15 años Pregrado en Ingeniería Matemática

Linealización con MATLAB

Guía Práctica

Daniel Rojas Díaz[1]

Carlos Mario Vélez Sánchez[2]

Departamento de Ciencias Matemáticas EAFIT

Índice

1. Conceptos de linealización

Simbólicamente podemos expresar una ecuación diferencial ordinaria (**EDO**) de n-ésimo orden con una variable dependiente mediante la forma general

$$F(x, y, y',, ..., y^{(n)}) = 0, \tag{1}$$

donde F es una función con valores reales de $n+2$ variables: $x, y, y', ..., y^{(n)}$ que puede resolverse para la mayor derivada en términos de las $n+1$ variables restantes

$$\frac{d^n y}{dx^n} = f(x, y, y', ..., y^{n-1}), \tag{2}$$

donde f es una función continua con valores reales que se conoce como la **forma normal** de la ecuación (1). Ahora bien, es importante recordar que las EDO pueden clasificarse por **linealidad**, así, se dice que una ecuación diferencial de n-ésimo orden (1) es lineal si F es lineal en $y, y', ...y^{(n)}$, es decir, cuando la ecuación (1) es de la forma $a_n(x)y^{(n)} + a_{n-1}(x)y^{(n-1)} + ... + a_1(x)y' + a_0(x)y - g(x) = 0$. De lo anterior es posible identificar que las propiedades características de una EDO lineal son las siguientes:

- La variable dependiente y y todas sus derivadas $y', y'', ..., y^{(n)}$ son de primer grado, es decir, la potencia de cada término que contiene y es igual a 1.
- Los coeficientes de $a_0, a_1, ..., a_n$ de $y, y', ..., y^{(n)}$ dependen de la variable independiente x.

Ejemplo 1.1: EDO lineal y no lineal

1. EDO lineales:
   $(y-x)dx + 4xdy = 0$ $\qquad y'' - 2y' + y = 0$ $\qquad x^3y''' + xy' - 5y = e^x$
2. EDO no lineales
   $(1-y)y' + 2y = e^x$ $\qquad y'' + \operatorname{sen} y = 0$ $\qquad y^{(4)} + y^2 = 0$

[1]Estudiante Pregrado en Biología - drojasd@eafit.edu.co
[2]Docente Ingeniería Matemática - cmvelez@eafit.edu.co

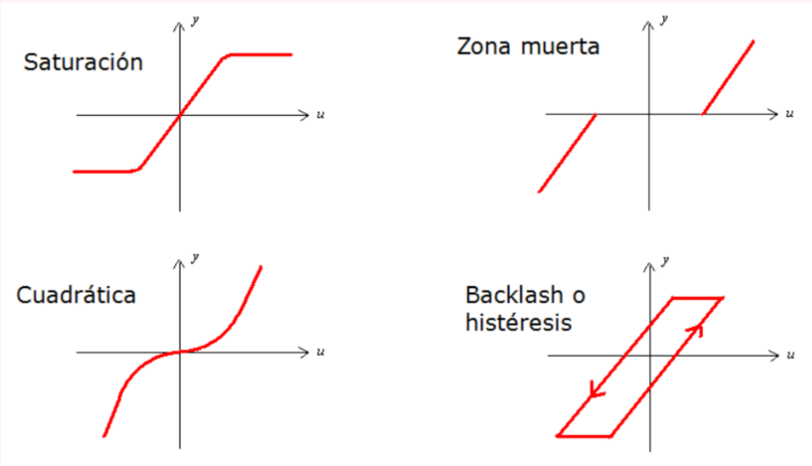
1.1. ¿Qué es y para qué linealizar?

Las ecuaciones diferenciales son fuentes de información ya que generalmente se emplean para describir el comportamiento de algunos sistemas o fenómenos de la vida real, ya sean físicos, químicos, biológicos, sociológicos e incluso económicos. En el modelamiento matemático mediante ecuaciones diferenciales, el objetivo subsecuente al planteamiento de la estructura del modelo es la extracción de información, es decir, solucionar el sistema de ecuaciones planteado y analizarlo. Muchos sistemas de EDO no son lineales y por ende no pueden resolverse analíticamente sino a través de métodos numéricos que se aproximan a la solución real.

Uno de los principales retos a la hora de trabajar con sistemas no lineales es que su comportamiento no puede predecirse extrapolando resultados previamente obtenidos, es decir que cada cambio sobre los parámetros o entradas implica resolver nuevamente el sistema. De ahora en adelante, nuestro interés se centrará en la relación entre la entrada y la salida del sistema (Ejemplo 1.1).

Ejemplo 1.2: Alinealidades en sistemas dinámicos

Tipos de alinealidades



Siendo la entrada (u), la salida (y) se entiende como el valor al que converge el sistema.

La linealización es un método de análisis de sistemas no lineales que consiste en obtener un equivalente lineal del sistema alrededor de un punto de equilibrio. La validez del sistema lineal depende mucho de la naturaleza del punto de operación (estable o inestable), lo cual determina el rango de linealidad.

Entre las principales ventajas de la linealización podemos mencionar que:

- Existen métodos matemáticos muy conocidos para resolver y analizar sistemas lineales.
- Es posible linealizar un sistema en tiempo discreto.
- El modelo lineal puede ser variable en el tiempo.
- Es posible analizar el modelo no lineal con todos los beneficios de los modelos lineales.
- Pueden desarrollarse controles lineales aplicables al sistema no lineal.

1.2. Curva de linealización

La curva de linealización es una herramienta gráfica que nos permite escoger un punto de operación adecuado. En general, es posible linealizar el sistema en cualquier punto de equilibrio para un vector de entradas $[u_0 = a_0, u_1 = a_1, ..., u_n = a_n]$, sin embargo, sin ningún conocimiento *a priori* es muy posible que el rango de linealidad obtenido no tenga sentido y nuestros posteriores análisis se vean viciados. Por ejemplo, si comparamos las curvas de linealidad del Ejemplo 1.1 con la curva de la Figura 1, podemos apreciar fácilmente que existen valores de u para los cuales no tiene sentido linealizar (en casos de alinealidades como saturación, zona muerta o histéresis).

