

RocketSim Uniandes - AEISS, CS

Juan José Mejía Álvarez
Dirección División Teórica, AEISS

26 de enero de 2024

I. Preámbulo

La presente es una propuesta para la realización de un software capaz de integrar las ecuaciones de movimiento de un cohete que está encaminado a la órbita terrestre. Su objetivo principal es otorgar las métricas dinámicas que optimizan el proceso de ascenso y minimizan las pérdidas y el combustible requerido. Los objetivos secundarios son: otorgar curvas de tiempo de parámetros dinámicos en un formato adecuado para su implementación en un computador de vuelo; determinar los esfuerzos aerodinámicos a los que será sometido el vehículo a lo largo de su ascenso; implementar una interfaz de usuario que permita el fácil acceso y uso por parte de personas con pocos conocimientos en programación; e implementar un algoritmo de correcciones en tiempo real para el ajuste en medio de la misión de vuelo.

II. Planteamiento del problema

Se presenta acá una breve contextualización del problema a tratar y las razones por las cuales se requieren algoritmos robustos de optimización y modelación numérico-computacionales para su solución.

La parte puramente técnica y su razón no es de interés para CS, solo la expongo para dar el contexto y las ecuaciones que van a regir el software en cuestión. No se maten la cabeza con esta parte, solo con que la lean y entiendan la idea general está bien. Cualquier parámetro que no esté definido, considérenlo como dado y conocido.

II.I. Ascenso de un cohete

Dadas las fuertes limitaciones en términos de manejo de propelente que presenta un cohete, el problema de optimización de trayectorias de ascenso es de gran interés para el sector espacial. Para esto, consideramos que la cantidad a minimizar es la masa de propelente del cohete (esto nos permite maximizar la carga máxima a llevar al espacio, y por lo tanto, el dinero recibido por la misión de vuelo), sometido a las ecuaciones diferenciales de movimiento (2), a una fuerza de empuje determinada por la naturaleza de los motores y a unas condiciones iniciales y finales sobre el estado dinámico del cohete.

II.I.1. Etapas de ascenso

El ascenso de un vehículo auto-propulsado al espacio exterior tiene numerosos aspectos fijos impuestos por la misión de vuelo particular: un punto de inicio fijo, una órbita final predeterminada y una arquitectura particular del cohete. En adición a esto, tenemos un ascenso que consiste en tres etapas principales:

1. **Ascenso:** Comienza con el encendido de los motores de la primera etapa, con un ángulo de 90° con respecto al horizonte, antiparalelo a la dirección de la gravedad. Esto implica que el cohete se encuentra apuntando directamente al cenit local. En esta fase, el vehículo se encuentra acelerando a través de la atmósfera planetaria hasta un punto de transición, momento en el cual ocurre el apagado de los motores principales (MECO - Main engine cut-off).

2. **Transferencia:** Una vez se llega a MECO, el vehículo procede su trayectoria a lo largo de una órbita de transferencia sin ningún tipo de empuje. En esta fase, las fuerzas aerodinámicas se vuelven despreciables por virtud de la nula densidad de gases atmosféricos.
3. **Ajuste orbital:** Una vez el vehículo ha llegado al apoápside de su órbita de transferencia, se inicia una segunda fase de empuje con la que se ajustan los parámetros orbitales hasta obtener la órbita objetivo.

