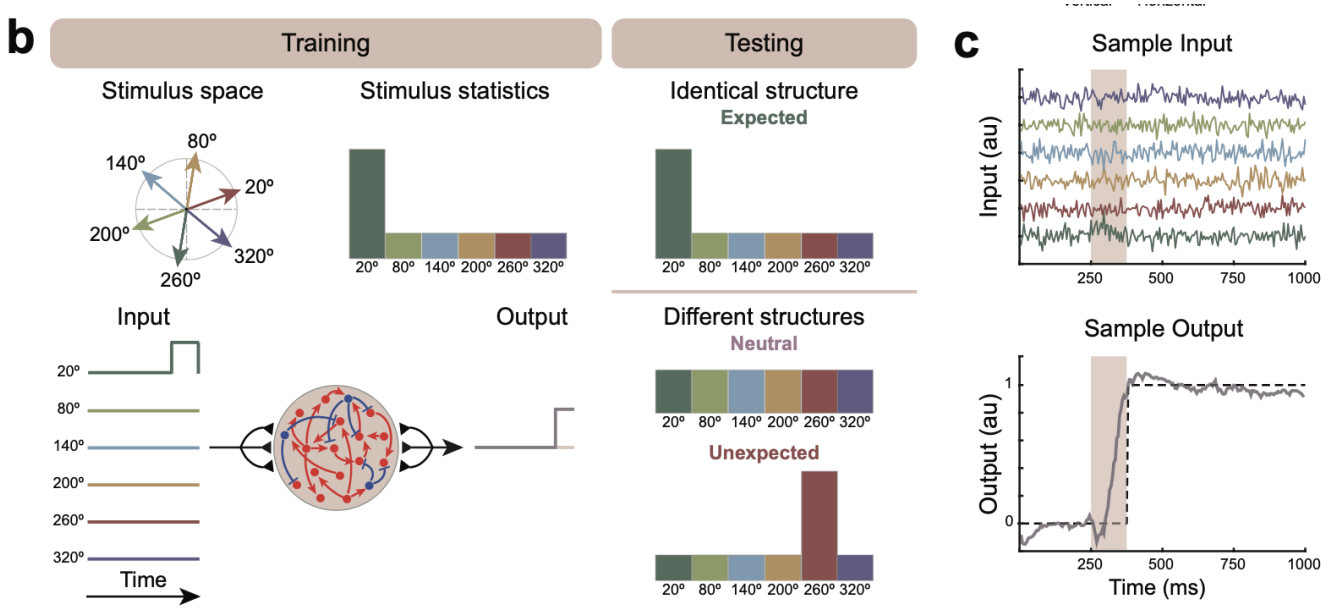
Manual

Task setup

6-AFC (Alternative Forced Choice) Task



Data directory setup

- `rdk_70_30_6afc`: **6-AFC**. Stim 1 was presented 70% of the time
  - `feedforward_only`: 3-layer RNNs without inter-layer feedback connections
    - `hi_coh`: high coherence condition (coherence constant set to 0.7)
    - `lo_coh`: low coherence condition (coherence constant set to 0.6)
  - `with_feedback`: 3-layer RNNs with all connections
    - `hi_coh`: high coherence condition (coherence constant set to 0.7)
    - `lo_coh`: low coherence condition (coherence constant set to 0.6)
- `rdk_80_20_6afc`: **6-AFC**. Stim 1 was presented 80% of the time. Similar folder setup as above.
- `rdk_70_30`: **2-AFC**. Stim 1 was presented 70% of the time. Similar folder setup as above.
- `rdk_80_20`: **2-AFC**. Stim 1 was presented 80% of the time. Similar folder setup as above.

MATLAB scripts

Main script is `generate_trials.m` which requires `fnc_generate_trials.m` and `fnc_eval_model.m`.

