# Tracker Module

The tracker module provides the model with a system that is able to estimate the best value for a quantity based on one or two feedback signals, a “good” result and a “bad” result, which may be noisy, have different weights, and which may not be directly associable to the actions which caused them. That estimated quantity is updated periodically by the tracker module for the model to use without any additional actions needed by the model, and multiple different quantities may be estimated simultaneously by the tracker module (each quantity will be estimated using a separate ‘tracker’).

This is currently implemented as a pseudo-“rational” function estimator which tries to fit a 3-parameter quadratic function relating the control value to the feedback. The choice of a quadratic function reflects a non-informed guess as to how learning extrapolates over the range of possible values. This is the process which it uses to do so:

1. When a tracker is created it starts (by default) with an initial evaluation that is uniform over the range of control values and chooses one at random from that range.

2. It determines the length of the update interval from an exponential distribution based on a mean duration which can be provided for the tracker (it defaults to 10 seconds). The choice of an exponential distribution for evaluation reflects a non-informed guess as to what the evaluation periods are. That interval is bounded to be at least 1 second, and no more than 60 seconds, and generated like this:

-d * log(random[0,1])

*d* is the mean duration

3. Good and bad events are recorded until the update interval completes.

4. It computes a mean rate of return from the good and bad events during the interval. That mean rate of return is added to the estimation of the quadratic function with a weight equal to the duration of the interval.

5. Using the updated estimate of the quadratic function it selects a new control value using a softmax rule. The temperature used in the softmax rule decreases as time progresses (while the tracker is active) according to the function:

init-temp/(1 + Active-Time/Scale-time)

Unless set otherwise the initial temperature (init-temp) is 1 and the time scale (scale-time) is 180 secs. This means that in 3-minute Space Fortress the temperature is 1 at the beginning of the first game, .5 at the beginning of the second game, .2 at the beginning of the fifth game, and .05 at the beginning of the 20th game (assuming the tracker is active throughout).

6. Return to step 2.

The control value is made available to the model through the slot of a chunk in a buffer, and the slot and buffer are specified when a tracker is created. There will be at most one active tracker for each control-slot, but a given control-slot may have multiple suspended trackers which are available to be restored (which suspends a currently active one). The good and bad feedback are also determined from slots of chunks in buffers which are specified when the tracker is created (both slots must be for a chunk in the same buffer). Whenever the chunk in a buffer which is associated with a tracker's feedback is set or modified through a scheduled event, the feedback will be recorded. If a chunk is set in the buffer then the value in both the good and bad slot (if available) are recorded. If a chunk is modified in the buffer only those slots which are modified are checked for a feedback event occurring.

After a tracker has been created, some of its components can be updated to adjust the process. When a modification happens that will initiate an immediate update to create a new control value.

## Parameters

All of the default components of how a tracker operates can be specified with parameters. The tracker module can also provide a detailed trace of the actions it performs if the parameter to do so is enabled, and there are parameters which can be set to apply a decay function to the use when updating the equation.

### :bad-scale

The bad-scale parameter can be set to a function, command name string, or **nil**. If it is set to a function or command string that function/command will be passed four values: the tracker's control-slot, the tracker's name, the value from the bad slot when it changed, and the tracker's bad-weight. The value returned should be the number which indicates the value to apply when updating the estimate. Note that the weight is only provided for reference and should not be applied for that value because the weight is applied later in the calculation by the module (it only applies the weight once after collecting all the values for a period). The default value is **nil** which means there is no scaling applied.

### :bad-weight

The bad-weight parameter can be set to a number, and that number is multiplied by the value of a bad feedback result (either the value from the slot or the result of the bad-scale if one exists) to produce the value for the event. The default value is -1.

### :control-buffer

The control-buffer parameter can be set to the name of a buffer, and indicated the default buffer in which the slot for the control value will be found. The default value is goal.

### :control-count

The control-count parameter indicates the default number of possible control results a tracker will use between the min and max values indicated (inclusive of both). The default value is 21.

### :control-max

The control-max parameter indicates the default maximum value for the range of control results. The default value is 1.

### :control-min

The control-min parameter indicates the default minimum value for the range of control results. The default value is 0.

