

## **Laboratorio 2:**

Estimación y control de actitud en sistemas  
aeroespaciales

Germán Quijada

7 de julio de 2023

# 1 Definición del problema

Se solicita caracterizar, simular y controlar el siguiente sistema dinámico

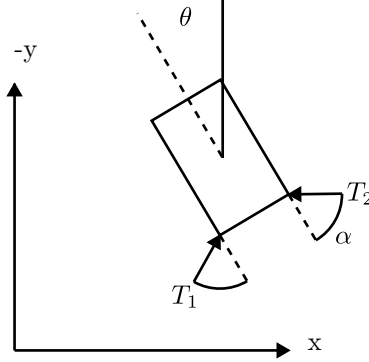


Figura 1: Representación de sistema a trabajar.

El sistema se trata de un pequeño satélite capaz de traslación y rotación en el plano por lo que su movimiento se puede describir con 3 grados de libertad,  $x$  e  $y$  para la traslación y  $\theta$  para la rotación. Además cuenta con 2 propulsores a un ángulo  $\alpha$  de los costados del satélite, el empuje que realizan son la acción de control disponible. Finalmente las variables  $w$  y  $h$  describen su ancho y alto respectivamente.

Se busca encontrar las ecuaciones del movimiento del sistema para poder realizar simulaciones con algún método de integración, luego lograr establecer un controlador automático y finalmente poner el controlador a prueba a ver si funciona si las lecturas del estado del satélite fuesen de sensores con ruido.

## 2 Ecuaciones del movimiento

Primero se determinan las fuerzas actuando en el eje horizontal, vertical y los momentos.

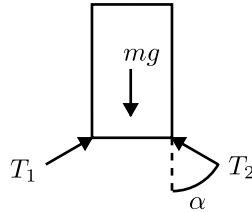


Figura 2: Diagrama de cuerpo libre.

Las fuerzas y momentos que actúan en el *body frame* debido a los propulsores son las siguientes

$$F_{bf} = \begin{bmatrix} (T_1 - T_2) \sin(\alpha) \\ (-T_1 - T_2) \cos(\alpha) \end{bmatrix}$$

$$T_{bf} = (T_1 - T_2) \left( -\frac{h \sin(\alpha)}{2} + \frac{w \cos(\alpha)}{2} \right)$$

Para obtener las ecuaciones en el sistema inercial, se multiplican las fuerzas por la matriz de rotación, el momento no es necesario rotarlo

$$R = \begin{bmatrix} \cos(\theta(t)) & -\sin(\theta(t)) \\ \sin(\theta(t)) & \cos(\theta(t)) \end{bmatrix}$$

También se aprovecha de agregar la fuerza de gravedad, que actúa en relación con el sistema inercial y no con el body frame

$$F = R \cdot F_{bf} + \mathbf{g}$$

$$F = \begin{bmatrix} -(-T_1 - T_2) \sin(\theta(t)) \cos(\alpha) + (T_1 - T_2) \sin(\alpha) \cos(\theta(t)) \\ gm + (-T_1 - T_2) \cos(\alpha) \cos(\theta(t)) + (T_1 - T_2) \sin(\alpha) \sin(\theta(t)) \end{bmatrix}$$

Luego para obtener las ecuaciones del movimiento, se aplica la segunda ley de Newton para las fuerzas y su equivalente para el momento

$$F = m \frac{d^2}{dt^2} \mathbf{x}(t)$$

$$T = I \frac{d^2}{dt^2} \theta(t)$$

Donde la inercia es la de un rectángulo *fino*

$$I = \frac{m(h^2 + w^2)}{12}$$

De esta forma las ecuaciones del movimiento son

$$\frac{d^2}{dt^2} x(t) = \frac{-(-T_1 - T_2) \sin(\theta(t)) \cos(\alpha) + (T_1 - T_2) \sin(\alpha) \cos(\theta(t))}{m}$$

$$\frac{d^2}{dt^2} y(t) = \frac{gm + (-T_1 - T_2) \cos(\alpha) \cos(\theta(t)) + (T_1 - T_2) \sin(\alpha) \sin(\theta(t))}{m}$$

$$\frac{d^2}{dt^2} \theta(t) = \frac{12.0 (T_1 - T_2) \left( -\frac{h \sin(\alpha)}{2} + \frac{w \cos(\alpha)}{2} \right)}{m(h^2 + w^2)}$$

