

Ingeniería Industrial
2016/2017

Trabajo Fin de Carrera

**Determinación De Orbita de
Sistemas de Satélites Ligados**

Luis Miguel González Job

Tutor/es

Manuel Sanjurjo Rivo

Carlos Santiuste Romero

Leganés 28/07/2017

Palabras clave: OD; Determinación de Órbita; Orbit Determination; TSS; Tether Satellite System; Sistema de Satélites Ligados.

Resumen: Este trabajo fin de carrera comienza presentando brevemente la historia y problemática de los satélites conectados. Posteriormente se postula el modelo dinámico considerado para los satélites conectados en este trabajo. Una vez definido el modelo dinámico se estudia la determinación de órbita de un satélite conectado, y se explica el método de generación de medidas para este trabajo. Tras esto se introduce una descripción del filtro utilizado seguido de los resultados debidos a la utilización de éste para la determinación de órbita de un satélite conectado conocidas las medidas realizadas sobre éste. Por último, se introduce un capítulo de conclusiones que evalúa el proyecto en general.



[Incluir en el caso de interés en su publicación en el archivo abierto]

Esta obra se encuentra sujeta a la licencia Creative Commons **Reconocimiento - No Comercial - Sin Obra Derivada**

Índice

1. Introducción	3
1.1. Misiones	4
1.1.1. Tether en misiones espaciales humanas:	4
1.1.2. Tethers en misiones con satélites	5
1.1.3. Vuelos de cohetes de sondeo	6
1.1.4. Misiones futuras y propuestas	6
2. Modelo Dinámico	7
2.1. Descripción del movimiento	8
2.2. Geometría de masas	9
2.3. Cinemática	10
2.4. Formulación de Newton-Euler	10
2.4.1. Movimiento del centro de masas	10
2.4.2. Movimiento relativo al centro de masas	11
2.4.3. Par gravitatorio	11
3. Determinación de órbita de un TSS conocido	13
4. Descripción del filtro	16
4.1. Estimación de mínimos cuadrados	16
4.2. Ecuaciones variacionales	17
5. Resultados	20
6. Conclusiones	24
7. Anexo	25
7.1. Main.m	25
7.2. Initial _{Conditions} .m	26
7.3. DataGeneration_v1.m	27
7.4. TSS_OD_v1.m	29
7.5. Filter _{v3} .m	30
7.6. dxdt2.m	33
7.7. dCdt3.m	33
8. Bibliography	36

Índice de figuras

1.	A Way to Stars	3
2.	Configuración esquemática del sistema que muestra los triedros utilizados en el desarrollo.	7
3.	Representación de las proyecciones del vector u en el sistema de referencia de direcciones inerciales y el triedro orbital.	8
4.	Trayectoria real del centro de gravedad del sistema	20
5.	Movimiento relativo	20
6.	Trayectoria de las masas con errores de medida	21
7.	Trayectorias real (azúl) y generada a partir de las medidas (rojo) . .	22
8.	Movimiento relativo: real (azúl) y generado a partir de las medidas (rojo)	22

1. Introducción

La utilización de cables en el espacio fue propuesta por vez primera por Tsiolkovskii en 1895 en su obra *'A Way to Stars'* [1]. En ella plantea la utilidad de estos dispositivos en diferentes aplicaciones espaciales. Por ejemplo, señala su funcionalidad a la hora de generar gravedad artificial, que se consigue uniendo con un cable dos secciones de una estación orbital y haciendo girar al conjunto alrededor de su centro de masas. También expone la posibilidad de realizar maniobras por intercambio de momento angular haciendo uso de un contrapeso ligado al vehículo espacial mediante un cable.

Desde este momento estos sistemas han ido apareciendo en la ciencia ficción como en la novela *'Fountains of Paradise'* [2] donde se especula con la idea original de Tsiolkovskii acerca de la posibilidad de construir una torre que conecte la superficie de la Tierra con la órbita geostacionaria. Los cables espaciales han constituido un área de desarrollo abierta dentro del campo de la tecnología espacial como se puede deducir del número de misiones que se han realizado en este campo, algunas de las cuales están enumeradas en el siguiente capítulo.



Figura 1: A Way to Stars

Las características particulares que presentan estos sistemas hacen necesario un estudio y análisis específico que se cubrirá en los capítulos consecutivos. Existen problemas dinámicos y de control tales como la estabilidad en la actitud de estos sistemas, el despliegue y retirada de los cables, así como las vibraciones de los mismos que han de ser estudiadas en profundidad. Uno de estos aspectos es la determinación de órbitas que estudiamos en este proyecto. La particularidad de este tipo de órbitas es que las secciones del sistema de satélites ligados no sigue un movimiento kepleriano.

El problema de los dos cuerpos [3] explica la determinación preliminar de órbitas clásica aunque no es el la mejor manera de abordar esta clase de sistemas. La determinación de órbitas moderna está basada en métodos de estimación pero no es capaz de proporcionar condiciones iniciales a los estimadores como hace la determinación preliminar de este tipo de sistemas. Por ello, es útil desarrollar un tratamiento *ad hoc* de la información que podemos obtener de un vehículo orbital que se encuentre ligado que nos permita detectar, seguir e identificar esta clase de sistemas. La aproximación a este estudio a partir del problema de los dos cuerpos se puede realizar desde distintos puntos de vista.

La primera manera supone que el centro de masas del sistema ligado completo describe un movimiento kepleriano. Hay que tener en cuenta que los vehículos observables no coinciden con su centro de masas por lo que su movimiento diferirá del obtenido del problema de los dos cuerpos en el movimiento relativo a su centro de masas. Por esto hay que realizar un tratamiento conjunto de ambas.

La segunda manera es considerar que son dos cuerpos independientes que sufren una fuerza de perturbación debida a la tensión del cable. De esta forma es posible utilizar los métodos clásicos con alguna modificación.

En este trabajo se propone el estudio de un sistema de satélites ligados haciendo uso de un modelo dinámico sencillo. Este modelo se describe en detalle en el apartado 2. En el apartado 3 se tratará de establecer métodos de POD (*Preliminary Orbit Determination*) válidos para sistemas de satélites ligados. El objetivo es el diseño y desarrollo de un filtro de mínimos cuadrados semejante al que aparece en la literatura consultada, en el que no existan limitaciones en el movimiento relativo al centro de masas; esto se encuentra en la sección 4. Por último, en 5, se estudiarán los resultados obtenidos en un caso práctico con dicho estimador.

1.1. Misiones

1.1.1. Tether en misiones espaciales humanas:

- Gemini 11: (1966) desplegó un tether de 30m que fué estabilizado mediante una rotación que generó una gravedad artificial de 0.00015 g. El propósito de la misión era demostrar la estabilización pasiva de la actitud de los dos objetos conectados por el tether, y crear gravedad artificial mediante el giro combinado del conjunto de vehículos. [12]
- Misiones Shuttle TSS
 - Misión TSS-1: (1992), desplegó un tether de 256m (de un objetivo de 20km). El propósito de la misión TSS-1 eran verificar el concepto de estabilización del gradiente de gravedad, y proveer un espacio de investigación para la física espacial y la electrodinámica en plasma. [12].
 - Misión TSS-1R: (1996) Desplegó un tether hasta 19.7 km antes de romperse (de un objetivo de 20.7km). La misión TSS-1R tenía como objetivo desplegar un tether de 20.7 km por debajo de la orbita y mantenerse en ese punto recopilando datos. La misión TSS-1R tenía como objetivo llevar a cabo experimentos exploratorios en la física espacial de plasma. Los datos recopilados indicaban que el movimiento

del tether conductor a través del campo magnético de la Tierra produciría un EMF que conduciría una corriente a través del sistema ligado [12].

1.1.2. Tethers en misiones con satélites

- Yo-Yo Despin: (2007) Sistema liega de pequeño tamaño usado en pruebas de satélites y sondas espaciales robóticas. En particular, los tether son usados en el mecanismo "yo-yo de-spin", a menudo utilizado en sistemas en los que una sonda se pone a girar durante un disparo cohete de un motor de inyección, pero se necesita que el giro se detenga durante el vuelo. En este mecanismo, los pesos en los extremos del cable de gran longitud son desplegados lejos del cuerpo del satélite girando. Cuando los cables son cortados, mucho del momento angular debido al giro es transferido a los cuerpos cortados. Como ejemplo, la misión Dawn utilizó dos cuerpos de 3kg desplegados en un cable de 12 metros [12].
- NASA Small Expendable Deployer System Experiments Los siguientes tres experimentos fueron los primeros vuelos satisfactorios de tether largos en orbita, y demostró el funcionamiento mecánico y electrodinámico del tether. [12].
 - SEDS I: (1993) desplegó un tether de 20km. [12].
 - EDS II: (1994) desplegó un tether de 20km. En estos experimentos, los modelos de tether fueron verificados, y los ensayos demostraron que un vehículo de reentrada pueden desplegarse descendientemente hasta una orbita de reentrada usando tethers [12].
 - PMG: (1993) desplegó un tether de 500m para demostrar operaciones electromecánicas en tethers [12].
- Experimentos NRL TiPS and ATEx
 - TiPS: (1996) Desplegó dos objetos conectados por tethers (Ralph and "Norton") con longitudes de 4km y se rompió en 2006 [12].
 - ATEx: (1999) desplegó 22m de un tether de 6km. Pretendía testear un nuevo método para el despliegue de tethers, nuevos materiales, control activo, y supervisión del sistema [12].
- Young Engineers' Satellite (YES)
 - YES: (1997) No se llegó a desplegar el tether (de 35km) [12].
 - YES2: (2007) Desplegó un tether de 32km. Parte de la misión Foton-M3 de la ESA. No se llegó a recibir ninguna señal. La capsula no se llegó a recuperar [12].
- Experimento KITE: (2016) No se llegó a desplegar el tether (de 700m). Se testeó la tecnología del tether en el vehículo de la estación espacial Japonesa HTV 6 [12].

- Misiones CubeSat: Las misiones CubeSats son pequeñas, satélites de bajo coste que son lanzadas típicamente como pesos secundarios en otras misiones, usualmente construidas y operadas por proyectos de estudiantes. Varias misiones CubeSat han intentado desplegar tethers, hasta ahora sin conseguirlo [12].
 - MAST: (2007) Desplegó un tether de 1km [12].
 - STARS: (2009) Disponía de un tether de 5km [12].
 - STARS-II: (2014) Tiene un tether de 300m. Un objetivo de este programa era demostrar posible tecnología para basura espacial [12].
 - ESTCube-1: (2013) La misión trataba de probar una vela eléctrica en órbita. Estaba diseñada para desplegar un tether usando fuerzas centrífugas [12].

