DINÁMICA DE VEHÍCULOS ESPACIALES

Práctica 1: Cinemática y Dinámica de la Actitud. Visualización de la actitud mediante MATLAB/STK. Estudio de la estabilidad.

En esta práctica se emplearán los conceptos de cinemática y dinámica de actitud aprendidos en la asignatura y se utilizarán Matlab y STK para visualizar y demostrar dichos conceptos y algunas aplicaciones avanzadas.

Arrancar STK, Matlab, y descomprimir el archivo practica1.zip en un directorio elegido para trabajar.

1. Representación de la actitud con STK

A continuación vamos a aprender las opciones básicas de representación de actitud de STK.
En primer lugar creamos un escenario nuevo con las opciones por defecto y en él creamos un satélite (insert default), al cual llamamos "Sat", modificamos su óbtia de forma que tenga de semieje mayor 8800 km y excentricidad 0.2, y el resto de sus elementos orbitales (que no son muy relevantes a efectos de esta demostración) a elección del alumno. Pinchamos en el icono "View From/To" de la ventana 3D, y pinchamos nuestro satélite en "View from" y en "View to". Cambiar la vista para ver desde el sistema de referencia inercial, pinchando en "Reference frame" la opción "Earth Inertial Axes". Pulsamos OK. Hacer zoom hacia el satélite y encuadrarlo en la pantalla. Animando la simulación, observemos que por defecto tiene una actitud que permanece prácticamente constante relativa a la órbita (girando con ella) y sus paneles solares siempre apuntan al sol. Esto se debe al perfil de actitud por defecto, que estudiamos a continuación. Cambiar el modelo del satélite (3D graphics->Model) al de Hubble para ver con más precisión lo que sigue.

Para estudiar adecuadamente la actitud, primero añadimos una representación de los ejes cuerpo (Body Axes), ejes de la órbita (VVLH, Vehicle Velocity Local Horizontal), ejes inerciales (J2000) y del vector del Sol: para ello hacemos doble click en el satélite para obtener sus propiedades y pinchamos en "3D->Vector". Pinchamos "Show" en Body Axes , en VVLH y en "Sun Vector". Pinchamos OK y simulamos.

Animar la simulación. Observar en primer lugar el comportamiento de los ejes VVLH: el eje Z apunta a la tierrra, el eje X está contenido en el plano de la órbita (coincidirá con ella en una órbita circular, pero no en una órbita elíptica) y el eje Y es normal a la órbita. Observemos que los ejes cuerpos permanecen con actitud casi constante respecto a estos ejes (sería totalmente constante si no hubiera excentricidad). Observamos también que los paneles solares siguen constantemente al vector del Sol. Los ejes inerciales se mueven de forma aparente debido a que estamos manteniendo una visión fija con respecto a la Tierra. En realidad, son ellos los que se mantienen fijos en el espacio.

Para cambiar el comportamiento por defecto, pinchamos "Attitude" en las propiedades del satélite. En la versión básica sólo se puede fijar la actitud de dos formas:

- Basic: Perfil predefinido por STK (no obedece necesariamente las ecuaciones de la dinámica de actitud). El tipo (type) que viene por defecto, Nadir alignment with ECF velocity constraint, alinea el satélite con la órbita, manteniendo el eje Z cuerpo apuntando hacia el nadir (eje Z de VVLH). De ahí viene "... nadir constraint". La otra dirección la determina la velocidad, que en una órbita excéntrica no es constante respecto a la órbita, de ahí la lenta variación del sistema de referencia. Hay otros tipos pero no son demasiado interesantes.

- Precomputado: Proporcionamos la información de actitud en el tiempo a través de un archivo de actitud con extensión .a. Este archivo contiene un vector de tiempos y una matriz de cuaterniones que da la actitud para cada instante de tiempo del vector de tiempos. Esta será la opción que usaremos en la práctica.