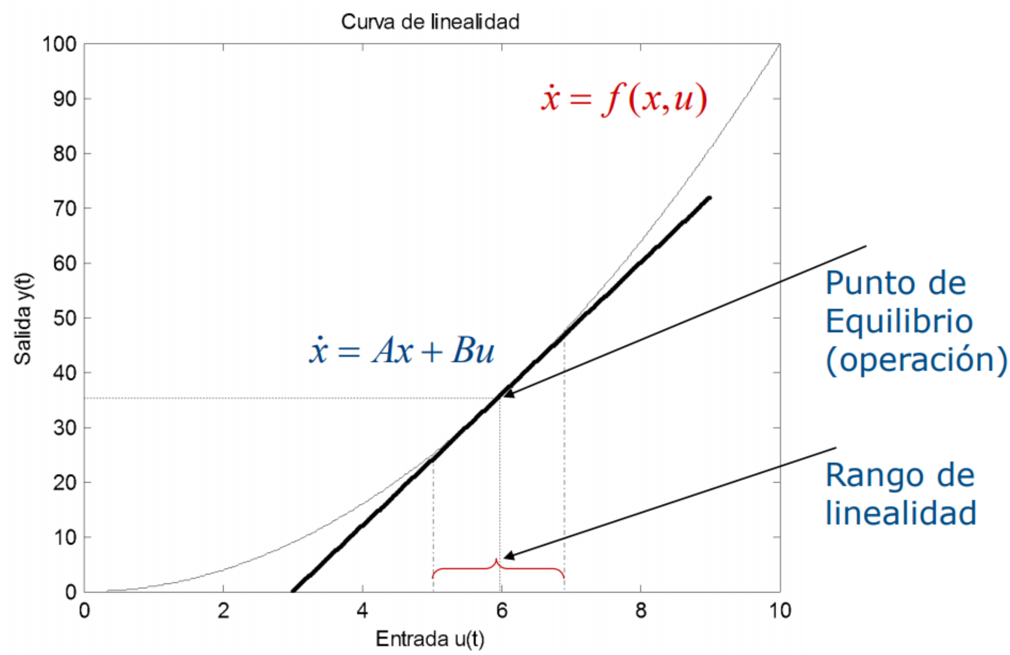


Figura 1: Al trazar la curva de linealidad para el sistema no lineal de primer orden $x' = f(x, u)$ podemos apreciar que una aproximación a través de derivadas en el punto de operación $u = 6$ tendría un rango de linealidad aproximado $[5, 7]$

1.3. Puntos críticos

Antes de abordar el tema de linealización propiamente, es prudente contextualizar otros conceptos básicos aún si no son propiamente necesarios ya que constituyen todo el soporte del método. En primer lugar, la teoría que abordaremos es válida principalmente para los sistemas autónomos, es decir, sistemas de ecuaciones que no dependen explícitamente de la variable independiente o lo que es igual: sistemas cuya razón de cambio solo depende de su estado actual.

Como debe haberse abordado previamente en el curso de Sistemas dinámicos o Modelación y simulación IV, una ecuación diferencial puede representarse como un sistema autónomo a través de cambios de variable, **por ejemplo**, en la EDO de segundo orden $x'' = g(x, x')$ podemos hacer $y_1 = x$ y $y_2 = x'$, así, $y_1' = y_2$ y $y_2' = g(y_1, y_2)$. Transformando la EDO de segundo orden en un sistema de dos ecuaciones de primer orden.

Ejemplo 1.3: Sistemas no autónomos

El sistema de EDO no lineales de primer orden

$$\frac{dx_1}{dt} = x_1 - 3x_2 + t^2$$

$$\frac{dx_2}{dt} = tx_1 \operatorname{sen} x_2$$

es un sistema no autónomo debido a la presencia explícita de t.

Ejemplo 1.4: Sistemas autónomos

Ejemplo de sistemas autónomos:

La ecuación del péndulo simple está dada por $\ddot{\theta} + \dfrac{f\dot{\theta}}{m} + \dfrac{g \operatorname{sen}\theta}{l} = \dfrac{F_{ext}}{ml}$

Donde θ es el ángulo, $\dot{\theta}$ es la velocidad angular, f es el coeficiente de fricción, m es la masa, g es la aceleración de la gravedad, l es la longitud de la cuerda y F_{ext} es la fuerza externa aplicada. Recuerde que las derivadas con respecto al tiempo $\dfrac{dx}{dt}$ se representan como $\dot{x}$.

Procedemos de la siguiente manera:

$$\begin{cases} x_1 = \theta \\ x_2 = \dot{\theta} \\ u = F_{ext} \end{cases}$$

De donde obtenemos las siguientes ecuaciones de estado

$$\begin{cases} \dot{x_1} = x_2 \\ \dot{x_2} = \ddot{\theta} = -\dfrac{fx_2}{m} - \dfrac{g \operatorname{sen} x_1}{l} + \dfrac{u}{ml} \end{cases}$$

Nota

Por razones prácticas, el resto de ejemplificaciones de esta sección se dará para sistemas de segundo orden, sin embargo, el mismo razonamiento es aplicable a sistemas de orden superior, tal y como se verá más adelante.

Un sistema como el expuesto en el Ejemplo 1.4 se denomina **sistema autónomo plano** y puede escribirse de la forma

$$\begin{aligned} \dot{x_1} &= P(x_1, x_2) \\ \dot{x_2} &= Q(x_1, x_2) \end{aligned} \tag{3}$$

lo cual define a su vez un campo vectorial $\mathbf{V}(x_1, x_2) = (P(x_1, x_2), Q(x_1, x_2))$ cuyas soluciones pueden interpretarse como la trayectoria resultante de una partícula que se mueve a través de la región del plano. Si $\mathbf{X}(t) = ((x_1(t), x_2(t))$ denota la posición de la partícula en el tiempo t, entonces $\mathbf{X}(t) = (x_1'(t), x_2'(t))$ es el vector velocidad $\mathbf{V}$. Así la trayectoria de la partícula es una solución del sistema que satisface la condición inicial $\mathbf{X}(0) = (x_1(0), x_2(0))$. La gráfica de x_1 vs x_2 recibe el nombre de plano de estado o espacio de estado (también diagrama de fase) para sistemas de orden n.

Si $P(x_1, x_2), Q(x_1, x_2)$ y las primeras derivadas parciales $\frac{\partial P}{\partial x_1}, \frac{\partial P}{\partial x_2}, \frac{\partial Q}{\partial x_1}, \frac{\partial Q}{\partial x_2}$ son continuas en una región R del plano, entonces una solución del sistema autónomo (3) que satisface $\mathbf{X}(0) = \mathbf{X}_0$ es única y se clasifica dentro de uno de los siguientes tipos básicos:

1. **Una solución constante**. Los valores de x_1 y x_2 no cambian en el tiempo. Se le conoce como **punto crítico** e implica que cuando la partícula se coloca en dicho punto permanece en él indefinidamente.

