[ASEN 5044]
Statistical Estimation of Dynamical Systems

Progress Report 1

Fall 2020

Group Members:

Joshua Malburg

Junior Sundar

Task Breakdown

For this project, we are going for an even allocation of tasks between the members. The breakdown is as shown in the table below:

Question Number	Allocated to
1	Joshua
2	Joshua
3	Junior
4	Joshua
5	Junior
6	Joshua/Junior

Project Status

We have completed Part 1, questions #1 through #3, and have both started working on Part 2, filter implementation and tuning. The following pages capture what we will be submitting for Part 1 of the final report. For question #3 our simulation plots match the plots provided in the progress report assignment. At this time we do not have any technical questions, but would like feedback on Part 1: is our submission adequate to receive full points? Are there any areas / questions we should elaborate on, etc?

Question #1

Continuous-time System Equations

The non-linear equations of motions of our UGV-UAV two-agent system are provided in the problem description as:

$$\dot{\xi}_g = v_g \cos\theta_g + \tilde{\omega}_{x,g} \tag{1}$$

$$\dot{\eta}_g = v_g \sin\theta_g + \tilde{\omega}_{y,g} \tag{2}$$

$$\dot{\theta}_g = \frac{v_g}{L}\tan\phi_g + \tilde{\omega}_{\omega,g} \tag{3}$$

$$\dot{\xi}_a = v_a \cos\theta_a + \tilde{\omega}_{x,a} \tag{4}$$

$$\dot{\eta}_a = v_a \sin\theta_a + \tilde{\omega}_{y,a} \tag{5}$$

$$\dot{\theta}_a = \omega_a + \tilde{\omega}_{\omega,a} \tag{6}$$

The inputs $[v_g, \phi_g, v_a, \omega_a]^T$ to our system are the UGV linear velocity (m/s), UGV steering angle (rad), UAV linear velocity (m/s) and UAV angular rate (rad/s). Our state vector $[\xi_g, \eta_g, \theta_g, \xi_a, \eta_a, \theta_a]^T$ is comprised of the easting position (m), northing position (m) and heading angle (rad) for both the UGV and UAV; each state equation is assumed to be corrupted by AWGN. For measurements we are provide the UAV easting and northing position along with the UGV-UAV relative azimuth angles and range; the output sensing equations are then:

$$\theta_{ga} = \tan^{-1}\left(\frac{\eta_a - \eta_g}{\xi_a - \xi_g}\right) - \theta_g + \tilde{v}_{\theta,ga} \tag{7}$$

$$r = \sqrt{(\xi_g - \xi_a)^2 + (\eta_g - \eta_a)^2} + \tilde{v}_r \tag{8}$$

$$\theta_{ag} = \tan^{-1}\left(\frac{\eta_g - \eta_a}{\xi_g - \xi_a}\right) - \theta_a + \tilde{v}_{\theta,ag} \tag{9}$$

$$\xi_a + \tilde{v}_{\xi,a} \tag{10}$$

$$\eta_a + \tilde{v}_{\eta,a} \tag{11}$$

Our system can be expressed in standard non-linear state-space form by stacking the NL state equations and measurements from above in to $\mathcal{F}$ and h matrices:

$$\dot{x} = \begin{bmatrix} \mathcal{F}_1(x,u,w,t) \\ \vdots \\ \mathcal{F}_n(x,u,w,t) \end{bmatrix} \tag{12}$$

$$y = \begin{bmatrix} h_1(x,u,v,t) \\ \vdots \\ h_n(x,u,v,t) \end{bmatrix} \tag{13}$$

To find the linear CT perturbation model of our system we linearize about the nominal operation point provided in the problem description and find the partial derivatives / Jacobians (see Appendix A for supporting derivations):

$$\left.\frac{\delta \mathcal{F}}{\delta x}\right|_{nom} = \begin{bmatrix} \frac{\delta \mathcal{F}_1}{\delta x_1} & \cdots & \frac{\delta \mathcal{F}_1}{\delta x_n} \\ \vdots & \ddots & \vdots \\ \frac{\delta \mathcal{F}_n}{\delta x_1} & \cdots & \frac{\delta \mathcal{F}_n}{\delta x_n} \end{bmatrix} = \begin{bmatrix} 0 & 0 & -u_1 \sin(x_3) & 0 & 0 & 0 \\ 0 & 0 & u_1 \cos(x_3) & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -u_3 \sin(x_6) \\ 0 & 0 & 0 & 0 & 0 & u_3 \cos(x_6) \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{14}$$

$$\left.\frac{\delta \mathcal{F}}{\delta u}\right|_{nom} = \begin{bmatrix} \frac{\delta \mathcal{F}_1}{\delta u_1} & \cdots & \frac{\delta \mathcal{F}_1}{\delta u_n} \\ \vdots & \ddots & \vdots \\ \frac{\delta \mathcal{F}_n}{\delta u_1} & \cdots & \frac{\delta \mathcal{F}_n}{\delta u_n} \end{bmatrix} = \begin{bmatrix} \cos(x_3) & 0 & 0 & 0 \\ \sin(x_3) & 0 & 0 & 0 \\ \frac{1}{L}\tan\phi_g & \frac{u_1}{L}(\tan^2(u_2+1)) & 0 & 0 \\ 0 & 0 & \cos(x_6) & 0 \\ 0 & 0 & \sin(x_6) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{15}$$