First, specify testing condition with `task_info`:

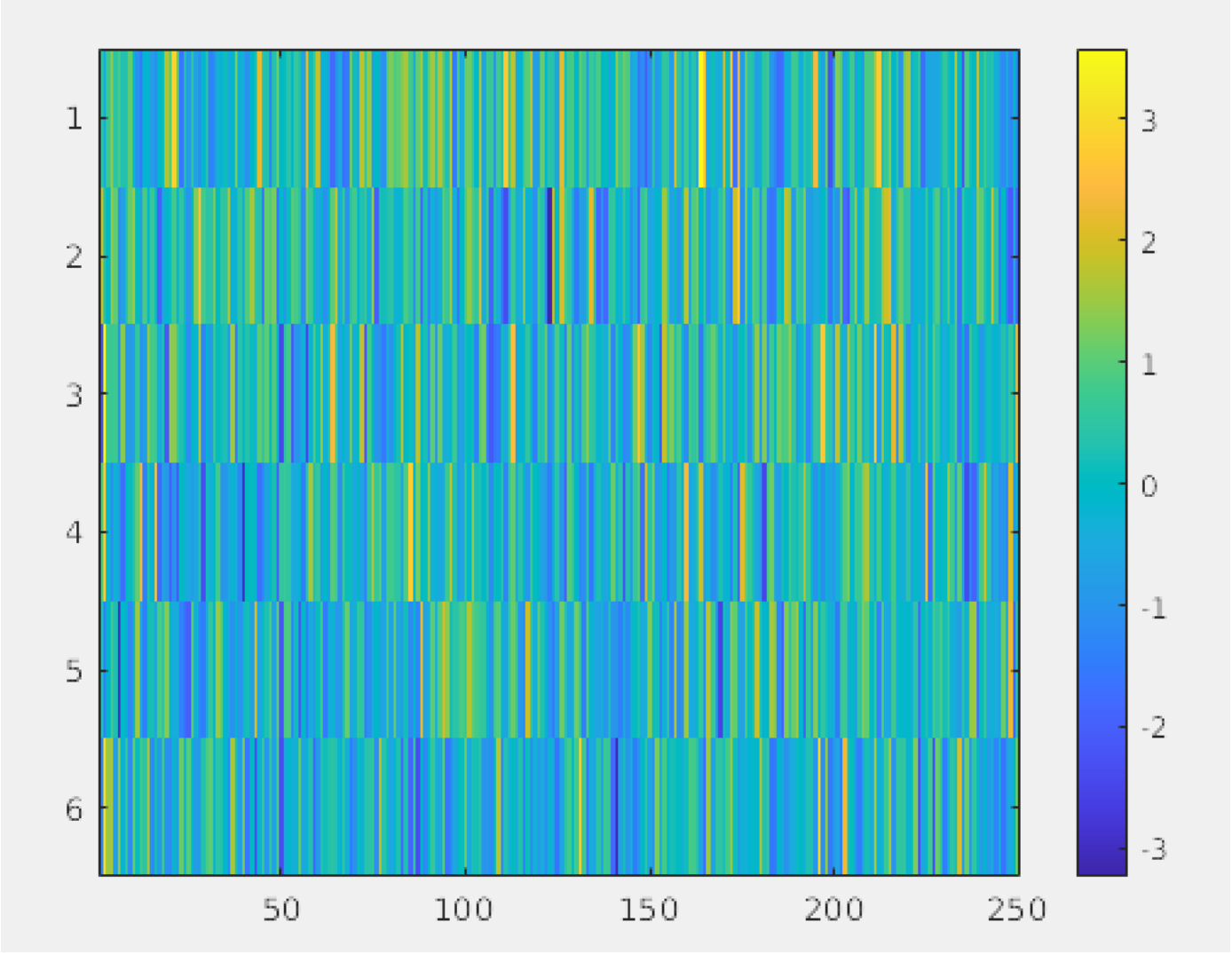
```
% **Testing task condition (70-30 6-AFC with high coherence)
task_info = struct();
task_info.trials = 100;
task_info.trial_dur = 250; % trial duration (timesteps)
task_info.stim_on = 80;
task_info.stim_dur = 50;
task_info.num_stims = 6; % 6AFC
task_info.pred = 1; % predominant stimulus is "1"
task_info.coh = 0.7; % hi_coh = 0.7 vs. lo_coh = 0.6
task_info.primary_prob = 0.70; % 70-30 split
```

Above will generate 100 testing 6-AFC trials where stim 1 will be presented in ~70% of the 100 trials. Use `primary_prob` to adjust the likelihood of stim 1 presentation. Coherence set to the high coherence setting (0.7).