1.1.3. Vuelos de cohetes de sondeo

- CHARGE 2 (1985) Lanzada para observar la colección actual junto con otros fenómenos. El objetivo principal era para medir la corriente durante emisiones de electrones. Objetivos secundarios estaban relacionados procesos de plasma asociados con corriente continua y pulsos de una fuente de un haz de electrones de baja carga. [12].
- OEDIPUS
 - OEDIPUS A: (1989) Desplegó un tether de 958m. Lanzada para medir campos eléctricos débiles en la aurora [12].
 - OEDIPUS C: (1995) Desplegó un tether de 1174m [12].
- T-Rex (2010) Desplegó un tether de 300m [12].

1.1.4. Misiones futuras y propuestas

- ProSEDS Propuesta para analizar el uso de una sección desnuda de una correa electrodinámica espacial para un dispositivo de recolección de electrones que ha sido sugerida como una alternativa prometedora a los colectores de electrones de cuerpo final para ciertas aplicaciones de amarre [12].
- STARS-C Tether de 100m. Después de ser entregado a la estación espacial internacional, el satélite será desplegado desde el módulo experimental japonés Kibo [12].
- CubeSat technology Tether de 1km de cinta conductora entrelazada. Se espera que el satélite sea capaz de conducir una corriente electrodinámica en ambas direcciones. Se espera que sea capaz de subir o bajar la órbita varios kilómetros al día, cambiar el estado de libración, cambiar el plano de la órbita. El satélite superó un test de despliegue en Mayo de 2010 [12].
- MiTEE Candidato para una de las misiones espaciales universitarias CubeSat [12].

2. Modelo Dinámico

El sistema de satélites ligados (de ahora en adelante TSS) se modelará como un cable en cuyos extremos se sitúan dos masas m_1 and m_2 . El cable será considerado una varilla rígida, homogénea, de masa m_T y longitud L . Esta configuración es la habitual en el estudio de este tipo de sistemas. Se analizará la dinámica del conjunto, que se supone orbita alrededor de la Tierra. Para el análisis dinámico se introducirán los siguientes sistemas ortogonales de referencia, tal y como se aprecia en la figura:

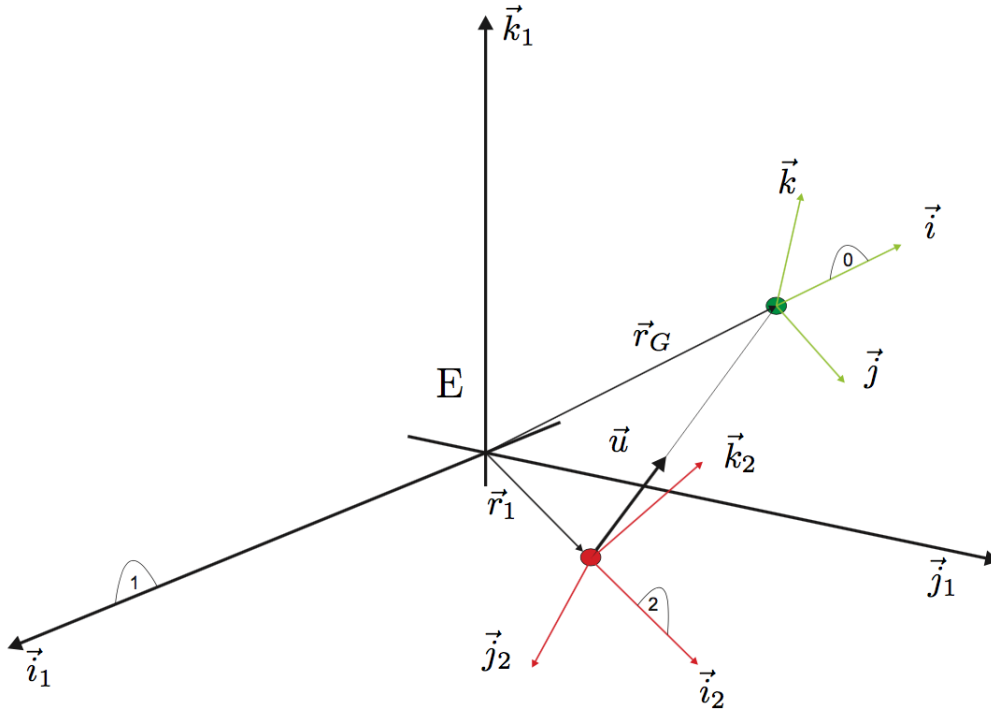


Figura 2: Configuración esquemática del sistema que muestra los triédros utilizados en el desarrollo.

1. Sistema Geocéntrico Ecuatorial Medio en J2000.0: $E x_1 y_1 z_1$. Este será nuestro sistema de referencia inercial.
2. El triédro orbital $Oxyz$ con origen en el centro de gravedad del sistema. Al movimiento del centro de masas se le puede asociar un triédro orbital $R = \{O; \vec{i}, \vec{j}, \vec{k}\}$ cuyo origen O coincide con G y cuyos vectores unitarios $(\vec{i}, \vec{j}, \vec{k})$ se definen como sigue:
 - \vec{i} , con la dirección y sentido del vector posición \vec{r}_G .
 - \vec{k} , contenido en el plano definido por E, \vec{r}_G y la velocidad $\vec{v}_G = \dot{\vec{r}}_G$ del centro de masas, y de sentido tal que $\vec{k} \cdot \vec{v}_G \geq 0$.
 - \vec{j} , se elige de forma que el triédro esté orientado a derechas.

Este sistema de referencia es semejante al 'horizontal local, vertical local' (LGLV) que se define en otros trabajos [4].

3. Un triedro auxiliar $O_2, \vec{i}_2, \vec{j}_2, \vec{k}_2$, cuyo origen coincide con la posición de la masa m_1 , y cuyos vectores unitarios se definen de manera análoga al triedro orbital:
- \vec{i}_2 , con la dirección y sentido del vector posición \vec{r}_1 de la masa m_1 .
 - \vec{j}_2 , contenido en el plano definido por E , \vec{r}_1 y la velocidad $\vec{v}_1 = \dot{\vec{r}}_1$ de la masa m_1 y de sentido tal que $\vec{k}_2 \cdot \vec{v}_1 \geq 0$.
 - \vec{j}_2 , se elige de forma que el triedro esté orientado a derechas.

Para definir la orientación del *tether* se utilizarán los sistemas 1 y 2. Por una parte un sistema de referencia centrado en O y paralelo al Geocéntrico Ecuatorial, en el que la posición del *tether* está definida por los ángulos α , en un plano β perpendicular a éste. Por otra parte, haciendo uso del triedro orbital, dicha posición queda definida por los ángulos θ y φ , tal y como se muestra en las figuras.

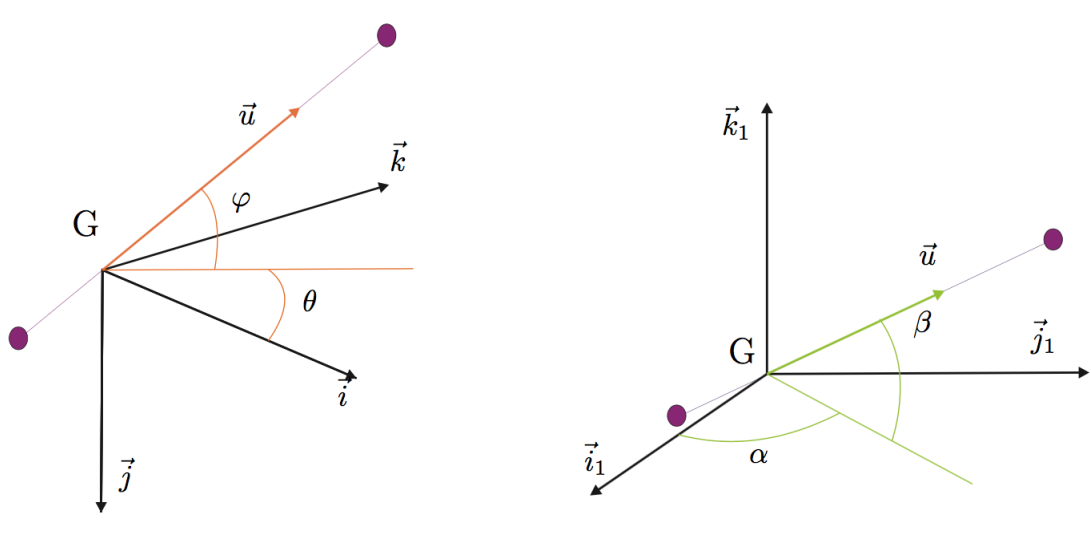


Figura 3: Representación de las proyecciones del vector u en el sistema de referencia de direcciones inerciales y el triedro orbital.

2.1. Descripción del movimiento

En esta sección se describirá el movimiento del sistema formado por el *tether* y sus dos masas extremas, respecto de la referencia geocéntrica $Ex_1y_1z_1$. Sean Las dimensiones transversales del tether se suponen despreciables por lo que el problema reduce su número de grados de libertad a 5. Utilizaremos 5 coordenadas generalizadas independientes como sigue:

- las tres coordenadas (ξ, η, ζ) del centro de masas G del sistema, que se resumen en el vector \vec{r}_G , y
- una pareja de ángulos (α, β) en el triedro inercial, que determinen un vector unitario \vec{u} dirigido de la masa m_1 (inferior) a masa m_2 (superior) que sirve para fijar la posición del *tether*.