Para comprobar que el sistema de ecuaciones es realmente así, se creó una simulación interactiva que permitiese verificar el movimiento. Esto fue muy útil,

ya que en primeras iteraciones del sistema de ecuaciones este no era correcto y comprobarlo de otra forma hubiera sido muy engorroso o simplemente no se hubiera notado.

<https://qgerman2.github.io/ecasa2/>

### 3 Vector estado

El vector estado será el siguiente

$$\mathbf{x}(t) = \begin{bmatrix} x(t) \\ y(t) \\ \theta(t) \\ \frac{d}{dt}x(t) \\ \frac{d}{dt}y(t) \\ \frac{d}{dt}\theta(t) \end{bmatrix}$$

Y las expresiones a integrar en el tiempo son

$$\frac{d}{dt}\mathbf{x}(t) = \begin{bmatrix} \frac{d}{dt}x(t) \\ \frac{d}{dt}y(t) \\ \frac{d}{dt}\theta(t) \\ \frac{T_1 \sin(\alpha) \cos(\theta(t))}{T_1 \sin(\alpha) \sin(\theta(t))} + \frac{T_1 \sin(\theta(t)) \cos(\alpha)}{T_1 \cos(\alpha) \cos(\theta(t))} - \frac{T_2 \sin(\alpha) \cos(\theta(t))}{T_2 \sin(\alpha) \sin(\theta(t))} + \frac{T_2 \sin(\theta(t)) \cos(\alpha)}{T_2 \cos(\alpha) \cos(\theta(t))} + g \\ - \frac{6.0T_1 h \sin(\alpha)}{h^2 m + mw^2} + \frac{6.0T_1 w \cos(\alpha)}{h^2 m + mw^2} + \frac{6.0T_2 h \sin(\alpha)}{h^2 m + mw^2} - \frac{6.0T_2 w \cos(\alpha)}{h^2 m + mw^2} \end{bmatrix}$$

Al tener grados de libertad dentro de funciones trigonométricas y multiplicación de acciones de control por grados de libertad, se considera que el sistema no es lineal.

## 4 “Diagrama de bloques”

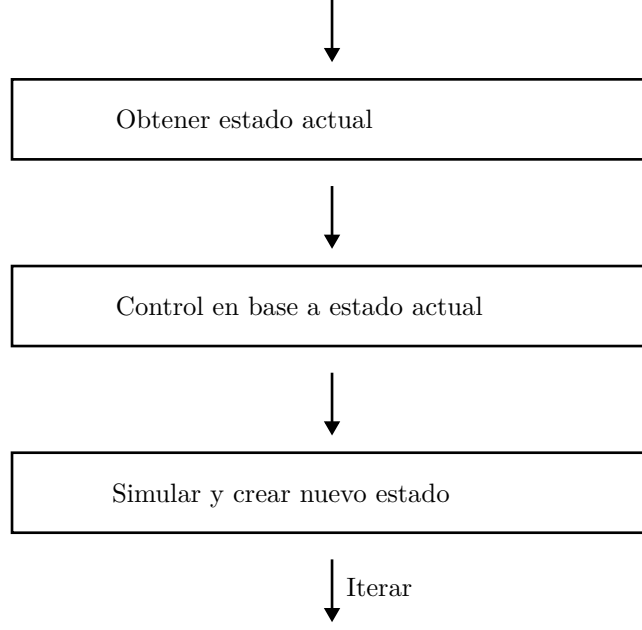


Figura 3: Intento de diagrama de bloques.

## 5 Linealización

Para posteriores procedimientos, es necesario linealizar el sistema de ecuaciones.

