

NEES/NIS Design Notes

NEES/NIS design based on model provided in class.

You are screen sharing

Stop share

Talking: Nisar Ahmed

1st Statistical Test for KF Performance: NEES Chi-square

- Use “truth model test” (TMT) simulation to assess validity of NEES at every time step over N Monte Carlo runs

TM = High-fidelity system dynamics + sensor model: can include all kinds of non-linearities and other perversions of the actual physical system that we want to consider

```
graph LR; TM[Truth Model (TM) Simulator] -- y_k --> KF[Kalman Filter code]; TM -- x_k (ground truth) --> GT[x_k (ground truth)]; KF -- "(x_k, P_k^+)_{k=1}^N" --> Out1[ ]; GT --> Out2[ ]; Out1 --- Out2;
```

Compute NEES $\epsilon_{x,k}$ and assess dynamical consistency conditions (1)-(3): do results look right?

ASEN 5044: Statistical Estimation for Dynamical Systems 14

From what I can gather TM is the ode45 work we have done. And Kalman Filter code is the Part4Wrapper.m. Do we just want to integrate NEES/NIS testing into Wrapper or make it a functional call?

$\epsilon_{x,k} = e_{x,k}^T (P_k^+)^{-1} e_{x,k} \rightarrow \text{Normalized estimation error squared (NEES) at time } k$

$\epsilon_{y,k} = e_{y,k}^T (S_k)^{-1} e_{y,k} \rightarrow \text{Normalized innovation squared (NIS) at time } k$

Code for ex,k and ey,k:

```
%  
% (NEES) error calculated here, that is,  
%  
% e_x,k = (xhat - xtrue)'*P^+_k*(xhat - xtrue)  
%  
%  
NEESsshist(k) = (xk_truehist(:,k) - mkp1_plus)'*invPkp1*(xk_truehist(:,k) - mkp1_plus);  
  
%  
% (NIS) error calculated here, that is,  
%  
% e_y,k = (y - yhat)'*R^(-1)_k*(y - yhat)  
%  
%  
NISsshist(k) = innov_kp1'*invPyypk1*innov_kp1;
```

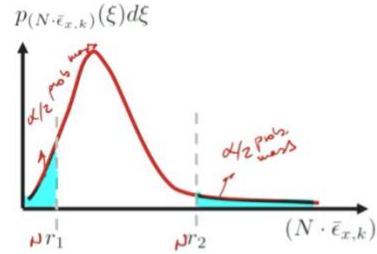
Setting our bounds in the Chi-square distro is done in MATLAB with chi2inv, see slide below.

You are screen sharing
Stop share

Talking: Nisar Ahmed

NEES Chi-square Hypothesis Test

• If we have that the random variable $(N \cdot \bar{\epsilon}_{x,k} \sim \chi^2_{N \cdot n})$, then it follows that $(N \cdot \bar{\epsilon}_{x,k})$ should be limited to $(N \cdot r_1) \leq (N \cdot \bar{\epsilon}_{x,k}) \leq (N \cdot r_2)$ for $(1 - \alpha) \cdot 100\%$ of the time where α is determined by $\int_{N \cdot r_1}^{N \cdot r_2} p_{(N \cdot \bar{\epsilon}_{x,k})}(\xi) d\xi = 1 - \alpha$, for $p_{(N \cdot \bar{\epsilon}_{x,k})}(\xi) d\xi = \chi^2_{N \cdot n}$



χ^2 hyp. test for given sig. level α ,
where α = prob. of declaring KF inconsistent when it is in fact consistent,
i.e. α = Type I error rate / 'False Positive' probability

⇒ This is just a χ^2 hypothesis test!

→ Given N simulation runs of length T steps and corresponding $\bar{\epsilon}_{x,k}$ values for each step $k = 1, 2, \dots, T$, we test the NEES statistics as follows:

*if $\epsilon_{x,k} \in [r_1, r_2]$: declare KF consistent with significance level α

*otherwise: declare KF inconsistent with significance level α

usually: we choose bounds r_1 and r_2 s.t. $\alpha = 0.05$ or 0.01

→ ** in Matlab: $r_1 = \text{chi2inv}(\frac{\alpha}{2}, N \cdot n) ./ N$
 $r_2 = \text{chi2inv}(1 - \frac{\alpha}{2}, N \cdot n) ./ N$

ASEN 5044: Statistical Estimation for Dynamical Systems
15

In Code:

```
%
% N = number of simulation runs
% n = total number of time step's k
%
% r1x and r2x are the lower and upper bounds of the confidence interval for the NEES statistic

% Compute the confidence intervals for the NEES and NIS statistics
r1x = chi2inv(alphaNEES/2, numberOfSimulationRuns*timeStepk) ./ numberOfSimulationRuns;
r2x = chi2inv(1 - alphaNEES/2, numberOfSimulationRuns*timeStepk) ./ numberOfSimulationRuns;

% r1y and r2y are the lower and upper bounds of the confidence interval for the NIS statistic
r1y = chi2inv(alphaNIS/2, numberOfSimulationRuns*timeStepk) ./ numberOfSimulationRuns;
r2y = chi2inv(1 - alphaNIS/2, numberOfSimulationRuns*timeStepk) ./ numberOfSimulationRuns;
```

Calculating mean for epsilon vectors: nis

; $\bar{\epsilon}_{x,k}$ values for *each* step $k = 1, 2, \dots, T$,

In code:

```
% Compute the mean and variance of the NEES and NIS samples  
NEESmean = mean(NEESsamps, 1);  
NISmean = mean(NISSamps, 1);
```

Consistency filter should look like the results on this slide:

Case 1: $Q_{KF} = \text{true } Q$ for system for $W=1$

