

Unicamp - Universidade Estadual de Campinas

FEEC - Faculdade de Engenharia Elétrica e Computação



EE400 - Métodos da engenharia elétrica - Turma A

Projeto D

- **Pedro Romero Santos - 288643**
- **Pedro Rafael Moreira Ataide Amorim - 278519**
 - **José Pedro Iglesias Garcia - 278472**

Link do código no Github:

<https://github.com/PedroRomeroSantos/EE400-Projeto-D-Grupo-P.git>

Parametrização das órbitas

Vamos usar um conjunto de dados conhecidos como elementos orbitais de Kepler. Esses elementos são parâmetros fundamentais utilizados para descrever com precisão a trajetória dos satélites ao redor da Terra.

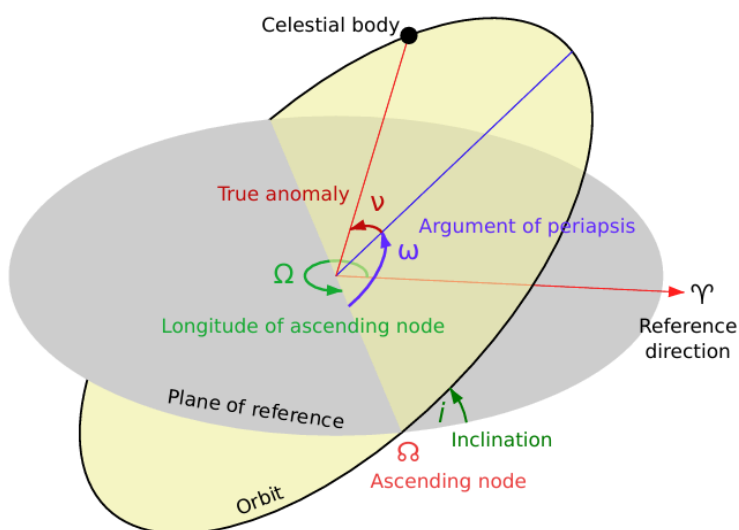


Figura 2: Elementos orbitais de Kepler

Parâmetro	Descrição
e	Excentricidade da órbita, varia entre 0 e 1
a	Semieixo maior, metade da distância entre apogeu e perigeu
ω	Argumento do perigeu, define a orientação da elipse no plano orbital
i	Inclinação, ângulo entre o plano equatorial da terra e o plano orbital
Ω	Longitude do nó ascendente, ângulo entre uma direção de referência e o nó ascendente
Δt	Tempo desde o perigeu, tempo decorrido desde que o satélite passou pelo perigeu
μ	Parâmetro de gravitação da terra, sendo G a constante universal da gravitação e M a massa da terra

a) Parametrização da forma no plano orbital: Vamos descrever a elipse no sistema de coordenadas perifocal. Nesse sistema, a elipse está contida no plano, com a origem localizada no foco mais próximo ao perigeu, e o eixo x orientado na direção do perigeu e o eixo y com ângulo $\nu = 90^\circ$

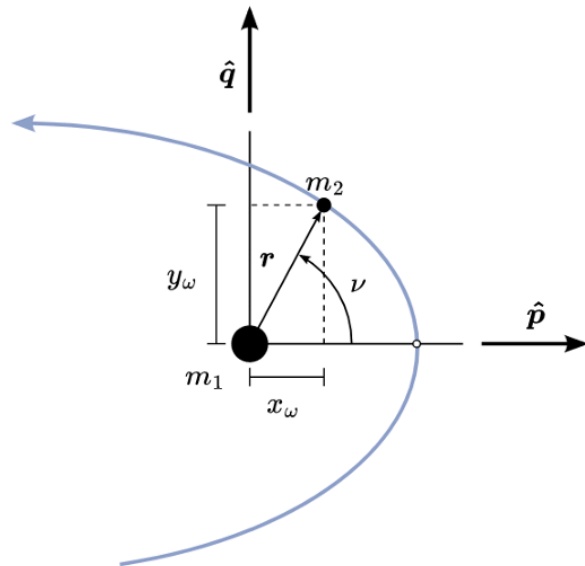


Figura 3: Sistema de coordenadas perifocal

Vamos descrever parametrização cartesiana em termos de $r(t)$, distância entre o satélite e o centro da terra, e o ângulo $\nu(t)$, chamado de anomalia verdadeira.

- $x = x(r, \nu) = r(t) \times \cos(\nu(t))$
- $y = y(r, \nu) = r(t) \times \sin(\nu(t))$

Baseando-se na elipse circunscrita a uma circunferência de raio igual ao semi-eixo maior, com o ângulo central E (anomalia excêntrica), é conveniente utilizar parametrizações em termos de E , a e e .