2.2. Geometría de masas

En nuestro sistema existen tres masas a considerar: m_1 , m_2 y m_T que corresponden a la masa del objeto menos pesado, la masa del objeto más pesado, y la masa del *tether* respectivamente. Utilizaremos parámetros adimensionales para definir la geometría de masas como sigue:

1. $m = m_1 + m_2 + m_T$ es la masa total del sistema.
2. $\Lambda = \frac{m_T}{m}$ es la masa del *tether* expresada como fracción de la masa total m . Obviamente, siempre se tendrá $\Lambda < 1$. Además, puesto que los *tether* suelen ser ligeros, con frecuencia se tiene $\Lambda \ll 1$.
3. para definir el ángulo ψ se introducen las masas reducidas ν_1 y ν_2 definidas por las expresiones:

$$m \cdot \nu_1 = m_1 + \frac{1}{2}m_T \quad (1)$$

$$m \cdot \nu_2 = m_2 + \frac{1}{2}m_T \quad (2)$$

y puesto que siempre se verifica $\nu_1 + \nu_2 = 1$, las dos masas pueden describirse mediante un único parámetro, el ángulo ψ , dado por las relaciones

$$\nu_1 = \cos^2\psi, \quad \nu_2 = \sin^2\psi \quad (3)$$

En definitiva, para describir las masas del sistema se usarán los valores adimensionales Λ y ψ , y m , la masa total del sistema. Si se desea, se puede obtener las siguientes expresiones de m_1 , m_2 y m_T en función de m , Λ y ψ , variables que se utilizarán a partir de ahora para describir la geometría de masas del sistema:

$$m_1 = m(\cos^2\psi - \frac{1}{2}\Lambda) \quad (4)$$

$$m_2 = m(\sin^2\psi - \frac{1}{2}\Lambda) \quad (5)$$

$$m_T = m\Lambda \quad (6)$$

Por lo tanto los vectores posición de las masas del sistema \vec{r}_1 y \vec{r}_2 en función de las coordenadas generalizadas elegidas quedarán:

$$\vec{r}_1 = \vec{r}_G - L\sin^2\psi\vec{u} \quad (7)$$

$$\vec{r}_2 = \vec{r}_G + L\cos^2\psi\vec{u} \quad (8)$$

El tensor central de inercia del sistema $\bar{\bar{I}}_G$ en ejes triortogonales, ligados al *tether* y cuyo primer vector sea \vec{u} resulta:

$$\bar{\bar{I}}_G = \begin{pmatrix} 0 & 0 & 0 \\ 0 & I_s & 0 \\ 0 & 0 & I_s \end{pmatrix} \quad (9)$$

donde I_s es el momento de inercia del sistema respecto a una recta que pasa por G y es perpendicular al *tether*.

$$I_s = I_{m_1} + I_{m_2} + I_{Tether} \quad (10)$$

donde cada uno de los términos resulta ser:

$$\begin{aligned} I_{m_1} &= m_1 L^2 \sin^4 \psi \\ I_{m_2} &= m_2 L^c \cos^4 \psi \\ I_{Tether} &= \frac{1}{12} m_T L^2 + m_T \left(\frac{L}{2} - L \sin^2 \psi \right)^2 \end{aligned}$$

Por lo tanto el momento de inercia utilizando los parámetros de la geometría de masas:

$$I_s = \frac{1}{12} m L^2 (3 \sin^2 \psi - 2\Lambda) \quad (11)$$

2.3. Cinemática

Ya que el vector \vec{u} está unido al *tether* y asumiendo que $\vec{\omega}_{31}$ es la velocidad angular del *tether* se puede verificar que:

$$\frac{d\vec{u}}{dt} = \vec{\omega}_{31} \times \vec{u} \quad (12)$$

de la que multiplicando vectorialmente por \vec{u} , se deduce el valor de $\vec{\omega}$:

$$\vec{\omega}_{31} = \vec{u} \times \dot{\vec{u}} + \sigma \vec{u} \quad (13)$$

Ya que el momento de inercia del sistema en la dirección de \vec{u} es nulo, el valor de σ resulta ser irrelevante para este problema. Puesto que $\vec{u} \times \dot{\vec{u}}$ es ortogonal a \vec{u} , $\vec{u} \times \dot{\vec{u}}$ lleva siempre la dirección principal de inercia en G , ya que el plano normal a la barra pro G es de direcciones principales. Por tanto, se verifican siempre las relaciones:

$$\bar{\bar{I}}_G \circ \vec{u} = \vec{0}, \quad \bar{\bar{I}}_G \circ \dot{\vec{u}} = I_s \dot{\vec{u}} \quad \bar{\bar{I}}_G \circ (\vec{u} \times \dot{\vec{u}}) = I_s (\vec{u} \times \dot{\vec{u}})$$

2.4. Formulación de Newton-Euler

En esta sección se estudiará el movimiento del centro de masas (ecuación de cantidad de movimiento) y el relativo al centro de masas (ecuación de momento cinético).

2.4.1. Movimiento del centro de masas

Las ecuaciones que gobiernan el movimiento del centro de masas del sistema de estudio son:

$$m \ddot{\vec{r}}_G = - \frac{m\mu}{|\vec{r}_G|^3} \vec{r}_G + \vec{F}_p \quad (14)$$

donde \vec{F}_p es la fuerza de perturbación, que se define como la resultante de todas las fuerzas externas que actúan sobre el sistema a excepción del término principal del potencial gravitatorio evaluado en el centro de masas. En este estudio no incluiremos ninguna fuerza de perturbación.

2.4.2. Movimiento relativo al centro de masas

La ecuación del momento cinético es:

$$\frac{d}{dt}(\vec{H}_G) = \vec{M}_p \quad (15)$$

siendo \vec{H}_G el momento cinético del sistema en el movimiento relativo al centro de masas (tomando momentos en el propio centro de masas G). \vec{M}_p es el momento respecto del centro de masas G de todas las fuerzas externas que actúan sobre el sistema. El momento cinético \vec{H}_G se puede escribir como:

$$\vec{H}_G = \bar{\bar{I}}_G \circ \vec{\omega}_{31} \quad (16)$$

donde $\bar{\bar{I}}_G$ es el tensor de inercia del sistema en el centro de masas G , y $\vec{\omega}_{31}$ es la velocidad angular del *tether* respecto a la referencia $Gx_1y_1z_1$ que se puede escribir:

$$\vec{\omega}_{31} = \vec{u} \times \dot{\vec{u}} + \sigma \vec{u} \quad (17)$$

Por lo tanto la ecuación (15) se puede escribir como:

$$\vec{u} \times \ddot{\vec{u}} = \frac{1}{I_s}(\vec{M}_p) \quad (18)$$

Utilizando como coordenadas generalizadas los ángulos α y β , el vector \vec{u} se puede escribir en función de α y β como sigue:

$$\vec{u} = \cos\alpha \cos\beta \vec{i}_1 + \sin\alpha \cos\beta \vec{j}_1 + \sin\beta \vec{k}_1 \quad (19)$$

La ecuación vectorial (15) se puede descomponer en tres ecuaciones escalares del as que sólo dos son independientes. Proyectando según las direcciones \vec{k}_1 y $\sin\alpha \vec{i}_1 - \cos\alpha \vec{j}_1$ obtenemos:

$$\cos^2\beta \ddot{\alpha} - \dot{\alpha} \dot{\beta} \sin(2\beta) = \frac{1}{I_s} M_z \quad (20)$$

$$\ddot{\beta} + \cos\beta \sin\beta \dot{\alpha}^2 = \frac{1}{I_s} (M_x \sin\alpha - M_y \cos\alpha) \quad (21)$$

Donde \vec{M} es un momento resultante genérico respecto al centro de masas del sistema. En este estudio solo se considerará el de mayor orden, esto es, el par gravitatorio.

2.4.3. Par gravitatorio

El par gravitatorio es la resultante de las fuerzas gravitatorias que actúan sobre el sistema y que provocan un momento en su centro de masas G y su valor está dado por la integral:

$$\vec{M}_{pG} = - \int_{\Omega_o} (\vec{r} - \vec{r}_G) \times \frac{\mu}{r^3}(\vec{r}) dm = \vec{r}_G \times \left(\int_{\Omega_o} \frac{m\mu}{r^3}(\mathbf{r}) d\mathbf{m} \right) \quad (22)$$

Donde Ω_o representa el volumen de control que ocupado por la masa del sistema; \vec{r} es el vector posición de un elemento de masa genérico del sistema, que se puede representar como:

$$\vec{r} = \vec{r}_G + \vec{s} \quad (23)$$

y ya que $|\vec{s}| \ll |\vec{r}_G|$ la contribución de \vec{s} se puede despreciar obteniendo una expresión aproximada para el momento:

$$\vec{M}_{pG} \approx \frac{3\mu}{|\vec{r}_G|^3} \vec{i} \times (\bar{I}_G \circ \vec{i}) \quad (24)$$

usando la identidad $\bar{I}_G \circ \vec{i} = I_s \vec{u} \times (\vec{i} \times \vec{u}) = I_s \{\vec{i} - \vec{u}(\vec{i} \cdot \vec{u})\}$ el par gravitatorio resulta:

$$\vec{M}_{pG} \approx \frac{3\mu}{|\vec{r}_G|^3} I_s (\vec{u} \times \vec{i})(\vec{u} \cdot \vec{i}) \quad (25)$$

Utilizando esta expresión en las ecuaciones (20) y (21) es posible completar la formulación que proporciona la evolución de la dinámica del *tether*:

$$\frac{d\alpha}{dt} = \dot{\alpha} = \vec{u}_1 \quad (26)$$

$$\frac{d\beta}{dt} = \dot{\beta} = \vec{u}_2 \quad (27)$$

$$\frac{du_1}{dt} = 2u_1 u_2 \tan\beta + \frac{3\mu \vec{u} \cdot (\vec{i} \times \vec{k}_1)(\vec{u} \cdot \vec{i})}{|\vec{r}_G|^3 \cos^2\beta} \quad (28)$$

$$\frac{du_2}{dt} = \frac{3\mu(\vec{u} \cdot \vec{i})}{|\vec{r}_G|^3} \left(\vec{u} \cdot (\vec{i} \times \vec{i}_1) \sin\alpha - \vec{u} \cdot (\vec{i} \cdot \vec{j}_1) \cos\alpha \right) - \cos\beta \sin\beta \vec{u}_1^2 \quad (29)$$

3. Determinación de órbita de un TSS conocido

En este apartado asumiremos que disponemos de las observaciones efectuadas y de los datos adicionales acerca del objeto rastreado. En particular, supondremos conocida la geometría de masas del sistema así como la longitud del cable. Las medidas son medidas de radar de ambos cuerpos. El archivo de datos de partida puede tener una configuración por filas semejante a la que aparece a continuación:

$$t_i \quad R_i^I \quad A_i^I \quad h_i^I \quad R_i^S \quad A_i^S \quad h_i^S \quad (30)$$

donde t_i es el instante en el que se produce la observación expresado en algún formato de tiempo concreto, R_i es la medida de distancia (*range*), A_i es la medida de azimuth mientras que h_i es la elevación. Los superíndices distinguen entre las observaciones debidas a uno y otro cuerpo. Tras realizar el oportuno cambio de referencia el archivo de datos con el que trabajaremos será similar a un conjunto de filas con el siguiente aspecto:

$$t_i \quad x_i^I \quad y_i^I \quad z_i^I \quad x_i^S \quad y_i^S \quad z_i^S \quad (31)$$

donde ahora aparecen por columnas las coordenadas de las masas inferior y superior en un sistema geocéntrico ecuatorial. Para poder utilizar el filtro necesitamos inicializarlos con unas condiciones iniciales. Por lo tanto es necesario desarrollar un algoritmo que proporcione al método de corrección diferencial una estimación inicial de la amplitud de la libración del cable y de su velocidad angular para asegurar que dicha estimación inicial se encuentra dentro del rango de convergencia. El vector de estado inicial consta de los siguiente elementos:

$$\left[x_G \quad y_G \quad z_G \quad v_{xG} \quad v_{yG} \quad v_{zG} \quad \alpha \quad \beta \quad \dot{\alpha} \quad \dot{\beta} \right] \quad (32)$$

donde

- x_G, y_G, z_G , son las coordenadas del centro de masas del sistema en el sistema geocéntrico ecuatorial al que están referidas las medidas.
- v_{xG}, v_{yG}, v_{zG} , son las coordenadas de la velocidad del centro de masas referidas al mismo sistema de referencia.
- α, β , son las variables angulares que fijan la posición del vector unitario \vec{u} respecto del sistema de referencia geocéntrico ecuatorial.
- $\dot{\alpha}, \dot{\beta}$, son las derivadas temporales de las variables angulares que determinan la velocidad angular de \vec{u} .