El hecho de que los paneles solares apunten al Sol viene dado por la propiedad "Model pointing" en 3D Graphics. Otros modelos permiten apuntar el cuerpo del satélite, antenas, etc.. a diferentes blancos.

2. Cinemática

Ahora pasamos a ver como podemos representar secuencias generadas por Matlab. Para simplificar la visualización, eliminar el vector Sol y el sistema de referencia VVLH. Para poder vigilar los cuaterniones y otros parámetros en tiempo real, en 3D graphics/Data Display se puede seleccionar "Attitude Quaternions" (también Euler Angles).

IMPORTANTE: Se sigue la convención de STK de forma que la parte escalar del cuaternión es la cuarta componente y la parte vectorial las primeras tres componentes.

Las siguientes funciones de Matlab están disponibles para cálculos cinemáticos:

- C=euler3132DCM(euler313) obtiene la DCM a partir de un vector de ángulos de Euler (en radianes) en la secuencia 3-1-3.
- euler313=DCM2euler313(C) obtiene un vector con los ángulos de Euler en la secuencia 3-1-3 a partir de la DCM. Se ha fijado que el segundo ángulo se encuentre entre -90 y +90.
- q=DCM2quat(C) a partir de la DCM genera el cuaternión correspondiente.
- C=quat2DCM(q) a partir del cuaternión genera la DCM correspondiente.
- [T,C]=EDCDCM(times,omegas,C0) integrador de la ecuación diferencial cinemática para la DCM, con condición inicial C0, para el vector de tiempo times y la velocidad angular omegas (cada fila de tres columnas representa las componentes de la velocidad angular en cada instante de times). El resultante es un cell con tantos componentes como la longitud de times.
- [T,q]=EDCq(times,omegas,q0) integrador de la ecuación diferencial cinemática para los cuaterniones, con condición inicial q0, para el vector de tiempo times y la velocidad angular omegas (cada fila de tres columnas representa las componentes de la velocidad angular en cada instante de times). El resultante es una matriz cuyas filas son los cuaterniones y con tantas columnas como la longitud de times.
- qp=qprod(q1,q2) da el producto de dos cuaterniones q1 y q2.
- qc=qconj(q) da el conjugado de un cuaternión.
- [eje,angulo]=q2euler(q) permite obtener el eje y ángulo de Euler a partir del cuaternión.
- q=euler2q(eje,angulo) permite obtener el cuaternión a partir del eje y ángulo de Euler.

Además se dispoone de la siguiente función:

- generar_archivo_actitud (nombre,T,Q), genera un archivo de actitud (.a) para usar con STK, recibe el nombre del archivo (string) sin extensión, un vector columna T con todos los tiempos en los que se ha calculado la actitud (supuesto ordenado, debe empezar por 0), y una matriz Q cada una de cuyas filas es el cuaternión de actitud correspondiente al tiempo T en dicha fila.

Ejemplo 1. En primer lugar, vamos a generar un perfil de actitud constante en el tiempo, respecto al sistema inercial J2000.
Para ello, seguir los siguientes pasos:

1. Generar una actitud inicial. Por ejemplo eligiendo unos ángulos de Euler en el conjunto 3-1-3, digamos 30º,45º,0º. Calculamos con los scripts la matriz correspondiente a este ángulo, la pasamos a cuaterniones y así obtenemos q0.
2. Generamos un vector de tiempos entre 0 y 3600, con paso 1 segundo, es decir, t=[0:3600].
3. Generamos una velocidad angular igual a cero para todos estos tiempos. Para ello, simplemente Omega=zeros(3601,3).
4. Integramos la ecuación diferencial de los cuaterniones. Como la velocidad angular es cero debe salir un cuaternión constante igual a q0 en todo tiempo: [T,Q]=EDCq(t,Omega,q0).
5. Generar el archivo .a: generar_archivo_actitud ('ejemplo1',T,Q).
6. Ver el archivo con un editor de texto y entender todas sus lineas.
7. En Sat1->propiedas->Basic,attitude->marcar precomputed, elegir el archivo en file y apply.

