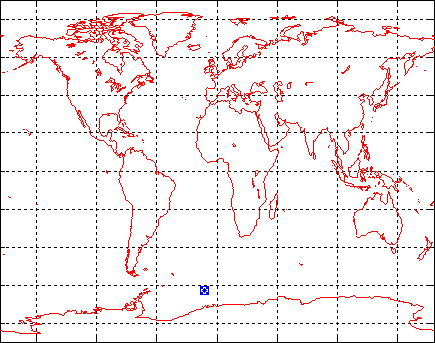
lunes, mayo 19, 2008

**[Órbita de Molniya o Molniya Orbit o молния](http://rocoblog.blogspot.com/2008/05/rbita-de-molniya-o-molniya-orbit-o.html)**

[[](http://www2.udec.cl/~franciscoroco/molniya.gif)](http://www2.udec.cl/~franciscoroco/molniya.gif)  
  
  
La [órbita de Molniya](http://es.wikipedia.org/wiki/%C3%93rbita_de_Molniya) es una órbita inventada por los camaradas de la ex [Unión Soviética](http://es.wikipedia.org/wiki/URSS) para utilizarla en sus satélites de comunicación. Antes de hacer esto, pensé que era fácil comunicar vía satélite a cualquier parte del globo, pero no es así, pues sólo para lugares que estan cercanos al Ecuador se puede poner un [satélite geoestacionario.](http://es.wikipedia.org/wiki/%C3%93rbita_geoestacionaria)  
  
¿Que pasa con los lugares que están cercanos a los polos?  
  
El satélite geoestacionario permenece sobre el punto que se requiere porque mantiene su velocidad igual a la de este punto. ¿Que características debe tener la órbita para cumplir con esta misión?  
  
La velocidad de rotación de la tierra es constante, y si quiero que el satélite siga a un punto en la superfice la velocidad de rotación - o velocidad angular - del satélite debe ser la misma. La órbita con la velocidad angular constante es la circular, que usan los geoestacionarios, en ella, la distancia del satélite a la tierra permanece igual siempre -no como el la órbita elíptica-.  
  
La órbita del satelite estará sobre un plano, este plano contendrá siempre el centro de la tierra, es decir , si quiero que mi satélite geostacionario pase por Concepción, defino el plano por el que esta esta ciudad, y me doy cuenta que al girar sobre este punto el satélite no pasará por el centro de la tierra, es decir no puedo poner un satelite geoestacionario ahí, ¿Dónde puedo? En el ecuador porque el plano de las ubicaciones que están ahi contiene el centro de la tierra.  
  
En ese momento es cuando el DT llama para que entre a la cancha a la órbita elíptica. La característica que tiene esta órbita es que tiene una distancia máxima, el [apogeo,](http://es.wikipedia.org/wiki/Apogeo) y una mínima, el [perigeo.](http://es.wikipedia.org/wiki/Perigeo) La gracia esta en que en el perigeo la velocidad del satélite  
  
La que Esta órbita es una buena solución para el problema de cubrir zonas de alta latitud, dicho de otra forma , de zonas cercanas a los polos. Los camarradas pensarón en esta órbita para su Madrrre RRRusia, pero también puede servir pa nuestro Chile querido porque tenemos territorio muy al sur del mundo.  
  
El objetivo de las siguientes lineas de sintaxis es reproducir por cuales zonas sobre el globo pasa nuestro satélite y de qué zonas cubre, o sea que zonas en ese instante es capaz de dar comunicación.  
  
Hay que decir que esta órbita tiene 63.4° de inclinación respecto del Ecuador, y tiene su apogeo cerca de Ciudad de Mejico, y algún lugar de Rusia.  
  
Lo notable de esto es que se hizo en tiempos donde no existían metodos como Matlab u Octave para hacer todos estos gráficos y cálculos, los camarradas la sacaron con lápiz, regla de cálculo y mucho ingenio. Debo confesar que esto me costo y si hubiese sido uno de esos rusos probablemente hubiese terminado en el [**Gulag**](http://es.wikipedia.org/wiki/Gulag) por no presetar cálculos a tiempo.

**Sintaxis en MATLAB**

clc;  
clear all;  
close all

**Datos**

rt=6400; %Longitud del Radio de la Tierra  
g0=9.8; %acel. gravedad  
rper=rt+600; %Perigeo  
betha=(rper/rt); %Periogeo Adimensional  
ii=-63.4\*pi/180; %angulo orbita -ecuador  
lon\_ap=-100; %longitud del apogeo  
lon\_ap=lon\_ap+90;  
uv=((rt\*g0/1000)^(0.5)); %Adimensional de velocidad  
ut=rt/uv; %Adimanensional de Tiempo  
T=12\*3600/ut; %Período de la órbita adimensional

**Cálculos**

Se calculan las constantes de la ecuacion polar de la orbita y se calcula el semi eje mayor

[mhtml:file://I:\ARCHIVOS%20MATLAB\Archivos%20HTML\RocoBlog-2.mht!http://2.bp.blogspot.com/_Ep6DnIy71qM/SDJBQf_s6BI/AAAAAAAAAEw/9URPq6s832E/s400/molniya_eq50705.png](http://2.bp.blogspot.com/_Ep6DnIy71qM/SDJBQf_s6BI/AAAAAAAAAEw/9URPq6s832E/s1600-h/molniya_eq50705.png)

a=(T/(2\*pi))^(2/3);  
  
  
e=1-betha/a;  
k=(a\*(1-e^2))^0.5;  
  
  
t=linspace(0,24,50\*24)\*3600/ut; %vector tiempo en seg  
E=zeros(1,length(t)); %anomamalía verdadera  
error=1e-8; %Tolerancia para el cálculo de E  
ini=0;