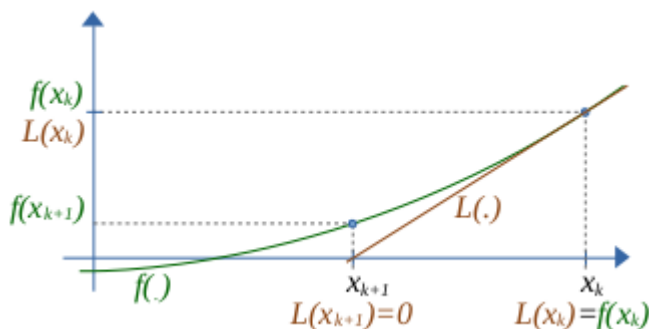
da equação da elipse, substituímos:

$$\Rightarrow \frac{(x+d)^2}{a^2} + \frac{y^2}{b^2} = 1, \text{ com } a = \frac{c}{1-e^2}, b = \frac{c}{(1-e^2)^{\frac{1}{2}}} \text{ e } d = \frac{c \times e}{1-e^2}$$

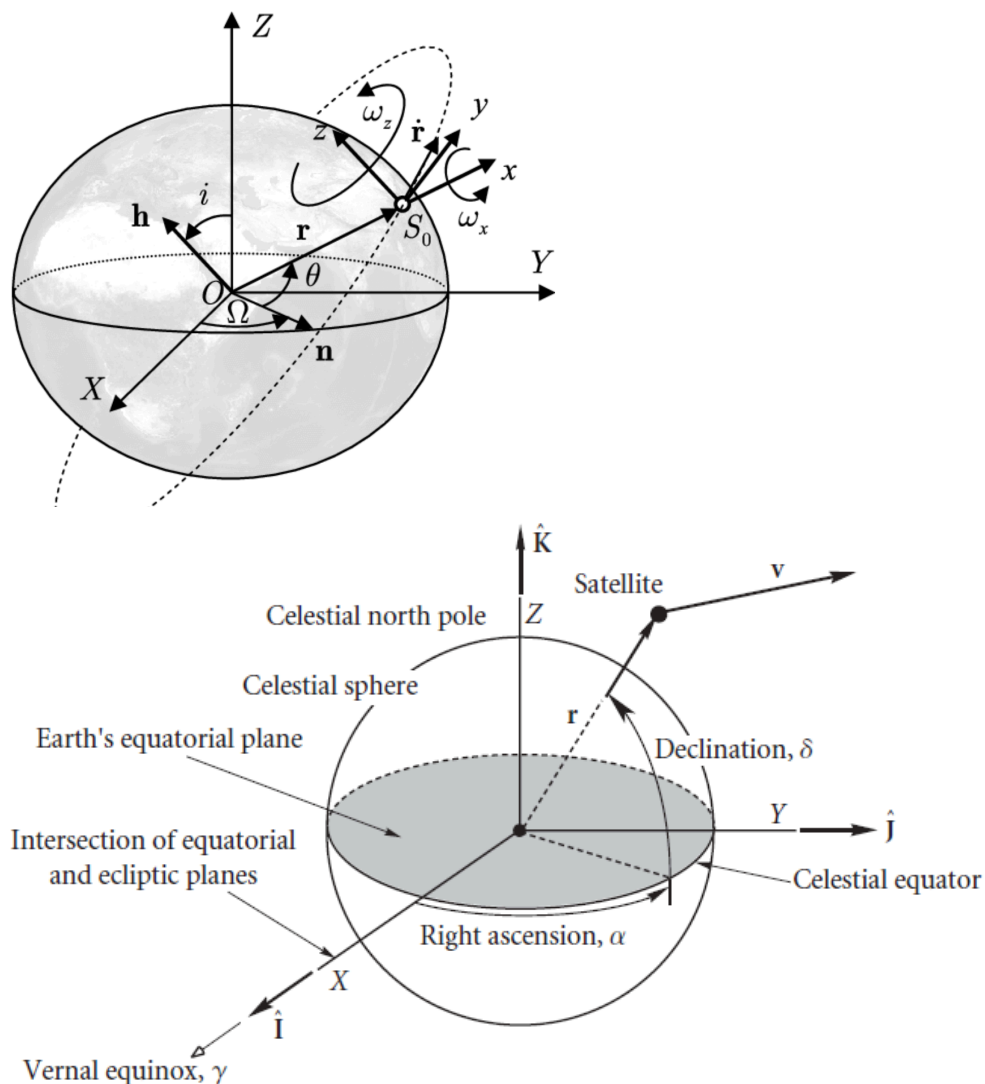
Para a circunferência, temos x_c e y_c que precisam ser ajustados para elipse de forma que o x_c precisa ser deslocado $d = a \times e$ e o y_c precisa ser comprimido $(1 - e^2)^{\frac{1}{2}}$.