Nuestro esquema POD (Preliminary Orbit Determination) comenzará por la lectura de las tres primeras medidas de las masas superior e inferior. Se analizará la satisfacción de una condición de coherencia con el sistema comprobando que $|\vec{r}_i^S - \vec{r}_i^I| < \delta$ para $i = 1, 2, 3$ siendo δ un umbral arbitrario. Una estimación inicial de la posición del centro de masas para el vector de estado inicial es:

$$\vec{r}_{Gi} = \vec{r}_i^S \sin^2 \psi + \vec{r}_i^I \cos^2 \psi \quad i = 1, 2, 3 \quad (33)$$

Como consideramos que el centro de masas únicamente sufre el efecto del término principal de las fuerzas gravitatorias, podemos tratar la estimación de la velocidad como un problema clásico de determinación de órbita kepleriana. Haciendo uso del algoritmo de Herrick-Gibbs [3] para el cálculo de la velocidad del centro de masas en el instante $t = t_2$.

$$G_1 = \frac{t_3 - t_2}{(t_2 - t_1)(t_3 - t_1)}; \quad G_3 = \frac{t_2 - t_1}{(t_3 - t_2)(t_3 - t_1)}; \quad G_2 = G_1 - G_3;$$

$$H_1 = \frac{\mu(t_3 - t_2)}{12}; \quad H_3 = \frac{\mu(t_3 - t_2)}{12}; \quad H_2 = H_1 - H_3;$$

Por lo que:

$$d_i = G_i + \frac{H_i}{r_i^3} \quad i = 1, 2, 3;$$

$$\vec{v}_{G2} = -d_1 \vec{r}_{G1} + d_2 \vec{r}_{G2} + d_3 \vec{r}_{G3} \quad (34)$$

Con la velocidad queda determinada la órbita del centro de masas. Las condiciones angulares se pueden obtener considerando de nuevo las ecuaciones (7) y (8). Para las variables α_0 y β_0 se tiene:

$$L\vec{u}_0 = \vec{r}_2^S - \vec{r}_2^J;$$

$$\alpha_0 = \arctan \left(\frac{\vec{u}_0 \cdot \vec{j}_1}{\vec{u}_0 \cdot \vec{i}_1} \right); \quad (35)$$

$$\beta = \arcsin(\vec{u}_0 \cdot \vec{k}_1); \quad (36)$$

En el caso de la velocidad angular es necesario realizar una estimación adicional ya que no disponemos de medidas que nos permitan calcular el vector velocidad de las masas inferior y/o superior en el instante $t = t_2$. Utilizaremos un esquema de diferencias centradas de tres nodos no equiespaciados. Para ello primero tendremos que adimensionalizar los valores de posición y tiempo. Los valores que utilizaremos para realizar la adimensionalización serán R_\oplus (radio ecuatorial terrestre WGS-84) para la posición y $\sqrt{R_\oplus/\mu}$ (siendo μ constante gravitacional JGM-3) para el tiempo. Denotaremos con $(\tilde{\cdot})$ las variables adimensionales.

$$\tilde{r} = \frac{r}{R_\oplus}$$

$$\tilde{v} = \sqrt{\frac{\mu}{R_\oplus}} \vec{v}$$

Las velocidades adimensionales en $t = t_2$:

$$\tilde{v}_2^X = \frac{\tilde{r}_3^X h_0^2 + \tilde{r}_2^X (h_0 + h_1)(H_1 - h_0) - \tilde{r}_1^X h_0^2}{h_0 h_1 (h_0 + h_1)} \quad (37)$$

con

$$h_0 = (t_2 - t_1) \cdot \sqrt{\frac{\mu}{R_\oplus^3}}$$

$$h_1 = (t_3 - t_2) \cdot \sqrt{\frac{\mu}{R_{\oplus}^3}}$$

Con $X \equiv S, I$. Posteriormente el vector $\dot{\vec{u}}$ puede obtenerse de dos formas alternativas:

$$L\dot{\vec{u}}_0 = \vec{v}_2^S - \vec{v}_2^I;$$

$$L\cos^2\psi\dot{\vec{u}}_0 = \vec{v}_2^S - \vec{v}_{G2};$$

Y por último:

$$\dot{\alpha}_0 = \frac{\cos\alpha_0\dot{\vec{u}}_0 \cdot \vec{j}_1 - \sin\alpha_0\dot{\vec{u}}_0 \cdot \vec{i}_1}{\cos\beta_0} \quad (38)$$

$$\dot{\beta}_0 = \frac{\dot{\vec{u}}_0 \cdot \vec{k}_1}{\cos\beta_0} \quad (39)$$

4. Descripción del filtro

En este capítulo se describirá las ecuaciones que se han utilizado para el desarrollo del filtro de mínimos cuadrados para la determinación de órbita de nuestro sistema. Puesto que el movimiento está gobernado por un sistema de ecuaciones diferenciales, dado un modelo, las condiciones iniciales determinan una solución para el vector de estado del sistema. El objetivo de este método es buscar el vector de estado inicial que una vez utilizado para realizar el cálculo de la órbita, ésta se la que mejor se aproxime a los datos medidos.

4.1. Estimación de mínimos cuadrados

En este capítulo se ha seguido el desarrollo teórico que se encuentra recogido en el apartado 8.1 de [7]. Para resolver este problema trataremos de hallar el vector de estado inicial que mejor aproxima el movimiento del sistema estudiado mediante la minimización de la siguiente función de coste:

$$J(\vec{x}_0) = \vec{\rho}^T \vec{\rho} = (\vec{z} - \vec{h}(\vec{x}_0))^T (\vec{z} - \vec{h}(\vec{x}_0)) \quad (40)$$

siendo $\vec{\rho}$ el vector diferencia entre las observaciones y la predicción del model. Esto es, \vec{z} es el vector de medida, en el que se agrupan las observaciones en los m instantes de tiempo considerados. Por lo tanto, este vector es de longitud $6 \times m$ y cada grupo de seis componentes responde a la estructura considerada en (31) (de donde se ha eliminado la columna de tiempos). El vector \vec{h} , por otra parte, denota el valor del modelo como función del vector de estado inicial \vec{x}_0 . Es decir, en nuestro caso:

$$\vec{h}_i = \begin{pmatrix} x_i(\vec{x}_0) - L \sin^2 \psi \cos(\alpha(\vec{x}_0)) \cos(\beta(\vec{x}_0)) \\ y_i(\vec{x}_0) - L \sin^2 \psi \sin(\alpha(\vec{x}_0)) \cos(\beta(\vec{x}_0)) \\ z_i(\vec{x}_0) - L \sin^2 \psi \sin(\beta(\vec{x}_0)) \\ x_i(\vec{x}_0) + L \cos^2 \psi \cos(\alpha(\vec{x}_0)) \cos(\beta(\vec{x}_0)) \\ y_i(\vec{x}_0) + L \cos^2 \psi \sin(\alpha(\vec{x}_0)) \cos(\beta(\vec{x}_0)) \\ z_i(\vec{x}_0) + L \cos^2 \psi \sin(\beta(\vec{x}_0)) \end{pmatrix} \quad (41)$$

con $i = 1, \dots, m$ Para realizar la linearización del problema alrededor de una condicona inicial de referencia, \vec{x}_0^{ref} , el vector $\vec{\rho}$ se puede aproximar como:

$$\vec{\rho} \approx \Delta \vec{z} - \mathbf{H} \Delta \vec{x}_0$$

siendo $\Delta \vec{x}_0 = \vec{x}_0 - \vec{x}_0^{ref}$ la diferencia entre \vec{x}_0 y el estado de referencia. Por otra parte, $\Delta \vec{z} = \vec{z} - \vec{h}(\vec{x}_0^{ref})$ denota la diferencia entre las observaciones obtenidas y las predichas por el modelo para la trayectoria de referencia. Por último, la matriz jacobiana

$$\mathbf{H} = \left. \frac{\partial \vec{h}(\vec{x}_0)}{\partial \vec{x}_0} \right|_{\vec{x}_0^{ref}} \quad (42)$$

proporciona las derivadas parciales de las observaciones predichas por el modelo respecto del vector de estado de referencia en el instante inicial $t = t_0$.

Considerando la definición de $\vec{h}(\vec{x}_0)$ la matriz \mathbf{H} se puede escribir como:

$$\mathbf{H} = \begin{bmatrix} \frac{\partial \vec{h}}{\partial \vec{x}_0} \\ \frac{\partial \vec{h}}{\partial \vec{x}_1} \cdot \frac{\partial \vec{x}_1}{\partial \vec{x}_0} \\ \dots \\ \frac{\partial \vec{h}}{\partial \vec{x}_m} \cdot \frac{\partial \vec{x}_m}{\partial \vec{x}_0} \end{bmatrix} \quad (43)$$

En esta expresión, podemos identificar por una parte la matriz de transición dado que:

$$\frac{\partial \vec{x}_i}{\partial \vec{x}_0} = \Phi(\vec{x}_i, \vec{x}_0)$$

Y por otra parte la matriz de observación, que podemos expresar de la siguiente forma:

$$\frac{\partial \vec{h}}{\partial \vec{x}_i} = \begin{bmatrix} I & 0 & U_1 & 0 \\ I & 0 & U_2 & 0 \end{bmatrix} = M_i$$

Donde $I_{3 \times 3}$ es la matriz identidad, mientras que las submatrices U_1, U_2 se definen como sigue:

$$\begin{aligned} U_1 &= -L \sin^2 \psi V \\ U_2 &= L \cos^2 \psi V \\ V &= \begin{bmatrix} -\sin \alpha_i \cos \beta_i & -\cos \alpha_i \cos \beta_i \\ \cos \alpha_i \cos \beta_i & -\sin \alpha_i \cos \beta_i \\ 0 & \cos \beta_i \end{bmatrix} \end{aligned}$$

Por lo tanto, la expresión de la matriz jacobiana \mathbf{H} está dada en función de la matriz de observación y de la matriz de transición.

$$\mathbf{H} = [M_i \Phi(\vec{x}_i, \vec{x}_0)]_{i=1, \dots, m} \quad (44)$$

Por tanto la función de coste que es necesario minimizar en la formulación linealizada es:

$$J(\Delta \vec{x}_0) = (\Delta \vec{z} - \mathbf{H} \Delta \vec{x}_0)^T (\Delta \vec{z} - \mathbf{H} \Delta \vec{x}_0) \quad (45)$$

Si las columnas del jacobiano \mathbf{H} son linealmente independientes, la condición de mínimo lleva a la solución general del problema de mínimos cuadrados que se puede escribir como:

$$\delta \vec{x}_0 = (\mathbf{H}^T \mathbf{H})^{-1} (\mathbf{H} \Delta \vec{z})$$

Este valor de $\delta \vec{x}_0$ se utilizará como paso iterativo en la búsqueda del vector de estado inicial. Finalizando cuando la diferencia entre pasos sea menor a un error admisible elegido.