**Cálculo de E en función del tiempo**

Se resuelve la ecuación

[mhtml:file://I:\ARCHIVOS%20MATLAB\Archivos%20HTML\RocoBlog-2.mht!http://1.bp.blogspot.com/_Ep6DnIy71qM/SDJBNP_s6AI/AAAAAAAAAEo/_CIV7nZsadM/s400/molniya_eq27602.png](http://1.bp.blogspot.com/_Ep6DnIy71qM/SDJBNP_s6AI/AAAAAAAAAEo/_CIV7nZsadM/s1600-h/molniya_eq27602.png)

Para ello se utiliza el método de Newton o bisección, para ubicar la solución de E, en la ecuación que la relaciona E con el tiempo

for i=1:length(t)  
ini=0;  
f=t(i)-T/(2\*pi)\*(E(i)-e\*sin(E(i))); %Función a buscar solución  
z=1;  
while abs(f)>error  
E(i)=ini;  
f=t(i)-T/(2\*pi)\*(E(i)-e\*sin(E(i)));  
fa=f;  
while sign(f) == sign(fa) %Búsqueda del cambio de signo  
E(i)=E(i)+pi/180\*10^(2-z);  
fa=f;  
f=t(i)-T/(2\*pi)\*(E(i)-e\*sin(E(i)));  
ini=E(i)-pi/180\*10^(2-z);  
end  
z=z+1;  
end  
end

**Determinación v y rho para cada tiempo**

La anomalía verdadera v, se obtiene a partir de E con:

[mhtml:file://I:\ARCHIVOS%20MATLAB\Archivos%20HTML\RocoBlog-2.mht!http://2.bp.blogspot.com/_Ep6DnIy71qM/SDJBUf_s6CI/AAAAAAAAAE4/8AInt3n4Ktw/s400/molniya_eq171326.png](http://2.bp.blogspot.com/_Ep6DnIy71qM/SDJBUf_s6CI/AAAAAAAAAE4/8AInt3n4Ktw/s1600-h/molniya_eq171326.png)

La ecuación de la Elipse, que da la posición del radio adimensional rho para cada ángulo de la anomalía verdadeda v es:

[mhtml:file://I:\ARCHIVOS%20MATLAB\Archivos%20HTML\RocoBlog-2.mht!http://2.bp.blogspot.com/_Ep6DnIy71qM/SDJBIf_s5_I/AAAAAAAAAEg/3E6UVDYe1Ho/s400/molniya_eq18538.png](http://2.bp.blogspot.com/_Ep6DnIy71qM/SDJBIf_s5_I/AAAAAAAAAEg/3E6UVDYe1Ho/s1600-h/molniya_eq18538.png)

v=2\*atan(((1+e)/(1-e))^0.5\*tan(E/2));  
  
  
r=k^2./(1+e\*cos(v));  
rm=r.\*rt;

**Paso a ejes cartesianos**

y=rm.\*cos(v).\*cos(ii);  
z=rm.\*cos(v).\*sin(ii);  
x=sqrt(rm.^2-y.^2-z.^2).\*(-sign(sin(v)));

**Paso a coordenadas polares**

tm=atan2(y,x)-pi/2;  
fim=atan2(z,sqrt(x.^2+y.^2));

**Se toma en cuenta el movimiento de la tierra y la posición del perigeo**

tm=tm-2\*pi/(24)\*t\*ut/3600+lon\_ap\*pi/180;

**Orden del los vectores para que queden entre -180 y 180**

for i=1:length(tm)  
while abs(tm(i))>pi  
tm(i)=tm(i)-2\*pi\*sign(tm(i));  
end  
end

**Ordenamos el Vector para que no aparezcan líneas cruzadas**

i=1;  
while i < class="keyword">if sign(tm(i))~=sign(tm(i+1)) && abs(tm(i))>pi/180\*100  
tm =[ tm(1:i) NaN tm(i+1:end)];  
fim=[fim(1:i) NaN fim(i+1:end)];  
i=i+2;  
end  
i=i+1;  
end

**Zonas de cobertura**Se calcula el angulo de cobertura para cada posición de la órbita y se ordenana los vectores para cuando cruzan de una parte del planeta a la otra

alfa=.57006302;  
g=pi/2-alfa-asin(sin(pi/2+alfa)\*rt./rm);  
for i=1:length(g)  
xz(i,:)=tm(i)+g(i).\*cos(linspace(0,2\*pi,60));  
yz(i,:)=fim(i)+g(i).\*sin(linspace(0,2\*pi,60));  
end

**Ordenamiento de la zona de cobertura**

[l,a]=size(yz);  
for i=1:l  
for j=1:a  
if abs(yz(i,j)) > pi/2  
yz(i,j)=pi-yz(i,j);  
xz(i,j)=-xz(i,j);  
end  
end  
end

**Gráfico de La Órbita**

Se carga la costa del mundo, la órbita y la zona de cobretura, además se grafica en forma de animación para ver la zona de cobertura en cada proyección de la orbita del satélite

load coast; %Carga costas del mundo  
for i=1:length(g)  
plot(tm\*180/pi,fim\*180/pi,'r') %Grafica Órbita  
hold on  
plot(long,lat,'k') %Carga mapa  
xlim([-180 180]) %Límites  
ylim([-90 90])  
plot(180/pi\*xz(i,:),180/pi\*yz(i,:),'.')  
plot(tm(i)\*180/pi,fim(i)\*180/pi,'o')  
pause(.01)  
hold off  
end