- $x_c = a \times \cos(E) \Rightarrow a \times \cos(E) + a \times e \Rightarrow a \times \cos(E) + d$
- $y_c = a \times \sin(E) \Rightarrow a \times \sin(E) \times (1 - e^2)^{\frac{1}{2}}$

b) Parametrização do tempo: Para incluir a dependência temporal no problema, partiremos da equação de Kepler, $E(t) - e \cdot \sin E(t) = M(t)$. Porém ela é uma equação transcendente e não pode ser representada como uma fração de polinômios, o que significa que não existe uma fórmula geral para resolvê-la e por isso vamos resolver por métodos numéricos. Para esse caso, utilizaremos a aproximação linear de Newton-Raphson (imagem abaixo). Nele, parte-se de uma aproximação inicial para a raiz da equação e então calcula-se a derivada da função no ponto correspondente. A cada iteração, o constrói-se uma reta tangente à curva da função nesse ponto. A interseção dessa reta tangente com o eixo x fornece uma nova aproximação para a raiz. Esse processo é repetido até que a solução se aproxime o suficiente do valor real da raiz, de acordo com uma margem de erro escolhida. Vamos usar a biblioteca SciPy em Python, que tem esse método como disponível.



c) Parametrização Espacial: Para descrever a órbita plana no espaço tridimensional, vamos escrever as coordenadas do sistema perifocal no sistema ECI (Earth-Centered Inertial), no qual tem a terra como origem do seu sistema de eixos.



É necessário rotacionar os eixos até que coincidam com o esperado na seguinte ordem:

1. Rotacionar em torno z_ω , até que o eixo x_ω esteja alinhado com o nó ascendente.
2. Rotacionar em torno do novo x'_ω , até que o plano orbital e equatorial estejam alinhados.
3. Rotacionar em torno de z' , para alinhar o nó ascendente com a direção de referência.

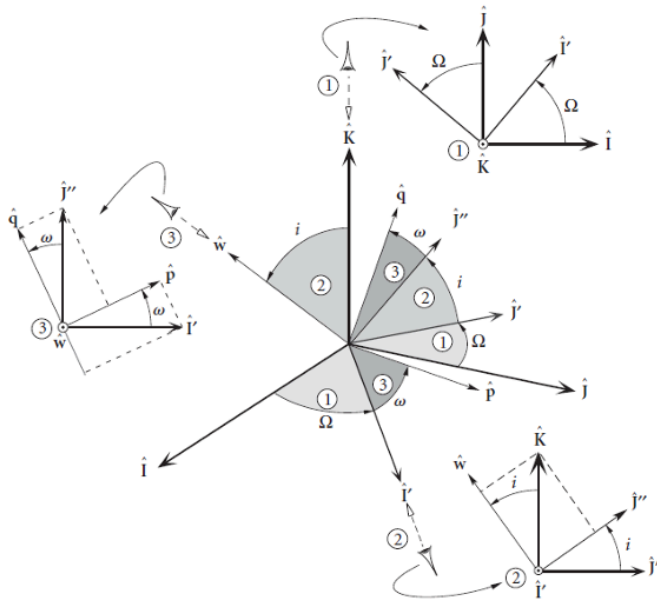


Figura 5: Rotações para alinhar os ângulos

- Matriz de rotação em x:

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) \\ 0 & \sin(\alpha) & \cos(\alpha) \end{pmatrix}$$

- Matriz de rotação em z:

$$\begin{pmatrix} \cos(\alpha) & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Observando os parâmetros de Kepler no item a), orientação, ordem de rotação e as imagens acima, temos:

Parâmetros de Kepler	Orientações e ordem
----------------------	---------------------

Ω	Rotacionar em torno z_{ω} , até que o eixo x_{ω} esteja alinhado com o nó ascendente
i	Rotacionar em torno do novo x'_{ω} , até que o plano orbital e equatorial estejam alinhados
ω	Rotacionar em torno de z' , para alinhar o nó ascendente com a direção de referência

Trilateração e gradiente descendente

Com base em um conjunto de medidas de distância em relação a vários transmissores, um receptor pode determinar sua posição ao resolver um problema geométrico de intersecção de círculos (ou esferas). Para resolver precisamos de 4 itens:

1. A Posição de cada um dos satélites (não é mais um problema graças a parte de parametrização das órbitas)
2. As distâncias relativas para cada um dos satélites
3. Equacionar cada uma das leituras
4. Resolver o sistema de equações resultante

a) As distâncias relativas para cada um dos satélites: A diferença entre o tempo de recebimento de um sinal, Time of arrival (TOA), e o tempo de emissão Time of transmission (TOT) faz com que seja possível determinar o Time of flight (TOF). Como a velocidade da luz é conhecida, podemos usá-la para mensurar essas diferenças de tempo de um sinal.

b) Sistema de equações:

- $TOF = TOA - TOT$
- $c = \frac{\rho_i}{TOF}$, com c sendo a velocidade da luz e ρ_i a distância do drone até o satélite i .