### :control-scale

The control-scale parameter can be set to a function, command name string, or **nil**. If it is set to a function or command string that function/command will be passed three values: the tracker's control-slot, the tracker's name, and the current choice for the value for the control result when it is updated. The value returned will be the one that is placed into the control slot. The default value is **nil** which means there is no adjustment to the value.

### :good-scale

The good-scale parameter can be set to a function, command name string, or **nil**. If it is set to a function or command string that function/command will be passed four values: the tracker's control-slot, the tracker's name, the value from the good slot when it changed, and the tracker's good-weight. The value returned should be the number which indicates the value to apply when updating the estimate. Note that the weight is only provided for reference and should not be applied for that value because the weight is applied later in the calculation by the module (it only applies the weight once after collecting all the values for a period). The default value is **nil** which means there is no scaling applied.

### :good-weight

The good-weight parameter can be set to a number, and that number is multiplied by the value of a good feedback result (either the value from the slot or the result of the good-scale if one exists) to produce the value for the event. The default value is 1.

### :initial-temp

The initial-temp parameter sets the value for the default initial temperature used by a tracker. The default value is 1.

### :temp-scale

The temp-scale parameter sets the default value used in the computation of the temperature decrementing for a tracker. The default value is 180.

### :trace-tracker

The trace-tracker parameter controls whether the module prints information in the model trace with all of the details about the tracker actions which occur. It can be set to **t** or **nil**, and the default value is **nil**.

### :tracker-decay

The tracker-decay parameter sets the decay value used when discounting the previous results when updating the a tracker. It can be set to a number or **nil**. The default value is **nil**, but it will be set automatically to .5 or .99 if it is **nil** when decay is enabled with a power law or exponential decay mechanism respectively using the :tracker-decay-method parameter.

### :tracker-decay-method

The tracker-decay-method parameter controls whether or not previous results are discounted when updating a tracker estimate. It can be set to **nil**, **power**, or **exponential**. If it is non-**nil** then the indicated decay mechanism is applied to the past estimates when updating a tracker. The default value is **nil**.

### :update-delay

The update-delay parameter sets the default value for the mean of the update interval (in seconds) used by a tracker. The default value is 10.

### :values-buffer

The values-buffer parameter can be set to the name of a buffer and indicates the default buffer in which the good and bad slots for a tracker will be located. The default value is imaginal.

## Chunk-types & Chunks

The tracker module defines the following chunk-types:

```
(chunk-type tracker-params control-scale good-slot bad-slot good-weight bad-weight
            good-scale bad-scale min max control-count temp scale delay x-bias y-bias)

(chunk-type tracker-modification control-scale good-slot bad-slot good-weight
            bad-weight good-scale bad-scale min max control-count delay)

(chunk-type (track-outcome (:include tracker-params))
            name control-slot good-buffer bad-buffer)

(chunk-type suspend-tracker name control-slot (suspend t))

(chunk-type resume-tracker name control-slot)

(chunk-type copy-tracker name control-slot new-name (copy t) reweight copy-temp)

(chunk-type tracker-query exists active available)
```

It creates no initial chunks.

## Tracker buffer

The tracker buffer is a searchable multi-buffer. It holds chunks which represent the information for all of the trackers which have been created and are currently active. Because there is only one active tracker for a specified control-slot, that is the recommended way to find a specific tracker in a production.

Activation spread parameter: :tracker-activation
Default value: 0.0

### Queries

The tracker buffer responds to the default queries like this:

‘State busy’ will always be **nil**.

‘State free’ will always be **t**.

'State error' will always be **nil**.

It has additional queries named exists, and active. Exists will return true if there is an existing tracker with a control-slot that matches the value provided and **nil** if not. Active will return true if there is an existing tracker with a control-slot that matches the value provided and that tracker is not suspended, or **nil** otherwise.