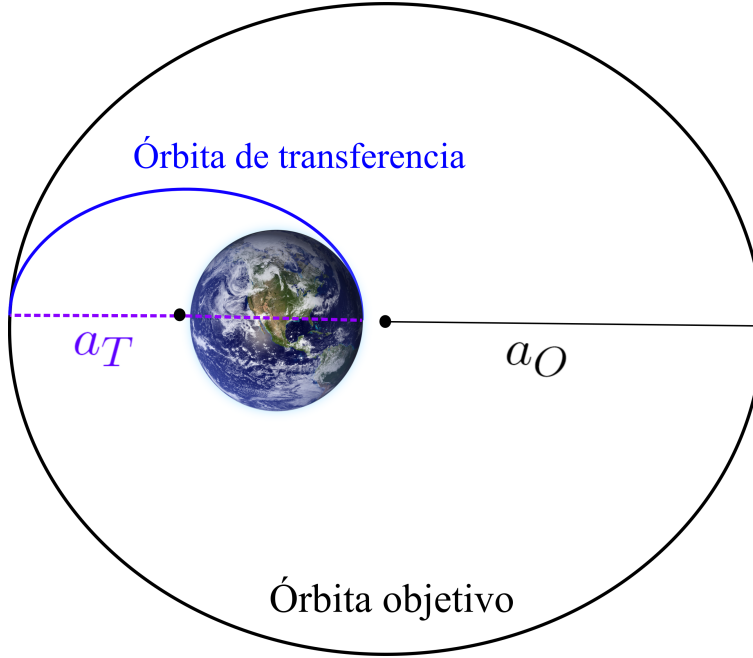


Figura 1: La órbita objetivo es determinada con antelación por la dirección de misión de vuelo, mientras que la órbita de transferencia es determinada por la órbita objetivo. Esto a su vez determina los parámetros de ascenso. a_T y a_O representan los semi-ejes mayores de ambas órbitas.

El proceso físico se observa en la figura 1. Vemos que el proceso de ascenso se traduce en la transición de una órbita Kepleriana de transferencia a otra órbita Kepleriana objetivo. Esto de manera automática impone que el periápside de la primera se encuentra en la plataforma de lanzamiento, mientras que su apoápside se encuentra tocando la órbita objetivo. Esta es, en efecto, una transferencia de Hohmann. Usando las ecuaciones de la mecánica orbital obtenemos el semi-eje mayor y la excentricidad de la transferencia:

$$a_T = \frac{r_a + r_p}{2} \qquad e_T = \frac{r_a - r_p}{r_a + r_p} \qquad (1)$$

(Recordar acá que una órbita Kepleriana es solo una sección cónica).

El proceso de ascenso, que es el que nos interesa a nosotros, se ve tal y como se observa en la figura 2. En esta imagen es fácil ilustrar cuáles son los parámetros de interés para nosotros: **Nos interesa modelar la curva que hace el cohete a medida que sale de la atmósfera, de tal forma que esta sea óptima en su gasto de combustible.**

II.II. Ecuación de movimiento

El sistema de ecuaciones diferenciales ordinarias con respecto al tiempo que obedece la dinámica de un cohete camino a la órbita terrestre están dados por medio de un sencillo análisis de fuerzas:



Figura 2: Ascenso orbital de un cohete Delta-II desde la base área de Vandenberg, en California. Se observa el proceso de ascenso, que comienza con el encendido de los motores y termina con MECO.

$$m(t)\dot{\mathbf{v}}(t) = \mathbf{F}_*(t) + m\mathbf{g}(\mathbf{r}(t)) + \mathbf{F}_D(\mathbf{v}, \mathbf{r}, t) + \mathbf{F}_L(\mathbf{v}, \mathbf{r}, t) \quad (2)$$

Donde \mathbf{v} es el vector velocidad del vehículo, \mathbf{r} es el vector posición, m es su masa (la cual varía con respecto al tiempo por virtud de ser un cohete), \mathbf{F}_* es la fuerza de empuje proporcionada por los motores, \mathbf{F}_D es la fuerza de arrastre y \mathbf{F}_L es la fuerza de sustentación aerodinámica. El punto encima de las cantidades indica derivada con respecto al tiempo: $\frac{d}{dt}(\cdot) := \dot{(\cdot)}$.

Notar que la masa del cohete depende de como sea el flujo másico \dot{m}_p de propelente a través de los motores. En general podemos escribir:

$$m(t) = m(t_0) - \int_{t_0}^t \dot{m}_p(t') dt' \quad (3)$$

La elección de un buen sistema de referencia con el cual escribir cada uno de los vectores mostrados en la ecuación (2) es esencial. En particular, nos interesa el sistema de referencia orbital NTW (normal-tangente-normal), cuya dirección más importante (llamada dirección T pero correspondiente al eje y) está dirigida a lo largo del vector unitario de velocidad (dirección instantánea de movimiento) y su tercera dirección (llamada W pero correspondiente al eje z) está dirigida a lo largo del momento angular orbital, definido por el vector $\mathbf{h} := \mathbf{r} \times \mathbf{v}$.