2. **Una solución que define un arco**.

3. **Una solución periódica**. En este caso, si p es el periodo de la solución, entonces $\mathbf{X}(t+p) = \mathbf{X}(t)$.

Para hallar los puntos críticos de un sistema de ecuaciones autónomo es necesario igualar a 0 los miembros de la derecha de dicho sistema y resolver las correspondientes igualdades (lo cual se hace normalmente mediante software). Ahora bien, el análisis realmente interesante radica en el comportamiento de las soluciones para los diagramas de fase cuando se escogen $\mathbf{X}_0$ cercanos a un punto crítico

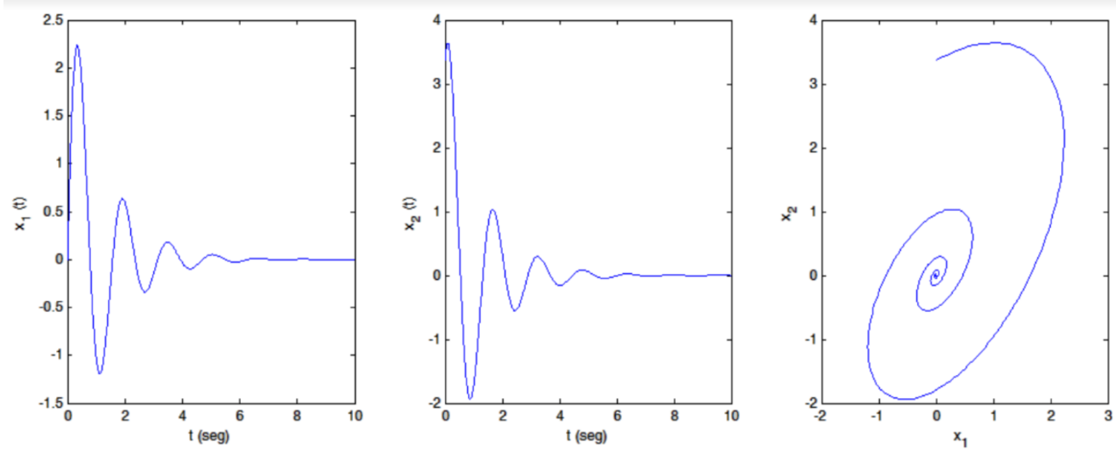


Figura 2: Solución del sistema de ecuaciones planteado en el Ejemplo 1.4, las dos primeras gráficas de la izquierda corresponden a las soluciones aproximadas para x_1 y x_2 en el tiempo. La gráfica de la derecha es el plano de estado correspondiente, siendo una solución que define un arco.

del sistema. Suponga que $\mathbf{X}_i$ es un punto crítico en un sistema autónomo plano y que $\mathbf{X} = \mathbf{X}(t)$ es una solución del sistema que satisface $\mathbf{X}(0) = \mathbf{X}_0$. Si se interpreta la solución como una trayectoria de una partícula en movimiento y $\mathbf{X}_0$ está cerca de $\mathbf{X}_i$ entonces:

1. Si $\lim_{t\to+\infty} \mathbf{X}(t) = \mathbf{X}_i$, entonces $\mathbf{X}_i$ es un punto crítico estable ya que el diagrama de fase converge a él.
2. Si $\lim_{t\to+\infty} \mathbf{X}(t) \neq \mathbf{X}_i$, entonces existen dos posibilidades: o bien $\mathbf{X}(t)$ se aleja de $\mathbf{X}_i$, en cuyo caso se trata de un punto crítico inestable, o $\mathbf{X}(t)$ describe circunferencias en torno al punto crítico, en cuyo caso el punto crítico es un **Centro**.

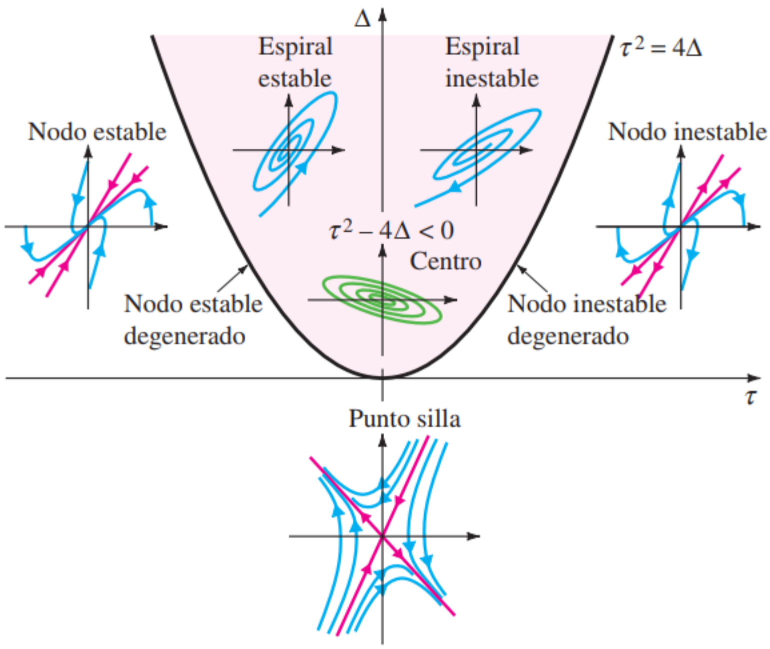


Figura 3: Resumen de la clasificación de puntos críticos. En general, los puntos críticos inestables $\mathbf{X}_i$ se caracterizan porque al escoger $\mathbf{X}_0$ cercanos, el diagrama de fase no tiende a $\mathbf{X}_i$. Por el contrario, en el caso de los puntos críticos estables $\mathbf{X}_n$, $\mathbf{X}_0$ tiende hacia ellos. Extraído de [Zill and Wright, 2015].