Pulsar reset y play en STK. Visualizar el resultado. Ojo: Verificar que la animación se visualiza en el periodo de tiempo correcto (desde el inicio de la simulación hasta una hora).

Ahora se van a repetir otros ejemplos. Para los ejemplos, se repiten los mismos pasos variando el procedimiento de generación de Q.

Ejemplo 2. Velocidad angular constante. Seleccionar por ejemplo omega=[0.1 0.2 0.4] todo el tiempo. Observar que también se puede visualizar el vector velocidad angular.

Ejemplo 3. Creamos ahora un perfil de actitud de maniobra.
La maniobra partirá de unas actitudes inicial y final dadas, en un tiempo de maniobra dado.

Para ello, usamos la siguiente propiedad: si partimos de un cuaternión inicial q_0 a otro final, q_F, existe un cuaternión que representa la rotación de velocidad angular mínima q_{rot} que cumple:

$$q_F = q_0 * q_{rot}$$

Por tanto:

$$q_{rot} = \frac{1}{q_0} * q_F = \frac{q_0^*}{\|q_0\|^2} * q_F = q_0^* * q_F$$

Estas multiplicaciones de cuaterniones pueden hacerse fácilmente en Matlab con los scripts proporcionados.

Una vez obtenemos el cuaternión de rotación, extraemos sus eje y ángulo de Euler, y construimos el cuaternión de rotación instantánea:

$$q_{rot}(t) = \begin{bmatrix} \vec{e}\sin\left(\frac{\omega t}{2}\right) \\ \cos\left(\frac{\omega t}{2}\right) \end{bmatrix}$$

donde la velocidad angular viene dada por:

$$\vec{\omega} = \vec{e}\frac{\theta}{tMAN},$$

siendo θ el ángulo de Euler y tMAN el tiempo de maniobra dado.

De esta manera, la actitud en función del tiempo es:

$$q(t) = q_0 * q_{rot}(t)$$

En este ejemplo se pide: partiendo de la misma actitud inicial que antes, generar una velocidad angular que permita llegar a la actitud (dada en ángulos de Euler en el conjunto 3-1-3) 60º,15º,180º a lo largo de una hora. Posteriormente "quedarse" en dicha actitud durante media hora.

Ejemplo 4: Generar un giro de forma que un vector en ejes cuerpo coincida con un vector en ejes inerciales.

Supongamos ahora que partimos de la misma actitud inicial y queremos que el eje z coincida con el vector Sol (aproximadamente constante en ejes inerciales). En este caso también usamos los conceptos de ángulo y eje de Euler, pero calculados usando el vector z (es decir [0 0 1], que llamamos v0) y el vector Sol en ejes cuerpo (v1). Para calcular el vector Sol en ejes cuerpo, tomamos su valor en ejes J2000 (usar Report&Graph manager), y con q0 lo pasamos a ejes cuerpo y lo normalizamos. Ahora, el eje de Euler será el producto vectorial normalizado de v1 por v0, y el ángulo de Euler será el ángulo entre ambos vectores (arcocoseno del producto escalar).

3. Dinámica de actitud. Visualización via Matlab/STK.

Todos los ejemplos vistos previamente tienen una dinámica "inventada". En los ejemplos que veremos ahora se van a visualizar ejemplos más realistas de la dinámica de actitud vistos en clase, que servirán para refrendar la teoría.
Para ello se usarán una serie de scripts de Matlab. Como antes visualizar el satélite directamente en la ventana 3-D, en el sistema de referencia inercial, y mostrar los ejes J2000 y ejes cuerpo.

Ejecutar el script "cuerpoLibre1.m". Este script genera simulaciones para una rotación en torno al eje menor, intermedio y mayor (las tres primeras gráficas), así como una rotación en torno al eje mayor con precesión (las otras gráficas). Se utiliza un modelo sin disipación de energía.