Se linealiza el sistema asumiendo una condición en la que está flotando quieto por lo que los propulsores están ejerciendo una aceleración equivalente a la aceleración de gravedad, además se asume que la rotación  $\theta$  es muy cercana a 0.

$$\begin{aligned}\frac{d^2}{dt^2}x(t) &= g\theta(t) + \frac{(T_1 - T_2) \sin(\alpha)}{m} \\ \frac{d^2}{dt^2}y(t) &= \frac{(-T_1 - T_2) \cos(\alpha)}{m} \\ \frac{d^2}{dt^2}\theta(t) &= \frac{12.0 (T_1 - T_2) \left( -\frac{h \sin(\alpha)}{2} + \frac{w \cos(\alpha)}{2} \right)}{m(h^2 + w^2)}\end{aligned}$$

Las ecuaciones son muy parecidas a las ecuaciones antes de rotarlas al sistema inercial, cabe destacar la inclusión del término con la aceleración de gravedad en la aceleración horizontal.

Al asumirse que los propulsores están ejerciendo empuje equivalente a la gravedad, esta se puede eliminar de la aceleración vertical, sin embargo para el movimiento horizontal esto tiene el efecto de que si el satélite está inclinado, existe un componente horizontal de los propulsores proporcional a la gravedad, por lo que se agrega a la ecuación.

## 6 Espacio estado

Para lograr el control de múltiples variables por medio de la acción de control, se opta por el método de posicionamiento de polos para un sistema representado por matrices espacio estado. Los detalles se extrajeron de <https://ctms.engin.umich.edu/CTMS/index.php?example=Introduction&section=SystemModeling>

$$\frac{d}{dt}\mathbf{x}(t) = A\mathbf{x}(t) + B\mathbf{u}(t)$$

Donde el vector  $\mathbf{u}$  es

$$\mathbf{u}(t) = \begin{bmatrix} T_1 \\ T_2 \end{bmatrix}$$

El sistema a convertir es el linealizado

$$\frac{d}{dt}\mathbf{x}(t) = \begin{bmatrix} \frac{d}{dt}x(t) \\ \frac{d}{dt}y(t) \\ \frac{d}{dt}\theta(t) \\ \frac{T_1 \sin(\alpha)}{m} - \frac{T_2 \sin(\alpha)}{m} + g\theta(t) \\ -\frac{T_1 \cos(\alpha)}{m} - \frac{T_2 \cos(\alpha)}{m} \\ -\frac{6.0T_1 h \sin(\alpha)}{h^2 m + mw^2} + \frac{6.0T_1 w \cos(\alpha)}{h^2 m + mw^2} + \frac{6.0T_2 h \sin(\alpha)}{h^2 m + mw^2} - \frac{6.0T_2 w \cos(\alpha)}{h^2 m + mw^2} \end{bmatrix}$$

y las matrices correspondientes son

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & g & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \frac{\sin(\alpha)}{m} & -\frac{\sin(\alpha)}{m} \\ -\frac{\cos(\alpha)}{m} & -\frac{\cos(\alpha)}{m} \\ -\frac{6.0h \sin(\alpha)}{h^2 m + mw^2} + \frac{6.0w \cos(\alpha)}{h^2 m + mw^2} & \frac{6.0h \sin(\alpha)}{h^2 m + mw^2} - \frac{6.0w \cos(\alpha)}{h^2 m + mw^2} \end{bmatrix}$$

## 7 Posicionamiento de polos

El control con posicionamiento de polos se realiza obteniendo una matriz de ganancia  $K$ , la cual multiplicada con el estado actual de la siguiente manera resulta en las acciones de control apropiadas para llevar el vector estado a 0. Los detalles se extrajeron de <https://ctms.engin.umich.edu/CTMS/index.php?example=Introduction&section=ControlStateSpace>

$$\mathbf{u}(t) = -K\mathbf{x}(t)$$

$$\frac{d}{dt}\mathbf{x}(t) = A\mathbf{x}(t) + B\mathbf{u}(t)$$

Para el posicionamiento de polos es necesario evaluar numéricamente las matrices, en función de lo solicitado en el laboratorio y observaciones durante el desarrollo se utilizan los siguientes valores.