The following line will generate **one** trial:

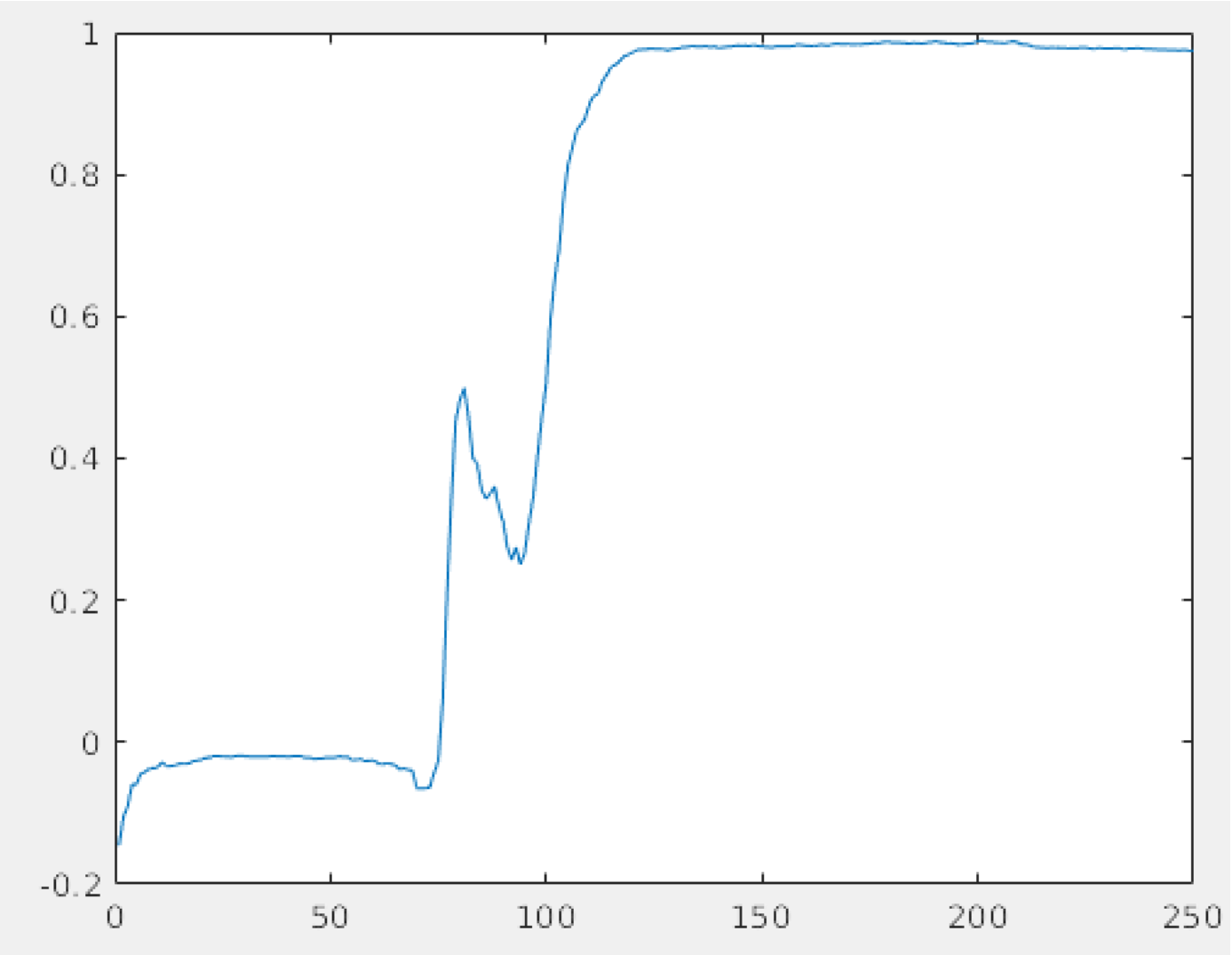
```
[u, lab] = fnc_generate_trials('rdk', task_info);
```

`u` contains the trial data. `lab` will tell you which stimulus was presented.

