

University of Toronto Vision Laboratory
Technical Report TR-98-1

The Trial Source Toolbox

Richard Murray
July 1998

We describe a MATLAB 5 toolbox that implements common psychophysical testing protocols, including the method of constant stimuli, UDTR staircases, and QUEST. The toolbox also provides methods for interleaving protocols.

The Trial Source Toolbox is an object-oriented MATLAB 5 implementation of common protocols for choosing stimulus levels in psychophysical experiments. It includes easy-to-use routines that implement the method of constant stimuli, UDTR staircases, and QUEST, as well as routines to interleave any combination of these protocols.

Trial sources for basic protocols

For each method of choosing stimulus levels (e.g., constant stimuli, staircase), the toolbox creates a MATLAB object, which we call a 'trial source'. A typical experiment might interleave one constant-stimuli trial source and two staircase trial sources. To create trial sources, the toolbox contains several 'constructor' functions. The arguments passed to a constructor specify the trial source's properties (e.g., What stimulus levels should the constant-stimuli trial source present? How many reversals should the staircase run for?), and the trial source is the object that is returned as the constructor's return argument. This section describes the constructors for the method of constant

stimuli, the UDTR staircase, QUEST, and a few less common protocols. Later sections show how to interleave protocols, and how to use trial sources in an experiment.

Method of constant stimuli.

```
src = constim( levels, trialsperlevel, id )
```

- `levels` is a vector listing the stimulus levels to be presented
- `trialsperlevel` is the number of times each stimulus level is to be presented (default=Inf)
- `id` is an ID code for the trial source (default=-1)

When this trial source is polled, the stimulus levels in `levels` are each returned `trialsperlevel` times, in round-robin fashion. That is, a permutation of `levels` is formed, each member is returned once, a new permutation is formed, and this is repeated `trialsperlevel` times. After each stimulus level has been returned `trialsperlevel` times, the trial source expires, and it returns a stimulus level of NaN (MATLAB's Not-a-Number constant) if polled again; set `trialsperlevel=Inf` for a trial source that never expires.

Psychophysics Toolbox QUEST.

```
src = pquest( guess, guesssd, thresh, beta, delta, gamma,
             mulstep, range, accuracy, id)
```

- `guess` is the initial estimate of the threshold
- `guesssd` is the standard error of the initial estimate (default=2*guess)
- `thresh` is the percent correct threshold level sought (default=0.75)
- `beta` is the Weibull slope parameter (default=2)
- `delta` is the guessing rate (default=0.01)
- `gamma` is the chance performance level (default=0.5)
- `mulstep` is the multiplicative difference between stimulus levels (default=1.05)
- `range` is the ratio of the largest to the smallest stimulus level (default=4)
- `accuracy` is the standard deviation of the final threshold estimate, expressed as a proportion (default=0)

- `id` is an ID code for the trial source (default=-1)

This trial source follows the protocol described in Watson and Pelli (1983). It is simply a wrapper around the Psychophysics Toolbox implementation of QUEST (= PQUEST). It does, however, take care of certain log-to-linear transformations required by PQUEST.

The stimulus levels that the trial source may return are determined by `guess`, `mulstep` and `range`. The stimulus levels are `guess`, and multiplicative steps upwards and downwards from `guess`, stepping by factors of `mulstep`. Enough steps are included so that the ratio of the highest stimulus level to the lowest stimulus level is approximately `range`. That is, the stimulus levels are $guess \times (mulstep^{-n}, \dots, mulstep^{-1}, 1, mulstep^1, \dots, mulstep^n)$, with n chosen so that $mulstep^n / mulstep^{-n} \approx range$. In some experiments, these parameters must be set with care so that meaningless stimulus levels are not returned (e.g., values below 1.0 if the stimulus dimension is a multiplicative increment). Alternatively, your code may use MATLAB's `max` and `min` to explicitly clip the stimulus levels returned by PQUEST.

The `guesssd` argument sets the width of the Bayesian prior on the threshold. The `accuracy` argument is expressed as a proportion, so `accuracy=0.05` means that the standard deviation of the final threshold estimate distribution should be 5% of the final threshold estimate. After this confidence interval has been attained, the trial source expires, and returns NaN if polled again; set `accuracy=0` for a trial source that never expires.