Parámetro	Valor
$g$	$9.81[m/s^2]$
$\alpha$	$30[grados]$
$m$	$24[kg]$
$w$	$0.2[m]$
$h$	$0.3[m]$

Cuadro 1: Parámetros de simulación.

Las matrices espacio estado evaluadas quedan

$$A = \begin{bmatrix} 0 & 0 & 0 & 1.0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.0 \\ 0 & 0 & 9.81 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0.020833333333333 & -0.020833333333333 \\ -0.0360843918243516 & -0.0360843918243516 \\ 0.0446251553017072 & -0.0446251553017072 \end{bmatrix}$$

Se definen los siguientes polos de manera arbitraria.

$$Polos = [-0.1 \quad -0.2 \quad -0.3 \quad -0.5 \quad -0.6 \quad -0.7]$$

La función del paquete *SciPy*, `scipy.place_poles(A, B, poles)` entrega la matriz de ganancia  $K$ .

$$K = \begin{bmatrix} 0.01021 & -1.66276 & 9.00668 & 0.17973 & -11.08511 & 17.31705 \\ -0.01376 & -1.66277 & -10.25376 & -0.22230 & -11.08513 & -18.34947 \end{bmatrix}$$

Finalmente se verifica que el control sea efectivo, realizando una simulación interactiva <https://qgerman2.github.io/ecasa2/>

## 8 Simulación sin ruido

Se busca llevar el satélite con la siguiente condición inicial al origen.

$$\begin{bmatrix} x(0) \\ y(0) \\ \theta(0) \\ \left. \frac{d}{dt}x(t) \right|_{t=0} \\ \left. \frac{d}{dt}y(t) \right|_{t=0} \\ \left. \frac{d}{dt}\theta(t) \right|_{t=0} \end{bmatrix} = \begin{bmatrix} 200 \\ 1000 \\ 0 \\ -1 \\ -5 \\ 0 \end{bmatrix}$$

Se simulan las ecuaciones del movimiento **no lineales** integrando por método de Euler a un paso de tiempo de  $1/60[s]$ , aplicando en cada paso el control por medio de la matriz de ganancia  $K$  y los resultados son los siguientes.

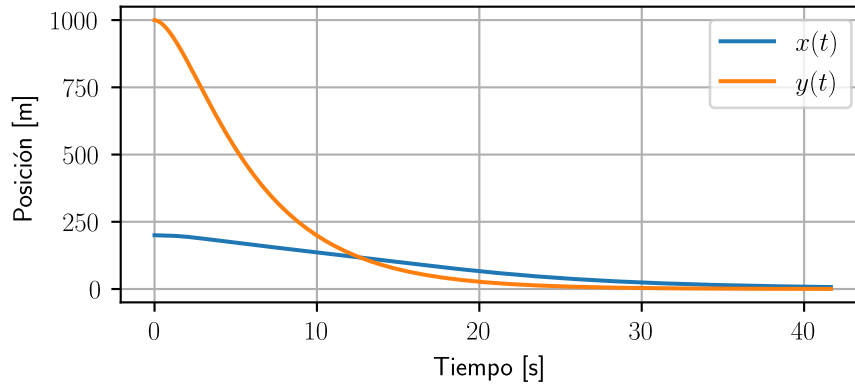


Figura 4: Simulación con control.

Y la acción de control es



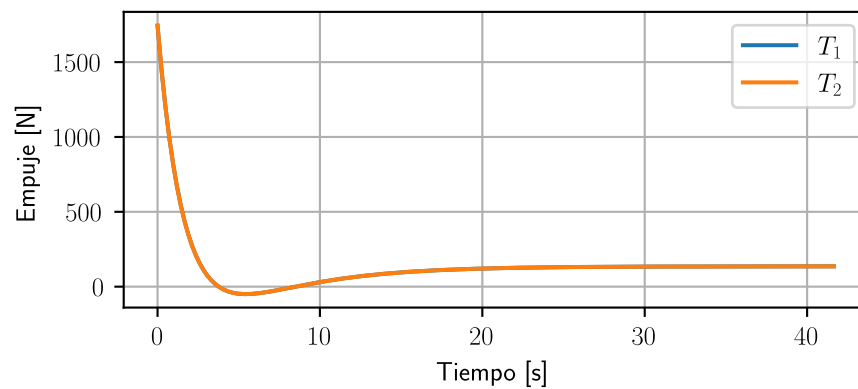


Figura 5: Acción de control.