Este sistema de referencia es de especial importancia ya que la fuerza de arrastre es siempre contraria a la dirección de movimiento y la fuerza de sustentación es siempre perpendicular a esta última. En la figura 3 se puede observar la configuración de cada uno de los vectores de fuerza, así como el sistema de

referencia en color magenta.

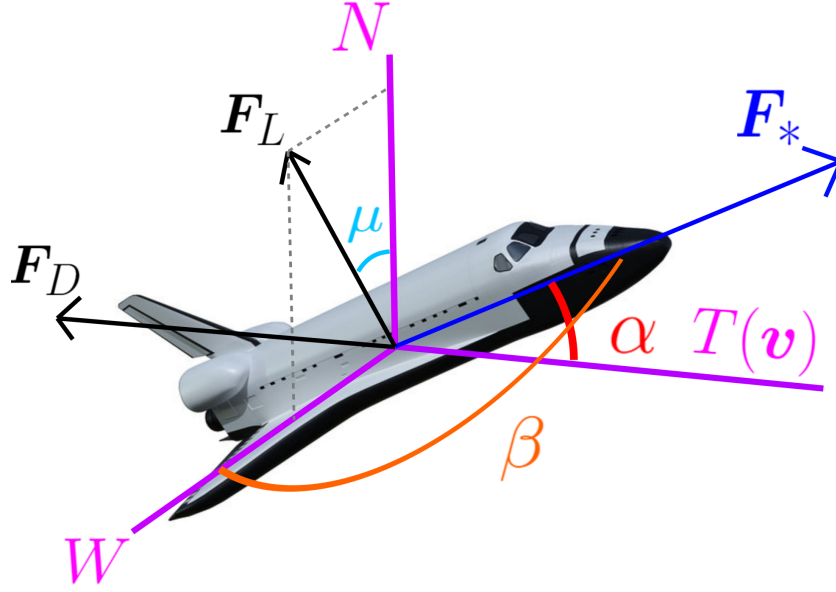


Figura 3: Configuración de fuerzas para una nave desplazándose en la atmósfera terrestre.

Un análisis cuidadoso de componentes en este sistema de referencia, así como del vector de aceleración $\dot{\mathbf{v}}$, otorga una primera forma vectorial explícita de las ecuaciones diferenciales a resolver:

$$\frac{m}{r} \begin{bmatrix} r\dot{\gamma}v - v^2 \cos \gamma \\ r\dot{v}_T \\ r\dot{v}_W \end{bmatrix} = F_* \begin{bmatrix} \sin \alpha \sin \beta \\ \cos \alpha \sin \beta \\ \cos \beta \end{bmatrix} + mg \begin{bmatrix} -\cos \gamma \\ -\sin \gamma \\ 0 \end{bmatrix} + F_L \begin{bmatrix} \cos \mu \\ 0 \\ \sin \mu \end{bmatrix} + F_D \begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix} \quad (4)$$

Donde γ es conocido como el ángulo de trayectoria de vuelo, y corresponde al ángulo que hace el vector unitario de velocidad del centro de masa de la nave con el horizonte local (la superficie equipotencial gravitacional). Este ángulo puede ser observado en la figura 4. De igual forma, \dot{v}_T y \dot{v}_W representan las aceleraciones del cuerpo a lo largo de las direcciones tangente (T) y perpendicular al plano orbital (W). Notar también que los ángulos α y β son tratados como localmente esféricos, esto es, representan un ángulo azimutal y polar respectivamente.

II.III. Ecuaciones computacionales

En el caso más general, las ecuaciones (4) deben ser integradas de manera numérica. Solo en el caso simplificado $\mu = 0$ y $\beta = \pi/2$, se pueden obtener soluciones analíticas. Para cualquier caso, siempre resulta conveniente tratar con cantidades adimensionales. Siguiendo la notación expuesta en [2], definimos:

$$v_\Delta := \sqrt{Rg_0} \quad (5)$$

Donde R y g_0 son el radio y la aceleración gravitacional promedio del planeta en cuestión respectivamente. En adición, retomamos la altura $h := r - R$ con respecto a la superficie del planeta y la distancia *globalmente* perpendicular x . Ambas cantidades obedecen las ecuaciones:

