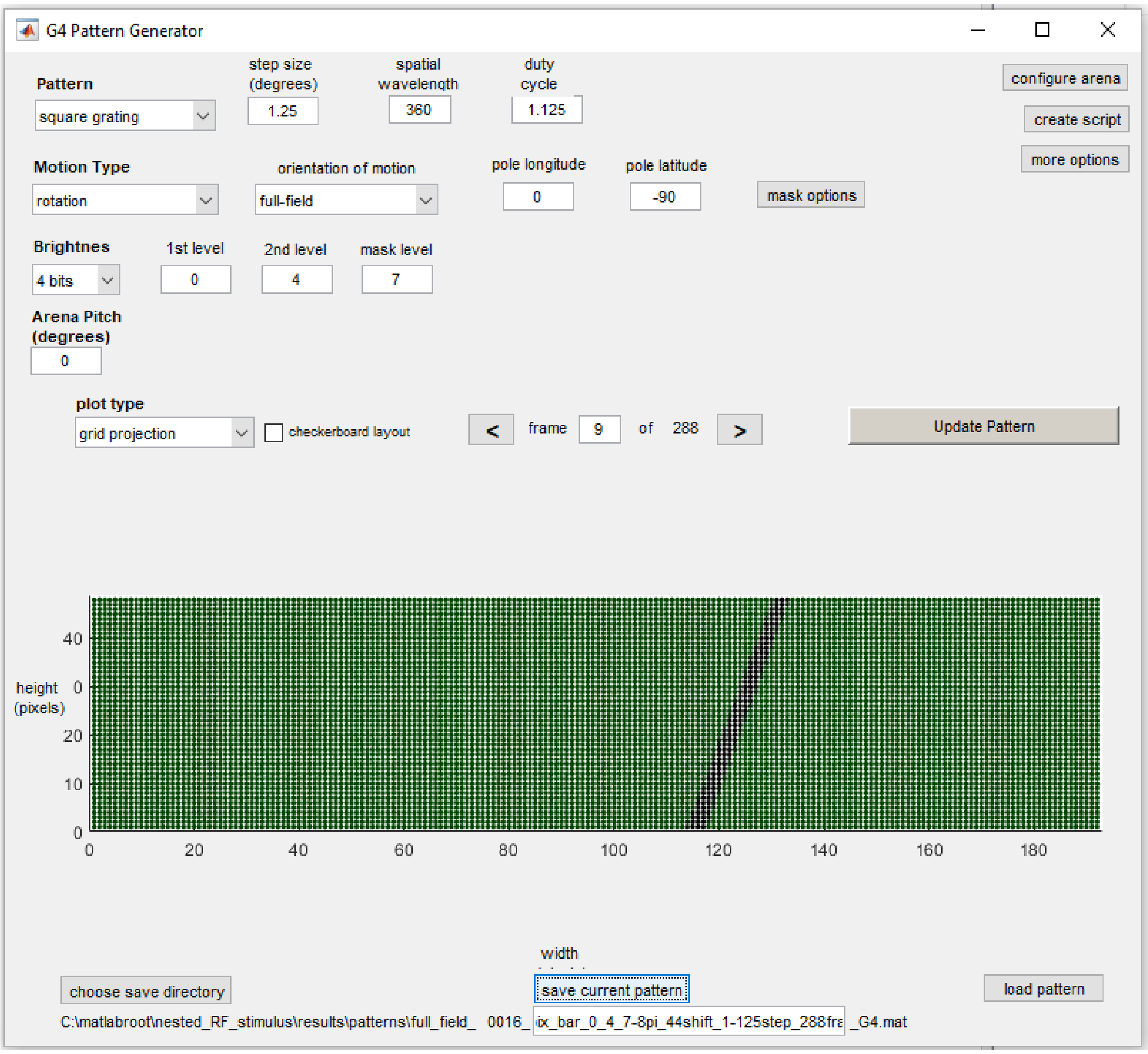
1.2 Making P2 from scratch.

Protocol 2 is made from scratch every time that you run the function `generate_protocol2`. HOWEVER, the cropped bar stimuli that are presented in P2 require premade patterns of full field bar stimuli. These premade patterns are found within the folder `\results\patterns\protocol2\full_field_bars` within the repository. However, the details of how these patterns were made are outlined below.

1.2.1 Pre-making the full field bar stimuli for P2

Using the `G4_pattern_generator_gui` full field bar stimuli were made. A single vertical bar stimulus was made, 4 pixels width, then the "rotation" of the arena was updated to get the patterns in the 8 different orientations (1/8 pi). These patterns were saved in `\results\patterns\protocol2\full_field_bars` or after July 2025 changed to lower contrast bars (background = 4 not 6) `\results\patterns\protocol2\full_field_bars4`.