Ejemplo 1.5: Cálculo de puntos críticos

Dadas las ecuaciones diferenciales de un levitador magnético

$$\begin{cases} m\ddot{y} = mg - \dfrac{i^2}{y} \\ e = Ri + L\dfrac{di}{dt} \end{cases}$$

De donde obtenemos las siguientes ecuaciones de estado mediante $x_1 = y, \quad x_2 = \dot{y}, \quad x_3 = i, \quad u = e(t)$.

$$\begin{cases} \dot{x_1} = x_2 \\ \dot{x_2} = g - \dfrac{x_3^2}{mx_1} \\ \dot{x_3} = -\dfrac{Rx_3}{L} + \dfrac{u(t)}{L} \end{cases}$$

Igualando las ecuaciones de estado a 0 y resolviendo obtenemos

$$\begin{cases} \dot{x_1} = x_2 = 0, \quad x_{2-0} = 0 \\ \dot{x_3} = -\dfrac{Rx_3}{L} + \dfrac{u(t)}{L} = 0, \quad x_{3-0} = \dfrac{u_0}{R} \\ \dot{x_2} = g - \dfrac{x_3^2}{mx_1} = 0, \quad x_{1-0} = \dfrac{u_0^2}{mgR^2} \end{cases}$$

1.4. Matriz Jacobiana

Una vez identificado el punto de operación podemos iniciar el proceso de linealización. Recuerde, del curso de cálculo, que la linealización de una función derivable $f(x)$ en el punto x_1 es la ecuación de la recta tangente a la gráfica de f en el punto:

$$y = f(x_1) + f'(x_1)(x - x_1).$$

Para valores de x cercanos a x_1, los puntos sobre la gráfica de f son cercanos a los puntos sobre la recta tangente (Rango de linealidad, Figura 1). De manera similar, la linealización de una función de dos variables $f(x_1, x_2)$ que es derivable en un punto $\mathbf{X}_1 = (x_{1(a)}, x_{2(b)})$ es la ecuación del **plano tangente** a la gráfica de f:

$$z = f(x_{1(a)}, x_{2(b)}) + \frac{\partial f}{\partial x_1}(x_{1(a)}, x_{2(b)})(x_1 - x_{1(a)}) + \frac{\partial f}{\partial x_2}(x_{1(a)}, x_{2(b)})(x_2 - x_{2(b)}) \tag{4}$$

Nota

Recuerde que las ecuaciones de estado pueden darse en forma matricial. Siendo

$$\mathbf{X}(t) = \begin{pmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_n(t) \end{pmatrix}, \qquad \mathbf{g}(\mathbf{X}) = \begin{pmatrix} g_1(x_1, x_2, \cdots, x_n) \\ g_2(x_1, x_2, \cdots, x_n) \\ \vdots \\ g_n(x_1, x_2, \cdots, x_n) \end{pmatrix}$$

Entonces, un sistema autónomo como el presentado en el Ejemplo 1.4 puede escribirse de la forma $\mathbf{X}' = \mathbf{g}(\mathbf{X})$.

La linealización consiste en reemplazar el término $\mathbf{g}(\mathbf{X})$ del sistema original $\mathbf{X}' = \mathbf{g}(\mathbf{X})$ por un término lineal de la forma $\mathbf{A}(\Delta\mathbf{X})$ que posteriormente da origen al espacio continuo lineal

$$\begin{cases} \dot{X} = \mathbf{AX} + \mathbf{Bu} \\ \mathbf{y} = \mathbf{CX} + \mathbf{Du} \end{cases}$$

Para calcular la matriz $\mathbf{A}$ procedemos de la siguiente manera (nótese la importancia de los conceptos introducidos hasta ahora en el documento). En el caso de un sistema de orden 2, $g(x_1, x_2)$ es la forma matricial del sistema de ecuaciones (3), por lo cual la ecuación del plano tangente (4) puede escribirse de la siguiente forma lineal dado que, cuando $\mathbf{X}_1$ es un punto crítico, por definición $P(x_1, x_2) = Q(x_1, x_2) = 0$ (Ejemplo 1.5):

$$\begin{aligned} \dot{x}_1 = P(x_1, x_2) \approx \left.\frac{\partial P}{\partial x_1}\right|_{(x_1,x_2)} (\Delta x_1) + \left.\frac{\partial P}{\partial x_2}\right|_{(x_1,x_2)} (\Delta x_2) \\ \dot{x}_2 = Q(x_1, x_2) \approx \left.\frac{\partial Q}{\partial x_1}\right|_{(x_1,x_2)} (\Delta x_1) + \left.\frac{\partial Q}{\partial x_2}\right|_{(x_1,x_2)} (\Delta x_2) \end{aligned} \tag{5}$$

A partir de (5) podemos definir a la matriz $\mathbf{A}(\Delta\mathbf{X})$ como

$$\mathbf{A} = \begin{pmatrix} \left.\frac{\partial P}{\partial x_1}\right|_{(x_1,x_2)} & \left.\frac{\partial P}{\partial x_2}\right|_{(x_1,x_2)} \\ \left.\frac{\partial Q}{\partial x_1}\right|_{(x_1,x_2)} & \left.\frac{\partial Q}{\partial x_2}\right|_{(x_1,x_2)} \end{pmatrix}$$

Y tras analizar la estabilidad, queda concluido el proceso de linealización.

Nota **Criterio de estabilidad para sistemas linealizados**

Sea $\mathbf{X}_i$ un punto crítico del sistema linealizado $\mathbf{X}' = \mathbf{A}(\Delta\mathbf{X})$

- Si los eigenvalores de $\mathbf{A} = \mathbf{g}'(\mathbf{X}_i)$ tienen parte real negativa, entonces $\mathbf{X}_i$ es un punto crítico asintóticamente estable.
- Si existe al menos un eigenvalor de $\mathbf{A} = \mathbf{g}'(\mathbf{X}_i)$ con parte real positiva, entonces $\mathbf{X}_i$ es un punto crítico asintóticamente inestable.

Ejemplo 1.6: Cálculo de la matriz Jacobiana

Continuando con el Ejemplo 1.5, podemos calcular la respectiva matriz Jacobiana $\mathbf{A}$ (Compruébelo usted mismo a través de la definición de $\mathbf{A}$):

Ecuaciones de estado:

$$\begin{cases} \dot{x_1} = x_2 \\ \dot{x_2} = g - \dfrac{x_3^2}{mx_1} \\ \dot{x_3} = -\dfrac{Rx_3}{L} + \dfrac{u(t)}{L} \end{cases}$$

El punto crítico está definido por $x_{2-0} = 0$, $x_{3-0} = \dfrac{u_0}{R}$ y $x_{1-0} = \dfrac{u_0^2}{mgR^2}$

Así, la Matriz Jacobiana es

$$\begin{pmatrix} 0 & 1 & 0 \\ \dfrac{mg^2R^2}{u^2} & -\dfrac{f}{m} & -\dfrac{2gR}{u} \\ 0 & 0 & -\dfrac{R}{L} \end{pmatrix}$$