## Requests

***track-outcome***

[**control-slot** *control* | **control-slot** *control* **name** *name* ]
[**good-slot** *good-slot* | **bad-slot** *bad-slot* | **good-slot** *good-slot* **bad-slot** *bad-slot* ]
{**good-buffer** *good-buffer*}
{**good-weight** *good-weight*}
{**good-scale** *good-scale*}
{**bad-buffer** *bad-buffer*}
{**bad-weight** *bad-weight*}
{**bad-scale** *bad-scale*}
{**min** *minimum-control-value*}
{**max** *maximum-control-value*}
{**control-count** *number-of-control-values*}
{**control-scale** *control-scale*}
{**temp** *initial-temperature*}
{**scale** *temperature-scale*}
{**delay** *update-delay*}
{**x-bias** *initial bias x values*}
{**y-bias** *initial bias y values*}

This request will create a new tracker, and if there is already a tracker for the indicated control-slot then it will be suspended automatically when the new one is created. A creation request must have a control-slot, and must have at least one of the good-slot or bad-slot values specified but may provide both. If a name is provided, it must be unique among all trackers for the indicated control-slot, and if not provided a unique name will be assigned to the tracker. The rest of the slots control the details for the tracker, and any which are not provided will be set to the current value from the corresponding parameter of the module (except x-bias and y-bias described below).

If all the provided parameters are valid a new tracker is created and a chunk which contains all of its information (except the initial temperature and temperature scale) will be added to the set of

chunks for the tracker buffer. It will then schedule an event to modify the control-slot of the chunk in the control-buffer with the initial guess for the control value:

```
     0.050   TRACKER                MOD-BUFFER-CHUNK GOAL
```

That event is scheduled at the time the tracker is created with a priority of -2000 so that other actions at that time e.g. other production requests, will have an opportunity to create the chunk in the control buffer before the tracker sets the initial value. If there is no chunk in the buffer when that modification happens then a warning will be printed and the initial value will not be set.

```
#|Warning: schedule-mod-buffer-chunk called with no chunk in buffer GOAL |#
```

When a tracker updates it will generate two or three events. There will always be an update-tracker event at the time of the update indicating the name of the tracker and the control slot that is being updated (only shown in high detail trace):

```
    31.946   TRACKER                update-tracker TRACKER0 VALUE
```

If there is a chunk in the control buffer it will generate a mod-buffer-chunk event to update the control-slot's value with the new guess:

```
    31.946   TRACKER                MOD-BUFFER-CHUNK GOAL
```

If there is not a chunk in the control buffer then it will output warnings to indicate that and then generate a set-buffer-chunk event to place a chunk which contains only the control slot into the control buffer:

```
#|Warning: Tracker update occurred with no chunk in the control buffer GOAL. |#
#|Warning: Creating a new chunk with only the VALUE slot. |#
     4.038   TRACKER                SET-BUFFER-CHUNK GOAL CHUNK2
```

In that case it will also create an event which is not displayed in the trace to remove the temporary chunk which was created to be placed into the buffer. That event would look like this if it were printed:

```
     4.038   TRACKER                Clean-up unneeded chunk
```

If there was a tracker for the indicated control-slot then it will be suspended and its chunk will be removed from the set of available chunks in the buffer. That will result in an event like this in the trace indicating the suspended tracker:

```
    31.800   TRACKER                update-and-suspend-tracker TRACKER0 VALUE
```

The x-bias and y-bias slots can be specified to start the tracker with a distribution that is not uniform across the choices before the initial value is determined. Both require a list of numbers as a value, and there are two ways to use them. If both the x-bias and y-bias are specified as lists with the same number of items then each corresponding x,y pair is treated as a value and mean rate of return from the good result to update the quadratic equation. Note, the x values do not have to be within the range of possible choices. If only the y-bias is specified then it must be a list no longer than the number of choices for the tracker. Each of the provided numbers is considered as the mean rate of return from the good result for the corresponding choice – the first number provided corresponds to the first choice value, etc. If there are fewer values than choices, each of the remaining choices is treated as having a mean rate of return of 0 for the initial update.