Las gráficas no dan mucha información, así que para visualizarlas usar el comando
generar_archivo_actitud ('dinamica1',times,quat);
sustituyendo "quat" por qMay, qMen, y qInt respectivamente, y luego cargarlo en STK (acordarse de darle a reload, apply) para ver los tres casos. **La simulación sólo dura 200 segundos por lo que hay que ralentizar considerablemente la velocidad de simulación.**

Ver ahora el caso de rotación con precesión, representado por qPrec
generar_archivo_actitud ('precesion',times, qPrec);
cargarlo con STK, y para ver los "conos" que representan el movimiento, hacer lo siguiente:
-Añadir la visualización del vector velocidad angular (en 3-D Graphics, Vector, Add y buscar AngVelocity Vector), seleccionar persistencia, y connect:trace.
-Copiar el Sat1 como otro satélite, y cambiar únicamente los ejes de la velocidad angular,

seleccionando Axes: J2000.

Si se han llevado a cabo estos pasos correctamente se visualizará un circulo deslizando sobre otro, son los vistos en teoría.

¿Es este el caso prolato u oblato? Modificar el tensor de inercia en el script para observar el otro caso. Borrar el satélite copiado al terminar.

Cerrar las gráficas de MATLAB y borrar los vectores de velocidad angular. Ejecutar ahora el script "cuerpoLibre2.m". Este script ejecuta un modelo con disipación de energía, tal como se vio en teoría (con una burbuja de combustible en el depósito que disipa energía por rozamiento).

La figura 1 muestra la estabilidad (asintótica) del eje mayor (x), frente a una pequeña perturbación. La figura 2 muestra la inestabilidad del eje menor (z). La figura 3 es del mismo caso y muestra como se disipa lentamente la energía (mientras que el emomento cinético permanece constante). La figura 4 muestra como las curvas polodia se vuelven convergentes al eje mayor, dibujando la elipse del momento cinético en el proceso. La figura 5 la inestabilidad del eje intermedio (y). En todos los casos el vehículo termina rotando en torno al eje mayor, aunque es imprevisible, si se comienza con una rotación intermedia o menor, si el resultado final será una rotación "positiva" o "negativa". Esto se ejemplifica con la figura 6, con unas condiciones muy muy parecidas a las de la figura 2 pero que termina en una rotación positiva (la figura 2 termina en una rotación negativa). Ésto es un ejemplo de caos.
Para ver estas simulaciones en STK se usa de nuevo el comando

```
generar_archivo_actitud ('dinamica2',times,quat);
```

cambiando quat por qMay, qMen, qInt, qMen2 respectivamente. Se puede visualizar la curva polodia (para los casos de ejes menor e intermedio) dibujando el vector velocidad angular y añadiendo persistencia en la traza, como antes. Aunque dichas curvas estarán dibujadas en ejes cuerpo y por tanto "se moverán".
Correr ahora el script cuerpoForz.m. Simula el satélite en rotación sometido a un par de perturbación. Se muestran tres figuras, la primera son las velocidades angulares, la segunda los ángulos evolucionando respecto al tiempo, y la tercera el ángulo uno frente al dos. Si la velocidad angular de rotación es suficientemente alta, el efecto del par perturbador es pequeño y la curva de los ángulos es un epiciclo de pequeña amplitud (si no es suficientemente alta, hay efectos no lineales y la curva es más compleja y de gran amplitud). La forma exacta del epiciclo depende de la relación entre los momentos de inercia. Visualizarlo en STK:

```
generar_archivo_actitud ('dinamica3',times,q);
```

Para ver bien en STK la curva que describen los ángulos y que se ve en la figura 3 de MATLAB, se puede hacer de la siguiente forma: en el satélite, 3D-Graphics, Vector, eliminar todos los vectores y sistemas de referencia presentes, y añadir el eje z de los ejes cuerpo. Marcar show en persistence, connect:trace, y como ejes los J2000.

Probar a modificar la velocidad angular del satélite, haciéndola menor (linea 20 del script cuerpoForz) o a cambiar las inercias (linea 12) para ver como se modifican los resultados. Si se baja lo suficiente la velocidad angular inicial (por ejemplo hasta 0.1), se ve que se pierde rigidez giroscópica.