2. Linealizando con MATLAB

Para este ejercicio de linealización utilizaremos el siguiente sistema de ecuaciones diferenciales de orden 3 planteado en [Esteva and Leyton, 2011].

$$\begin{aligned}\dot{s} &= \beta_s s[1-(s+r)] - \alpha_s cs - \mu_s s\\ \dot{r} &= \beta_r r[1-(s+r)] + q\alpha_s cs - \mu_r r\\ \dot{c} &= \mu_c - \mu_c c\end{aligned} \tag{6}$$

Que describe la dinámica poblacional de bacterias sensibles ($\dot{s}$) y resistentes ($\dot{r}$) a un antibiótico que se administra a una tasa ($\dot{c}$). Como puede apreciarse, las ecuaciones no dependen explícitamente del tiempo. Además, se trata de ecuaciones previamente normalizadas, lo cual implica que su gráfica estará acotada en el intervalo [0,1]. Es necesario aclarar que el parámetro α_s corresponde a $\alpha_s \dfrac{\Lambda}{\mu_c}$, siendo Λ la entrada del sistema. Observe que (6) ya se encuentra dado en forma de ecuaciones de estado y por ende podemos representarlo fácilmente en Simulink[1].

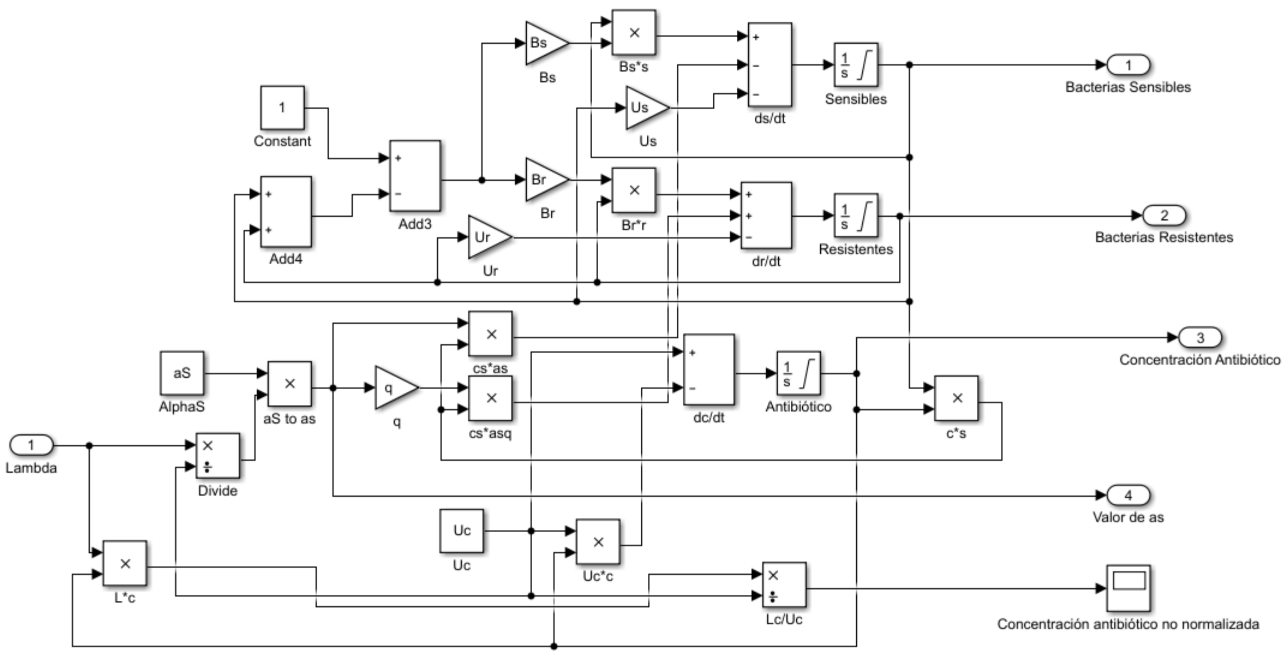


Figura 4: Representación en simulink del sistema de ecuaciones (6).

Parámetros	Definición	Valor
β_s	Tasa de crecimiento de S	0.4 minuto^{-1}
β_r	Tasa de crecimiento de R	0.1 minuto^{-1}
α_S	Eliminación de S	0.3960 (min.mg)$^{-1}$
K	Capacidad de carga de S y R	10^{12} bac.
q	Fracción de S que adquieren resist.	4^{-3} esc.
μ_s	Tasa de muerte natural de S	0.2 minuto^{-1}
μ_r	Tasa de muerte natural de R	0.09 minuto^{-1}
μ_c	Tasa de degradación de C	0.0083 minuto^{-1}
Λ	Dosis inicial de C	0.8328 mg/minuto

Figura 5: Valores de los parámetros del sistema de ecuaciones (6). Extraído de [Esteva and Leyton, 2011].

[1] Click para recurrir a las diapositivas de Espacio de estado y Simulación

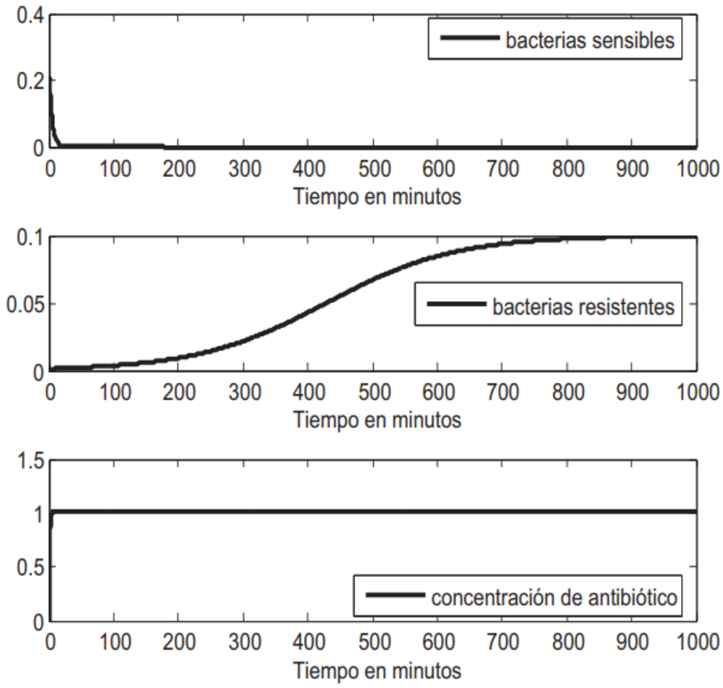


Figura 6: Transcurso temporal de las soluciones del sistema (6). Estas soluciones representan las densidades de las poblaciones de bacterias sensibles, bacterias resistentes y concentración de antibiótico. En esta simulación $\alpha_s \dfrac{\Lambda}{\mu_c} = 39{,}7$, $S_0 = 0{,}18$ y $R_0 = 0$. Extraído de [Esteva and Leyton, 2011].