Note LEB: The patterns 0015 and 0016 (OFF bars at 6/8 and 7/8pi) were updated in OCTOBER 2025 (15/10/2025). Previously, they were moving in the opposite direction to the other bar stimuli. This has been corrected now. The processing scripts will also need to be updated.



Pattern: square grating

step size (degrees): 1.25

spatial wavelength: 360

duty cycle: 1.125

Motion Type: rotation

orientation of motion: full-field

pole longitude: 0

pole latitude: -90

Brightnes: 4 bits

1st level: 0

2nd level: 4

mask level: 7

Arena Pitch (degrees): 0

plot type: grid projection

□ checkerboard layout

frame 9 of 288

C:\matlabroot\nested_RF_stimulus\results\patterns\full_field_ 0016_ ix_bar_0_4_7-8pi_44shift_1-125step_288fra _G4.mat

Figure 2: G4_pattern_generator_gui view when making the full field bar stimuli.

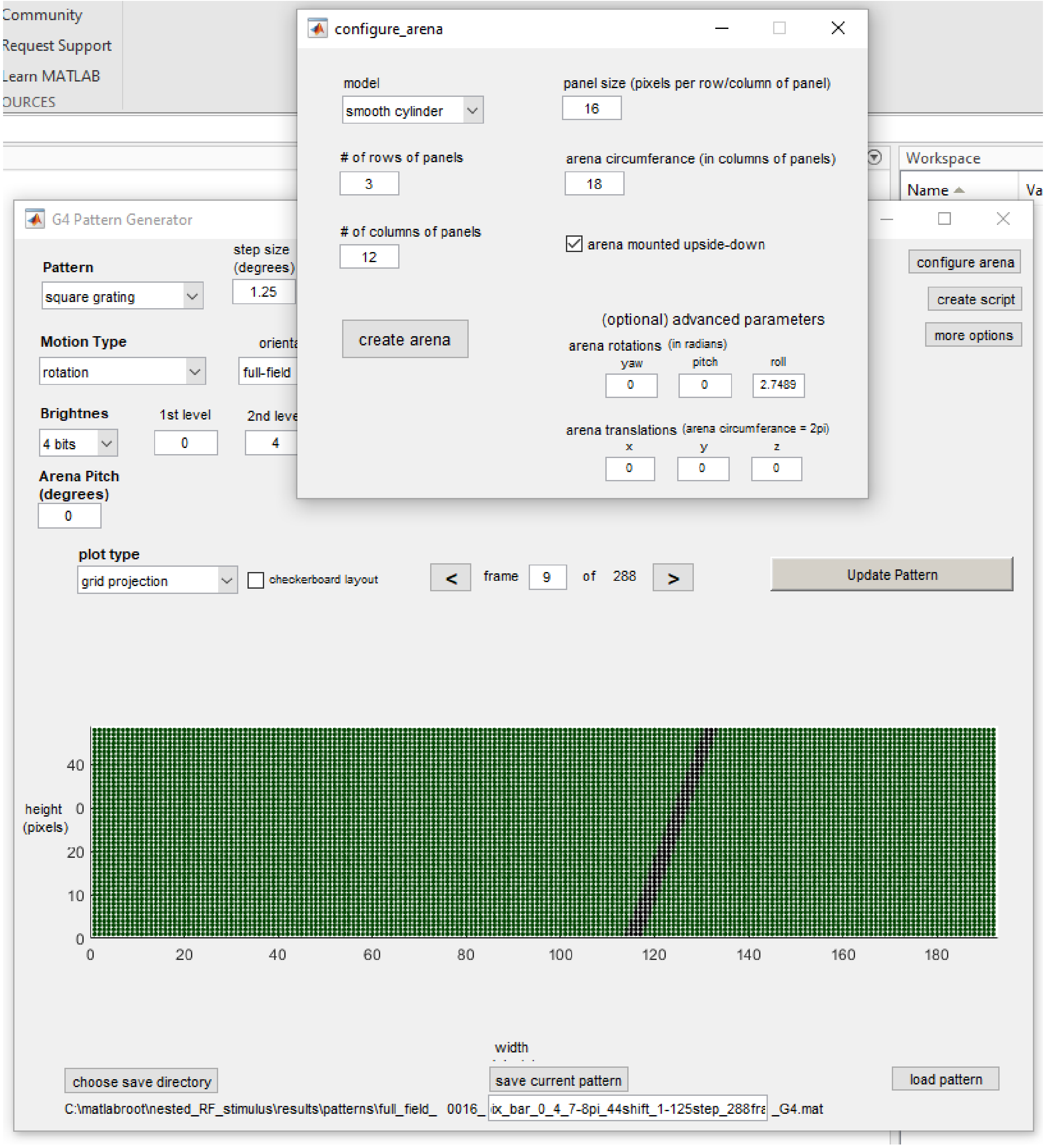
- Pattern: Square grating
- Step size (deg): 1.25
- Spatial wavelength: 360
- Duty cycle: 1.125
- Motion Type: Rotation
- Orientation of motion: full-field
- Pole longitude: 0
- Pole latitude: -90
- Brightness: 4 bits
- 1st level: 0

- 2nd level: 4

288 frames.