$$\left.\frac{\delta h}{\delta u}\right|_{nom} = \begin{bmatrix} \frac{\delta h_1}{\delta u_1} & \cdots & \frac{\delta h_1}{\delta u_n} \\ \vdots & \ddots & \vdots \\ \frac{\delta h_n}{\delta u_1} & \cdots & \frac{\delta h_n}{\delta u_n} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{16}$$

$$\left.\frac{\delta h}{\delta x}\right|_{nom} = \begin{bmatrix} \frac{\delta h_1}{\delta x_1} & \cdots & \frac{\delta h_1}{\delta x_n} \\ \vdots & \ddots & \vdots \\ \frac{\delta h_n}{\delta x_1} & \cdots & \frac{\delta h_n}{\delta x_n} \end{bmatrix}$$
$$= \begin{bmatrix} \frac{x_5 - x_2}{(x_1-x_4)^2+(x_2-x_5)^2} & \frac{x_1 - x_4}{(x_1-x_4)^2+(x_2-x_5)^2} & -1 & \frac{x_2 - x_5}{(x_1-x_4)^2+(x_2-x_5)^2} & \frac{x_4 - x_1}{(x_1-x_4)^2+(x_2-x_5)^2} & 0 \\ \frac{x_1 - x_4}{\sqrt{(x_1-x_4)^2+(x_2-x_5)^2}} & \frac{x_2 - x_5}{\sqrt{(x_1-x_4)^2+(x_2-x_5)^2}} & 0 & \frac{x_4 - x_1}{\sqrt{(x_1-x_4)^2+(x_2-x_5)^2}} & \frac{x_5 - x_2}{\sqrt{(x_1-x_4)^2+(x_2-x_5)^2}} & 0 \\ \frac{x_5 - x_2}{(x_1-x_4)^2+(x_2-x_5)^2} & \frac{x_1 - x_4}{(x_1-x_4)^2+(x_2-x_5)^2} & 0 & \frac{x_2 - x_5}{(x_1-x_4)^2+(x_2-x_5)^2} & \frac{x_4 - x_1}{(x_1-x_4)^2+(x_2-x_4)^2} & -1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{17}$$

The resulting CT linear matrices F, G and H are $n \times n$, $n \times m$ and $n \times p$, where n equals the number of states (6), m equals the number of inputs (4) and p equals the number of measurements (5).

Question #2

Linear Discrete-time System Equations

If our time-step is small we can use Euler integration to approximate the state transition function which enables us to define the DT linear matrices as a function of the CT Jacobians found in Question #1. For the provided nominal state vector $x_{nom} = \left[10, 0, \frac{\pi}{2}, -60, 0, -\frac{\pi}{2}\right]^T$ and input vector $\left[2, -\frac{\pi}{18}, 12, \frac{\pi}{25}\right]^T$ our DT linearized matrices are then:

$$\tilde{F}_k = I + \triangle T \frac{\delta \mathcal{F}}{\delta x}\bigg|_{nom[k]} = \begin{bmatrix} 1 & 0 & -0.2 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1.2 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{18}$$

$$\tilde{G}_k = \triangle T \frac{\delta \mathcal{F}}{\delta u}\bigg|_{nom[k]} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0.1 & 0 & 0 & 0 \\ -0.0353 & 0.4124 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -0.1 & 0 \\ 0 & 0 & 0 & 0.1 \end{bmatrix} \tag{19}$$

$$\widetilde{H}_k = \frac{\delta h}{\delta x}\bigg|_{nom[k]} = \begin{bmatrix} 0 & 0.0143 & -1 & 0 & -0.0143 & 0 \\ 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0.0143 & 0 & 0 & -0.0143 & -1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{20}$$

Our system is not time-invariant because our matrices are a function of input and state and therefore the nominal point is different at each time step.

Question #3

DT Nonlinear Model

We simulate the nonlinear model using the '`ode45()`' function on MATLAB. We define the nonlinear dynamics model function in the code provided in Appendix C '`NL_DynModel`'. The resulting NL state dynamics simulation assuming no process and measurement noise is shown in Figure 1. Here we set the initial state equal to the specification provided:

$$x(0) = \begin{bmatrix} 10 & 0 & \frac{\pi}{2} & -60 & 0 & -\frac{\pi}{2} \end{bmatrix}^T \tag{21}$$

This represents the nominal state trajectory.

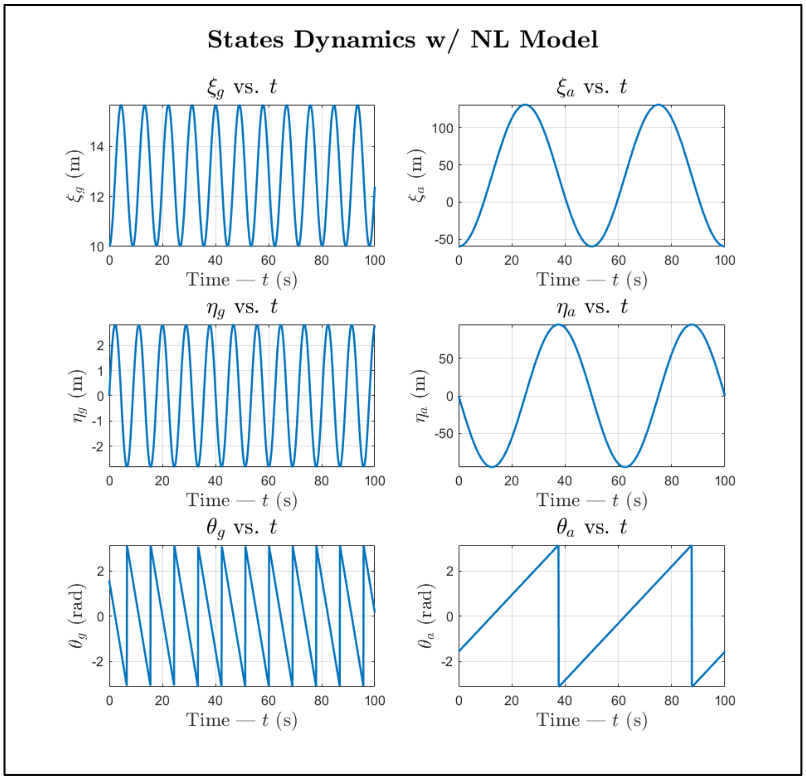

Figure 1 – State dynamics simulation with nonlinear model (using ode45)