## 9 “Diagrama de bloques (con filtro de Kalman)”

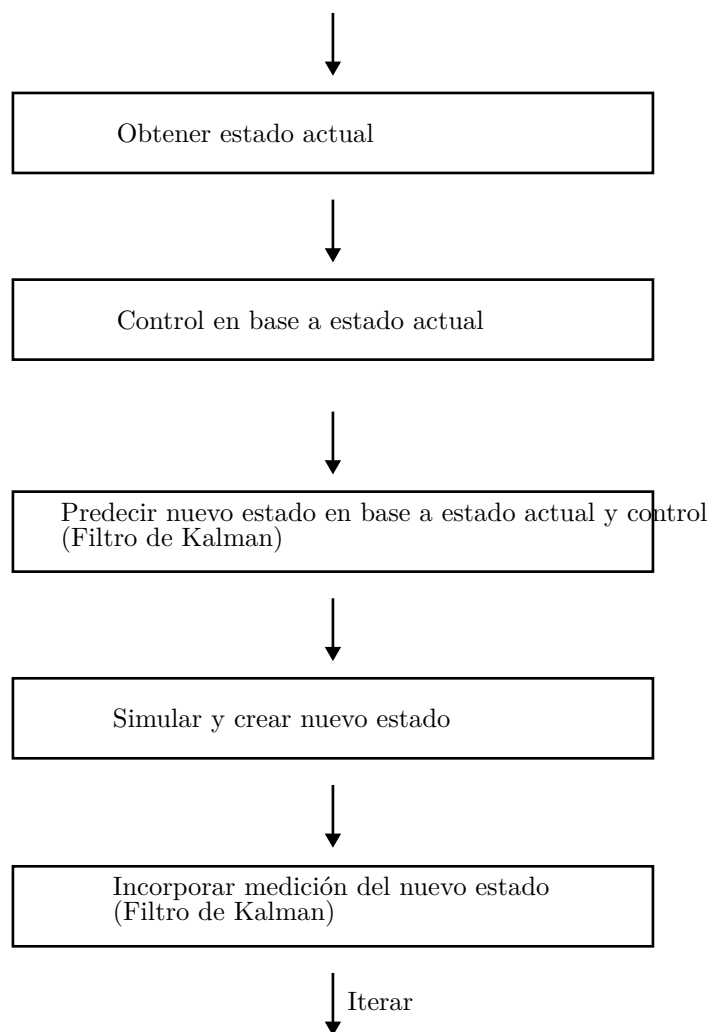


Figura 6: Intento de diagrama de bloques (con filtro de Kalman).

## 10 Filtro de Kalman

Para implementar un filtro de Kalman, es necesario construir las siguientes matrices

*Belief* de estado actual  $\mathbf{x}$  y la matriz de covarianza  $\mathbf{P}$  que representa que tan seguro se está del estado actual. Para  $\mathbf{x}$  se ingresan los mismos valores que la condición inicial y para  $\mathbf{P}$  se utiliza la varianza de los sensores simulados, los

valores de covarianza se dejan inicialmente en 0.

Matriz de proceso  $\mathbf{F}$  que corresponde al sistema lineal muy similar a la matriz  $\mathbf{A}$  del espacio estado, y también la matriz de control  $\mathbf{B}$  correspondiente al efecto de la acción de control sobre el sistema, similar a  $\mathbf{B}$  del espacio estado.