2.1. Trazar y analizar la curva de linealización

Para trazar la curva de linealización (Figura 1) de nuestro sistema dinámico debemos realizar varias simulaciones con distintos valores para la entrada y almacenar los valores en los cuales se estabilizan los estados de interés ($\dot{s}$ y $\dot{r}$) . **Es absolutamente necesario estar seguro de que los valores que se almacenan corresponden al valor de convergencia de las salidas de interés** (Si su sistema no converge, continúe a la sección 2.2) .

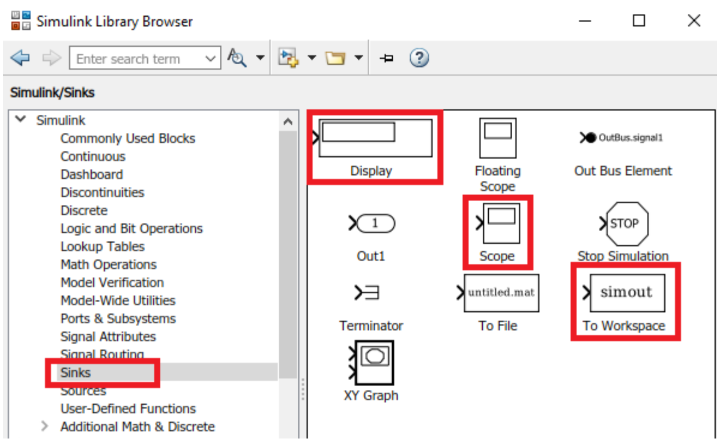


Figura 7: Simulink Library ofrece algunos bloques que pueden serle de utilidad para extraer los valores que empleará en la curva de linealización. Por ejemplo, Display le mostrará el valor actual de la salida, Scope le permite ver la solución numérica para identificar si el sistema está convergiendo y To Workspace le permite almacenar los valores de la salida en Matlab.

La Figura 11 es un excelente ejemplo de que trazar la curva de linealidad en muchas ocasiones es un proceso interactivo: Para valores de $u = \Lambda > 0{,}0035$ el sistema presenta una alinealidad tipo saturación (Ejemplo 1.2), para valores de $0 < u < 0{,}0015$ el sistema responde muy linealmente y para valores

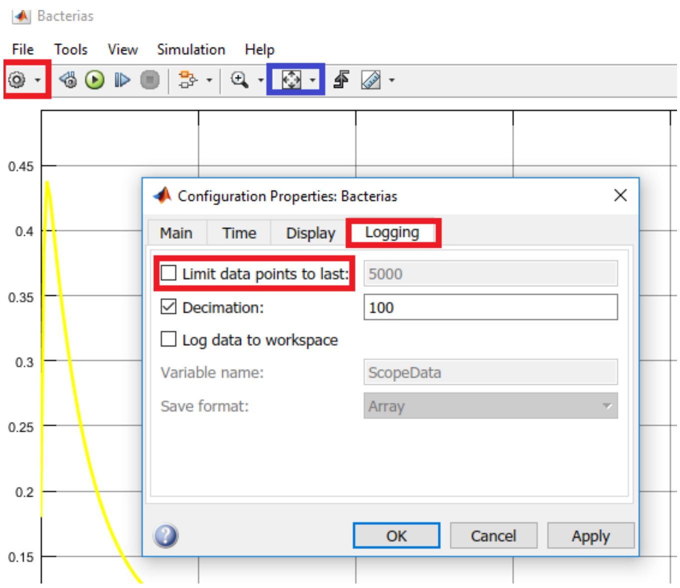


Figura 8: Recuerde que si desea ver todo el comportamiento del sistema que analiza a través del tiempo debe deseleccionar la opción de *Limit data points to last* a través del menú *Configuration* del bloque Scope. En muchas ocasiones, los problemas de visualización de la gráfica se solucionan haciendo click en el *Scale X and Y* (recuadro azul).

de $0{,}0015 < u < 0{,}0035$ el sistema presenta una alinealidad cuadrática. Dado lo anterior, se intuye que para cada sistema es necesario realizar múltiples simulaciones que nos permitan comprender la relación entre la entrada y la salida. Con un conocimiento *a priori* de nuestro sistema podrían haberse escogido 10 valores de u igualmente espaciados en el intervalo $[0{,}5]$ y obtendríamos como resultado una línea recta debido a la saturación.

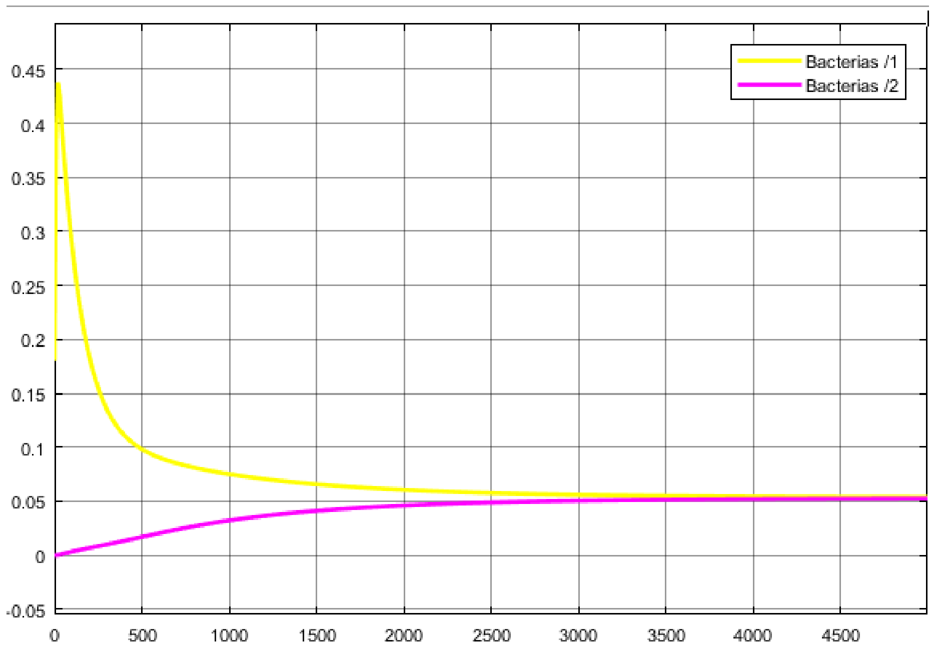


Figura 9: En la Figura 6 se aprecia que el sistema converge para un tiempo inferior a 1000, sin embargo, para ciertos valores de $u = \Lambda$, el sistema converge en tiempos superiores a 4000. Lo anterior evidencia la importancia de monitorear la simulaciones con las que se trazará la curva de linealización.