$$\dot{x} = v \cos \gamma \quad \dot{h} = v \sin \gamma \quad (6)$$

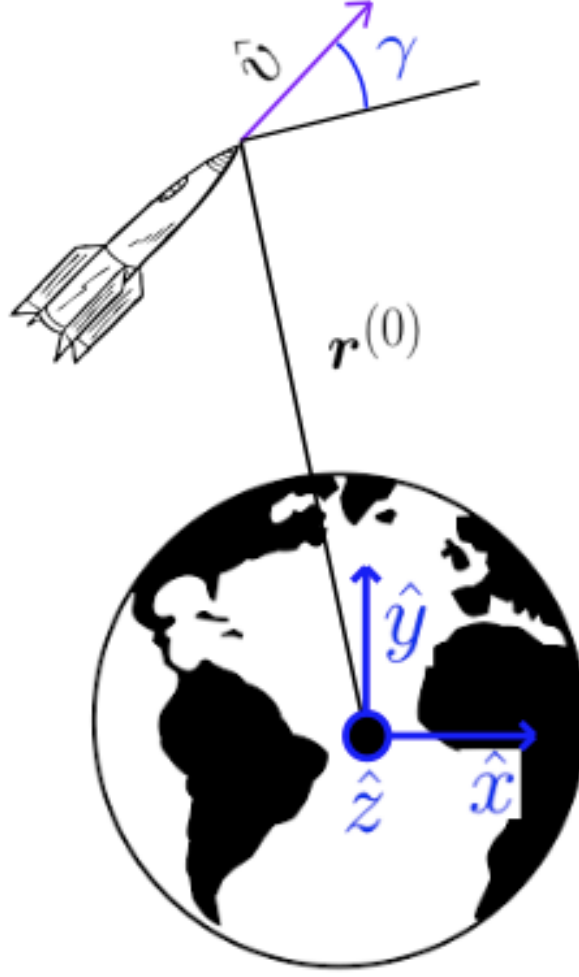


Figura 4: El horizonte local se observa como la línea negra perpendicular a la dirección radial.

Cantidades que adquieren un significado inmediato al observar la figura 4.

Veamos ahora la definición de las fuerza de arrastre: $F_D(\mathbf{r}, \mathbf{v}, t) := \frac{1}{2} \rho(\mathbf{r}) v^2(t) A_{\perp} C_D$, donde $\rho(r)$ es la densidad de la atmósfera, la cual decae exponencialmente con la altura, A_{\perp} es el área de contacto de la nave con el flujo de viento y C_D es el coeficiente de arrastre (estos últimos son parámetros dados, y no tienen que preocuparse por ellos, solo tómenlos como floats de input). La fuerza de sustentación tiene una forma similar, pero solo cambia el coeficiente $C_D \rightarrow C_L$, el cual es el coeficiente de sustentación: $F_L(\mathbf{r}, \mathbf{v}, t) = \frac{1}{2} \rho(\mathbf{r}) v^2(t) A_{\perp} C_L$. Usando la ecuación (2) y aplicando la siguiente prescripción adimensional:

$$\sigma := \frac{v}{v_{\Delta}} \quad \eta := \frac{h}{H} \quad \xi := \frac{x}{H} \quad \Phi := \frac{F_*}{mg_0} \quad \tau := \frac{g_0 t}{v_{\Delta}} \quad (\dots)' := \frac{d(\dots)}{d\tau} \quad (7)$$

Se pueden obtener las ecuaciones computacionales de ascenso orbital, las cuales están optimizadas para su integración computacional:

$$\begin{aligned}
\sigma\gamma' - \frac{\sigma^2 R \cos \gamma}{\eta H + R} &= \Phi \sin \alpha \sin \beta - \left(\frac{R}{\eta H + R} \right)^2 \cos \gamma + \frac{Re^{-\eta} \sigma^2 \kappa_L \cos \mu}{H} \\
\sigma'_T &= \Phi \cos \alpha \sin \beta - \left(\frac{R}{\eta H + R} \right)^2 \sin \gamma - \frac{Re^{-\eta} \sigma^2 \kappa_D}{H} \\
\sigma'_W &= \Phi \cos \beta + \frac{Re^{-\eta} \sigma^2 \kappa_L \sin \mu}{H}
\end{aligned} \tag{8}$$