Note here that the angles θ_g and θ_a have been wrapped to within $[-\pi, \pi]$.

The nonlinear measurement model function is defined in Appendix C '`NL_MeasModel`'. This function takes in the states at step $'k'$ and outputs the sensor readings.

The resulting NL measurement dynamics without process and measurement noise is shown in Figure 2. This represents the nominal measurements trajectory.

Again the angles γ_{ag} and γ_{ga} have been wrapped within $[-\pi, \pi]$.

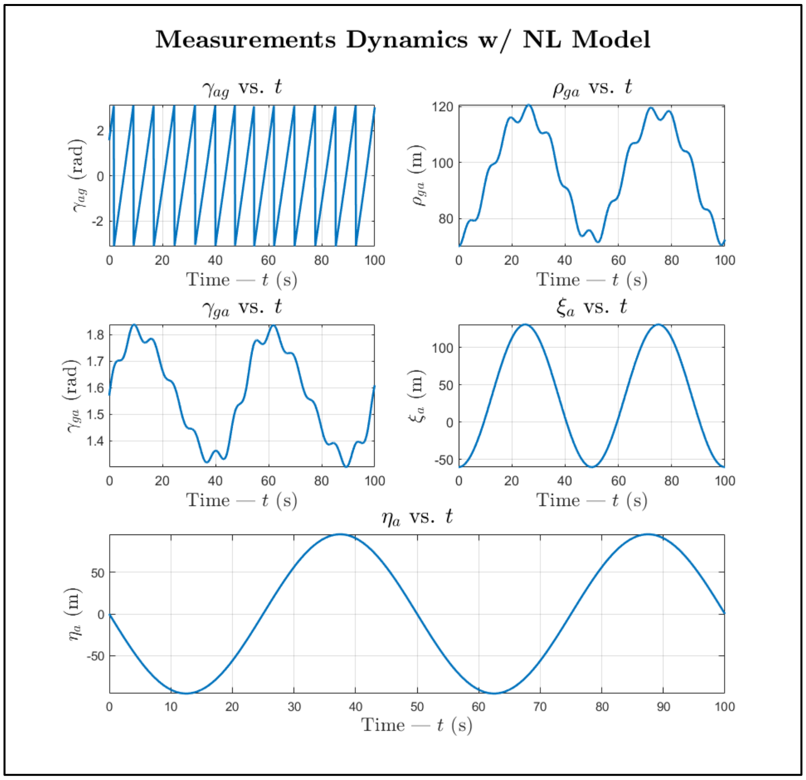

Figure 2 – Measurement dynamics simulation with nonlinear model

DT Linearized Model

We simulate the linearized DT model using the methodology prescribed in Question #1 and Question #2. To perform the linearization, we define the following from specification:

$$x_{nom}(0) = \begin{bmatrix} 10 & 0 & \frac{\pi}{2} & -60 & 0 & -\frac{\pi}{2} \end{bmatrix}^T \tag{22}$$

$$u_{nom} = \begin{bmatrix} 2 & -\frac{\pi}{18} & 12 & \frac{\pi}{25} \end{bmatrix}^T \tag{23}$$

$$\delta x(0) = [0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0.1]^T \tag{24}$$

The transition matrices for the DT linearization model at step ‘k’ are calculated in Appendix C ‘`Linearize`’ using $x_{nom}(k)$ and u_{nom}. The linearized state and measurement dynamics and perturbations are calculated in Appendix C ‘`DT_L_Model`’. $\delta x(0)$ in Equation (24) was selected because its small enough to ensure the linearization does not deviate from nominal trajectory

The graph in Figure 3 plots the state evolution of the linearized DT model. Although the state evolution closely matches the NL model’s nominal trajectory, there is in fact a perturbation from the nominal trajectory shown in Figure 5. The graph in Figure 4 plots the measurement dynamics of the linearized DT model and the sensor readings perturbations are shown in Figure 6.

