

---

# trainRewardModel.m

## Table of Contents

High-level steps .....	1
Code .....	1

**Summary:** Script to train the reward model.

## High-level steps

1. Set up GP structure and train
2. Add noise and re-train while SNR is high
3. Check length-scales of the model

## Code

```
% 1. Set up and train reward GP structure
clear reward_dynmodel;
reward_dynmodel.fcn = @gp1d;
reward_dynmodel.train = @train;
reward_dynmodel.induce = zeros(5000,0,1);
reward_dynmodel.inputs = r_x;
reward_dynmodel.targets = reward2loss(r_y);

reward_dynmodel = reward_dynmodel.train(reward_dynmodel); % train dynamics GP

% 2. Add noise and train again while SNR is high
r_hyp = reward_dynmodel.hyp;
highSNR = any(exp(r_hyp(end-1,:)-r_hyp(end,:)) > 500);

while highSNR
    target_std = std(reward_dynmodel.targets);
    reward_dynmodel.targets = reward_dynmodel.targets + 0.01*target_std*randn(size(reward_dynmodel.targets));
    reward_dynmodel = reward_dynmodel.train(reward_dynmodel); % train dynamics GP
    r_hyp = reward_dynmodel.hyp;
    highSNR = exp(r_hyp(end-1)-r_hyp(end)) > 500;
    disp(['Reward model SNR = ' num2str(exp(r_hyp(end-1)-r_hyp(end)))]);
end

% 3. Check length-scales
if r_hyp(end-1) > log(10)
    fprintf('Reward log-signal-std hyperparameter is %.4f, greater than log(10)\n', r_hyp(end-1));
end

if r_hyp(end-1) < log(0.1)
    fprintf('Reward log-signal-std hyperparameter is %.4f, less than log(0.1)\n', r_hyp(end-1));
end
```

```
input_std = std(reward_dynmodel.inputs)';  
fprintf('Reward input lengthscales and input std:\n');  
disp(num2str([exp(r_hyp(1:12)), input_std]));
```

*Published with MATLAB® R2014a*