Donde $\kappa_D := \frac{C_D A_\perp H \rho_0}{2m}$ y $\kappa_L := \frac{C_L A_\perp H \rho_0}{2m} = \kappa_D \frac{C_L}{C_D}$ son los coeficientes de arrastre y sustentación reducidos respectivamente. Este conjunto de ecuaciones diferenciales es complementado por la versión adimensional de las ecuaciones (6):

$$\eta' = \frac{\mu R \sin \gamma}{H} \qquad \xi' = \frac{\mu R \cos \gamma}{H} \tag{9}$$

Las cinco ecuaciones diferenciales ordinarias acopladas mostradas son las ecuaciones computacionales de vuelo atmosférico. Solo en el caso en el que se hace la aproximación $\eta H + R \approx R$ y se considera $\beta = \pi/2$ y $\mu = 0$, se obtienen las ecuaciones normalizadas expuestas en [2]. En este sentido, (8) y (9) representan la descripción dinámica más general para una nave espacial desplazándose en un cuerpo planetario con atmósfera sin componentes de viento cruzado. Esta prescripción adimensional puede ser integrada de manera directa usando algún método numérico, como Runge-Kutta.

II.IV. Problema de optimización

Queda ahora la pregunta de como ajustar los parámetros de misión de vuelo α, β y γ para hacer la transición entre las dos primeras fases de ascenso lo más continua y óptima posible. También, resulta importante determinar la posición y el tiempo de MECO que requiere la menor cantidad de propelente. Por último, pero no menos importante, se busca minimizar la cantidad de propelente necesaria para completar la transición. Todo esto constituye un problema de control óptimo, que revisaremos de manera superficial acá tomando inspiración de aquello discutido por [1, 2].

Para empezar, tenemos en cuenta que las componentes de aceleración a lo largo de la dirección W son despreciables en comparación con la dinámica experimentada a lo largo de la dirección T . Esto nos permite ignorar la tercera ecuación de (4). Tomando entonces $\beta = \pi/2$ y $\mu = 0$, procedemos a integrar la segunda componente de (4) desde un tiempo inicial de despegue t_0 hasta un tiempo final t_f de ajuste orbital:

$$\begin{aligned}
\int_{t_0}^{t_f} \dot{v} dt &= \int_{t_0}^{t_f} \left(\frac{F_*}{m} \cos \alpha - g \sin \gamma - \frac{F_D}{m} \right) dt \\
v_f - v_0 &= \underbrace{v_* \ln \left(\frac{m_0}{m_f} \right)}_{\text{Cohete libre}} - \underbrace{\int_{t_0}^{t_f} \left(\frac{F_*(1 - \cos \alpha)}{m} + \frac{g_0 R^2 \sin \gamma}{r^2} + \frac{\kappa_D v^2 e^{-(r-R)/H}}{H} \right) dt}_{\text{Funcional de pérdidas}}
\end{aligned} \tag{10}$$

Donde en la segunda línea se usa la identidad $\int_{t_0}^{t_f} \frac{\dot{m} v_*}{m} dt = v_* \int_{m_0}^{m_f} \frac{dm}{m} = -\Delta v_0$ y la definición de coeficiente de arrastre reducido. Es claro del integrando del funcional de pérdidas que existen tres contribuciones que reducen la ganancia en velocidad:

- El término $F_*(1 - \cos \alpha)/m$ corresponde a pérdidas por direccionamiento. Esto representa trabajo realizado por los propulsores a lo largo de componentes perpendiculares a la dirección de movimiento. Esto produce una aceleración que no contribuye a incrementar la velocidad final.
- El término $g \sin \alpha$ corresponde a las pérdidas por aceleración gravitacional.
- El término aerodinámico es relevante únicamente para la primera fase de ascenso, pero resulta ser la razón por la cual no se puede empezar con propulsión a un ángulo de trayectoria de vuelo nulo (eliminando por completo las pérdidas gravitacionales): Las estructuras de un cohete acelerando a través de una atmósfera están sujetas a momentos de torsión directamente proporcionales al ángulo de ataque y a la presión dinámica. De esta manera, a medida que la nave espacial asciende y los términos aerodinámicos se vuelven despreciables, se puede reducir el ángulo γ .