4.2. Ecuaciones variacionales

En este apartado veremos cómo calcular el vector de estado así como las derivadas parciales de ese vector de estado respecto de las condiciones iniciales. La matriz de transición se define como: $\Phi(t, t_0) = (\partial \vec{x}(t) / \partial \vec{x}(t_0))$. La evolución de la matriz de transición se puede obtener de la siguiente ecuación diferencial:

$$\frac{d}{dt} \Phi(t, t_0) = \mathcal{A}(\vec{x}, t) \cdot \Phi(t, t_0) \quad (46)$$

$$\Phi(t \equiv t_0) = I; \quad (47)$$

$$\text{siendo } \mathcal{A}(\vec{x}, t) = \frac{(\vec{x}, t)}{\partial \vec{x}}$$

La dinámica en variables adimensionales viene dada por las siguientes ecuaciones diferenciales:

$$\frac{d\vec{x}(\tau)}{d\tau} = \frac{d}{d\tau} \begin{pmatrix} \tilde{v}_{xG} \\ \tilde{v}_{yG} \\ \tilde{v}_{zG} \\ -\tilde{x}/\tilde{r}^3 \\ -\tilde{y}/\tilde{r}^3 \\ -\tilde{z}/\tilde{r}^3 \\ \dot{\tilde{\alpha}} \\ \dot{\tilde{\beta}} \\ 2\dot{\tilde{\alpha}}\dot{\tilde{\beta}} + \frac{3}{\tilde{r}_G^5 \cos^2 \beta} \vec{u} \cdot (\vec{r}_G \times \vec{k}_1) (\vec{u} \cdot \vec{r}_G) \\ \frac{3}{\tilde{r}_G^5} (\vec{u} \cdot \vec{r}_G) \left(\vec{u} \cdot (\vec{r}_G \times \vec{i}_1) \sin \alpha - \vec{u} \cdot (\vec{r}_G \times \vec{j}_1) \cos \alpha \right) - \cos \beta \sin \beta \dot{\tilde{\alpha}}^2 \end{pmatrix} \quad (48)$$

Siendo $\vec{r}_G = \tilde{x}\vec{i}_1 + \tilde{y}\vec{j}_1 + \tilde{z}\vec{k}_1$ y $\tilde{r} = |\vec{r}_G|$. Por lo tanto, la expresión de la matriz $\mathcal{A}(\vec{x}_0, t)$ se puede escribir de la siguiente forma:

$$\mathcal{A}(\vec{x}_0, t) = \begin{bmatrix} 0 & I_{3 \times 3} & 0 & 0 \\ \mathcal{B} & 0 & 0 & 0 \\ 0 & 0 & 0 & I_{2 \times 2} \\ \mathcal{C} & 0 & \mathcal{D} & \mathcal{E} \end{bmatrix}$$

Las submatrices que aparecen en la igualdad se definen tal y como se expone a continuación:

- La matriz $\mathcal{B} \in \mathcal{M}_{3 \times 3}$

$$\mathcal{B} = -\frac{1}{\tilde{r}_G^5} \begin{bmatrix} 1 - 3\tilde{x}^2/\tilde{r}_G^2 & 0 & 0 \\ 0 & 1 - 3\tilde{y}^2/\tilde{r}_G^2 & 0 \\ 0 & 0 & 1 - 3\tilde{z}^2/\tilde{r}_G^2 \end{bmatrix}$$

- La matriz $\mathcal{C} \in \mathcal{M}_{2 \times 3}$, cuyos elementos:

$$\mathcal{C}_{11} = \frac{3}{\tilde{r}_G^5} \left[-5 \frac{\tilde{x}(\vec{r}_G \cdot \vec{i})}{\tilde{r}_G^2 \cos \beta} c_1 + \cos \alpha c_1 - \sin \alpha (\vec{r}_G \cdot \vec{i}) \right]$$

$$\mathcal{C}_{12} = \frac{3}{\tilde{r}_G^5} \left[-5 \frac{\tilde{y}(\vec{r}_G \cdot \vec{i})}{\tilde{r}_G^2 \cos \beta} c_1 + \sin \alpha c_1 + \cos \alpha (\vec{r}_G \cdot \vec{i}) \right]$$

$$\mathcal{C}_{13} = \frac{3}{\tilde{r}_G^5} \left[-5 \frac{\tilde{z}(\vec{r}_G \cdot \vec{i})}{\tilde{r}_G^2 \cos \beta} c_1 + \tan \beta c_1 \right]$$

$$\mathcal{C}_{21} = \frac{3}{\tilde{r}_G^5} \left[-5 \frac{\tilde{x}(\vec{r}_G \cdot \vec{i})}{\tilde{r}_G^2} c_2 + \cos \alpha \left(\cos \beta c_2 - (\vec{r}_G \cdot \vec{i}) \sin \beta \right) \right]$$

$$C_{22} = \frac{3}{\tilde{r}_G^5} \left[-5 \frac{\tilde{y}(\vec{r}_G \cdot \vec{i})}{\tilde{r}_G^2} c_2 + \sin\alpha \left(\cos\beta c_2 - (\vec{r}_G \cdot \vec{i}) \sin\beta \right) \right]$$

$$C_{23} = \frac{3}{\tilde{r}_G^5} \left[-5 \frac{\tilde{z}(\vec{r}_G \cdot \vec{i})}{\tilde{r}_G^2} c_2 + \sin\beta c_2 + (\vec{r}_G \cdot \vec{i}) \cos\beta \right]$$

Llamando $c_1 = (\tilde{y}\cos\alpha - \tilde{x}\sin\alpha)$ y $c_2 = \tilde{z}\cos\beta - (\tilde{x}\cos\alpha + \tilde{y}\sin\alpha)\sin\beta$

- La matriz $\mathcal{D} \in \mathcal{M}_{2 \times 2}$, cuyos elementos:

$$\mathcal{D}_{11} = \frac{3}{\tilde{r}_G^5} \left[c_1^2 - (\tilde{x}\cos\alpha + \tilde{y}\sin\alpha) \frac{(\vec{r}_G \cdot \vec{i})}{\cos\beta} \right]$$

$$\mathcal{D}_{12} = \frac{1}{\cos^2\beta} \left[2\dot{\tilde{\alpha}}\dot{\tilde{\beta}} + \frac{3}{\tilde{r}_G^5} \tilde{z} \right]$$

$$\mathcal{D}_{21} = \frac{3}{\tilde{r}_G^5} (\tilde{x}\sin\alpha - \tilde{y}\cos\alpha) \left(\sin\beta(\vec{r}_G \cdot \vec{i}) - \cos\beta c_2 \right)$$

$$\mathcal{D}_{22} = \dot{\tilde{\alpha}}^2 (\sin^2\beta - \cos^2\beta) + \frac{3}{\tilde{r}_G^5} \left[c_2^2 + (\vec{r}_G \cdot \vec{i}) (\tilde{x}\cos\alpha + \tilde{y}\sin\alpha - \tilde{z}\tan\beta) \right]$$

- La matriz $\mathcal{E} \in \mathcal{M}_{2 \times 2}$

$$\mathcal{E} = \begin{bmatrix} 2\dot{\tilde{\beta}}\tan\beta & 2\dot{\tilde{\alpha}}\tan\beta \\ -2\dot{\tilde{\alpha}}\cos\beta\sin\beta & 0 \end{bmatrix}$$

Con el problema así planteado, es necesario integrar (48) (con condición inicial $\vec{x}(t = t_0, \vec{x}_0)$) conjuntamente con (46) y (47) para el cálculo de la matriz jacobiana \mathbf{H} .

5. Resultados

Para el estudio del filtro primero se generarán los datos utilizando el archivo matlab "DataGeneration_v1.m" para ello se debe introducir un vector de estado inicial:

$$X_0 = [36000 \quad 0 \quad 0 \quad 0 \quad 3,3275 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]$$

este vector de estado inicial generará la siguiente órbita:

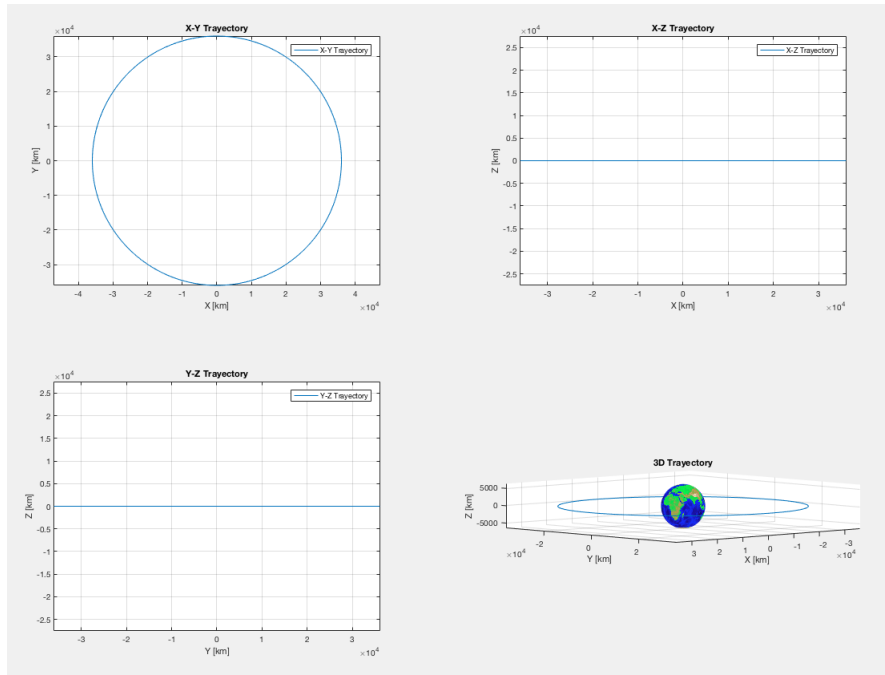


Figura 4: Trayectoria real del centro de gravedad del sistema

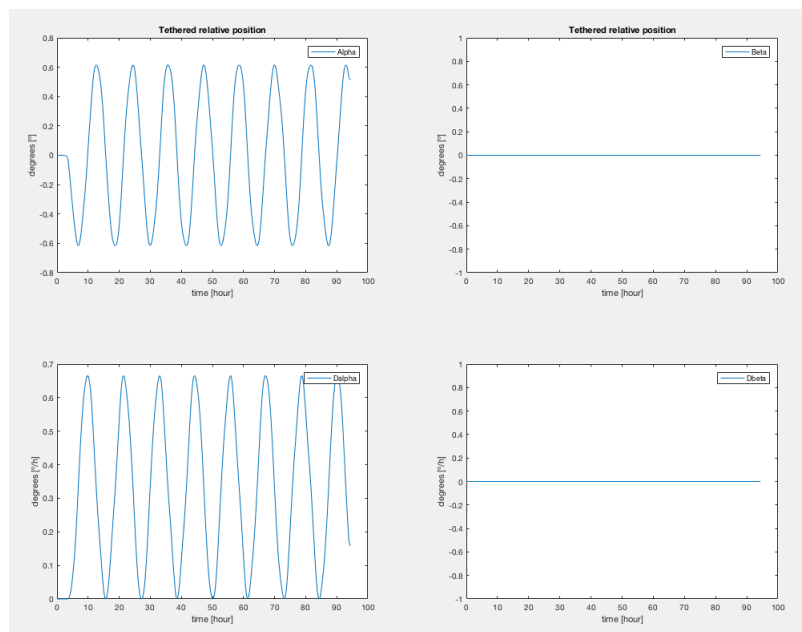


Figura 5: Movimiento relativo

Una vez se tiene la trayectoria real del sistema de satélites se calcula la matriz de medidas de cada uno de las masas (inferior y superior) y se le aplica un error aleatorio a cada una de las medidas estableciendo un máximo error. En la siguiente figura se representan las órbitas de las dos masas una vez aplicado el error aleatorio.

