Statistical Estimation for Dynamical Systems: Final Project
Progress Report 1

Patrick M. Bodine-Ellison*
University of Colorado Boulder, Boulder, CO, 80309

Robert Bacon†
University of Colorado Boulder, Boulder, CO, 80309

Sayan Chatterjee‡
University of Colorado Boulder, Boulder, CO, 80309

I. Spitting Up of Work

Each team member is completing the math and programming of each section independently with periodic meetings to help each other work through sticking points. This helps each team member learn the full scope of the lesson provided by the assignment. Team members then split up into subsections to actually write the report into the LaTeXdocument.

II. Deterministic System Analysis

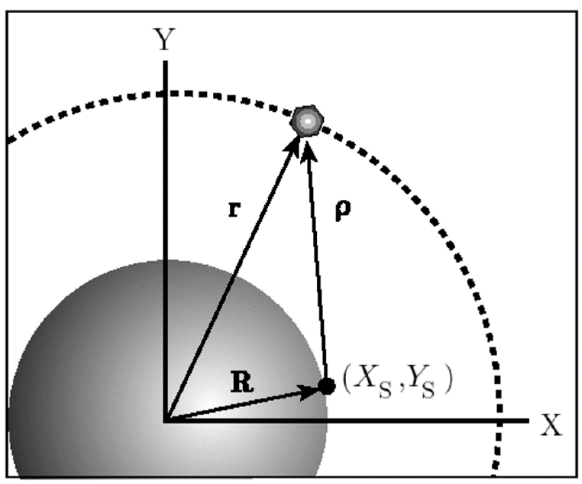

Fig. 1 Basic orbit determination problem setup

In this project we will consider a spacecraft orbiting in a 2D plane at an altitude of 300km with motion defined in an Earth centered inertial (ECI) x and y position defined by $X(t)$ and $Y(t)$ and ECI velocities defined by $\dot{X}(t)$ and $\dot{Y}(t)$. The non-linear equations of motion the define how the spacecraft moves over time are defined as

$$\ddot{X} = -\frac{\mu X}{r^3} + u_1 + \tilde{\mathbf{w}}_1 \qquad \textbf{AND} \qquad \ddot{Y} = -\frac{\mu Y}{r^3} + u_2 + \tilde{\mathbf{w}}_2 \tag{1}$$

where $r = \sqrt{X^2 + Y^2}$ represents the distance of the spacecraft from the ECI coordinate system origin at Earth's center, $\mu = 398800$ km^3/s^{2}is the standard gravitation parameter for Earth, u_1 and u_2 are control accelerations, and $\tilde{w}_1$ and $\tilde{w}_2$ are disturbances (process noise). The full set of states, inputs, and disturbances are defined as

*Graduate Student, Mechanical Engineering, University of Colorado Boulder, Boulder, CO, 80309
†Graduate Student, Aerospace Engineering, University of Colorado Boulder, Boulder, CO, 80309
‡Graduate Student, Aerospace Engineering, University of Colorado Boulder, Boulder, CO, 80309

$$x(t) = \begin{bmatrix} X \\ \dot{X} \\ Y \\ \dot{Y} \end{bmatrix} \quad \textbf{AND} \quad u(t) = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad \textbf{AND} \quad \tilde{\mathbf{w}}(t) = \begin{bmatrix} \tilde{\mathbf{w}}_1 \\ \tilde{\mathbf{w}}_2 \end{bmatrix} \tag{2}$$

The initial condition of the spacecraft is at it's maximum distance in X and maximum velocity in $\dot{Y}$ with the initial state vector where $r_0 = 6678$ is our nominal orbital radius.

$$x(0) = \begin{bmatrix} r_0 \\ 0 \\ 0 \\ r_0\sqrt{\frac{\mu}{r_0^3}} \end{bmatrix} \tag{3}$$

As seen in Fig(1) the spacecraft is sensed by noisy measurements of relative range ($\rho(t)^i$) in km, range rate ($\dot{\rho}^i(t)$) in km/s, and elevation angle ($\phi^i(t)$) from 12 ground stations where $i \in [1, 12]$ the station ID for each station. Each station i is located at position ($X_s^i(t)$,$Y_s^i(t)$) in the orbital plane on the Earth's surface rotating counter clockwise in the XY plane at a rate of $\omega_E = \frac{2\pi}{86400}$ rad/s. The data are modeled as

$$y^i(t) = \begin{bmatrix} \rho^i(t) \\ \dot{\rho}^i(t) \\ \phi^i(t) \end{bmatrix} + \tilde{\mathbf{v}}^i(t) \tag{4}$$

where $\tilde{\mathbf{v}}^i(t) \in \mathbb{R}^3$ is the measurement error vector at station i and

$$\rho^i(t) = \sqrt{(X(t) - X_s^i(t))^2 + (Y(t) - Y_s^i(t))^2} \tag{5}$$

$$\dot{\rho}^i(t) = \frac{[X(t) - X_s^i(t)][\dot{X}(t) - \dot{X}_s^i(t)] + [Y(t) - Y_s^i(t)][\dot{Y}(t) - \dot{Y}_s^i(t)]}{\rho^i(t)} \tag{6}$$

$$\phi^i(t) = \tan^{-1}\left(\frac{Y(t) - Y_s^i(t)}{X(t) - X_s^i(t)}\right) \tag{7}$$