La ecuación del filtro de Kalman para realizar una predicción es la siguiente:

$$x_{k+1} = \mathbf{F}x_k + \mathbf{B}u$$

Recordando la muy parecida ecuación del espacio estado:

$$\frac{d}{dt}\mathbf{x}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t)$$

Podemos obtener las matrices  $\mathbf{F}$  y  $\mathbf{B}$  del filtro de Kalman realizando el siguiente procedimiento, primero se multiplica ambos lados por  $dt$  y se suma el estado anterior, considerando  $x_k$  y  $x(t)$  equivalentes.

$$\frac{d}{dt}\mathbf{x}(t) \cdot dt + x_k = \mathbf{A}x_k \cdot dt + \mathbf{B}\mathbf{u}(t) \cdot dt + x_k$$

$$x_{k+1} = \mathbf{A}x_k \cdot dt + \mathbf{B}\mathbf{u}(t) \cdot dt + x_k$$

Al lado derecho, se puede agrupar  $\mathbf{A}$  con  $x_k$  y  $dt$  de la siguiente manera

$$x_{k+1} = (\mathbf{A} \cdot dt + \mathbf{I})x_k + (\mathbf{B} \cdot dt)u$$

De esta forma se obtienen las matrices  $\mathbf{F}$  y  $\mathbf{B}$  a partir de las matrices  $\mathbf{A}$  y  $\mathbf{B}$  del espacio estado.

La matriz de imprecisión del proceso  $\mathbf{Q}$  se considera 0, la matriz de conversión de estado a espacio medición  $\mathbf{H}$  se considera como diagonal de unos, indicando que se tienen sensores que entregan las variables tal cual como están representadas en el vector estado, incluyendo un tipo de giroscopio para la variable de rotación.

Finalmente la matriz  $\mathbf{R}$  de ruido es una diagonal con las varianzas de ruido presente en los sensores como indicado en las instrucciones del laboratorio.

$$\mathbf{x} = \begin{bmatrix} 200 \\ 1000 \\ 0 \\ -1 \\ -5 \\ 0 \end{bmatrix}$$

$$\mathbf{P} = \begin{bmatrix} 2500.0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2500.0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.01 & 0 & 0 & 0 \\ 0 & 0 & 0 & 25.0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 25.0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.01 \end{bmatrix}$$

$$\begin{aligned}
\mathbf{F} &= \begin{bmatrix} 1.0 & 0 & 0 & 0.01666 & 0 & 0 \\ 0 & 1.0 & 0 & 0 & 0.01666 & 0 \\ 0 & 0 & 1.0 & 0 & 0 & 0.01666 \\ 0 & 0 & 0.1635 & 1.0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.0 \end{bmatrix} \\
\mathbf{B} &= \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0.000347222222222222 & -0.000347222222222222 \\ -0.00060140653040586 & -0.00060140653040586 \\ 0.000743752588361787 & -0.000743752588361787 \end{bmatrix} \\
\mathbf{H} &= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \\
\mathbf{R} &= \begin{bmatrix} 2500.0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2500.0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.01 & 0 & 0 & 0 \\ 0 & 0 & 0 & 25.0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 25.0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.01 \end{bmatrix}
\end{aligned}$$

Notar que se incorpora un sensor tipo giroscopio muy preciso, esto es porque durante el desarrollo se observó que la rotación del satélite afecta muchísimo al movimiento.

Para el desarrollo de las demás ecuaciones del filtro de Kalman, se utilizó la librería *filterpy* como está descrito en el siguiente libro <https://github.com/rllabbe/Kalman-and-Bayesian-Filters-in-Python>