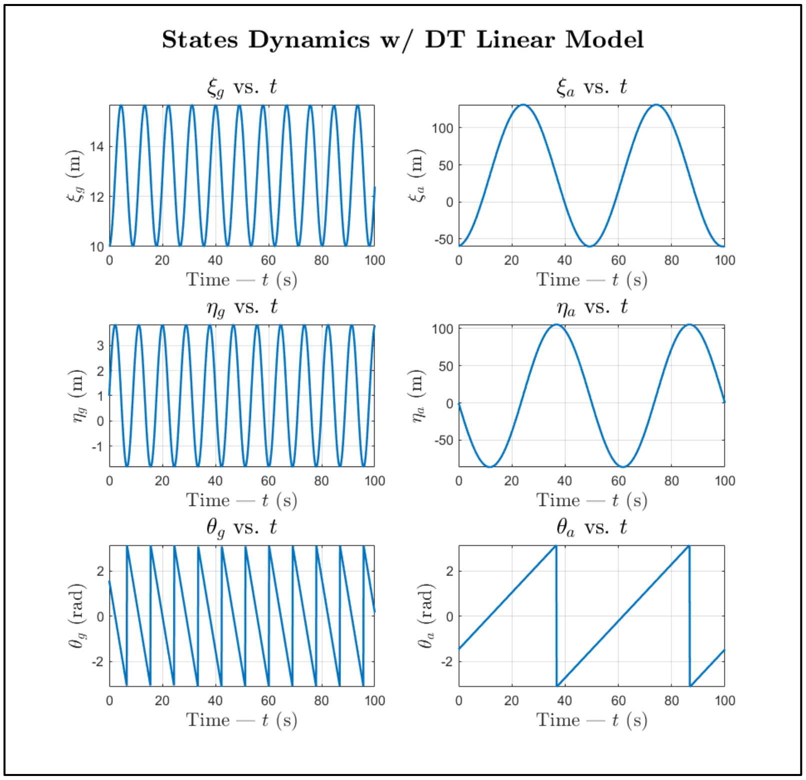

Figure 3 – State dynamics simulation with DT linearized model

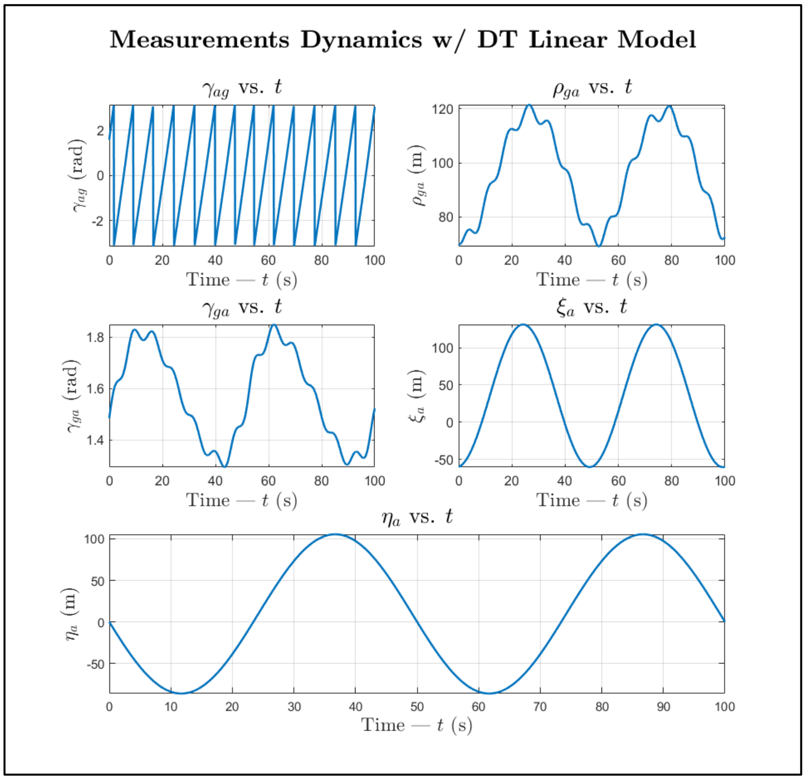

Figure 4 –Measurement dynamics simulation with DT linearized model

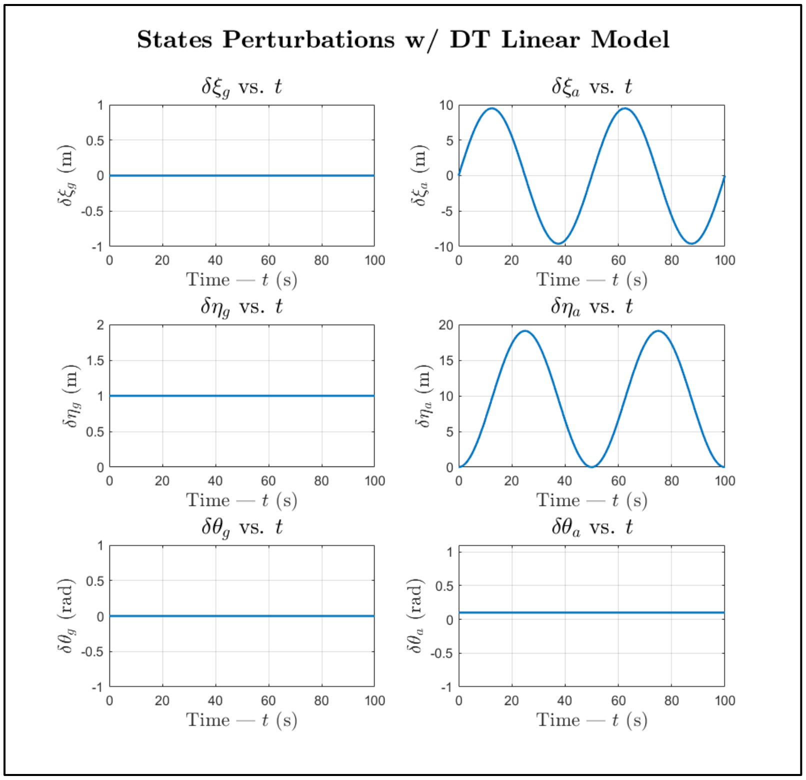

Figure 5 – State dynamics perturbations with DT linearized model

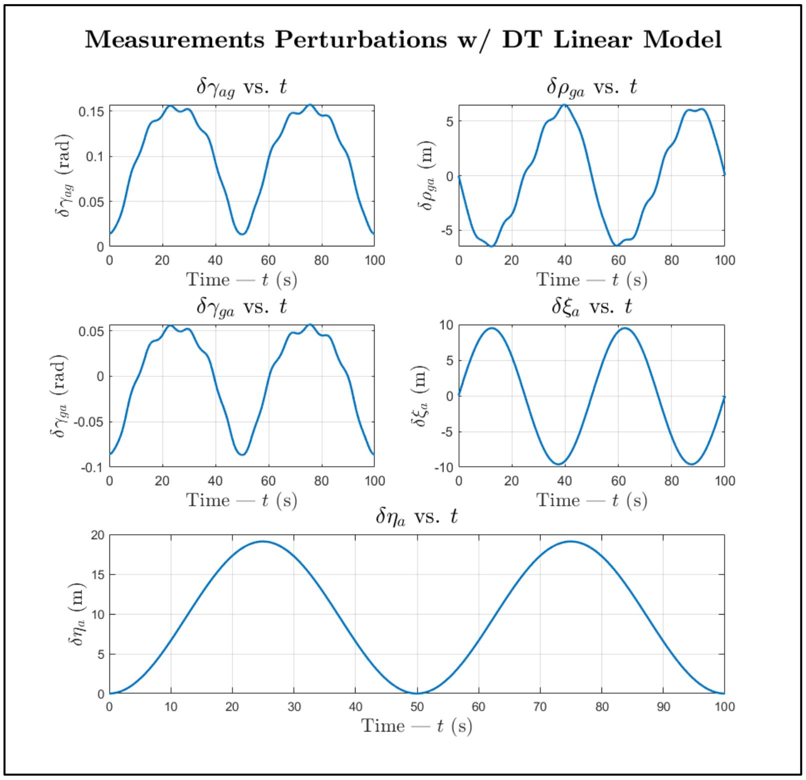

Figure 6 - Measurement dynamics perturbations with DT linearized model\

The state perturbation graphs show that ξ_g, η_g, θ_g and θ_a are constant and unchanging while ξ_a and η_a are varying in an oscillatory manner. This perturbation is transferred through to the sensor readings as well as all the sensor outputs are calculated using ξ_a and/or η_a.

We can conclude that the linearization only results in varying degrees of perturbation in the UAV's states from their nominal trajectory while the UGV's states deviate by a constant value equal to the initial perturbation. And as a result, the sensor readings relating to UAV's states contain major perturbations, while the relative sensor readings have a lower amplitude in their variation.

Appendix A – Supporting derivation for Jacobians

$\frac{\partial h_1}{dx_1} = \frac{\partial}{dx_1}\left(\tan^{-1}\left(\frac{x_5 - x_2}{x_4 - x_1}\right) - x_3 + \tilde{v}_{\theta,ga}\right) = \frac{\partial}{dx_1}\left(\tan^{-1}\left((x_5 - x_2)(-x_1 + x_4)^{-1}\right)\right)$	(A.1)
$\frac{\partial h_1}{dx_2} = \frac{\partial}{dx_2}\left(\tan^{-1}\left(\frac{x_5 - x_2}{x_4 - x_1}\right) - x_3 + \tilde{v}_{\theta,ga}\right) = \frac{\partial}{dx_2}\left(\tan^{-1}\left(\frac{-1}{x_4 - x_1}x_2 + \frac{x_5}{x_4 - x_1}\right)\right)$	(A.2)
$\frac{\partial h_1}{dx_4} = \frac{\partial}{dx_4}\left(\tan^{-1}\left(\frac{x_5 - x_2}{x_4 - x_1}\right) - x_3 + \tilde{v}_{\theta,ga}\right) = \frac{\partial}{dx_4}\left(\tan^{-1}\left((x_5 - x_2)(x_4 - x_1)^{-1}\right)\right)$	(A.3)