El verdadero problema de optimización consiste entonces en minimizar el funcional de pérdidas. Esto se puede lograr minimizando del intervalo de tiempo $\Delta t = t_f - t_0$ y considerando funciones óptimas $\alpha = \alpha(t)$ y $\gamma = \gamma(t)$ que reducen el valor de la integral. Las razones por las cuales esto es altamente no-trivial son, entre otras: la dependencia del término aerodinámico en la distancia radial r , el hecho que la evolución temporal de esta última es directamente proporcional a $\sin \gamma$ (ver (6)), las limitaciones estructurales del cohete de soportar esfuerzos de flexión, la dependencia explícita que la masa total tiene en el tiempo (lo que hace por extensión que los coeficientes aerodinámicos reducidos dependan también en el tiempo), y las condiciones iniciales y finales impuestas por la misión de vuelo (ver (12) y (13)).

En adición, podemos obtener el comportamiento numérico del ángulo de trayectoria de vuelo despejando $\dot{\gamma}$ de la segunda ecuación de (4) e integrando sobre el mismo intervalo temporal considerado anteriormente para obtener:

$$\begin{aligned} \int_{t_0}^{t_f} \dot{\gamma} dt &= \int_{t_0}^{t_f} \left(\frac{F_* \sin \alpha}{mv} - \frac{g \cos \gamma}{v} + \frac{F_L}{mv} + \frac{v \cos \gamma}{r} \right) dt \\ \gamma_f - \gamma_0 &= \int_{t_0}^{t_f} \left(\frac{F_* \sin \alpha}{mv} + \frac{\kappa_L v e^{-(r-R)/H}}{H} + \frac{v \cos \gamma}{r} - \frac{g_0 R^2 \cos \gamma}{vr^2} \right) dt \end{aligned} \quad (11)$$

En este caso, observamos distintas contribuciones al cambio en el ángulo de trayectoria de vuelo:

- El término $F_* \sin \alpha / (mv)$ corresponde al direccionamiento dado por el sistema de propulsión.
- El término aerodinámico de sustentación también es solo relevante para la primera fase de ascenso. Se puede notar que en este caso particular no hay pérdidas en la ganancia de velocidad y la fuerza de sustentación puede ser usada para cambiar la dirección de ascenso sutilmente en las capas más altas de la heterósfera.
- $v \cos \gamma / r$ representa la contribución por fuerza centrífuga.
- $g \cos \gamma / v$ es el cambio en el ángulo de trayectoria de vuelo por virtud de la fuerza gravitacional. El hecho que sea el único término con signo negativo tiene consecuencias importantes, en particular, que la gravedad puede ser usada para generar un direccionamiento **sin gastar combustible**.

II.IV.1. Restricciones

Las condiciones físicas reales del problema de ascenso de una nave espacial imponen restricciones sobre las condiciones iniciales: el cohete debe iniciar su recorrido apuntando directamente al cénit local, con velocidad cero y distancia radial aproximadamente igual al radio del cuerpo planetario desde el cual se realiza el ascenso. En ecuaciones, esto se lee de la forma:

$$v(t_0) = v_0 = 0 \quad \gamma(t_0) = \gamma_0 = \pi/2 \quad r(t_0) = r_0 \approx R \quad (12)$$

Entretanto, las condiciones finales son dictadas por el hecho que la nave espacial debe llegar al punto de contacto con la órbita de transferencia en MECO. De forma matemática, podemos hacer uso de las ecuaciones de la mecánica orbital para escribir las condiciones finales:

$$\begin{aligned} v(t_f) = v_f &= \sqrt{\frac{\mu(1 + 2e_T \cos f_T + e_T^2)}{a_T(1 - e_T^2)}} \\ \cos \gamma(t_f) = \cos \gamma_f &= \frac{1 + e_T \cos f_T}{\sqrt{1 + 2e_T \cos f_T + e_T^2}} \\ r(t_f) = r_f &= \frac{a_T(1 - e_T^2)}{1 + e_T \cos f_T} \end{aligned} \quad (13)$$

Donde f_T es la anomalía verdadera correspondiente a MECO (**Entiendan la anomalía verdadera como un ángulo de referencia orbital, como un float**) y $\mu := GM$ es el parámetro gravitacional estándar del sistema, donde G es la constante de gravitación estándar. Los parámetros a_T y e_T están dados por (1).