## 10.1 Agregando ruido y filtrándolo de los resultados anteriores

Recordemos el resultado de la simulación previa

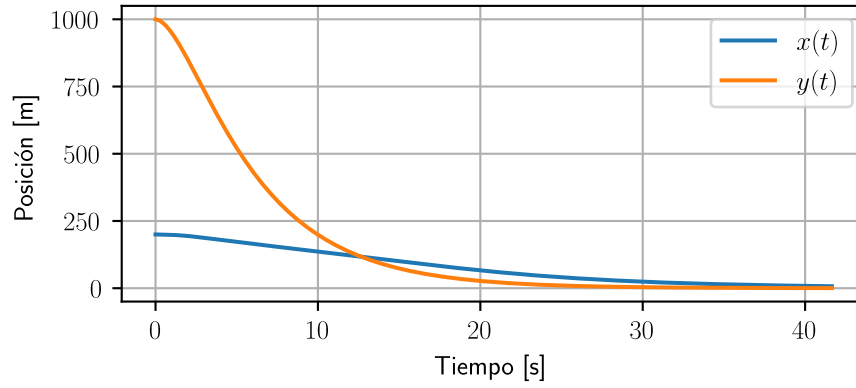


Figura 7: Estado sin ruido.

Si se le agrega ruido, se tiene

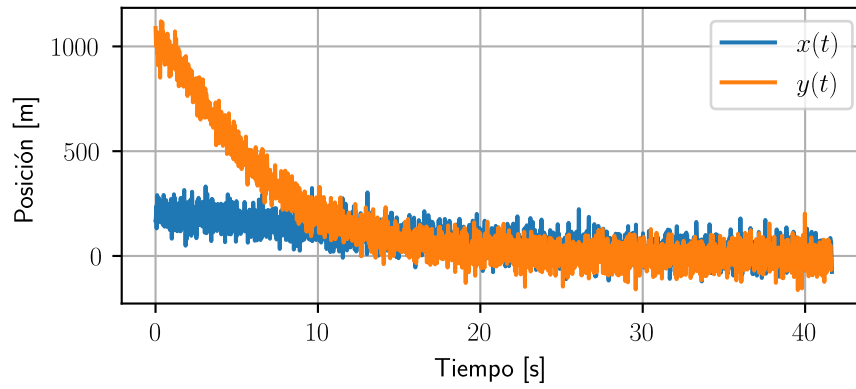


Figura 8: Estado con ruido.

Incorporando el filtro de Kalman

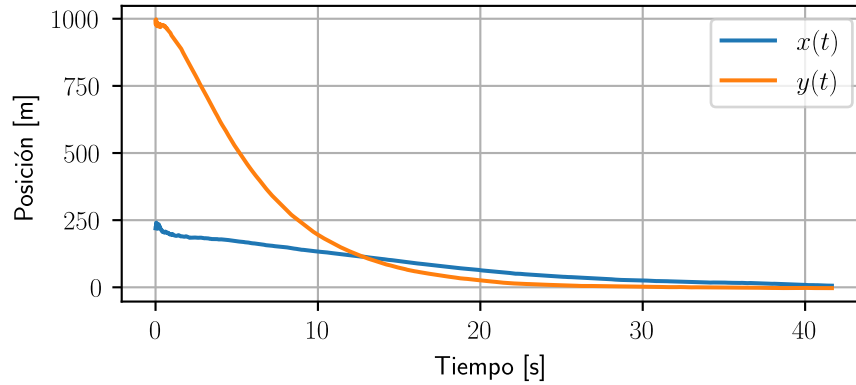


Figura 9: Estado con ruido filtrado.

## 11 Simulación con ruido

Para cada paso de la simulación, se extrajo el estado real del satélite y se le incorporó ruido de acuerdo a la varianza establecida. Esta medición con ruido es la que se introduce al filtro de Kalman y es con la salida del filtro que luego se evalúa la acción de control con la matriz de ganancia  $K$ , por lo que se tiene el peligro de que si el filtro no funciona, tampoco funcionara el control.

Los resultados son los siguientes:

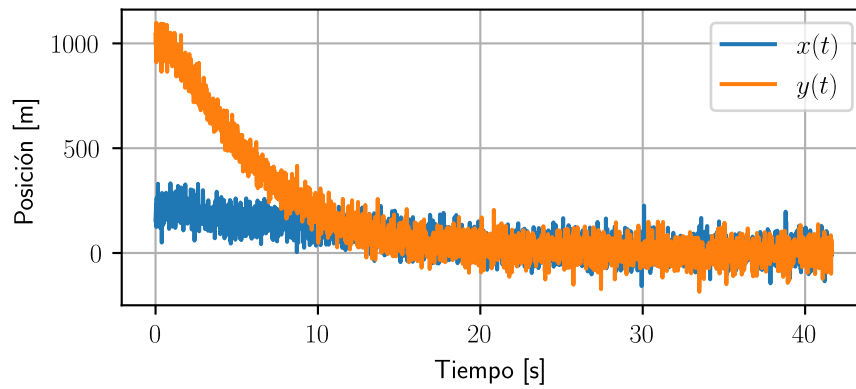


Figura 10: Mediciones de posición obtenidas.

