ASEN 5044 Statistical Estimation for Dynamical Systems
Fall 2022

Final Project Assignment

Out: Thursday 11/17/2022 (posted on Canvas + Gradescope)

Due: Tuesday 12/13/2022, 11:59 pm

(***GROUP SUBMISSION* TO GRADESCOPE!!!**)

You are to submit solutions as a project group and will receive a group grade (out of 100 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****. Please adhere to the following guidelines:*

- *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.*
- *Limit the write up and explanation of your results (main text and results figures) to between 15-30 pages (this does not include code or cover/contribution pages or advanced questions – any/all code may be attached as an Appendix at the end of the report). Label all plots and legends appropriately and report units.* ***All reports must be typed/typeset and submitted via Gradescope as a group (self-signup).***
- *Please list team member names on the cover page, and provide a short list of each team member's specific contributions on the first page of the report.*
- *Use and improve upon the material and feedback from the two interim progress reports to help save you time –* ***do not wait until the last minute to write up the final report!***

There are two parts to this final assignment for the system that you and your partner have selected. Provide clear, concise, and comprehensive answers to both parts.

Good luck!!

Part I. ***DETERMINISTIC SYSTEM ANALYSIS*** [20 pts - 3 questions]:

1. Find the required CT Jacobians needed to obtain CT linearized model parameters. Show the key steps and variables needed to find the Jacobian matrices and state the sizes of the results. DO NOT use a symbolic solver or software to find the Jacobians (though you may use these to do a final check of answers).

2. Linearize your system about its specified equilibrium/nominal operating point (given in the description) and find the corresponding DT linearized model matrices (from the corresponding DT nonlinear model Jacobians) for a suitable sampling time (use $\Delta T = 10$ sec for the orbit determination problem; use $\Delta T = 0.1$ sec otherwise). **If applicable**, discuss the observability and stability properties of your time-invariant system approximation around the linearization point (if you have a time-varying result, then note this and skip the analysis).

3. Simulate the linearized DT dynamics and measurement models near the linearization point for your system, assuming a reasonable initial state perturbation from the linearization point (report the perturbation you chose) and assuming no process noise, measurement noise, or control input perturbations. Use the results to compare and validate your Jacobians and DT model against a full nonlinear simulation of the system dynamics and measurements using ode45 in Matlab (or a similar numerical integration routine), starting from the same initial conditions for the total state vector and again assuming no process noise, no measurement noise, and no additional control inputs. Provide suitable labeled plots to report and compare your resulting states and measurements from the linearized DT and full nonlinear DT model. (For the orbit determination problem: simulate at least one full orbit period; otherwise, simulate at least 400 time steps).

Part II. ***STOCHASTIC NONLINEAR FILTERING*** [80 pts - 3 questions]:

4. [30 pts] Implement and tune a linearized KF using the specified nominal state trajectory (with nominal control inputs) and covariance matrix values posted on Canvas for the DT AWGN process noise and measurement noise for your selected nonlinear 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 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:

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

b. Plots of the NEES test statistic points from all Monte Carlo simulations vs. time, comparing resulting averages to computed upper/lower bounds for the NEES chi-square test. Explain precisely how the test was set up and how you interpreted the results.

c. Plots of the NIS test statistic points from all Monte Carlo simulations vs. time, comparing resulting averages to computed upper/lower bounds for the NIS chi-square test. Explain precisely how the test was set up and how you interpreted the results.

Some useful hints/tips:

- 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 $\hat{\delta x}^{+}(0)$ 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 $\hat{\delta x}^{+}(0)$ and $P^{+}(0)$ to tune and test against (you actually may want to check that your filter works consistently well for multiple values of $\hat{\delta x}^{+}(0)$ 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).

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 for the linearized KF, your full state estimates and recorded measurements (and hence state estimation errors and full measurement innovations) need to correctly account for the contribution of the nominal state trajectory that you are linearizing about at each time step (so, be sure to explain how you correctly accounted for this in the NEES and NIS tests).

5. [30 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 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:

a. 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).

b. 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.

c. 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.

Hints/tips: 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).

6. [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. ***Submit this as part your project assignment writeup (i.e. DO NOT e-mail/submit this part separately).***

AQ13. (10 pts) Write a haiku about estimation. Alternative forms of short poetry, verse, etc. are also acceptable. [1]

AQ14. (20 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.

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