En adición a esto, el comportamiento de la fuerza de empuje y la evolución de la masa total de la nave con respecto al tiempo están predeterminadas por las características del propulsor. En particular, se tiene que (3) fija $m = m(t)$. **De esta forma, observamos del funcional de pérdidas en (10) que solo $\alpha(t)$ y f_T están completamente sometidos al control óptimo.**

De esta manera tenemos que este problema, en gran medida restringido por la fijación de los ángulos β y μ y el no considerar vientos cruzados, se puede plantear alternativamente de la siguiente forma, tal y como plantea [2]: **Se busca la minimización de la demanda de combustible**, representada por un funcional de costo $J = m_p(t_f)$. Esto se logra con una selección adecuada de f_T y una función $\alpha(t)$ óptima, todo sometido a las ecuaciones (4), con condiciones iniciales (12) y finales (13) y a un sistema de propulsión pre-determinado.

III. Forma general del algoritmo esperado

El software esperado debería consistir de tres fases. La primera consiste en la preparación para la pre-ignición, la cual consiste en determinar los parámetros óptimos de misión que se programan en un computador de vuelo como *curvas de control*. La segunda consiste en la modulación del empuje para que el cohete sobreviva a Max-Q (máximo esfuerzo aerodinámico). La tercera y última consiste en la corrección iterativa de control de vuelo con base a las curvas de control determinadas en la primera fase.

Algoritmo 1 Primera fase: Optimización pre-ignición

- 1: INPUT: Parámetros de misión de vuelo orbital (e_T , a_T) - floats
 - 2: INPUT: Elemento de clase COHETE: Número de etapas (int), ISP - impulso específico - para cada etapa (lista de floats), Coeficientes aerodinámicos reducidos κ_D y κ_L para cada etapa (listas de floats), Coeficientes máscos para cada etapa (listas de floats).
 - 3: **Caja negra: magia de CS, optimización funcional y modelación de trayectoria.**
 - 4: OUTPUT: Curvas de tiempo $\alpha(t)$ y $\gamma(t)$, intervalo de tiempo $\Delta t := t_f - t_0$ y anomalía verdadera f_T que minimizan el funcional de perdidas y la masa de propelente.
-

Algoritmo 2 Segunda fase: Max-Q y esfuerzos aerodinámicos

- 1: INPUT: Output de primera fase.
 - 2: INPUT: Curvas de consumo de propelente $\dot{m}_p(t)$ a lo largo del intervalo de tiempo determinado en el algoritmo 1.
 - 3: **Caja negra: Magia de CS, modelación de los esfuerzos aerodinámicos con base a ecuaciones pre-establecidas.**
 - 4: OUTPUT: Curvas de tiempo de esfuerzos aerodinámicos.
-

Algoritmo 3 Tercera fase: Corrección en tiempo real

INPUT: Datos en tiempo real de acelerómetro y giroscopio.

Magia negra de CS: Aplicación de leyes de control pre-establecidas para retornar los parámetros dinámicos del cohete a las curvas ideales del algoritmo 1.

OUTPUT: Señal de control a sistemas de actuación dinámica.