$\frac{\partial h_1}{dx_5} = \frac{\partial}{dx_5}\left(\tan^{-1}\left(\frac{x_5 - x_2}{x_4 - x_1}\right) - x_3 + \tilde{v}_{\theta,ga}\right) = \frac{\partial}{dx_5}\left(\tan^{-1}\left(\frac{1}{x_4 - x_1}x_5 + \frac{-x_2}{x_4 - x_1}\right)\right)$	(A.4)
$\frac{\partial h_2}{dx_1} = \frac{\partial}{dx_1}\left(\sqrt{(x_1 - x_4)^2 + (x_2 - x_5)^2} + \tilde{v}_r\right) = \frac{\partial}{dx_1}\left(\sqrt{(x_1 - x_4)^2 + (x_2 - x_5)^2}\right)$	(A.5)
$\frac{\partial h_2}{dx_1} = \frac{\partial}{dx_2}\left(\sqrt{(x_1 - x_4)^2 + (x_2 - x_5)^2} + \tilde{v}_r\right) = \frac{\partial}{dx_2}\left(\sqrt{(x_1 - x_4)^2 + (x_2 - x_5)^2}\right)$	(A.6)
$\frac{\partial h_2}{dx_1} = \frac{\partial}{dx_4}\left(\sqrt{(x_1 - x_4)^2 + (x_2 - x_5)^2} + \tilde{v}_r\right) = \frac{\partial}{dx_4}\left(\sqrt{(-x_4 + x_1)^2 + (x_2 - x_5)^2}\right)$	(A.7)
$\frac{\partial h_2}{dx_1} = \frac{\partial}{dx_5}\left(\sqrt{(x_1 - x_4)^2 + (x_2 - x_5)^2} + \tilde{v}_r\right) = \frac{\partial}{dx_5}\left(\sqrt{(x_1 - x_4)^2 + (-x_5 + x_2)^2}\right)$	(A.8)
$\frac{\partial h_3}{dx_1} = \frac{\partial}{dx_1}\left(\tan^{-1}\left(\frac{x_2 - x_5}{x_1 - x_4}\right) - x_6 + \tilde{v}_{\theta,ag}\right) = \frac{\partial}{dx_1}\left(\tan^{-1}\left((x_2 - x_5)(x_1 - x_4)^{-1}\right)\right)$	(A.9)
$\frac{\partial h_3}{dx_2} = \frac{\partial}{dx_2}\left(\tan^{-1}\left(\frac{x_2 - x_5}{x_1 - x_4}\right) - x_6 + \tilde{v}_{\theta,ag}\right) = \frac{\partial}{dx_2}\left(\tan^{-1}\left(\frac{1}{x_1 - x_4}x_2 + \frac{-x_5}{x_1 - x_4}\right)\right)$	(A.10)
$\frac{\partial h_3}{dx_4} = \frac{\partial}{dx_4}\left(\tan^{-1}\left(\frac{x_2 - x_5}{x_1 - x_4}\right) - x_6 + \tilde{v}_{\theta,ag}\right) = \frac{\partial}{dx_4}\left(\tan^{-1}\left((x_2 - x_5)(-x_4 + x_1)^{-1}\right)\right)$	(A.11)
$\frac{\partial h_3}{dx_5} = \frac{\partial}{dx_5}\left(\tan^{-1}\left(\frac{x_2 - x_5}{x_1 - x_4}\right) - x_6 + \tilde{v}_{\theta,ag}\right) = \frac{\partial}{dx_5}\left(\tan^{-1}\left(\frac{-1}{x_1 - x_4}x_5 + \frac{x_2}{x_1 - x_4}\right)\right)$	(A.12)

