
trainHelicopterModel.m

Table of Contents

High-level-steps	1
Code	1

Summary: Script to learn the dynamics model. If SNR is too high, white noise is added to the GP targets and model is trained again.

High-level-steps

1. Set up GP structure
2. Train one GP for each variable separately and merge results into the complete GP
3. Check length-scales

Code

```
% 1. Set up GP inputs and targets
Du = length(policy.maxU); Da = length(plant.angi); % no. of ctrl and angles
xaug = [x(:,dyno) x(:,end-Du-2*Da+1:end-Du)]; % x augmented with angles
dynmodel.inputs = [xaug(:,dyni) x(:,end-Du+1:end)]; % use dyni and ctrl
dynmodel.targets = y(:,dyno);
dynmodel.targets(:,difi) = dynmodel.targets(:,difi) - x(:,dyno(difi));

dynmodel.hyp = zeros([nVar+nU+2, nVar]);

% 2. For each variable, train a separate GP until SNR falls below the limit
for k=1:nVar
    disp(['Training GP for variable ' num2str(k)]);
    auxDynmodel.fcn = dynmodel.fcn;
    auxDynmodel.train = dynmodel.train;
    auxDynmodel.induce = dynmodel.induce;
    auxDynmodel.inputs = dynmodel.inputs;
    auxDynmodel.targets = dynmodel.targets(:,observationIdx(k));
    auxDynmodel.hyp = dynmodel.hyp(:,k);

    auxDynmodel = auxDynmodel.train(auxDynmodel, plant, trainOpt);

    Xh = auxDynmodel.hyp;
    SNR = exp(Xh(end-1)-Xh(end));
    disp(['SNR = ' num2str(SNR)]);
    while SNR>500
        disp('SNR too high. Add noise and re-train.');
```

```
        SNR = exp(Xh(end-1)-Xh(end));
        disp(['Now SNR is ' num2str(SNR)]);
    end

% 2.1. Merge the trained hyper-parameters with the complete GP
dynmodel.hyp(:,k) = auxDynmodel.hyp;

end

% 3. Show the result of various tests based on the model's length-scales
Xh = dynmodel.hyp;
log_signal_std = (Xh(end-1,:) < log(0.1)) + (Xh(end-1,:) > log(10));
if any(log_signal_std)
    fprintf('Helicopter model log-signal-std hyperparameter is out of range for va
end

input_std = std(dynmodel.inputs);
if any(Xh(1:16,:) > 100*repmat(input_std',[1, nVar]))
    fprintf('Length-scales are much bigger than input std\n');
end

if any(Xh(1:16,:) < 0.01*repmat(input_std',[1, nVar]))
    fprintf('Length-scales are much smaller than input std\n');
end
```

Published with MATLAB® R2014a