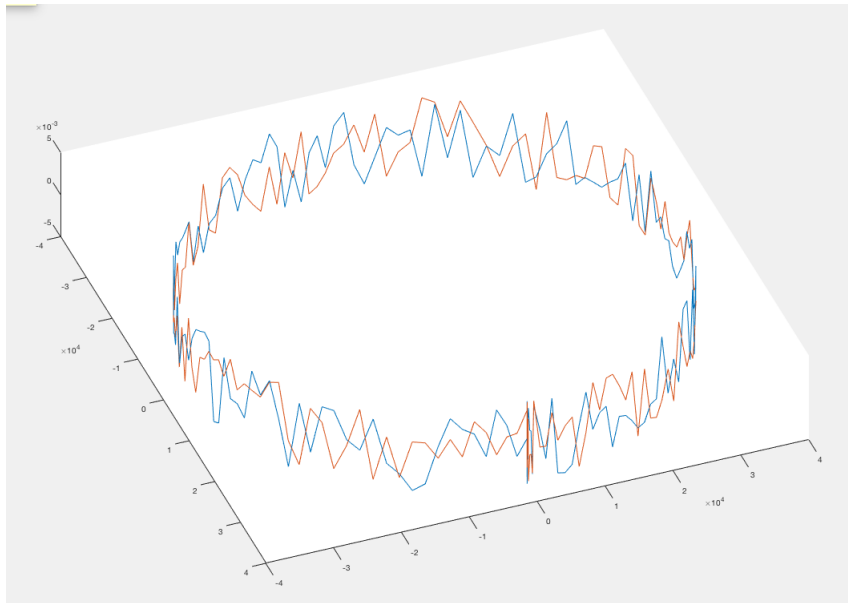


Figura 6: Trayectoria de las masas con errores de medida

Como se ha descrito previamente se cogerán las tres primeras medidas de la matriz de medidas de las masas superior e inferior y se calculará un vector de estado inicial para realizar el cálculo de la trayectoria. Utilizando el archivo matlab "TSS_OD_v1.m." obtenemos el siguiente vector de estado inicial para las medidas recopiladas" (cabe mencionar que este vector variará cada vez que se corra el programa ya que proviene de un generador de medidas que utiliza un valor aleatorio para generarlas):

$$X_0 = [3598,97 \quad 0,0012 \quad -0,0001 \quad -0,0026 \quad 0,3281 \quad -0,0002 \quad 0 \quad 0 \quad 0 \quad 0]$$

Este vector lo utilizaremos para propagar la órbita utilizando el modelo dinámico descrito previamente. La órbita generada a partir de este vector de estado inicial generado con las medidas de los satélites superior e inferior se puede observar en el siguiente gráfico.

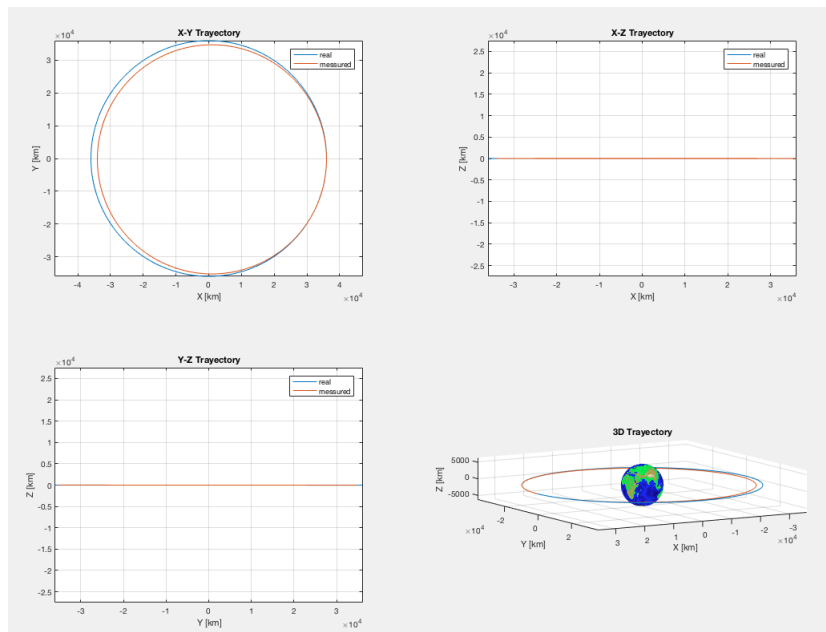


Figura 7: Trayectorias real (azúl) y generada a partir de las medidas (rojo)

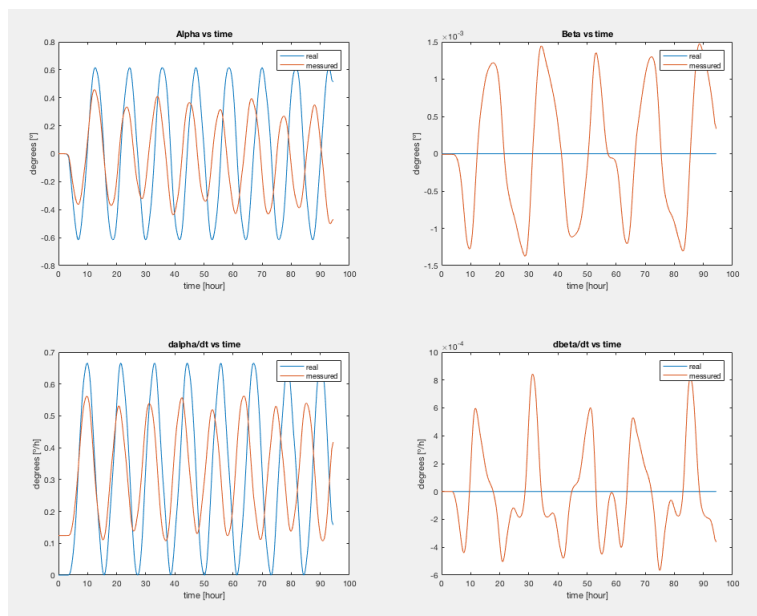


Figura 8: Movimiento relativo: real (azúl) y generada a partir de las medidas (rojo)

Una vez que se tiene la órbita generada con el vector de estado inicial y las medidas utilizaremos la función de minimización para buscar el vector de estado inicial que mejor aproxime la órbita generada a la de las medidas. Esto se realiza utilizando la función de matlab "Filter_v3.m". La función programada ha dado el siguiente error:

iWarning: Failure at t=3.587337e+01. Unable to meet integration tolerances without reducing the step size below the smallest value allowed (1.136868e-13)

at time t .”

que no ha sido posible resolver satisfactoriamente para la utilización del programa. Las ecuaciones matemáticas han sido reevaluadas y el código analizado sin poder encontrar el fallo por lo que no se ha podido verificar el correcto funcionamiento de éste.

Igualmente se han incluido los código matlab en el Anexo que podrían ser reevaluadas en el futuro para poder validar y usar el código generado.

6. Conclusiones

Se ha podido evaluar matemáticamente el problema de estudio y desarrollar

- las ecuaciones necesarias para la generación de datos para la posterior evaluación
- la dinámica de sistemas de satélites ligados para la generación de órbitas planetarias
- el filtro de mínimos cuadrados necesario para calcular el vector de estado inicial que represente mejor los valores obtenidos de medidas de las masas superior e inferior del sistema de estudio.

También se ha podido implementar satisfactoriamente en matlab las funciones para los estudios listados anteriormente excepto para el filtro. La función para el filtro ha sido implementada aunque el error que genera no ha podido evaluar satisfactoriamente el desarrollo matemático expuesto en este trabajo.

7. Anexo

En esta sección se incluyen los códigos matlab utilizados para la evaluación del desarrollo matemático descrito en este trabajo.