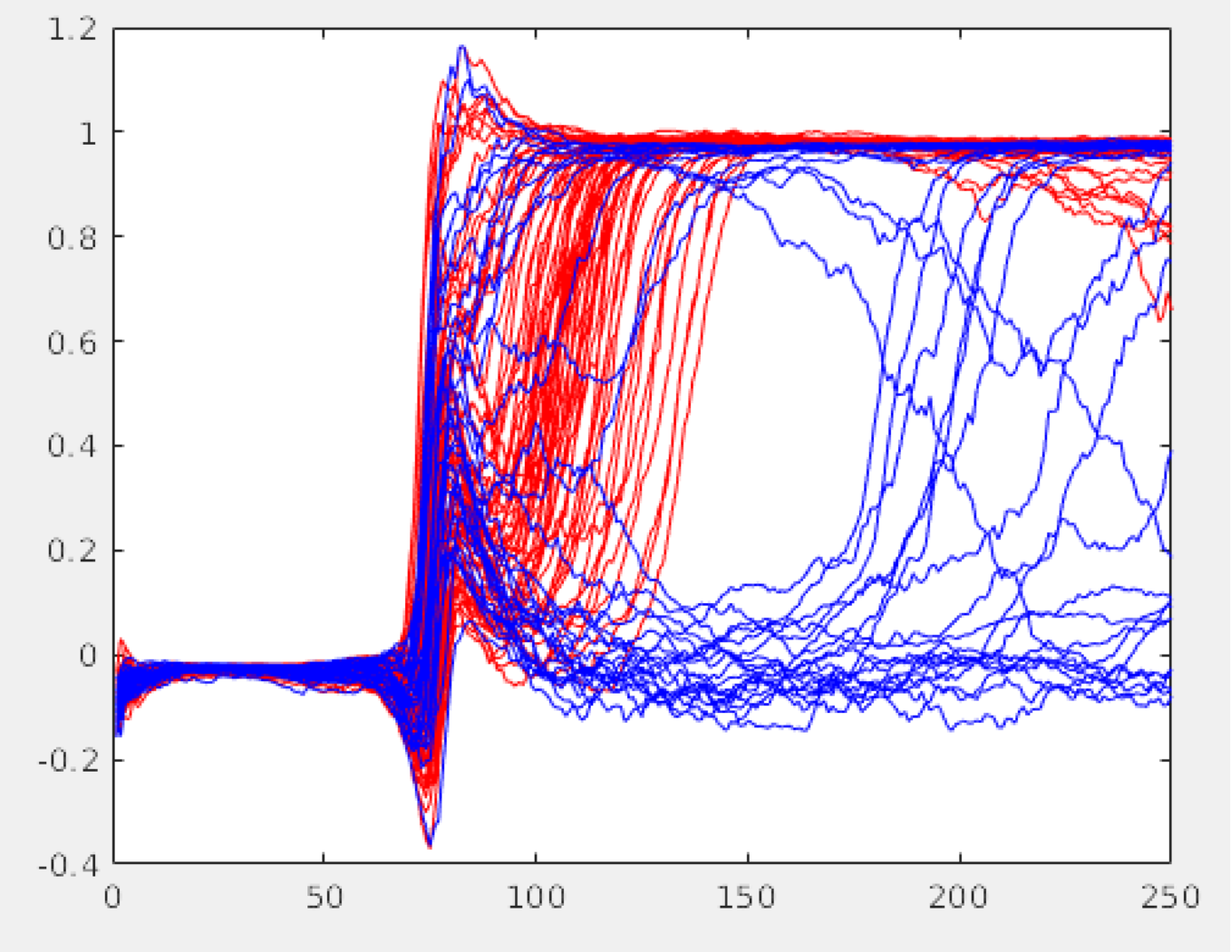


This will now feed the above trial data into a trained RNN to generate the RNN's output:

```
out = fnc_eval_model(model_path, u, feedback);
plot(out('O'))
```



Output signals from 100 trials (using the for-loop section in `generate_trials.m`):



Red lines are the output when stim 1 was presented, while the blue lines are the output signals for the other stims.

Output variables

Given a single trial, `fnc_eval_model.m` will generate a MATLAB container variable:

```
output_signal = out('O');

% Synaptic current variables
X1 = out('X1');
X2 = out('X2');
X3 = out('X3');

% Firing rate estimate variables
R1 = out('R1');
R2 = out('R2');
R3 = out('R3');
```

`X1` is the synaptic current variable data for the **first layer** and its size is [Time steps] × [Number of neurons].
`R1` is the firing-rate estimate variable data for the **first layer** and its size is the same as `X1`.