***suspend-tracker***

[**control-slot** *control* | **name** *name* | **control-slot** *control* **name** *name* ]
**suspend t**

This request is used to suspend an active tracker explicitly. If there is an active tracker with the specified control-slot, a single active tracker with the given name, or a tracker with the combination of control-slot and name given then it is suspended. A suspended tracker is updated immediately, but does not set the control slot's value. It does not collect any good or bad results while suspended, but does still decay as time passes (if enabled). The following action will be shown in the trace when the tracker is suspended (indicating the tracker's name):

```
31.800   TRACKER                update-and-suspend-tracker TRACKER0 VALUE
```

If no active tracker matches the provided references then a warning is printed indicating the information provided and nothing else happens:

```
#|Warning: Tracker request failed because no active tracker with name BAD-NAME. |#

#|Warning: Tracker request failed because no active tracker with control-slot BAD-SLOT
found. |#

#|Warning: Tracker request failed because no active tracker with name BAD-NAME and
control-slot VALUE found. |#
```

***resume-tracker***

[**control-slot** *control* | **name** *name* | **control-slot** *control* **name** *name* ]

This request will resume a suspended tracker. A request with only a control-slot or name value provided will resume a tracker if there is a single suspended tracker which matches the provided value. If both values are provided then a tracker which matches on both will be resumed. When resumed the tracker will update immediately and its chunk will be placed into the buffer set. Here are the events which will be generated for that:

```
 63.150   TRACKER                update-and-resume-tracker TRACKER0 VALUE
```

If there is an active tracker for the same control-slot then it will be suspended before the prior one is resumed which will generate these two events for the trackers:

```
121.227   TRACKER                update-and-suspend-tracker NEW-TRACKER VALUE
121.227   TRACKER                update-and-resume-tracker TRACKER0 VALUE
```

If there is no suspended tracker with the given name, no suspended tracker with a control-slot of control, or more than one suspended tracker with that control value, then a warning will be printed and nothing else will change. One of these warnings will be shown depending upon the reason for the failure:

```
#|Warning: Tracker request failed because could not find a unique tracker named BAD-NAME to resume. |#

#|Warning: Tracker request failed because could not find a unique tracker with control-slot BAD-SLOT to resume. |#

#|Warning: Tracker request failed because no tracker with name BAD-NAME and control-slot VALUE found to resume. |#
```

***copy-tracker***

[**control-slot** *control* | **name** *name* | **control-slot** *control* **name** *name* ]
**copy t**
{ **new-name** *new-name* }
{ **reweight** *scale* }
{ **copy-temp** *temp* }

This request is used to create a new tracker like the track-outcome request described above. The difference is that instead of supplying the details of the tracker the new tracker is created as a copy of an existing tracker which is found based on the name and/or control-slot which are provided. If a unique tracker is found based on that information then a new tracker is created with the current state of that tracker (excluding any experiences that have not yet been updated). If the new-name is provided that will be used for the name of the new tracker otherwise a name will be generated automatically for it. If that previous tracker was active then it is suspend, or if

there is a different tracker active for that control-slot then it will be suspended. If the scale value is specified as **t** or a number then the history for the new tracker will be adjusted. If the scale is **t** then the history will be scaled to represent only one occurrence of each possible choice (an N equal to the number of choices). If it is a number then the history will be scaled to have that weight as the N value. If the scale value is **:new** then the weights will be set as if this were a new tracker instead of copying the history. If the temp value is provided as **t** then the values that control the temperature setting will also be copied for this tracker, otherwise it will start with the initial temperature and temperature scale as set by the **:initial-temp** and **:temp-scale** parameters.

### Modification Requests

#### *tracker-modification*

{**good-slot** *good-slot*}
{**bad-slot** *bad-slot*}
{**good-weight** *good-weight*}
{**good-scale** *good-scale*}
{**bad-weight** *bad-weight*}
{**bad-scale** *bad-scale*}
{**min** *minimum-control-value*}
{**max** *maximum-control-value*}
{**control-count** *number-of-control-values*}
{**control-scale** *control-scale*}
{**delay** *update-delay*}

A modification request to the tracker buffer will update the information of the tracker which is currently in the tracker buffer. The control slot may not be adjusted and neither can the buffers in which the good and bad slots are located, but all other tracker values may be changed, with the constraint that there still must be either a good or bad slot in the tracker.

After the changes are made to the tracker the tracker buffer is erased (since that chunk no longer reflects the current values for that tracker) and an immediate update of the tracker occurs.