The tracking station position are perfectly known for all time t such that

$$X_s^i(t) = R_E\cos(\omega_E t + \theta^i(0)) \quad \textbf{AND} \quad Y_s^i(t) = R_E\sin(\omega_E t + \theta^i(0)) \tag{8}$$

where $R_E = 6378$ km is the radius of the Earth and $w_E = \frac{p\pi}{86400}$ rad/s, as previously defined, is the Earth rotation rate. Each station is initially located at

$$(X_s^i(0), Y_s^i(0)) = (R_E\cos(\theta^i(0)), R_E\sin(\theta^i(0))) \quad \textbf{WHERE} \quad \theta^i(0) = (i-1)\cdot\frac{\pi}{6} \tag{9}$$

Since our ground stations cannot sense the spacecraft through the Earth each station can only produce a valid measurement of the spacecraft for a a limited time-varying range of possible $\phi^i(t)$ angles. Station i only generates a data vector $y^i(t)$ at time t if

$$\phi^i(t) \in [-\frac{\pi}{2} + \theta^i(t), \frac{\pi}{2} + \theta^i(t)] \quad \textbf{WHERE} \quad \theta^i(t) = \tan^{-1}\left(\frac{Y_s^i(t)}{X_s^i(t)}\right) \tag{10}$$

For the purpose of recording these measurements in software, all $y^i(t)$ where the spacecraft is out of range of ground station i will be recorded as NaNs.

A. CT Jacobians

To build a linearized system state-space model for our 2-D spacecraft system we must first define the non-linear state-space equations $\mathcal{F}(x, u, \tilde{\mathbf{w}})$ and $h(t, x, u, \tilde{\mathbf{v}}, i)$ as follows

$$\begin{bmatrix} \mathcal{F}_1 \\ \mathcal{F}_2 \\ \mathcal{F}_3 \\ \mathcal{F}_4 \end{bmatrix} = \begin{bmatrix} x_2 \\ -\frac{\mu x_1}{(x_1^2+x_3^2)^{\frac{3}{2}}} + u_1 + \tilde{\mathbf{w}}_1 \\ x_4 \\ -\frac{\mu x_3}{(x_1^2+x_3^2)^{\frac{3}{2}}} + u_2 + \tilde{\mathbf{w}}_2 \end{bmatrix} \quad \textbf{AND} \quad \begin{bmatrix} h_1 \\ h_2 \\ h_3 \end{bmatrix} = \begin{bmatrix} \sqrt{(x_1 - X_s^i(t))^2 + (x_3 - Y_s^i(t))^2} \\ \frac{[x_1 - X_s^i(t)][x_2 - \dot{X}_s^i(t)] + [x_3 - Y_s^i(t)][x_4 - \dot{Y}_s^i(t)]}{\sqrt{(x_1 - X_s^i(t))^2 + (x_3 - Y_s^i(t))^2}} \\ \tan^{-1}\left(\frac{x_3 - Y_s^i(t)}{x_1 - X_s^i(t)}\right) \end{bmatrix} \tag{11}$$

We can then build our CT Jacobian matrices $\bar{A}$, $\bar{B}$, $\bar{C}^i$, and $\bar{D}^i$ as follows. Note that there is a separate $\bar{C}^i$ and $\bar{D}^i$ matrix for each station i.