7.1. Main.m

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%   Proyecto Final de Carrera      %
%   Luis Miguel González Job      %
%
%   OD_Tethered_System            %
%
%   Date: 2017.07.28              %
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Ordered list of function to run the program

%Theoretical orbit
    Initial_Conditions      %Load initial conditions

        options = odeset('RelTol',1e-6,'AbsTol', 1e-6);
        [t,x]=ode45(@(t,x) dxdt2(t,x), tspan, X0, options);
        %Solution of the ODE for x,y&z

    %TOG_v5                  %Plots the orbit data
                            %Now we have the matrix x with the data

%Generating Data for inferior and superior mass with and without noise

    DataGeneration_v1      %Data generator for x_IS and x_IS_n

%Calculating the Initial conditions for the noise data

    TSS_OD_v1              %Calculates the vector X0_n

max_error=1;                %Maximum error to skip iteration
%while (iter_error<=max_error)
%Calculates the orbit with the initial conditions with noise (X0_n)

        options = odeset('RelTol',1e-6,'AbsTol', 1e-6);
        [t_n,x_n]=ode45(@(t,x) dxdt2(t,x), tspan, X0_n, options);
        %Solution of the ODE for x,y&z

```

```

%TOG_v5                                %Plots the orbit data
                                         %Now we have the matrix x with the data

for zz=1:7
%Clear variables
%   clearvars alphas beta1 DX0 Dz hX0 I2 I3 ii M PhiCom SolCom state0
%stateModel tout U1 U2 V z Z22 Z23 Z32 Z33
%Calling Filter calculates the approximation using least mean squares
    Filter_v3
    %[DX0,stateModel]=Filter_v2(X0_n,tspan,psi,T_l,t,x_IS_n);
%Update of state vector for the noise values
    X0_n=X0_n+DX0;

%Error of the increment to know if the iteration has converge
    iter_error=abs(norm(X0_n)-norm(X0_n-DX0));

%Plots
    Plots
    hgexport(gcf, strcat('figure ',num2str(zz)), hgexport('factorystyle'), 'Format
    %saveplotlyfig(figure, 'your_image_filename.png')
end
%end

```

7.2. InitialConditions.m

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                               %
%   Projecto Final de Carrera   %
%   Luis Miguel González Job    %
%                               %
%   OD_Tethered_System         %
%                               %
%   Date: 2017.07.28           %
%                               %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Initialazing program
    close all;
    clear all;
    clc;

%Declaring and initialazing variables
    muE=398600.440;             %[km^3/s^2]      Graviational constant      JFM-3
    r0 = 36000;                 %[km]
    vC = sqrt(muE/r0);          %[km/s]
    omega = sqrt(muE/r0^3);
    Per = 2 * pi / omega% * 5;  %[s] %multiply by 2.2 to get 1 orbit

```

```

%time span for each loop
    tspan = [0 Per];

%Initial conditions
    %Position
    x0=r0;
    y0=0;%3/4*r0;
    z0=0;
    %Velocity
    vx0 = 0;
    vy0 = vC;
    vz0 = 0;%vC/10;
    %Orientation
    alpha=0;
    beta=0;
    dalpha=0;
    dbeta=0;
    X0=[x0;y0;z0;vx0;vy0;vz0;alpha;beta;dalpha;dbeta];

```

7.3. DataGeneration_v1.m

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                               %
%   Proyecto Final de Carrera   %
%   Luis Miguel González Job    %
%                               %
%   OD_Tethered_System         %
%                               %
%   Date: 2017.07.28           %
%                               %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Call TOG - Theoretical Orbit Generator
    %TOG_v4
    close all

%Satelite properties
    Properties

%Change the data to readings from position of both satelites
%Data in the form: ti xiI yiI ziI xiS yiS ziS
    %ti:    epoch of time when the observation is done
    %x,y,z: Coordinates of the position
    %I:    Inferior mass satelite
    %S:    Superior mass satelite

```

```

%Tethered orientation (vector u):
    i1=[1;0;0];
    j1=[0;1;0];
    k1=[0;0;1];
    %
    u=zeros(size(x,1),3);
    for i=1:size(x,1)
        u_i=cos(x(i,7))*cos(x(i,8))*i1+sin(x(i,7))*cos(x(i,8))*j1+sin(x(i,8))*k1;
        u(i,:)=u_i.';
    end

%Coordinates of inferior and superior satellite
%Coordinates of the center of mass of the TSS
    rG=[x(:,1) x(:,2) x(:,3)]; %[km]
%Coordinates of the inferior satellite
    x_I=rG-I_l*u;                %[km]
%Coordinates of the superior satellite
    x_S=rG+S_l*u;                %[km]
%Data for the TSS
    x_IS=[t x_I x_S];           %[s Km]

%Coordinates of inferior and superior satellite with noise
%Distance noise
    dist_error=0.0005; %[km]
%Data for the TSS with noise
    x_IS_n=[t x_IS(:,2:7)+dist_error*(rand(size(x,1),6)*2-ones(size(x,1),6))];

graficas_DataGeneration=0;
if graficas_DataGeneration==1
    figure
    plot3(x_IS(:,2),x_IS(:,3),x_IS(:,4))
    hold on
    plot3(x_IS(:,5),x_IS(:,6),x_IS(:,7))

    figure
    plot3(x_IS_n(:,2),x_IS_n(:,3),x_IS_n(:,4))
    hold on
    plot3(x_IS_n(:,5),x_IS_n(:,6),x_IS_n(:,7))
end

%Data validation
Data_validation=0;
if Data_validation==1
    for i=1:3
        T_l_n=norm([x_IS_n(i,2),x_IS_n(i,3),x_IS_n(i,4)]-[x_IS_n(i,5),x_IS_n(i,6),x_IS_n(i,7)]);
        if abs(T_l_n-T_l)<2*dist_error
    end
end

```

```

        disp('Data OK')
    else
        disp('Data not OK')
    end
end
end
end

```

7.4. TSS_OD_v1.m

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%   Proyecto Final de Carrera
%   Luis Miguel González Job
%
%   OD_Tethered_System
%
%   Date: 2017.07.28
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Data validation
if Data_validation==1
    for i=1:3
        T_l_n=norm([x_IS_n(i,2),x_IS_n(i,3),x_IS_n(i,4)]-[x_IS_n(i,5),x_IS_n(i,6)]);
        if abs(T_l_n-T_l)<2*dist_error
            disp('Data OK')
        else
            disp('Data not OK')
        end
    end
end

%Adimensional mass parameters
Lambda=T_m/Tot_m;    %[-] Adimensional parameter for the Tether mass
nu1=(I_m+0.5*T_m)/Tot_m;    %[-] Adimensional parameter that defines the re
nu2=(S_m+0.5*T_m)/Tot_m;    %[-] Adimensional parameter that defines the re
psi=acos(sqrt(nu1));    %[-] Adimensional parameter that defines the re
psi2=asin(sqrt(nu2));    %[-] Adimensional parameter that defines the re

%Calculating the initial conditions for the TSS_OD with noise
%Calculating Center of mass with noise
rG_n=zeros(3,3);
for i=1:3
    rG_n(i,:)= [x_IS_n(i,2) x_IS_n(i,3) x_IS_n(i,4)]*(sin(psi))^2+[x_IS_n(i,5) x_IS_n(i,6)]*(cos(psi))^2;
end

%Calculating the Velocity of the Center of mass with noise

```

```

        G1=(t(3)-t(2))/(t(2)-t(1))/(t(3)-t(1));
        G3=(t(2)-t(1))/(t(3)-t(2))/(t(3)-t(1));
        G2=G1-G3;
    G=[G1 G2 G3];
        H1=muE*(t(3)-t(2))/12;
        H3=muE*(t(2)-t(1))/12;
        H2=H1-H3;
    H=[H1 H2 H3];
        d=zeros(1,3);
    for i=1:3
        d(i)=G(i)+H(i)/(norm(rG_n(i,1:3)))^3;
    end
    vG_2_n=-d(1)*rG_n(1,:)+d(2)*rG_n(2,:)+d(3)*rG_n(3,:);

%Calculating alpha and beta with noise
    u0_n=(x_IS_n(i,2:4)-x_IS_n(i,5:7))/T_l;
    alpha0_n=atan((u0_n*j1)*(u0_n*i1));
    beta0_n=asin(u0_n*k1);

%Calculating the angular velocity (dalpha & dbeta) with noise
%Non-dimensional time and position values
    r_adim_n=x_IS_n(1:3,2:7)/Ec_Rad;
%Non-dimensional time steps (from 1-2 and from 2-3)
    h0=(t(2)-t(1))*sqrt(muE/Ec_Rad^3);
    h1=(t(3)-t(2))*sqrt(muE/Ec_Rad^3);
%Non-dimensional velocity for the inferior mass and superior mass in 2
    v_adim_I_2_n=(r_adim_n(3,1:3)*h0^2+r_adim_n(2,1:3)*(h0+h1)*(h1-h0)-r_adim_n(1,1:3)*h1*(h0+h1))/h0*h1*(h0+h1);
    v_adim_S_2_n=(r_adim_n(3,4:6)*h0^2+r_adim_n(2,4:6)*(h0+h1)*(h1-h0)-r_adim_n(1,4:6)*h1*(h0+h1))/h0*h1*(h0+h1);
%Velocity for the inferior mass and superior mass in 2
    v_I_2_n=v_adim_I_2_n*sqrt(muE/Ec_Rad);
    v_S_2_n=v_adim_S_2_n*sqrt(muE/Ec_Rad);
    du0_n=(v_S_2_n-v_I_2_n)/T_l;
%And lastly the values for dalpha & dbeta:
    dalpha0_n=(cos(alpha0_n)*du0_n*j1-sin(alpha0_n)*u0_n*i1)/cos(beta0_n);
    dbeta0_n=du0_n*k1/cos(beta0_n);

%Initial conditions vector to calculate the orbit of the TSS:
    X0_n=[rG_n(2,:) vG_2_n alpha0_n beta0_n dalpha0_n dbeta0_n]';

```

7.5. Filter_{v3.m}

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                    %
% Proyecto Final de Carrera                                         %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```



```

% Luis Miguel González Job      %
%                                %
% OD_Tethered_System           %
%                                %
% Date: 2017.07.28             %
%                                %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%Uses non-dimensional function for iteration%%

%function [DX0,stateModel]=Filter_v2(X0_n,tspan,psi,T_1,t,x_IS_n)
%Transition Matrix (Phi)
state0 = zeros([10,11]);
state0(:,1) = X0_n;
state0(1:10, 2:11) = eye(10);

state0 = reshape(state0, 110,1, []);

%tstep=(tspan(2)/500);
[tout, SolCom] = ode45(@dCdt3, tspan, state0);
%[tout, SolCom] =ode15s(@dCdt3,tspan,state0);
%[tout, SolCom] = ode45(@dCdt3, 1:tstep:tspan(2), state0);

stateModel = SolCom(:,1:10);

%for ii = 1:length(tout)
PhiCom = reshape(SolCom(:,11:110), 10, 10, []);
%end

%Observation Matrix (M)
M=zeros(6,10,size(stateModel,1));
I2=eye(2);
I3=eye(3);
Z33=zeros(3,3);
Z23=zeros(2,3);
Z32=zeros(3,2);
Z22=zeros(2,2);
for i=1:size(stateModel,1)
    alphai=stateModel(i,7);
    betai=stateModel(i,8);
    V=[-sin(alphai)*cos(betai)    -cos(alphai)*cos(betai);...
        cos(alphai)*cos(betai)   -sin(alphai)*cos(betai);...
        0                          cos(betai)];
    U1=-T_1*(sin(psi))^2*V;
    U2=-T_1*(cos(psi))^2*V;
    M(:, :, i)=[I3 zeros(3,3) U1 zeros(3,2);...
                I3 zeros(3,3) U2 zeros(3,2)];
end