## Commands

### print-tracker-stats

**Syntax:**

**print-tracker-stats** {*id* {*name?*}} -> [ ((name control temp equation)*) | **nil** ]
**print-tracker-stats-fct** {*id* {*name?*}} -> [ ((name control temp equation)*) | **nil** ]

## Remote command name:

**print-tracker-stats**

## Arguments and Values:

id ::= the name or control slot to use to find a tracker
name? ::= if this is true then the id must match a name otherwise a control slot
temp ::= the current temperature of the tracker
equation ::= (c b a)
a,b,c ::= the coefficients in the currently estimated equation for that tracker: $c + bx + ax^2$
name ::= the name of the tracker
control ::= the tracker's control slot

## Description:

Print-tracker-stats can be used to output the details of a tracker or multiple trackers to the command output trace. If an id is provided that will be used to find all trackers with the indicated control slot if the name? parameter is not provided or is **nil**, and if the name? parameter is provided as a true value then the id will be used to find all trackers with that name. If matching trackers can be found then the details of those trackers are printed and a list containing lists for each tracker's coefficients of their current estimated value equation are returned. If the id given does not correspond to a tracker in the current model a warning is printed and **nil** is returned. If no id is specified then the details of all the currently created trackers in the current model are printed and the list of coefficient lists is returned. If there is no current model then a warning is printed and **nil** is returned.

## Examples:

```
1> (load-act-r-model "ACT-R:extras;tracker;simple.lisp")
T

2E> (print-tracker-stats)
#|Warning: There are no trackers available. |#
NIL

3> (run-test 50)
...
T
"response"

4> (print-tracker-stats-fct 'value)
Tracker named TRACKER0 for slot VALUE (active)
```

```
 Good slot is GOOD in buffer IMAGINAL with weight 1
 Bad slot is BAD in buffer IMAGINAL with weight -1
  Current temperature is: 0.7870986
  Current equation is: -1.4593911 + 1.6013756x + -0.39073476x^2
  Choices: 1.000 2.000 3.000
  Probs:   0.261 0.451 0.288
  Current setting is: 2
((TRACKER0 VALUE 0.7870986 (-1.4593911 1.6013756 -0.39073476)))

5> (module-request 'tracker (define-chunk-spec control-slot size good-slot color))
...
T

6> (print-tracker-stats)
Tracker named TRACKER1 for slot SIZE (active)
 Good slot is COLOR in buffer IMAGINAL with weight 1
  Current temperature is: 1.0
  Current equation is: 0 + 0x + 0x^2
  Choices: 0.000 0.050 0.100 ... 0.900 0.950 1.000
  Probs:   0.048 0.048 0.048 ... 0.048 0.048 0.048
  Current setting is: 19/20
Tracker named TRACKER0 for slot VALUE (active)
 Good slot is GOOD in buffer IMAGINAL with weight 1
 Bad slot is BAD in buffer IMAGINAL with weight -1
  Current temperature is: 0.7899796
  Current equation is: -0.8470351 + 0.8375704x + -0.20702161x^2
  Choices: 1.000 2.000 3.000
  Probs:   0.299 0.394 0.307
  Current setting is: 3
((TRACKER1 SIZE 1.0 (0 0 0))
 (TRACKER0 VALUE 0.7899796 (-0.8470351 0.8375704 -0.20702161)))

7> (print-tracker-stats-fct 'tracker1 t)
Tracker named TRACKER1 for slot SIZE (active)
 Good slot is COLOR in buffer IMAGINAL with weight 1
  Current temperature is: 1.0
  Current equation is: 0 + 0x + 0x^2
  Choices: 0.000 0.050 0.100 ... 0.900 0.950 1.000
  Probs:   0.048 0.048 0.048 ... 0.048 0.048 0.048
  Current setting is: 19/20
((TRACKER1 SIZE 1.0 (0 0 0)))

E> (print-tracker-stats-fct 'bad)
#|Warning: There are no trackers associated with the BAD slot. |#
NIL

E> (print-tracker-stats bad t)
#|Warning: There are no trackers with the name BAD. |#
NIL

E> (print-tracker-stats)
#|Warning: No current model available for printing tracker stats. |#
NIL
```