Una vez obtenida la curva de linealidad, podemos elegir nuestro punto de operación. En principio, el criterio de selección para dicho punto depende de nuestras necesidades y objetivos. Entre más lineal sea la curva de linealidad alrededor del punto de operación, entonces el rango de linealidad será mayor

```
X=[0,0.0005,0.001,0.0015,0.0018,0.0021,0.0024,0.0027,0.003,0.0031,0.0032,0.0033,0.0034,0.005];
S=[0.5, 0.4391, 0.3782, 0.317, 0.2802, 0.2431, 0.2056, 0.1671, 0.1259, 0.1107, 0.0887, 0.0535, 0,0];
R=[0,0.0012,0.0026,0.0041,0.0052,0.0065,0.0082,0.011,0.0164, 0.0197,0.0295,0.0528,0.1,0.1];
hold on
plot(X,S,'-mo');
title('vs sensibles')
plot(X,R,'-mo');
title('Bacterias sensibles y resistentes vs antibiótico')
hold off
```

Figura 10: La curva de linealidad puede trazarse mediante un script simple una vez se han obtenido los valores de convergencia de las salidas para distintos valores de entrada.

y el sistema equivalente arrojará buenas aproximaciones en un rango más amplio de u. Elegir un punto de operación una vez ha ocurrido la saturación no tiene sentido, además, elegir un punto muy cercano a la zona donde se presenta la saturación implica un rango de validez reducido (al menos hacia un extremo) para nuestro sistema lineal.

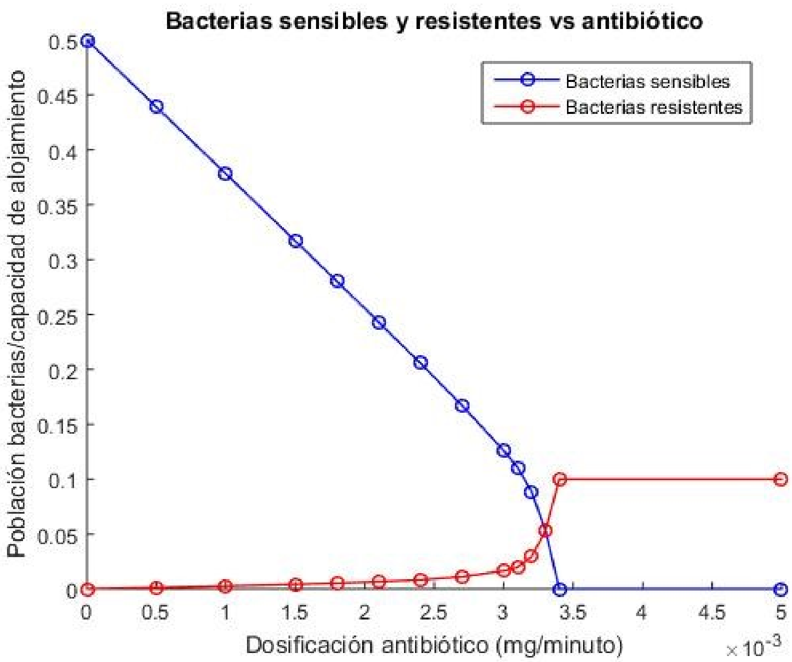


Figura 11: La curva de linealidad de nuestro sistema ha sido trazada para los estados $\dot{s}$ y $\dot{r}$. Hemos despreciado a $\dot{c}$ dado que por la naturaleza de dicha ecuación normalizada, es completamente insensible a $u = \Lambda$. Nótese la existencia de una alinealidad de tipo saturación para valores de u superiores a 0.0035.

2.2. Búsqueda de puntos críticos con *vpasolve*

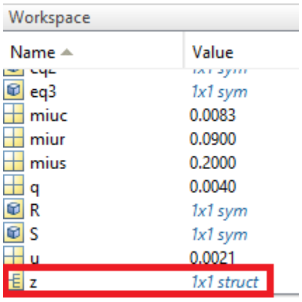
La función *vpasolve*[2] nos permite calcular numéricamente los puntos críticos de un sistema de ecuaciones (tal y como se hizo en el Ejemplo 1.5). Si bien es mucho más rápido, sencillo y preciso que trazar la curva de linealidad, no nos arroja información respecto al rango de linealidad o al tipo de alinealidad entre las entradas y salidas.

Usaremos *vpasolve* de la forma $z = vpasolve(ecuaciones == 0, variables)$:

[2]Click para abrir la documentación

```
Puntos_equilibrio.m
syms S R C %Definimos las variables simbólicas
%%Definimos los parámetros
u=0.0021;
Bs=0.4;
Br=0.1;
q=0.004;
mius=0.2;
miur=0.09;
miuc=0.0083;
alphas=0.396*u/miuc;
%%Definimos las ecuaciones a resolver
eq1=Bs*S*(1-(S+R))-alphas*C*S-mius*S;
eq2=Br*R*(1-(S+R))+q*alphas*C*S-miur*R;
eq3=miuc-miuc*C;
%%Resolvemos las ecuaciones
z=vpasolve(eq1==0,eq2==0,eq3==0,S,R,C);
```

Al ejecutar un código como el anterior, nos debe aparecer la variable z en el workspace, podemos acceder a los resultados a través de los comandos $z.S$, $z.R$ y $z.C$.




```
>> z.S

ans =

 0.24300453

>>
```

COMPARACIÓN VPASOLVE VS CURVA LINEALIZACIÓN

Punto de operación	PUNTOS DE EQUILIBRIO	Curva de linealización
0	S=0.5, R=0	S=0.5, R=0
2.1x10^{-3}	S=0.2429, R=0.0065	S=0.2431, R=0.0065
3.4x10^{-3}	S=0, R=0.1	S=0, R=0.1

S=Bacterias sensibles, R=Bacterias resistentes.