Up-down transformed response (UDTR) staircase.

```
src = udtr( levels, criterion, maxreversals, id )
```

- **levels** is a vector of stimulus levels that may be presented
- **criterion** is a string specifying the percent correct threshold level sought; valid values are '50%', '71%', '74%', '79%' and '83%' (default='79%')
- **maxreversals** is an integer specifying the number of reversals after which the staircase expires (default=Inf)
- **id** is an ID code for the trial source (default=-1)

This trial source follows the well-known staircase protocol described in Wetherill and Levitt (1965).

Stimulus series.

```
src = sersrc( levels, trialsperlevel, id )
```

- **levels** is a vector of stimulus levels to be presented
- **trialsperlevel** is the number of times each stimulus level is to be presented (default=1)
- **id** is an ID code for the trial source (default=-1)

This trial source presents the first stimulus level in `levels`, `trialsperlevel` times, then presents the second stimulus level, `trialsperlevel` times, and continues in this manner for each stimulus level. After all stimulus levels have been exhausted, the trial source expires, and returns NaN if polled again.

Trial sources for interleaved protocols

Two routines in the toolbox, `tslist` and `addtslist`, combine existing trial sources to produce a single new trial source. The new trial source, when consulted, polls each of the constituent trial sources in round-robin fashion, and returns the stimulus level and ID code given by the trial source currently at the head of the queue. The list expires when every constituent trial source has expired.

Making a trial source list.

```
list = tslist( src1, src2, ... )
```

The trial sources passed as arguments are combined into a single trial source list. The arguments may themselves be trial source lists created with `tslist` or `addtslist`.

Adding to a trial source list.

```
newlist = addtslist( list, src1, src2, ... )
```

The first argument is a trial source list created with `tslist` or `addtslist`. The remaining arguments are appended to this list.

Getting a trial source from a list

```
src = getsource( list, n )
```

Using trial sources

The trial sources created with the above constructors, both the basic protocols and the trial source lists, are all used via the four functions `gettrial`, `telltrial`, `trialnum` and `threshest`.

Getting a stimulus level.

```
[ newsrc, level, id ] = gettrial( src )
- src is a trial source
- newsrc is the updated version of the trial source
- level is the stimulus level to be presented
- id is the ID code of the trial source from which the stimulus level was
  obtained
```

This function obtains new stimulus levels from a trial source. If the trial source is a list, the ID code returned is the ID of the constituent trial source from which the stimulus level was obtained.

Reporting a trial result.

```
newsrsc = telltrial( src, level, correct )
```

- `src` is the trial source previously consulted to get the stimulus level
- `level` is the stimulus level that was presented
- `correct` is 1 if the subject gave the correct response, 0 if not
- `newsrsc` is the updated version of the trial source

Adaptive protocols (e.g., QUEST) need to know the results of trials. Results are reported through `telltrial`, which updates the trial source appropriately. Non-adaptive protocols (e.g., constant stimuli) can also have results reported to them through `telltrial`, though this has no effect.

Getting the trial number.

```
n = trialnum( src )
```

- `src` is a trial source
- `n` is the number of stimulus levels polled from the trial source so far

Getting a threshold estimate.

```
est = threshest( src )
```

- `src` is a trial source
- `est` is the trial source's estimate of the subject's threshold

The PQUEST and UDTR trial sources, once exhausted, provide estimates of the subject's threshold through this function. If these trial sources are polled before reading the required confidence interval size, or if other trial source types (which do not support this routine) are called, the return argument is either unreliable, or NaN.

Example

```

function experiment

% create individual trial sources
stimlev=[ 1 2 3 4 5 ];
ts1 = constim( stimlev, Inf, 1 );
ts2 = udtr( stimlev, '71%', 6, 2 );
ts3 = udtr( stimlev, '79%', 6, 3 );

% combine into a single interleaved trial source
src = tslist( ts1, ts2, ts3 );

% present trials
while 1,

    % get next stimulus level
    [ src, level, id ] = gettrial( src );
    if isnan(level),
        break
    end

    % present trial to subject
    correct = mypresenttrial( id, level );

    % report result to trial source
    src = telltrial( src, level, correct );

end

return

```

References

Watson, A. B., & Pelli, D. G. (1983). QUEST: A Bayesian adaptive psychometric method. *Perception & Psychophysics*, 33(2), 113-120.

Wetherill, G. B., & Levitt, H. (1965). Sequential estimation of points on a psychometric function. *British Journal of Mathematical and Statistical Psychology*, 18, 1-10.