```

```

end

%Jacobian Matrix (H)
%H=zeros(1,1);
H=M(:,:,1)*PhiCom(:,:,1);
for ii=2:size(M,3)
    H=[H;M(:,:,ii)*PhiCom(:,:,ii)];
end

%Calculus of h
%hX0=zeros(1,1);
hX0=[stateModel(1,1)-T_l*(sin(psi))^2*cos(stateModel(1,7))*cos(stateModel(1,8));...
stateModel(1,2)-T_l*(sin(psi))^2*sin(stateModel(1,7))*cos(stateModel(1,8));...
stateModel(1,3)-T_l*(sin(psi))^2*sin(stateModel(1,8));...
stateModel(1,1)+T_l*(cos(psi))^2*cos(stateModel(1,7))*cos(stateModel(1,8));...
stateModel(1,2)+T_l*(cos(psi))^2*sin(stateModel(1,7))*cos(stateModel(1,8));...
stateModel(1,3)+T_l*(cos(psi))^2*sin(stateModel(1,8))];
for ii=2:size(stateModel,1)
hX0=[hX0;...
stateModel(ii,1)-T_l*(sin(psi))^2*cos(stateModel(ii,7))*cos(stateModel(ii,8))
stateModel(ii,2)-T_l*(sin(psi))^2*sin(stateModel(ii,7))*cos(stateModel(ii,8))
stateModel(ii,3)-T_l*(sin(psi))^2*sin(stateModel(ii,8));...
stateModel(ii,1)+T_l*(cos(psi))^2*cos(stateModel(ii,7))*cos(stateModel(ii,8))
stateModel(ii,2)+T_l*(cos(psi))^2*sin(stateModel(ii,7))*cos(stateModel(ii,8))
stateModel(ii,3)+T_l*(cos(psi))^2*sin(stateModel(ii,8))];
end

%Vector of measurements
z=zeros(size(tout,1),6);
for ii=1:6
    z(:,ii) = interp1(t,x_IS_n(:,ii+1),tout);
end

%z0=zeros(1,1);
z0=z(1,:);
for ii=2:size(z,1)
    z0=[z0;z(ii,:)]';
end
Dz=z0-hX0;

%Minimize the function:
%x=lsqlin(H,);
DX0=(H.'*H)\(H.'*Dz); %!!!!!!!Reemplazar por la función de arriba

%end

```

7.6. dxdt2.m

```
%%%%%%%%%
```

```
function dydt = dxdt2(t,x)
```

```
% En principio el valor de rG se pasa por ventana
%rG=36000;
```

```
muE=398600.440; %[km^3/s^2] (JGM-3)
```

```
dydt = zeros(size(x));
```

```
%
```

```
rG = sqrt( x(1) * x(1) + x(2) * x(2) + x(3) * x(3) );
```

```
%
```

```
i=[x(1);x(2);x(3)]/rG;
```

```
i1=[1;0;0];
```

```
j1=[0;1;0];
```

```
k1=[0;0;1];
```

```
%
```

```
u=cos(x(7))*cos(x(8))*i1+sin(x(7))*cos(x(8))*j1+sin(x(8))*k1;
```

```
%
```

```
dydt(1) = x(4);
```

```
dydt(2) = x(5);
```

```
dydt(3) = x(6);
```

```
dydt(4) = -muE / (rG^3) * x(1);
```

```
dydt(5) = -muE / (rG^3) * x(2);
```

```
dydt(6) = -muE / (rG^3) * x(3);
```

```
dydt(7) = x(9); %u1=alfaPrima /// x(7)=alfa
```

```
dydt(8) = x(10);%u2=betaPrima /// x(8)=beta
```

```
dydt(9) = 2 * x(9) * x(10) * tan(x(8))+...
```

```
(3*muE*u'*cross(i,k1)*(u'*i))/...
```

```
(rG^3*(cos(x(8)))^2);
```

```
dydt(10)=(3*muE*(u'*i)/rG^3)...
```

```
*(u'*cross(i,i1)*sin(x(7))-u'*cross(i,j1)*cos(x(7)))...
```

```
-cos(x(8))*sin(x(8))*(x(9))^2;
```

```
%%%%%%%%%
```

7.7. dCdt3.m

```
%%%non-dimensional function for iteration%%%%%%%%%
```

```
function dComdt = dCdt3(t,x)
```

```
%Non dimensional state vector
```

```
%Constants to adimensionalize the matrix
```

```
muE=398600.440;          %[km^3/s^2]      Graviational constant      JGM-3
```

```
RE=63783.137;           %[km]          Ecuatorial Radius          WGS-84
```

```
TE=sqrt(RE^3/muE);     %[s]          Characteristic time for non-dimens
```

```

        VE=sqrt(muE/RE);          %[km/s]          Characteristic velocity for non-di

%Calculating the non-dimensional matrix x_n_adim
        x(1:10)=[x(1:3)/RE; x(4:6)/VE; x(7:8); x(9:10)*TE];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
x      = reshape(x, 10,11, []);
state = x(:,1);
Phi    = x(1:10,2:11);
%
muE=398600.440; %[km^3/s^2] (JGM-3)
dComdt = zeros(size(x));
%
rG = sqrt( state(1) * state(1) + state(2) * state(2) + state(3) * state(3) );
vec_rG = [state(1) state(2) state(3)];
%
i=[state(1);state(2);state(3)]/rG;
i1=[1;0;0];
j1=[0;1;0];
k1=[0;0;1];
%
u=cos(state(7))*cos(state(8))*i1+sin(state(7))*cos(state(8))*j1+sin(state(8))*k1;
%
dComdt(1,1) = state(4);
dComdt(2,1) = state(5);
dComdt(3,1) = state(6);
dComdt(4,1) = -state(1)/ (rG^3);
dComdt(5,1) = -state(2)/ (rG^3);
dComdt(6,1) = -state(3)/ (rG^3);
dComdt(7,1) = state(9); %u1=alfaPrima    ///    state(7)=alfa
dComdt(8,1) = state(10);%u2=betaPrima   ///    x(8)=beta
dComdt(9,1) = 2 * state(9) * state(10) +...
        (3*u'*cross(vec_rG,k1)'*(u'*vec_rG'))/...
        (rG^5*(cos(state(8)))^2);
dComdt(10, 1)=(3*(u'*vec_rG')/rG^5)...
        *(u'*cross(vec_rG,i1)'*sin(state(7))-u'*cross(vec_rG,j1)'*cos(state(7)))...
        -cos(state(8))*sin(state(8))*(state(9))^2;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Calculating the Matrix B
% B = zeros([3,3]);
B =-1/rG^5*diag([1-3*state(1)^2/rG^2,...
                1-3*state(2)^2/rG^2,...
                1-3*state(3)^2/rG^2]);

%Calculating the Matrix C
c1=state(2)*cos(state(7))-state(1)*sin(state(7));

```

```

c2=state(3)*cos(state(8))-(state(1)*cos(state(7))+state(2)*sin(state(7)))*sin(state(8));
C11=-5*state(1)*(vec_rG*i)/rG^2/cos(state(8))*c1+cos(state(7))*c1-sin(state(7))*(vec_rG*i)/rG;
C12=-5*state(2)*(vec_rG*i)/rG^2/cos(state(8))*c1+sin(state(7))*c1+cos(state(7))*(vec_rG*i)/rG;
C13=-5*state(3)*(vec_rG*i)/rG^2/cos(state(8))*c1+tan(state(8))*c1;
C21=-5*state(1)*(vec_rG*i)/rG^2*c2+cos(state(7))*(cos(state(8))*c2-(vec_rG*i)*sin(state(8)));
C22=-5*state(2)*(vec_rG*i)/rG^2*c2+sin(state(7))*(cos(state(8))*c2-(vec_rG*i)*sin(state(8)));
C23=-5*state(3)*(vec_rG*i)/rG^2*c2+sin(state(8))*c2+(vec_rG*i)*cos(state(8));
C =3/rG^5*[C11 C12 C13; C21 C22 C23];

%calculating the Matrix D
D11=3/rG^5*(c1^2-(state(1)*cos(state(7))+state(2)*sin(state(7)))*(vec_rG*i)/cos(state(8)));
D12=1/(cos(state(8)))^2*(2*state(9)*state(10)+3/rG^5*state(3));
D21=3/rG^5*(state(1)*sin(state(7))-state(2)*cos(state(7)))*(sin(state(8))*(vec_rG*i)/rG);
D22=state(9)^2*((sin(state(8)))^2-(cos(state(8)))^2)+3/rG^5*(c2^2+(vec_rG*i)*cos(state(8)));
D = [D11 D12; D21 D22];

%calculating the Matrix E
E = [2*state(10)*tan(state(8)) 2*state(9)*tan(state(8));...
     -2*state(9)*cos(state(8))*sin(state(8)) 0];

%Calculating the Matrix A
I2=eye(2);
I3=eye(3);
Z33=zeros(3,3);
Z23=zeros(2,3);
Z32=zeros(3,2);
Z22=zeros(2,2);
%
A=[Z33 I3 Z32 Z32;...
   B Z33 Z32 Z32;...
   Z23 Z23 Z22 I2;...
   C Z23 D E];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
dPhi = A*Phi;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
dComdt(1:10,2:11) = dPhi;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
dComdt = reshape(dComdt, 110,1, []);

```

8. Bibliography

Referencias

- [1] Tsilkovskii, K.E. *A Way to Stars*, Moscú, Izdatelstvo AN USSR, 1961 (en ruso).
- [2] Clarke, A.C. *Fountains of Paradise*, Ballantine Books, 1978.
- [3] Escobal, P.R. *Methods of Orbit Determination*, John Wiley & Sons, Inc., Nueva York, 1976.
- [4] Alfriend, K.T., Barnds, W.J., Coffey, S.L., Stuhrenberg, L.M., *Attitude and Orbit Determination of A Tethered Satellite System*. ASS Paper 95-351, 1995.
- [5] S. Cho, J.E. Cochran and D.A. Cicci. *Identification and orbit determination of a tethered satellite system*. Applied Mathematics and Computation 117 (2001) 301-312.
- [6] D.A. Cicci, C. Qualls and T.A. Lovell. *A look at tethered satellite identification using ridge-type estimation methods*. Applied Mathematics and Computation 119 (2001) 297-316.
- [7] Montenbruck, O., Gill, E., *Satellite Orbits*, Springer. Verlag, Berlin, Alemania, 2001.
- [8] Yuji Sakamoto, Kazuhiko Yotsumoto, Kazunori Sameshima, Masanori Nishio and Tetsuo Yasaka. *Methods for the orbit determination of tethered satellites in the project QPS*. Acta Astronautica 62 (2008) 151-158.
- [9] He Yong, Liang Bin, Xu WenFu and Li Cheng. *The Research of Real-Time Estimation Method on Tether Parameters for Tethered Satellite*. International Conference on Information and Automation June 22 -25, 2009, Zhuhai/Macau, China.
- [10] D.A. Cicci and C. Qualls. *Preliminary Orbit Determination of a Tethered Satellite Using the p-Iteration Method*. J of Astronaut Sci (2016) 63:158-173.
- [11] Orbit Determination Toolbox (ODTBX), <http://odtbx.sourceforge.net>. 2017-04-12. NASA Open Source Agreement (NOSA).
- [12] Space Tether Missions, https://en.wikipedia.org/wiki/Space_tether_missions. 2017-04-18.