Se muestra a continuación un ejemplo de clase COHETE.

```
1  class Cohete: #Se define la clase cohete, la cual se construye unicamente con los
    atributos pedidos al usuario con anterioridad.
2  def __init__(self,n,ISP,EPS,dv,F_star,M_0,MU,GAMMA):
3      self.etapas = n #N mero de etapas
4      self.isp = ISP #Valores de impulso especifico
5      self.eps = EPS #Valores de raz n de masa estructural
6      self.dv = dv #Valor de delta v requerido
7      self.fuerza_prop = F_star #Fuerzas de empuje
8      self.M_0 = M_0
9      self.MU = MU
10     self.GAMMA = GAMMA
11
12     def delta_v(self,x): #Se define la funcion de optimizacion a la cual se le quieren
    encontrar las raices (los valores de optimizacion)
13         delta = 0
14         for i in range(n):
15             delta = delta + self.isp[i]*np.log(np.abs((self.isp[i]-x)/(self.eps[i]*self
    .isp[i])))
16             i += 1
17         return dv-delta
18
19     def integral_gravedad(self):
20         t_combustion = []
21         sin_gamma = []
22         valor_ints = []
23         for i in range(self.etapas):
24             delta_t = self.M_0[i]*(1-self.MU[i])*g_0*self.isp[i]/(self.fuerza_prop[i]*1
    e+3)
25             t_combustion.append(delta_t)
26             s_gamma = np.sin(self.GAMMA[i])
27             sin_gamma.append(s_gamma)
28             i+=1
29         for i in range(self.etapas):
30             val_ints = sin_gamma[i]*t_combustion[i]*g_0
31             valor_ints.append(val_ints)
32         return valor_ints
33
34     def delta_v_gravedad(self,x):
35         delta = 0
36         for i in range(n):
```

```

37         delta = delta + self.isp[i]*np.log(np.abs((self.isp[i]-x)/(self.eps[i]*self
    .isp[i]))) - self.integral_gravedad()[i]
38         i += 1
39         return dv-delta
40
41     def d_delta_v(self,x): #Se obtiene la derivada numerica de la funcion anterior
42         h = 1e-6
43         return ((self.delta_v(x+h)-self.delta_v(x))/h)
44
45     def d_delta_v_gravedad(self,x):
46         h = 1e-6
47         return (self.delta_v_gravedad(x+h)-self.delta_v_gravedad(x))/h
48
49     def Newton_Raphson_sin_gravedad(self,x_0): #Se implementa un metodo de Newton-
    Raphson para la determinacion de las raices.
50         n_i = 100
51         datos=[]
52         iteraciones = []
53         for i in range(n_i):
54             x = x_0 - self.delta_v(x_0)/self.d_delta_v(x_0)
55             x_0 = x
56             plt.plot(iteraciones,datos)
57             plt.xlabel("Iteraci n")
58             plt.ylabel("Convergencia")
59             return x
60
61     def Newton_Raphson_con_gravedad(self,x_0):
62         n_i = 100
63         datos=[]
64         iteraciones = []
65         for i in range(n_i):
66             x = x_0 - self.delta_v_gravedad(x_0)/self.d_delta_v_gravedad(x_0)
67             x_0 = x
68             iteraciones.append(i)
69             datos.append(x)
70             plt.plot(iteraciones,datos)
71             plt.xlabel("Iteraci n")
72             plt.ylabel("Convergencia")
73             return x
74
75     def lambda_op(self,x_0):
76         varphi_g_not = self.Newton_Raphson_sin_gravedad(x_0)
77         varphi_g_aye = self.Newton_Raphson_con_gravedad(x_0)
78
79         lambda_op_g_not = []
80         lambda_op_g_aye = []
81
82         for i in range(self.etapas):
83             lambda_g_not = varphi_g_not*self.eps[i]/(self.isp[i]*g_0*(1-self.eps[i]) -
    varphi_g_not)
84             lambda_op_g_not.append(lambda_g_not)
85             lambda_g_aye = varphi_g_aye*self.eps[i]/(self.isp[i]*g_0*(1-self.eps[i]) -
    varphi_g_aye)
86             lambda_op_g_aye.append(lambda_g_aye)
87
88         def_g_not = 1
89         def_g_aye = 1
90
91         for i in range(self.etapas):
92             def_g_not = def_g_not * lambda_op_g_not[i]/(1+lambda_op_g_not[i])
93             def_g_aye = def_g_aye * lambda_op_g_aye[i]/(1+lambda_op_g_aye[i])

```

```
94  
95     return def_g_not, def_g_aye
```

Script 1: Ejemplo de clase COHETE.

Referencias

- [1] Anthony J. Calise and Martin S.K. Leung. Optimal guidance law development for an advanced launch system, 1995. NASA - Langley Research Center.
- [2] Ulrich Walter. *Astronautics: The Physics of Space Flight*, volume 3. Springer, Gewerbestrasse 11, 6330 Cham, Suiza, 2019.