$$\bar{A} = \begin{bmatrix} \frac{\partial \mathcal{F}_1}{\partial x_1} & \frac{\partial \mathcal{F}_1}{\partial x_2} & \frac{\partial \mathcal{F}_1}{\partial x_3} & \frac{\partial \mathcal{F}_1}{\partial x_4} \\ \frac{\partial \mathcal{F}_2}{\partial x_1} & \frac{\partial \mathcal{F}_2}{\partial x_2} & \frac{\partial \mathcal{F}_2}{\partial x_3} & \frac{\partial \mathcal{F}_2}{\partial x_4} \\ \frac{\partial \mathcal{F}_3}{\partial x_1} & \frac{\partial \mathcal{F}_3}{\partial x_2} & \frac{\partial \mathcal{F}_3}{\partial x_3} & \frac{\partial \mathcal{F}_3}{\partial x_4} \\ \frac{\partial \mathcal{F}_4}{\partial x_1} & \frac{\partial \mathcal{F}_4}{\partial x_2} & \frac{\partial \mathcal{F}_4}{\partial x_3} & \frac{\partial \mathcal{F}_4}{\partial x_4} \end{bmatrix} \quad \bar{B} = \begin{bmatrix} \frac{\partial \mathcal{F}_1}{\partial u_1} & \frac{\partial \mathcal{F}_1}{\partial u_2} \\ \frac{\partial \mathcal{F}_2}{\partial u_1} & \frac{\partial \mathcal{F}_2}{\partial u_2} \\ \frac{\partial \mathcal{F}_3}{\partial u_1} & \frac{\partial \mathcal{F}_3}{\partial u_2} \\ \frac{\partial \mathcal{F}_4}{\partial u_1} & \frac{\partial \mathcal{F}_4}{\partial u_2} \end{bmatrix} \quad \bar{C}^i = \begin{bmatrix} \frac{\partial h_1}{\partial x_1} & \frac{\partial h_1}{\partial x_2} & \frac{\partial h_1}{\partial x_3} & \frac{\partial h_1}{\partial x_4} \\ \frac{\partial h_2}{\partial x_1} & \frac{\partial h_2}{\partial x_2} & \frac{\partial h_2}{\partial x_3} & \frac{\partial h_2}{\partial x_4} \\ \frac{\partial h_3}{\partial x_1} & \frac{\partial h_3}{\partial x_2} & \frac{\partial h_3}{\partial x_3} & \frac{\partial h_3}{\partial x_4} \end{bmatrix} \quad \bar{D}^i = \begin{bmatrix} \frac{\partial h_1}{\partial u_1} & \frac{\partial h_1}{\partial u_2} \\ \frac{\partial h_2}{\partial u_1} & \frac{\partial h_2}{\partial u_2} \\ \frac{\partial h_3}{\partial u_1} & \frac{\partial h_3}{\partial u_2} \end{bmatrix} \tag{12}$$

First, we can quickly evaluate the values in these Jacobian matrices that go to 0 or 1 to yield

$$\bar{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{\partial \mathcal{F}_2}{\partial x_1} & 0 & \frac{\partial \mathcal{F}_2}{\partial x_3} & 0 \\ 0 & 0 & 0 & 1 \\ \frac{\partial \mathcal{F}_4}{\partial x_1} & 0 & \frac{\partial \mathcal{F}_4}{\partial x_3} & 0 \end{bmatrix} \quad \bar{B} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \quad \bar{C}^i = \begin{bmatrix} \frac{\partial h_1}{\partial x_1} & 0 & \frac{\partial h_1}{\partial x_3} & 0 \\ \frac{\partial h_2}{\partial x_1} & \frac{\partial h_2}{\partial x_2} & \frac{\partial h_2}{\partial x_3} & \frac{\partial h_2}{\partial x_4} \\ \frac{\partial h_3}{\partial x_1} & 0 & \frac{\partial h_3}{\partial x_3} & 0 \end{bmatrix} \quad \bar{D}^i = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{13}$$

We will have to individually evaluate the partial derivatives for the remaining terms. Starting with $\frac{\partial \mathcal{F}_2}{\partial x_1}$

$$\frac{\partial \mathcal{F}_2}{\partial x_1} = \frac{\partial}{\partial x_1}\left(-\frac{\mu x_1}{(x_1^2 + x_3^2)^{\frac{3}{2}}}\right) \tag{14}$$

Applying the quotient rule

$$\frac{\partial \mathcal{F}_2}{\partial x_1} = -\frac{\frac{\partial}{\partial x_1}(\mu x_1)(x_1^2 + x_3^2)^{\frac{3}{2}} - \frac{\partial}{\partial x_1}((x_1^2 + x_3^2)^{\frac{3}{2}})(\mu x_1)}{(x_1^2 + x_3^2)^3} \tag{15}$$

Further evaluating

$$\boxed{\frac{\partial \mathcal{F}_2}{\partial x_1} = -\frac{(\mu)(x_1^2 + x_3^2)^{\frac{3}{2}} - \mu 3 x_1^2 (x_1^2 + x_3^2)^{\frac{1}{2}}}{(x_1^2 + x_3^2)^3}} \tag{16}$$

Fortunately, $\frac{\partial \mathcal{F}_4}{\partial x_3}$ has the same form and evaluates the same way to yield

$$\boxed{\frac{\partial \mathcal{F}_4}{\partial x_3} = -\frac{(\mu)(x_1^2 + x_3^2)^{\frac{3}{2}} - \mu 3 x_3^2 (x_1^2 + x_3^2)^{\frac{1}{2}}}{(x_1^2 + x_3^2)^3}} \tag{17}$$

We can then move on to $\frac{\partial \mathcal{F}_4}{\partial x_1}$

$$\frac{\partial \mathcal{F}_4}{\partial x_1} = \frac{\partial}{\partial x_1}\left(-\frac{\mu x_3}{(x_1^2 + x_3^2)^{\frac{3}{2}}}\right) \tag{18}$$

Applying the chain rule yields

$$\frac{\partial \mathcal{F}_4}{\partial x_1} = \left(\frac{3\mu x_3}{2(x_1^2 + x_3^2)^{\frac{5}{2}}}\right)\frac{\partial}{\partial x_1}\left(x_1^2 + x_3^2\right) \tag{19}$$