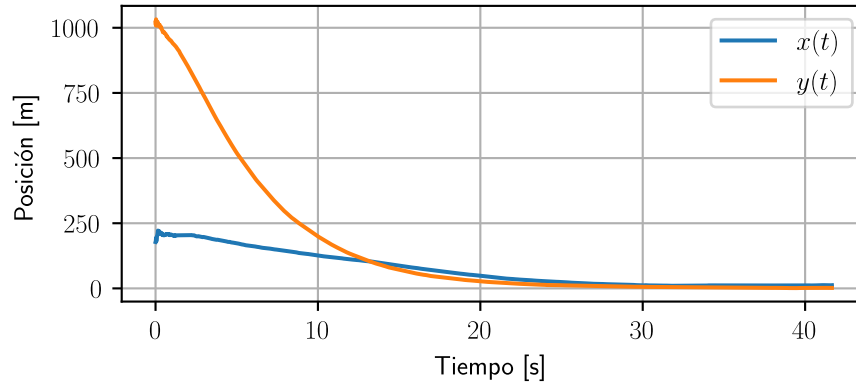


Figura 11: Mediciones de posición filtradas.

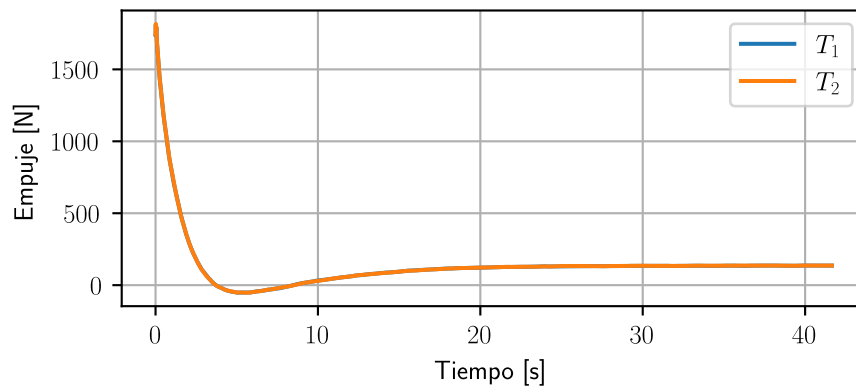


Figura 12: Acción de control.

## 12 Conclusiones

El código puede encontrarse aquí <https://github.com/qgerman2/ecasa2/blob/main/lab.ipynb>

Aún me sorprende lo efectivo del método de posicionamiento de polos para controlar sistemas de múltiples variables, lo encontré mágico en el ramo de sistemas de control y todavía es así, incluso para polos con números sin sentido el control usualmente funciona.

Creo que las formas en que se presenta el filtro de Kalman en el material educativo parte muy simple con ejemplos como de mover un vehículo, pero rápidamente abruma con la notación matemática y las matrices y varios conceptos

previos que se deben manejar. Encontrar el libro <https://github.com/rlabbe/Kalman-and-Bayesian-Filters-in-Python> fue un milagro y me gusta mucho como presentan el filtro, pude implementarlo aquí solo leyendo hasta la mitad del libro que es donde recién introducen a Kalman multivariable para sistemas lineales, pero que de todas formas aplico para el sistema no lineal.

Este laboratorio representa todo lo que esperaba de este ramo, simular, inventar sensores y luego hacer control, lamentablemente fue al final del ramo donde se nos puso un desafío en mi opinión abordable, ya que el antiguo del cohete 3D resulto bastante complejo.