All solutions for equations A.2, A.4, A.10, A.12 all are of the generic form:

$$\frac{\partial}{dx_a}\left(\tan^{-1}(ax_a + b)\right) = \frac{1}{1 + (ax_a + b)^2}\frac{\partial}{dx_a}(ax_a + b) = \frac{a}{1 + (ax_a + b)^2}$$

All solutions for equations A.1, A.3, A.9, A.11 all are of the generic form:

$$\frac{\partial}{dx_a}\left(\tan^{-1}(c(ax_a + b)^{-1})\right) = \frac{1}{1 + (c(ax_a + b)^{-1})^2}\frac{\partial}{dx_a}(c(ax_a + b)^{-1}) =$$

$$\frac{-ac}{1+(c(ax_a+b)^{-1})^2}(ax_a+b)^{-2} = \frac{\frac{-ca}{(ax_a+b)^2}}{1+\left(\frac{c}{ax_a+b}\right)^2} = \frac{\frac{-ca}{(ax_a+b)^2}}{\left(\frac{(ax_a+b)^2+c}{ax_a+b}\right)^2}$$

$$= \frac{-ca}{(ax_a+b)^2+c^2}$$

All solutions for equations A5 through A8 all are of the generic form:

$$\frac{\partial}{dx}(c+(ax+b)^2)^{1/2} = \frac{(c+(ax+b)^2)^{-1/2}}{2}\frac{\partial}{dx}((ax+b)^2)$$

$$= \frac{(c+(ax+b)^2)^{-1/2}}{2}(2a)(ax+b) = \frac{a(ax+b)}{\sqrt{c+(ax+b)^2}}$$

The rest of the math was done by hand, substituting the parameters (a,b,c) and simplifying:

$$\frac{a}{1+(ax_a+b)^2}$$

$\boxed{A.2}$ $a = \frac{-1}{x_4-x_1}$ $b = \frac{x_5}{x_4-x_1}$ $x_a = x_2$

$$\frac{\frac{-1}{x_4-x_1}}{1+\left[\left(\frac{-1}{x_4-x_1}\right)x_2+\frac{x_5}{x_4-x_1}\right]^2} = \frac{\frac{-1}{x_4-x_1}}{1+\left(\frac{x_5-x_2}{x_4-x_1}\right)^2} = \frac{\frac{-1}{\cancel{(x_4-x_1)}}}{\frac{(x_4-x_1)^2+(x_5-x_2)^2}{(x_4-x_1)^{\cancel{2}}}}$$

$$= \frac{-(x_4-x_1)}{(x_4-x_1)^2+(x_5-x)^2}$$

$\boxed{A.4}$ $a = \frac{1}{x_4-x_1}$ $b = \frac{-x_2}{x_4-x_1}$ $x_a = x_5$

$$\frac{\frac{1}{x_4-x_1}}{1+\left[\frac{1}{(x_4-x_1)}x_5+\left(\frac{-x_2}{x_4-x_1}\right)\right]^2} = \frac{\frac{1}{x_4-x_1}}{1+\frac{(x_5-x_2)^2}{(x_4-x_1)^2}} = \frac{\frac{1}{\cancel{(x_4-x_1)}}}{\frac{(x_4-x_1)^2+(x_5-x_2)^2}{(x_4-x_1)^{\cancel{2}}}} = \frac{(x_4-x_1)}{(x_4-x_1)^2+(x_5-x_2)^2}$$