Configure Arena:

- Arena rotation (roll): 2.7489 radians for (7/8*pi)



configure_arena

model: smooth cylinder

panel size (pixels per row/column of panel): 16

# of rows of panels: 3

arena circumferance (in columns of panels): 18

# of columns of panels: 12

☑ arena mounted upside-down

(optional) advanced parameters

arena rotations (in radians): yaw 0, pitch 0, roll 2.7489

arena translations (arena circumferance = 2pi): x 0, y 0, z 0

Pattern: square grating

step size (degrees): 1.25

Motion Type: rotation

orienta: full-field

Brightnes: 4 bits; 1st level: 0; 2nd leve: 4

Arena Pitch (degrees): 0

plot type: grid projection; □ checkerboard layout

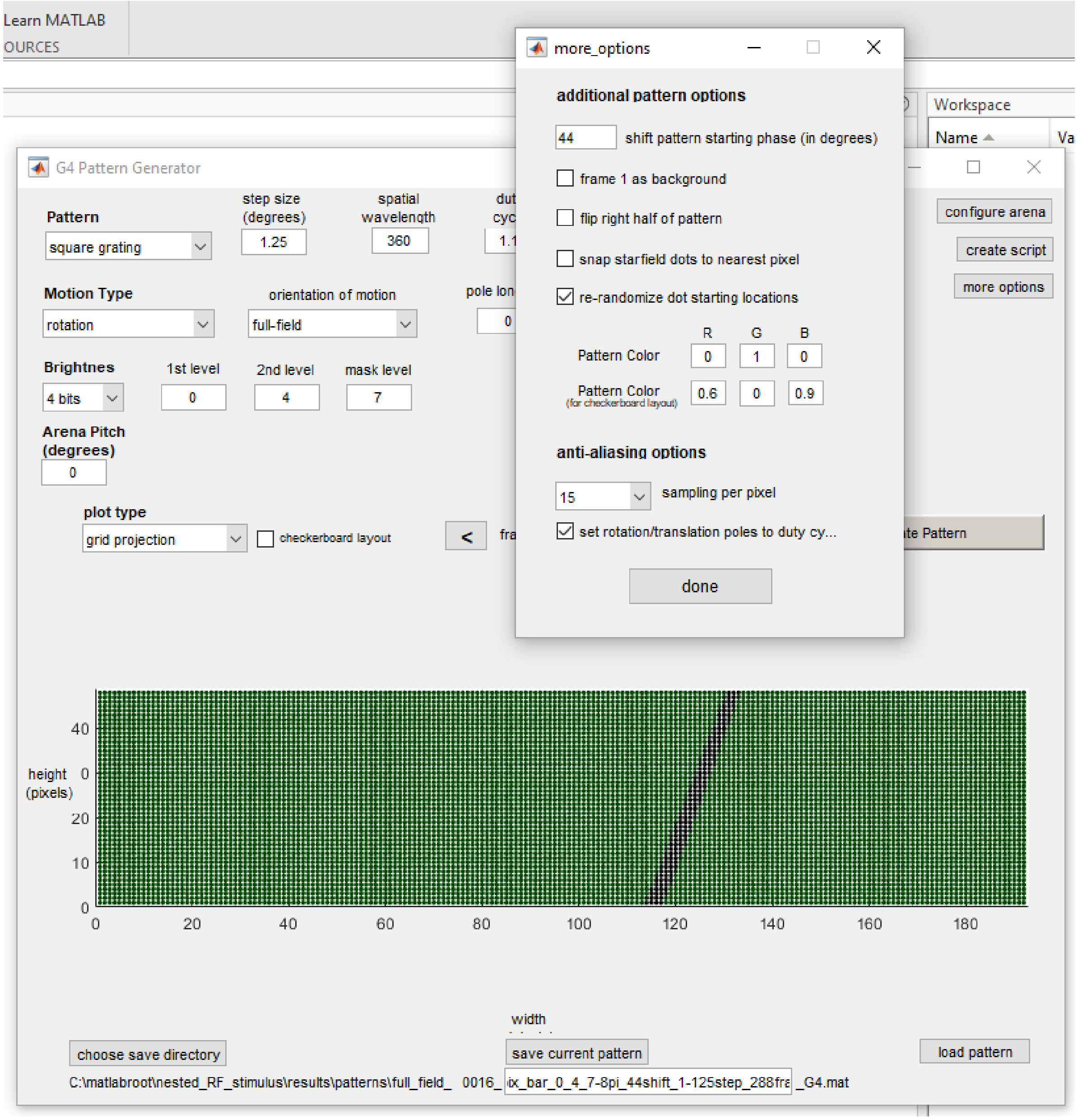
< frame 9 of 288 >

C:\matlabroot\nested_RF_stimulus\results\patterns\full_field_ 0016_ ix_bar_0_4_7-8pi_44shift_1-125step_288fra _G4.mat