Which evaluates to

$$\boxed{\frac{\partial \mathcal{F}_4}{\partial x_1} = \frac{3\mu x_3 x_1}{(x_1^2 + x_3^2)^{\frac{5}{2}}}} \tag{20}$$

$\frac{\partial \mathcal{F}_2}{\partial x_3}$ is evaluated the same way yielding the exact same result

$$\boxed{\frac{\partial \mathcal{F}_2}{\partial x_3} = \frac{3\mu x_3 x_1}{(x_1^2 + x_3^2)^{\frac{5}{2}}}} \tag{21}$$

That gives us all of the values we need to complete the $\bar{A}$ Jacobian matrix so we can move on to the elements of the $\bar{C}^i$ Jacobian matrix. First evaluating $\frac{\partial h_1}{\partial x_1}$

$$\frac{\partial h_1}{\partial x_1} = \frac{\partial}{\partial x_1}\left(\sqrt{(x_1 - X_s^i(t))^2 + (x_3 - Y_s^i(t))^2}\right) \tag{22}$$

Using the chain rule yields

$$\frac{\partial h_1}{\partial x_1} = \frac{1}{2}\left((x_1 - X_s^i(t))^2 + (x_3 - Y_s^i(t))^2\right)^{-\frac{1}{2}}\frac{\partial}{\partial x_1}(x_1 - X_s^i(t))^2 \tag{23}$$

$$\boxed{\frac{\partial h_1}{\partial x_1} = \frac{x_1 - X_s^i(t)}{\sqrt{(x_1 - X_s^i(t))^2 + (x_3 - Y_s^i(t))^2}}} \tag{24}$$

$\frac{\partial h_1}{\partial x_3}$ has a similar form and evaluates to

$$\boxed{\frac{\partial h_1}{\partial x_3} = \frac{x_3 - Y_s^i(t)}{\sqrt{(x_1 - X_s^i(t))^2 + (x_3 - Y_s^i(t))^2}}} \tag{25}$$

Next, we can evaluate all of the $\frac{\partial h_2}{\partial x}$ derivatives there are two forms here, the x_2 and x_4 derivatives are much simpler while the x_1 and x_3 derivatives are more complex. We can start with the simpler ones

$$\frac{\partial h_2}{\partial x_2} = \frac{\partial}{\partial x_2}\left(\frac{[x_1 - X_s^i(t)][x_2 - \dot{X}_s^i(t)] + [x_3 - Y_s^i(t)][x_4 - \dot{Y}_s^i(t)]}{\sqrt{(x_1 - X_s^i(t))^2 + (x_3 - Y_s^i(t))^2}}\right) \tag{26}$$

This derivative is extremely simple and evaluates to

$$\boxed{\frac{\partial h_2}{\partial x_2} = \frac{[x_1 - X_s^i(t)]}{\sqrt{(x_1 - X_s^i(t))^2 + (x_3 - Y_s^i(t))^2}}} \tag{27}$$

$\frac{\partial h_2}{\partial x_4}$ takes a similar form with

$$\boxed{\frac{\partial h_2}{\partial x_4} = \frac{[x_3 - Y_s^i(t)]}{\sqrt{(x_1 - X_s^i(t))^2 + (x_3 - Y_s^i(t))^2}}} \tag{28}$$

Then the more complicated derivatives

$$\frac{\partial h_2}{\partial x_1} = \frac{\partial}{\partial x_1}\left(\frac{[x_1 - X_s^i(t)][x_2 - \dot{X}_s^i(t)] + [x_3 - Y_s^i(t)][x_4 - \dot{Y}_s^i(t)]}{\sqrt{(x_1 - X_s^i(t))^2 + (x_3 - Y_s^i(t))^2}}\right) \tag{29}$$

For ease of derivation this can be separated into two derivatives

$$\frac{\partial h_2}{\partial x_1} = \frac{\partial}{\partial x_1}\left(\frac{[x_1 - X_s^i(t)][x_2 - \dot{X}_s^i(t)]}{\sqrt{(x_1 - X_s^i(t))^2 + (x_3 - Y_s^i(t))^2}}\right) + \frac{\partial}{\partial x_1}\left(\frac{[x_3 - Y_s^i(t)][x_4 - \dot{Y}_s^i(t)]}{\sqrt{(x_1 - X_s^i(t))^2 + (x_3 - Y_s^i(t))^2}}\right) \tag{30}$$

The term on the right is familiar to us from the derivation of $\frac{\partial h_1}{\partial x_1}$

$$\frac{\partial h_2}{\partial x_1} = \frac{\partial}{\partial x_1}\left(\frac{[x_1 - X_s^i(t)][x_2 - \dot{X}_s^i(t)]}{\sqrt{(x_1 - X_s^i(t))^2 + (x_3 - Y_s^i(t))^2}}\right) - \frac{[x_1 - X_s^i(t)][x_3 - Y_s^i(t)][x_4 - \dot{Y}_s^i(t)]}{((x_1 - X_s^i(t))^2 + (x_3 - Y_s^i(t))^2)^{\frac{3}{2}}} \tag{31}$$