Finalmente ejecutar el script cuerpoGG. Se simula el efecto del gradiente gravitatorio en tres casos: sin otros momentos perturbadores (qg1, primera gráfica), sometido a un momento perturbador constante durante diez segundos iniciales (qg2, segunda gráfica), y sometido al mismo momento pero con un volante de inercia rotando con eje de giro perpendicular al plano orbital (qg3, tercera gráfica). Comparar los tres casos y visualizarlo en STK, **teniendo en pantalla los ejes cuerpo y los ejes VVLH y fijando satellite ICR axis en la visualización**. Además hay que modificar los elementos orbitales del satélite para ajustarlos al de la simulación: poner como semieje mayor 7000 y como excentricidad 0. Además, para poder simular en STK es necesario escribir el cuaternión de actitud (qg1,qg2 o qg3) en relación a la órbita para lo cual hay que ejecutar

```
qg1_orb=qLVLH(T_1,a,raan,inc,omegaperi,theta0,qg1);
```

sustituyendo los valores de los elementos orbitales, y ya se puede visualizar en STK:

```
generar_archivo_actitud ('dinamica2',T_2, qg1_orb);
```

Igualmente para qg2 y qg3, pero el vector de tiempo correcto para estos vectores de actitud es T. ¿Cuál es la posición de equilibrio (qué ejes están apuntando en qué dirección)? ¿A qué se debe la diferencia entre el caso 2 y 3?

EJERCICIOS PARA ENTREGAR (HASTA +1 PUNTO)

Elegir una matriz de inercias asimétrica y simular las siguientes situaciones:

a. Un satélite en órbita afectado por el gradiente gravitatorio y que se encuentre girando en torno a su eje mayor, fuera de la posición de equilibrio. ¿Cuál es el efecto del gradiente gravitatorio? Probar para diferentes valores de la velocidad angular (pequeñas, grandes).

b. Un satélite con disipación de energía y con un volante de inercia rotando a una cierta velocidad, girando en torno a su eje intermedio. Comprobar las reglas obtenidas en clase que permiten predecir la estabilidad de este giro, comparándolas con las reglas sin disipación de energía.

c. Elegir una actitud inicial y una final. Para un vehículo modelado como sólido rígido (sin ruedas ni perturbaciones), y partiendo del reposo (velocidad angular cero) y pretendiendo llegar al reposo en la actitud final, encontrar (analíticamente) unos pares (M) que realicen dicha transición en diez minutos. ¿Sería posible que el par fuera nulo al principio y al final también? Simularlo y verificar que efectivamente el satélite se comporta adecuadamente. Si el modelo no es perfecto (por ejemplo, si hay disipación de energía y gradiente gravitario), ¿el comportamiento es el mismo o muy parecido, o varía bastante?

d. Finalmente, y para el caso anterior, utilizando el modelo con tres ruedas de reacción (tema 3 transparencia 32), suponiendo que no hay perturbaciones (M=0), determinar los valores que tendrían que tomar los pares de los motores internos (J_1,J_2,J_3) para realizar la maniobra pedida.

Entregar un máximo de 10 páginas describiendo los cuatro apartados, incluyendo las gráficas que se consideren convenientes para la explicación. No hay extensión mínima. Se admiten entregas parciales, si bien puntuarán menos. Para aprobar las prácticas al menos hay que entregar dos apartados.

Grabar un vídeo con STK del caso que se considere más interesante (ver boletín práctica 1 de Mecánica Orbital en http://aero.us.es/move/files/PR1STK.pdf) y subirlo a Youtube o un servicio similar. Mandar el enlace al profesor.

Plazo: tres semanas exactas a partir de la fecha de la práctica.

Se puntuará: claridad de conceptos y calidad de la redacción, extracción de conclusiones y comentarios personales. No es necesario adjuntar código. No se aceptarán plagios.