ASEN 5044 Statistical Estimation for Dynamical Systems
Fall 2019
Final Project Assignment - Part 2

Out: Tuesday 12/3/2019 (posted on Canvas)

Due: Wednesday 12/18/2019, 5:00 PM

(***GROUP SUBMISSION*** **TO CANVAS!**)

You are to submit solutions as a project group and will receive a group grade (out of 130 total points, not counting the advanced questions). Please clearly and succinctly address each part and each question in your write up – ***as always show all your work and explain your reasoning – you are being graded on demonstrating your understanding of the course material, not just getting things to work.*** *Remember that good basic programming practices will save you a lot of time/effort, and neatness, completeness and clarity in your submission will go a long way to helping your grade. Please limit the write up and explanation of your results (main text and results figures) to between 15-30 pages (this does not include code – this may be attached as an Appendix or submitted as a separate file).* ***Good luck!!***

Part I. BASIC SYSTEM ANALYSIS: [30 pts] Report your results for HW 8 Problem 2 parts (a)-(c) here. Note that this is your chance to rectify any mistakes/oversights or misunderstandings from HW 8. Unlike on HW 8, your answers for all parts will be graded this time (i.e. for more than just completion), so be sure to provide clear, concise, and comprehensive answers to all parts.

Part II. NONLINEAR FILTERING:

1. [40 pts] Implement and tune a linearized KF using the specified nominal state trajectory (along with any control input values that are required in addition to nominal inputs[1]) and covariance matrix values posted on Canvas for the DT nonlinear AWGN process noise and measurement noise for your selected system. Use NEES and NIS chi-square tests based on Monte Carlo truth model test (TMT) simulations to tune and validate your linearized KF's performance (note: be sure to explain how the relevant variables in each test can be adapted to the linearized KF). Choose a sufficiently large number of Monte Carlo runs and sample trajectory simulation length to perform the tests, and choose the α value for running each test (provide some justification for your choices).

Explain how you tuned your linearized KF's guess of the process noise, and provide appropriate plots/illustrations to show that your filter is working properly, namely:

i. Plots for a single 'typical' simulation instance, showing the noisy simulated ground truth states, noisy simulated data, and resulting linearized KF state estimation errors.

[1] this should apply for the Skycrane system only; the Cooperative Localization and Orbit problems require no additional control inputs apart from the nominal values

ii. Plots of the NEES test statistic points from all Monte Carlo simulations vs. time, comparing the resulting averages to computed upper/lower bounds for the NEES chi-square test.

iii. Plots of the NIS test statistic points from all Monte Carlo simulations vs. time, comparing the resulting averages to computed upper/lower bounds for the NIS chi-square test.

Note that:

- to do TMT, you will have to generate multiple sets of simulated ground truth trajectories using the nonlinear dynamics models (with process noise included along the same nominal trajectory) and corresponding measurements (with measurement noise included) using the nonlinear measurement models. The simulated measurements will be used as input to your linearized KF to produce state estimates and predicted measurements for the NEES and NIS tests, respectively.
- for each test trajectory sample in the NEES and NIS tests, you should initialize the filter with exactly the same initial perturbation state estimate $\delta\hat{x}(0)$ and covariance $P^+(0)$, and use these to randomly instantiate the ground truth state $x(0)$ at the start of each full nonlinear trajectory simulation for truth model testing. So, this means will need to come up with reasonable values for $\delta\hat{x}(0)$ and $P^+(0)$ to tune and test against (you actually may want to check that your filter works consistently well for multiple values of $\delta\hat{x}(0)$ and $P^+(0)$, but you only need to report one set of values that you think are representative for your application).
- your linearized KF will need to be tuned with a certain 'guessed' process noise covariance Q_{KF} , which in general will not be the same as the actual process noise in the full nonlinear TMT simulation (in reality, the latter would be unknown and thus 'hidden' from your filter).

Hint: The example code posted for NEES and NIS tests for linear KFs in the lecture notes might be useful to look at. Keep in mind that since you are dealing with linearized models to examine the δx states, your full state estimates and recorded measurements (and hence state estimation errors and full measurement innovations) need to account for the contribution of the nominal state trajectory that you are linearizing about at each time step.

1. [40 pts] Implement and tune an extended KF (EKF) using the specified nominal state trajectory along with the control input values and covariance matrix values posted on Canvas for the DT nonlinear process noise and measurement noise for your selected system. Use NEES and NIS chi-square tests based on Monte Carlo truth model test (TMT) simulations to tune and validate your EKF's performance (be sure to explain how the relevant variables in each test can be adapted to the EKF). Choose a sufficiently large number of Monte Carlo runs and sample trajectory simulation length to perform the tests, and choose the α value for running each test (provide some justification for your choices).

Explain how you tuned your EKF's guess of the process noise, and provide appropriate plots/illustrations to show that your filter is working properly, namely:

i. Plots for a single 'typical' simulation instance, showing the noisy simulated ground truth states, noisy simulated data, and resulting EKF state estimation errors for each state (with 2σ bounds).

ii. Plots of the NEES test statistic points from all Monte Carlo simulations vs. time, comparing the resulting averages to computed upper/lower bounds for the NEES chi-square test.

iii. Plots of the NIS test statistic points from all Monte Carlo simulations vs. time, comparing the resulting averages to computed upper/lower bounds for the NIS chi-square test.

Hint: Many of the same notes/hints given above for the linearized KF apply here for the EKF, with some obvious changes, e.g. you must pick and consistently initialize with $\hat{x}^+(0)$ for the total state instead of just the perturbation state, and account for the fact that you are estimating the total state and looking at the total measurement innovation for NEES and NIS assessments in the EKF (i.e. not just state/measurement perturbations).

3. [20 pts] Implement your linearized KF and EKF to estimate the state trajectory for the observation data log posted for your system on Canvas. Turn in plots of the estimated states and 2σ (make sure these are legible/readable). Compare your results – do you think the linearized KF or EKF does a better job (justify)?

Advanced Questions *Extra credit - partial credit only applies if you submit good work on time.* ***You can submit this as part your project assignment writeup.***

C1. [10 pts] Write a haiku about estimation. Alternative forms of short poetry, verse, etc. are also acceptable. [2]

C2. [30 pts] Implement the Unscented Kalman Filter (UKF, aka the sigma point filter or SPF) for your system, using the data log provided. Explain how you tuned the filter, and perform NIS tests to validate the performance. Compare the results to those obtained using the linearized KF and EKF, and comment on the results.

[2] must be G-rated/family-friendly, so no limericks about frustrated people from Nantucket...