Using the product rule we can then find the left term to be

$$\begin{aligned}\frac{\partial h_2}{\partial x_1} = {} & \frac{[x_2 - \dot{X}_s^i(t)]}{\sqrt{(x_1 - X_s^i(t))^2 + (x_3 - Y_s^i(t))^2}} - \\ & \frac{[x_1 - X_s^i(t)]^2[x_2 - \dot{X}_s^i(t)]}{((x_1 - X_s^i(t))^2 + (x_3 - Y_s^i(t))^2)^{\frac{3}{2}}} - \frac{[x_1 - X_s^i(t)][x_3 - Y_s^i(t)][x_4 - \dot{Y}_s^i(t)]}{((x_1 - X_s^i(t))^2 + (x_3 - Y_s^i(t))^2)^{\frac{3}{2}}}\end{aligned} \tag{32}$$

$\frac{\partial h_2}{\partial x_3}$ will have a similar form

$$\begin{aligned}\frac{\partial h_2}{\partial x_3} = {} & \frac{[x_4 - \dot{Y}_s^i(t)]}{\sqrt{(x_1 - X_s^i(t))^2 + (x_3 - Y_s^i(t))^2}} - \\ & \frac{[x_3 - Y_s^i(t)]^2[x_4 - \dot{Y}_s^i(t)]}{((x_1 - X_s^i(t))^2 + (x_3 - Y_s^i(t))^2)^{\frac{3}{2}}} - \frac{[x_3 - Y_s^i(t)][x_1 - X_s^i(t)][x_2 - \dot{X}_s^i(t)]}{((x_1 - X_s^i(t))^2 + (x_3 - Y_s^i(t))^2)^{\frac{3}{2}}}\end{aligned} \tag{33}$$

Finally, we can look at the $\frac{\partial h_3}{\partial x}$ derivatives

$$\frac{\partial h_3}{\partial x_1} = \frac{\partial}{\partial x_1}\left(\tan^{-1}\left(\frac{x_3 - Y_s^i(t)}{x_1 - X_s^i(t)}\right)\right) \tag{34}$$

$$\frac{\partial h_3}{\partial x_1} = \left(\frac{1}{\left(\frac{x_3 - Y_s^i(t)}{x_1 - X_s^i(t)}\right)^2 + 1}\right)\frac{\partial}{\partial x_1}\left(\frac{x_3 - Y_s^i(t)}{x_1 - X_s^i(t)}\right) \tag{35}$$

$$\frac{\partial h_3}{\partial x_1} = -\left(\frac{1}{\left(\frac{x_3 - Y_s^i(t)}{x_1 - X_s^i(t)}\right)^2 + 1}\right)\left(\frac{x_3 - Y_s^i(t)}{(x_1 - X_s^i(t))^2}\right) \tag{36}$$

$$\boxed{\frac{\partial h_3}{\partial x_1} = -\frac{x_3 - Y_s^i(t)}{(x_3 - Y_s^i(t))^2 + (x_1 - X_s^i(t))^2}} \tag{37}$$

Now the next

$$\frac{\partial h_3}{\partial x_3} = \frac{\partial}{\partial x_1}\left(\tan^{-1}\left(\frac{x_3 - Y_s^i(t)}{x_1 - X_s^i(t)}\right)\right) \tag{38}$$

$$\frac{\partial h_3}{\partial x_3} = \left(\frac{1}{\left(\frac{x_3 - Y_s^i(t)}{x_1 - X_s^i(t)}\right)^2 + 1} \right) \frac{\partial}{\partial x_3} \left(\frac{x_3 - Y_s^i(t)}{x_1 - X_s^i(t)} \right) \tag{39}$$

$$\frac{\partial h_3}{\partial x_3} = \left(\frac{1}{\left(\frac{x_3 - Y_s^i(t)}{x_1 - X_s^i(t)}\right)^2 + 1} \right) \left(\frac{1}{x_1 - X_s^i(t)} \right) \tag{40}$$

$$\boxed{\frac{\partial h_3}{\partial x_3} = \frac{x_1 - X_s^i(t)}{(x_3 - Y_s^i(t))^2 + (x_1 - X_s^i(t))^2}} \tag{41}$$

B. Linearized DT System

The nominal trajectory for the spacecraft is a prograde equatorial circular orbit at an altitude of 300 km. The spacecraft's initial state is as follows:

$$x_{nom}(0) = \begin{bmatrix} r_0 \\ 0 \\ 0 \\ r_0 n_0 \end{bmatrix} \tag{42}$$

where r_0 is the nominal orbit radius (6678 Km) and $n_0 = \sqrt{\mu / r_o^2}$ is the mean motion of the nominal orbit. Solving the equations of motion in Eq.(1) with this initial condition gives a nominal trajectory of the form

$$x_{nom}(t) = \begin{bmatrix} r_0 \cos(n_0 t) \\ -r_0 n_0 \sin(n_0 t) \\ r_0 \sin(n_0 t) \\ r_0 n_0 \cos(n_0 t) \end{bmatrix} \tag{43}$$