$\boxed{A.10}$ $a = \frac{1}{x_1-x_4}$ $b = \frac{-x_5}{x_1-x_4}$ $x = 2$

$$\frac{\frac{1}{x_1-x_4}}{1+\left[\frac{1}{(x_1-x_4)}x_2+\frac{-x_5}{(x_1-x_4)}\right]^2} = \frac{\frac{1}{x_1-x_4}}{1+\frac{(x_2-x_5)^2}{(x_1-x_4)^2}} = \frac{\frac{1}{\cancel{x_1-x_4}}}{\frac{(x_1-x_4)^2+(x_2-x_5)^2}{(x_1-x_4)^{\cancel{2}}}} = \frac{x_1-x_4}{(x_1-x_4)^2+(x_2-x_5)^2}$$

$\boxed{A.12}$ $a = \left(\frac{-1}{x_1-x_4}\right)$ $b = \frac{x_2}{x_1-x_4}$ $x = x_5$

$$\frac{\frac{-1}{x_1-x_4}}{1+\left[\left(\frac{-1}{x_1-x_4}\right)x_5+\left(\frac{x_2}{x_1-x_4}\right)\right]^2} = \frac{\frac{-1}{x_1-x_4}}{1+\frac{(x_2-x_5)^2}{(x_1-x_4)^2}} = \frac{\frac{-1}{\cancel{x_1-x_4}}}{\frac{(x_1-x_4)^2+(x_2-x_5)^2}{(x_1-x_4)^{\cancel{2}}}} = \frac{-(x_1-x_4)}{(x_1-x_4)^2+(x_2-x_5)^2}$$

$$\frac{-ca}{(ax+b)^2+c^2}$$

1) $a=-1 \quad b=x_4 \quad c=(x_5-x_2) \quad x=x_1$

$$\frac{+(x_5-x_2)(+1)}{(-x_1+x_4)^2+(x_5-x_2)^2} = \frac{x_5-x_2}{(x_4-x_1)^2+(x_5-x_2)^2}$$

3 $a=1 \quad b=-x_1 \quad c=x_5-x_2 \quad x=x_4$

$$\frac{(-1)(x_5-x_2)(1)}{(x_4+-x_1)^2+(x_5-x_2)^2} = \frac{x_2-x_5}{(x_4-x_1)^2+(x_5-x_2)}$$

A.9 $a=1 \quad b=-x_4 \quad c=x_2-x_5 \quad x=x_1$

$$\frac{(-1)(x_2-x_5)(1)}{(x_1-x_4)^2+(x_2-x_5)^2} = \frac{x_5-x_2}{(x_1-x_4)^2+(x_2-x_5)^2}$$

A.11 $a=-1 \quad b=x_1 \quad c=(x_2-x_5) \quad x=x_4$

$$\frac{(-1)(x_2-x_5)(-1)}{(-1x_4+x_1)^2+(x_2-x_5)^2} = \frac{(x_2-x_5)}{(x_1-x_4)^2+(x_2-x_5)^2}$$

$$\frac{a(ax+b)}{\sqrt{c+(ax+b)^2}}$$

A5 $a=1 \quad b=-x_4 \quad c=(x_2-x_5)^2 \quad x=x_1$

$$\frac{(1)(x_1+-x_4)}{\sqrt{(x_2-x_5)^2+(x_1-x_4{}^2)}}$$

A6 $a=1 \quad b=-x_5 \quad c=(x_1-x_4)^2 \quad x=x_2$

$$\frac{(1)(x_2+-x_5)}{\sqrt{(x_1-x_4)^2+(x_2-x_5)^2}}$$

A7 $a=-1 \quad b=x_1 \quad c=(x_2-x_5)^2 \quad x=x_4$

$$\frac{(-1)((-1)x_4+x_1)}{\sqrt{(x_2-x_5)^2+(x_4-x_1)^2}} = \frac{x_4-x_1}{\sqrt{(x_2-x_5)^2+(x_4-x_1)^2}}$$

A8 $a=-1 \quad b=x_2 \quad c=(x_1-x_4)^2 \quad x=x_5$

$$\frac{(-1)((-1)x_5+x_2)}{\sqrt{(x_1-x_4)^2+(x_5-x_2)^2}} = \frac{x_5-x_2}{\sqrt{(x_1-x_4)^2+(x_5-x_2)^2}}$$