- $\Rightarrow \rho_i = \sqrt{(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2}$

É possível observar que a relação entre as variáveis acima não é linear, então vamos precisar usar algum método iterativo para encontrar a solução desse sistema.

c) Encontrando soluções: Além do sistema não ser linear, as medidas apresentam erros, então precisamos de uma função que otimize esse problema. Vamos usar uma função $J(\theta)$ de critério que depende de uma solução θ e nosso problema consiste em encontrar o valor de θ que minimiza essa função;

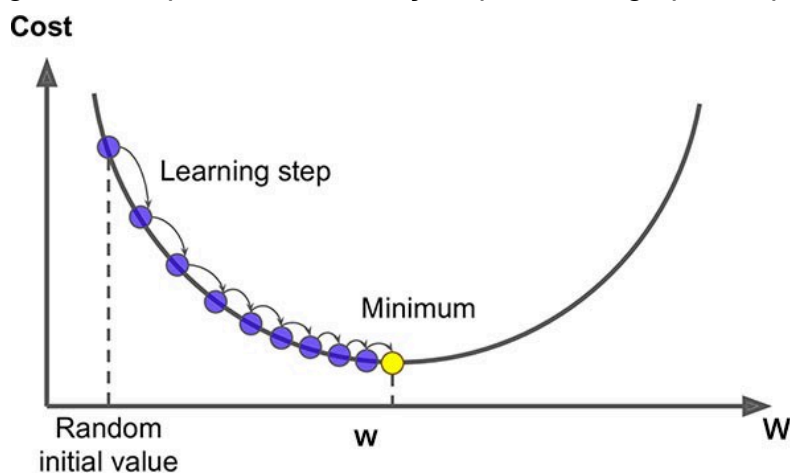
- $\theta = \operatorname{argmin} J(\theta)$

Um critério possível para nosso problema de localização do drone é uma função $J(r)$ que depende de um "chute" inicial de posição r e avalia a diferença entre a medida e o valor calculado usando o chute inicial.

- $J(r) = \frac{1}{2} \times \sum_{i=1}^N (\|r_i - r\| - \rho_i)^2$, com N sendo o conjunto de medidas

Precisamos avaliar quão preciso foi o "chute" observando a diferença $(r_i - r)$, que sendo pequena, será aproximadamente o módulo de ρ_i . Logo se $J(r)$ estiver próximo de zero, os métodos de otimização serão aplicados no sistema para que ele convirja para a posição real do drone.

d) Gradiente Descendente: O conceito de vetor gradiente de uma função multivariável indica a direção de máxima variação da função. Nesse sentido, o método de Gradiente Descendente (imagem abaixo) faz com que os parâmetros do modelo sejam ajustados iterativamente na direção oposta ao gradiente, buscando minimizar a função de custo e aproximar-se de um mínimo local. A dinâmica desse método atende exatamente ao problema que enfrentamos, pois o objetivo é minimizar o erro. Isso indica que o "chute" é uma $J(r)$ solução da função, com seu gradiente apontando na direção que converge para o ponto $J(r)$ desejado.



- $\nabla f = (\frac{\delta f}{\delta x}, \frac{\delta f}{\delta y}, \frac{\delta f}{\delta z})$

- $\Rightarrow \nabla f \simeq 0$
- $\frac{\delta J(r)}{\delta x} = (1 - \frac{\rho}{\|r_i - r\|})(x - x_i)$
- $\frac{\delta J(r)}{\delta y} = (1 - \frac{\rho}{\|r_i - r\|})(y - y_i)$
- $\frac{\delta J(r)}{\delta z} = (1 - \frac{\rho}{\|r_i - r\|})(z - z_i)$
- $R = (1 - \frac{\rho}{\|r_i - r\|})$
- $\nabla J(r) = R(r_i - r)$
- $\Rightarrow r = r_i - l \times r \cdot \nabla J(r)$, aplicação do método iterativo

Podemos implementar o método iterativo para que o valor se aproxime do ponto desejado. Envolve calcular , onde cada atualização dos parâmetros do gradiente é $r = r_i - l \times r \cdot \nabla J(r)$ feita utilizando um novo valor de "chute" ao longo de N iterações.

Referências

- https://www.ime.usp.br/~leo/intr_prog/img_zero_funcoes/img_newton_raphson_3.png
- [Earth Centred Inertial Frame – ADCS For Beginners](#)
- [Earth Centered Inertial \(ECI\) and LVLH coordinates | Download Scientific Diagram](#)
- [A Beginners Guide to Gradient Descent Algorithm for Data Scientists!](#)