We can now linearize Eq. (1) about this nominal trajectory by taking our continuous time Jacobians (which are functions of x and possibly t) from the previous section and plugging in the nominal trajectory $x_{nom}(t)$ for x. $\bar{B}$, Γ, and $\bar{D}$ are all constant (do not depend on the state x, control input u, or noise w) so stay the same as in section.

$$\bar{A}_{nom}(t) = \bar{A}(x_{nom}(t)) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{\mu}{r_0^3}\left(1 - 3\cos^2(n_0 t)\right) & 0 & \frac{3\mu \sin(2n_0 t)}{2r_0^3} & 0 \\ 0 & 0 & 0 & 1 \\ \frac{3\mu \sin(2n_0 t)}{2r_0^3} & 0 & -\frac{\mu}{r_0^3} \sin(1 - 3\sin^2(n_0 t)) & 0 \end{bmatrix} \tag{44}$$

$$\bar{C}^i_{nom}(t) = \bar{C}^i(t, x_{nom}(t)) = \bar{C}^i(t)|_{x_{nom}} \tag{45}$$

When evaluated about the nominal trajectory, $\bar{A}$ becomes dependant only on time and is periodic with a period of π/n_0 or half the period of the nominal orbit. Due to the time dependence of the ground station positions in the ECI frame, $\bar{C}$ is not periodic in the same way as $\bar{A}$. If the ratio of ω_E and n_0 is rational then $\bar{C}$ is periodic with period $\mathrm{LCM}(2\pi/n_0, 2\pi/\omega_E)$, otherwise it is aperiodic.

To convert the continuous time linearized model parameters to their discrete time counterparts, a simple Euler approximation is used. With the assumptions that the CT model matrices are only slowly time varying (can be treated as constant over the $\Delta t = 10$ second time step) and that Δt is small enough that higher order term can be ignored, the standard matrix exponential method for converting continuous, time invariant A matrices to discrete time can be simplified in the following way to get $\bar{F}[k]$:

$$\bar{F}[k] \approx e^{\bar{A}_{nom}(t_k)\Delta t} = \sum_{j=0}^{\infty} \frac{\Delta t^j \bar{A}_{nom}(t_k)^j}{j!} \approx I + \Delta t \bar{A}_{nom}(t_k) \tag{46}$$

$$\bar{F}[k] \approx \begin{bmatrix} 1 & \Delta t & 0 & 0 \\ -\Delta t \frac{\mu}{r_0^3}\left(1 - 3\cos^2(n_0 t)\right) & 1 & \Delta t \frac{3\mu \sin(2n_0 t)}{2r_0^3} & 0 \\ 0 & 0 & 1 & \Delta t \\ \Delta t \frac{3\mu \sin(2n_0 t)}{2r_0^3} & 0 & -\Delta t \frac{\mu}{r_0^3} \sin(1 - 3\sin^2(n_0 t)) & 1 \end{bmatrix} \tag{47}$$

Similarly,

$$\bar{G}[k] \approx \Delta t \bar{B}_{nom}(t_k) = \Delta t \bar{B} = \begin{bmatrix} 0 & 0 \\ \Delta t & 0 \\ 0 & 0 \\ 0 & \Delta t \end{bmatrix} \tag{48}$$

$$\bar{\Omega}[k] \approx \Delta t \bar{\Gamma}_{nom}(t_k) = \Delta t \bar{\Gamma} = \begin{bmatrix} 0 & 0 \\ \Delta t & 0 \\ 0 & 0 \\ 0 & \Delta t \end{bmatrix} \tag{49}$$

The measurement equations are algebraic and so stay the same from continuous to discrete time:

$$\bar{H}^i[k] = \bar{C}^i_{nom}(t_k) \tag{50}$$

C. Simulated Dynamics

With the derivations out of the way, the nonlinear system dynamics can be modelled using Eq.(11) for the initial conditions perturbed by the vector $[0.0, 0.075, 0.0, -0.021]^T$. Modelling the dynamics requires the use of MATLAB's ODE45 to integrate equations of motion from Eq.(1) without input or process noise.

```
tol = 1e-12;
options = odeset('RelTol',tol,'AbsTol',[tol tol tol tol]);

[nonLinearT,nonLinearState] = ode45(@propDyDt,tVec,state0,options);

function DyDt = propDyDt(t,state)
    mu = 398600;           % [km^3/s^2] Earth Gravitional Parameter
    x1 = state(1);
    x2 = state(2);
    x3 = state(3);
    x4 = state(4);

    r = norm([x1 x3]);

    DyDt = [x2; -mu*x1/r^3; x4; -mu*x3/r^3];
end
```

Graphing the results from ODE45 will yield the following results.