Figure 3: G4_pattern_generator_gui view when making the full field bar stimuli. Configure arena pop-up.

More options:

- Shift pattern in starting phase: 44



G4 Pattern Generator

Pattern: square grating

step size (degrees): 1.25

spatial wavelength: 360

dut cyc: 1.1

Motion Type: rotation

orientation of motion: full-field

pole lon: 0

Brightnes: 4 bits

1st level: 0

2nd level: 4

mask level: 7

Arena Pitch (degrees): 0

plot type: grid projection

□ checkerboard layout

C:\matlabroot\nested_RF_stimulus\results\patterns\full_field_ 0016_ ix_bar_0_4_7-8pi_44shift_1-125step_288fra _G4.mat

more_options

additional pattern options

44 shift pattern starting phase (in degrees)

□ frame 1 as background

□ flip right half of pattern

□ snap starfield dots to nearest pixel

☑ re-randomize dot starting locations

	R	G	B
Pattern Color	0	1	0
Pattern Color (for checkerboard layout)	0.6	0	0.9

anti-aliasing options

15 sampling per pixel

☑ set rotation/translation poles to duty cy...

Figure 4: G4_pattern_generator_gui view when making the full field bar stimuli. More options pop-up.

1.2.2 How the cropped bar patterns are generated

As generate_protocol2 is being run, there is a function that crops these full field patterns based on the "window size" specified by params used when running the function.

Within the function generate_bar_patterns_xy these full field bar patterns are cropped around the centre of the pattern, corresponding to pixel numbers [24, 96]. The full field patterns are cropped based upon the value of px_crop which determines the size of the "window size" within which the bar stimuli are presented. This cropped area is then added to the final pattern in a location centred upon the centre of the flash stimulus from protocol 1 that elicited the greatest response. The cropped patterns are saved within the experiment folder.

[Img of what the cropped bar pattern looks like.]

Each pattern is made up of 288 frames... Not 100% sure where this number comes from.

The actual frames shown is determined by the position function.

1.2.3 How the static bar flash patterns are generated

For pattern 0001 - vertical bars.

Middle of the arena [48, 192] = [24, 96]. Frame 30 = bar in the middle.

13 = off left. 47 = off right

The static function is made by generate_static_function within create_protocol2.

1.2.4 How the position functions are generated

The position functions that determine the presentation of the bar stimuli are generated by the function generate_bar_pos_fns. This script generates the position functions programmatically, not using the GUI, but you can view the position function structure in the GUI after if desired.

[Img of what the position function generator looks like.]

The same position function can be used for the patterns moving in different directions. Different position functions must be created for different speeds, different px_crop "window sizes" and for the two different directions. The position functions used for the bar stimuli are relatively simple, comprising a 1s static period followed by a sawtooth movement that goes from the start frame or `fr_low` to the end frame `fr_high` within a given duration `dur`. Setting the parameter `flip` to 1 reverses the direction of movement and so will move the pattern from `fr_high` to `fr_low`.

1.2.5 Modifying the parameters of P2

The parameters that can be modified for P2 are:

- `px_intensity` - The pixel intensity values to be used for the stimuli. This parameter should be a [1 x 3] array corresponding to [bkg_color, off_color, on_color] and should be three integer values between 0 and 15. 0 means the LED is off and 15 is the maximum intensity.
- `px_crop_flash` - Side length (in pixels) of the area within which to present the flash stimuli. This is currently set to 30 pixels.
- `px_crop_bar` - Side length (in pixels) of the area within which to present the bar stimuli. This is currently set to 30 pixels.
- `flash_dur_slow` - The duration of the slower flash stimuli (seconds).
- `int_dur_slow` - The duration of the interval between flash stimuli during the slow flashes (seconds).
- `px_flash` - The size of the flash stimuli (pixels).

The range of pixel values of the display are set using the following parameters:

- `screen_width_start = 17;`
- `screen_width_end = 192;`
- `screen_height_start = 1;`
- `screen_height_end = 48;`

In July 2025, the code that generates P2 was updated to include flashes of two sizes