Appendix C – Codes accompanying Question 3

```
function xdot = NL_DynModel(t,x,u,Pnoise)
%NL_DynModel
%   input: t - time model; x - state vector; u - control input vector;
%          Pnoise - process noice vector
%   output: ___
%   Function input for ode45 for NL dynamics model

% u = [v_g, phi_g, v_a, w_a]';
% x = [xi_g eta_g theta_g xi_a eta_a theta_a]';
% ~w = [w_x,g w_y,g w_w,g w_x,a w_y,a w_w,a]';
L = 0.5;

xdot = [u(1)*cos(x(3)) + Pnoise(1);
        u(1)*sin(x(3)) + Pnoise(2);
        (u(1)/L)*(tan(u(2))) + Pnoise(3);
        u(3)*cos(x(6)) + Pnoise(4);
        u(3)*sin(x(6)) + Pnoise(5);
        u(4) + Pnoise(6)];
end
```

```
function [y] = NL_MeasModel(x,Mnoise)
%NL_MeasModel
%   input: x - state vector; Mnoise - measurement noice vector
%   output: y - sensor readings;
%   Uses NL state inputs and measurement noise vector to get sensor
%   readings

% y = [gamma_ag rho_ga gamma_ga xi_a eta_a]';
% x = [xi_g eta_g theta_g xi_a eta_a theta_a]';
% x = [1    2     3       4    5     6      ]';

y = [atan2(x(5)-x(2),x(4)-x(1)) - x(3);
     sqrt((x(1)-x(4))^2 + (x(2)-x(5))^2);
     atan2(-x(5)+x(2),-x(4)+x(1)) - x(6);
     x(4);
     x(5)];

y = y + Mnoise;

end
```

```
function [A_t,B_t,C_t] = Linearize(x,u)
```

```
%Linearize
%   input: x - nominal state vector; u - nominal control input;
%   output: A_t - A tilde Matrix; B_t - B tilde Matrix; C_t - C tilde
%           Matrix
%   Obtain the CT linearized state perturbation matrices

% u = [v_g, phi_g, v_a, w_a]';
% x = [xi_g eta_g theta_g xi_a eta_a theta_a]';

L = 0.5;

A_t = [0 0 -u(1)*sin(x(3)) 0 0 0;
       0 0 u(1)*cos(x(3))  0 0 0;
       0 0 0               0 0 0;
       0 0 0               0 0 -u(3)*sin(x(6));
       0 0 0               0 0 u(3)*cos(x(6));
       0 0 0               0 0 0];



B_t = [cos(x(3))   0                    0         0;
       sin(x(3))   0                    0         0;
       tan(u(2))/L (u(1)/L)*sec(u(2))^2 0         0;
       0           0                    cos(x(6)) 0;
       0           0                    sin(x(6)) 0;
       0           0                    0         1];

% x = [xi_g eta_g theta_g xi_a eta_a theta_a]';

C11 = (x(5)-x(2))/((x(5)-x(2))^2 + (x(4)-x(1))^2);
C12 = -(x(4)-x(1))/((x(5)-x(2))^2 + (x(4)-x(1))^2);
C13 = -1;
C14 = -(x(5)-x(2))/((x(5)-x(2))^2 + (x(4)-x(1))^2);
C15 = (x(4)-x(1))/((x(5)-x(2))^2 + (x(4)-x(1))^2);
C21 = (x(1)-x(4))*((x(1)-x(4))^2 + (x(2)-x(5))^2)^-0.5;
C22 = (x(2)-x(5))*((x(1)-x(4))^2 + (x(2)-x(5))^2)^-0.5;
C24 = -(x(1)-x(4))*((x(1)-x(4))^2 + (x(2)-x(5))^2)^-0.5;
C25 = -(x(2)-x(5))*((x(1)-x(4))^2 + (x(2)-x(5))^2)^-0.5;
C31 = -(x(2)-x(5))/((x(2)-x(5))^2 + (x(1)-x(4))^2);
C32 = (x(1)-x(4))/((x(2)-x(5))^2 + (x(1)-x(4))^2);
C34 = (x(2)-x(5))/((x(2)-x(5))^2 + (x(1)-x(4))^2);
C35 = -(x(1)-x(4))/((x(2)-x(5))^2 + (x(1)-x(4))^2);
C36 = -1;
```

```
C44 = 1;
C55 = 1;

C_t = [C11 C12 C13 C14 C15 0;
       C21 C22 0   C24 C25 0;
       C31 C32 0   C34 C35 C36;
       0   0   0   C44 0   0;
       0   0   0   0   C55 0];
end
```

```
function     [x_DTL,     dx_DTL,     y_DTL,     dy_DTL,     F,     H,     O]     =
DT_L_Model(t,Dt,x_NL,y_NL,x_pert,u_nom)
%DT_L_Model
%   input: t - time; Dt - time increment; x_NL - nominal state trajectory;
%          y_NL - nominal sensor readings; x_pert - initial state
%          perturbation; u_nom = nominal control input;
%   output: x_DTL = discrete time state; dx_DTL = discrete time state
%           perturbation; y_DTL = discrete time sensor; dy_DTL = discrete
%           time sensor perturbation
%   Function for linear DT model

% u = [v_g, phi_g, v_a, w_a]';
% x = [xi_g eta_g theta_g xi_a eta_a theta_a]';
% ~w = [w_x,g w_y,g w_w,g w_x,a w_y,a w_w,a]';

x_nominal = [x_NL(1,:); x_NL(2,:);
             wrapToPi(x_NL(3,:)); x_NL(4,:);
             x_NL(5,:); wrapToPi(x_NL(6,:))];
y_nominal = y_NL;

F = zeros(6,6,length(t));
H = zeros(5,6,length(t));
O = zeros(6,6,length(t));

% Evaluate F and H matrices with predef nominal state trajectories
for i=1:length(t)
    [A_t, ~, C_t] = Linearize(x_nominal(:,i),u_nom);
    F(:,:,i) = eye(6) + A_t*Dt;
    H(:,:,i) = C_t;
    O(:,:,i) = Dt*eye(6,6);
end

dx_DTL = zeros(6,length(t));
```

```
dy_DTL = zeros(5,length(t));
x_DTL = zeros(6,length(t));
y_DTL = zeros(5,length(t));
dx_DTL(:,1) = x_pert;
dy_DTL(:,1) = H(:,:,1)*x_pert;
x_DTL(:,1) = x_nominal(:,1)+dx_DTL(:,1);
y_DTL(:,1) = y_nominal(:,1)+dy_DTL(:,1);

for i=2:length(t)
    dx_DTL(:,i) = F(:,:,i)*dx_DTL(:,i-1);
    dy_DTL(:,i) = H(:,:,i)*dx_DTL(:,i);
    x_DTL(:,i) = x_nominal(:,i)+dx_DTL(:,i);
    y_DTL(:,i) = y_nominal(:,i)+dy_DTL(:,i);
end
end
```