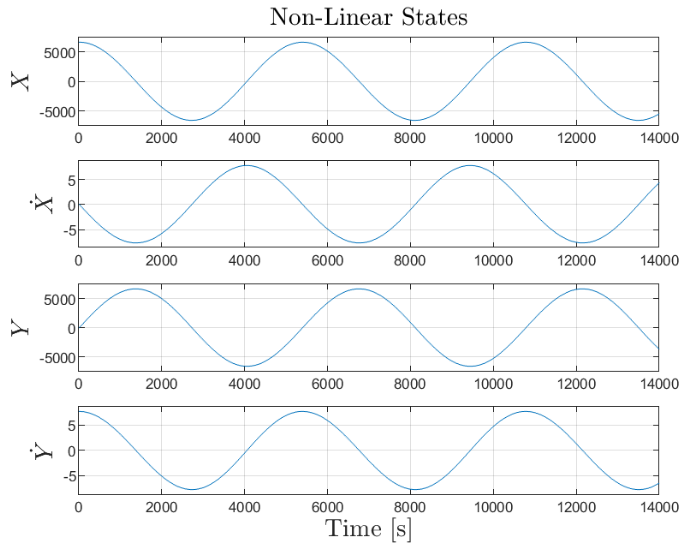

Fig. 2 Non-Linear State Trajectory

Referring back to previous section regarding the linearized DT system, the linearized equations requires the $\bar{A}$ matrix to be evaluated throughout time. Code regarding the evaluation of $\bar{A}$ is listed below.

```
function [A] = findANominal(x,mu)
A = [0,1,0,0;
    -mu*(x(3)^2-2*x(1)^2)/((x(1)^2+x(3)^2)^(5/2)),0,3*mu*x(1)*x(3)/((x(1)^2+x(3)^2)^(5/2)),0;
    0,0,0,1;
    3*mu*x(3)*x(1)/((x(1)^2+x(3)^2)^(5/2)),0,-mu*(x(1)^2-2*x(3)^2)/((x(1)^2+x(3)^2)^(5/2)),0];
end
```

Recall the formulation of a discrete time system requires the evaluation of $\bar{F}_k \approx I + \Delta T \cdot \bar{A}|_{nom[k]}$ which is then applied to $x_{nom}(t)$. Evaluation of the perturbed linear states only requires the addition of the perturbation vector listed before. Sample code is provided below.

```
% Find the Linear Pertub Solution
tVec = linspace(0,14000,1401); % Time Steps
for kk = 1:(length(tVec)-1)
    A = findANominal(xNominal(kk,:)); % Atilde Matrix
    F = eye(4) + detlaT*A;            % F Matrix for DT System
    xLin(kk+1,:) = xNominal(kk+1,:) + (F*pertub(kk,:)')'; % Linearized Trajectory
    pertub(kk+1,:) = F*pertub(kk,:)';                     % Perturbed Trajectory
end
```

Graphing the results will yield the following.

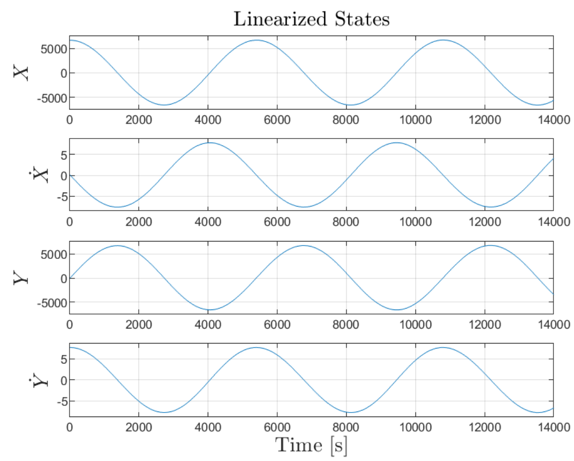

Fig. 3 Linearized State Trajectory

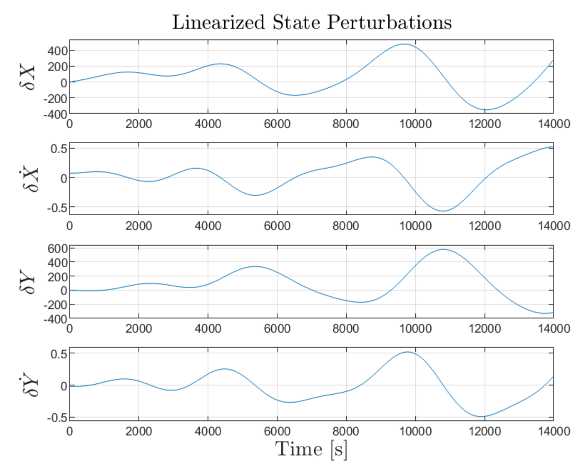

Fig. 4 Linearized State Trajectory with Small Perturbations

Based on the figures above, it is apparent that the linearized model deviates greatly compared to the nonlinear model. Meaning, the nonlinear dynamics predict perturbations better than the linearized dynamics since the linear model was linearized off the nominal trajectory.

Recall back to Eg.(4) regarding the measurements generated by the ground stations as a relationship of relative rate, range rate, and elevation angle. In addition, there are 12 stations divided up across the Earth, as referenced in Eq. (9). The following represents the evaluation of the measurements without measurement errors.

```
for nn = 1:12 % Iterate Through all Stations
        thi = wrapTo2Pi((nn - 1)*pi/6);      % [rad] Theta0 for ground station i
        Xis = Re*cos(we*t(kk)+thi);          % [km] X Position of Station
        Yis = Re*sin(we*t(kk)+thi);          % [km] Y Position of Station
        dXis = -Re*we*sin(we*t(kk)+thi);     % [km/s] Velocity X of Station
        dYis = Re*we*cos(we*t(kk)+thi);      % [km/s] Velocity Y of Station
        x_stat(:,kk,nn) = [Xis;dXis;Yis;dYis];

        thLim = we*t(kk)+thi;

        yNL(:,kk,nn) = [sqrt((xNL(1,kk)-Xis)^2+(xNL(3,kk)-Yis)^2);
                    ((xNL(1,kk)-Xis)*(xNL(2,kk)-dXis)+(xNL(3,kk)-Yis)*(xNL(4,kk)-dYis))/...
                    (sqrt((xNL(1,kk)-Xis)^2+(xNL(3,kk)-Yis)^2));
```

```
                    atan2((xNL(3,kk)-Yis),(xNL(1,kk)-Xis))]; % Form the Measurements Per Station
end
```

Note the prior code block will be evaluated through time. It is important to note that stations should only be able to provide measurements if there is proper visibility to the satellite. Visibility can be found from elevation angle ($\phi^i(t)$) and factoring in which side of the Earth the satellite is in view. The following is added in addition to the MATLAB script above. Basically, if the satellite is not in view, the measurements generated before will be replaced with NaN.

```
inView = and(angdiff(yNL(3,kk,nn),thLim) ≤ pi/2,angdiff(yNL(3,kk,nn),thLim) ≥ -pi/2);
if ¬inView
    yNL(:,kk,nn) = [NaN;NaN;NaN];
    visSatsNL(nn,kk) = NaN;
end
```

By graphing the results, the following figures demonstrates the station measurements for the non-linear and linear trajectories.

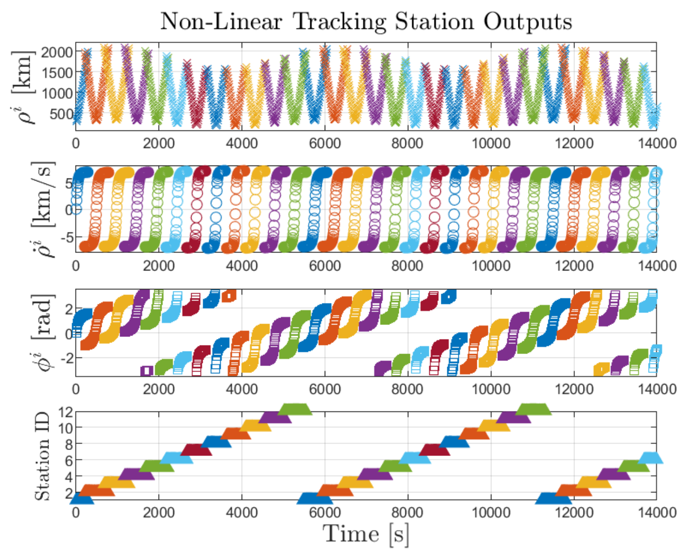

Fig. 5 Overall Station Tracking Non-Linear

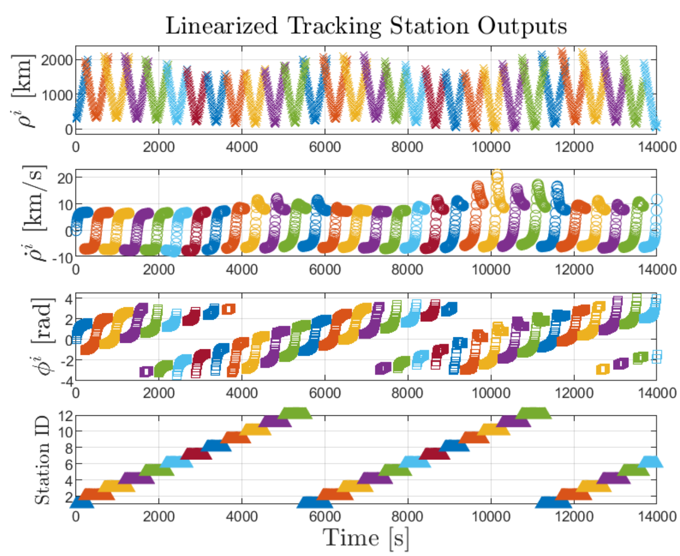

Fig. 6 Overall Station Tracking Linearized

Overall, the graphs are consistent with one another. Again, implying the linearized trajectory provides a a decent approximation to the nonlinear dynamics.