2.3. Linealización con Función linmod

La función *linmod*[3] nos permite el cálculo inmediato de las matrices **A, B, C, D** en un punto de operación. Para ello la emplearemos de la forma [A,B,C,D]=linmod('modelo_a_linealizar',[Estados en el punto de operacion],[Entradas en el punto de operacion]). En este orden de ideas, lo primero que debemos hacer es preparar el modelo que linealizaremos, para ello, reemplazamos las salidas y entradas por bloques *out* e *in* respectivamente. Todos los demás parámetros deben permanecer sin cambios. Por otro lado, si hemos venido trabajando con *vpasolve*, entonces puede ser conveniente ejecutar los siguientes comandos

```
>> vec=[z.S(1),z.R(1),z.C(1)];
>> vec=double(vec);
>> vec

vec =

    0.2430    0.0065    1.0000

>>
```

[3]Click para abrir la documentación

Gracias a ellos los resultados obtenidos con la función *vpasolve* quedarán almacenados en un vector que podemos emplear con facilidad. Note que z almacena 3 familias de soluciones incluida la trivial, por ello es necesario saber qué familia extraer.

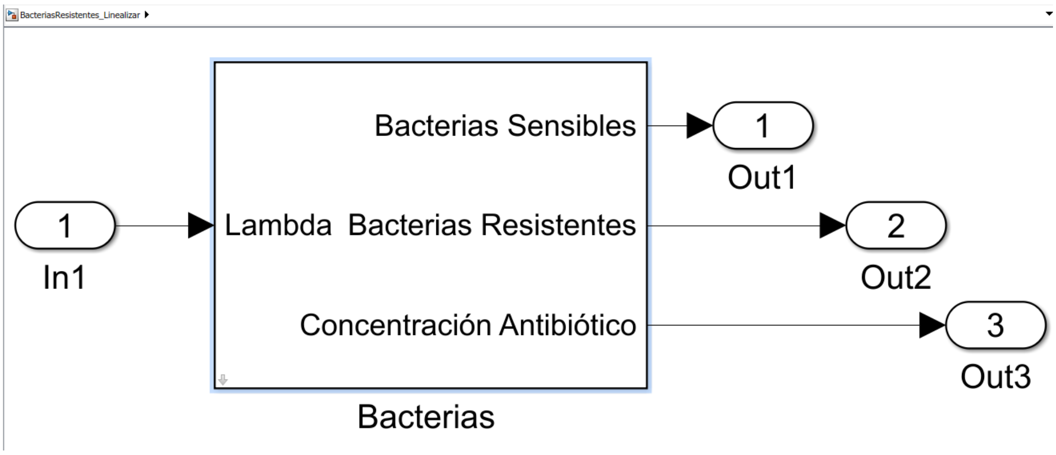


Figura 12: Ejemplo de modelo listo para linealizar, dentro de la máscara "Bacterias" podemos encontrar el subsistema de la Figura 4.

```
>> [A,B,C,D]=linmod('BacteriasResistentes_Linealizar',vec,0.0021)

A =

   -0.0972   -0.0972   -0.0243
   -0.0003   -0.0156    0.0001
         0         0   -0.0083


B =

  -11.5940
    0.0464
         0


C =

     1     0     0
     0     1     0
     0     0     1


D =

     0
     0
```

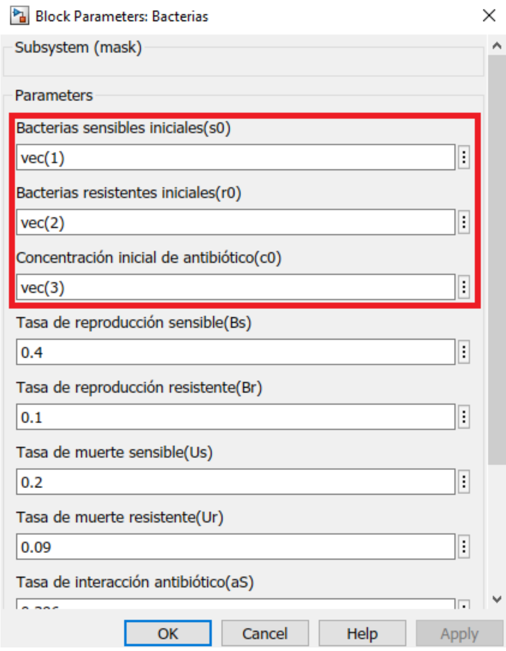
Figura 13: Ejemplo de comando para la linealizacion con linmod. Se aprecia que **C** es una matriz identidad y **D** es un vector de ceros, por ende, las salidas del sistema son los estados $\mathbf{AX} + \mathbf{Bu}$ (recuerde la forma del espacio continuo lineal expuesta en la sección 1.4).

2.4. Comparación entre los modelos no lineal y lineal

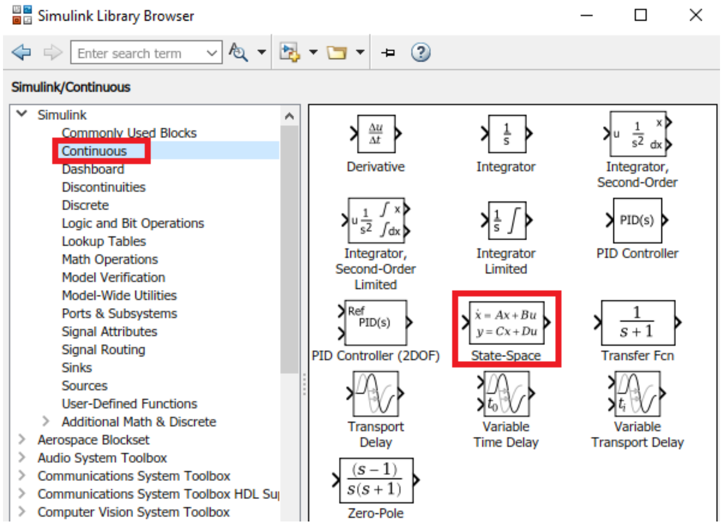
Ahora podemos preparar nuestro modelo de simulink para validar los resultados obtenidos. Para ahorrar trabajo, es recomendable hacer una copia del modelo que empleamos para obtener la curva de linealización y proseguir con las siguientes modificaciones:

1. Cree una máscara del sistema no lineal dejando las entradas y salidas por fuera.

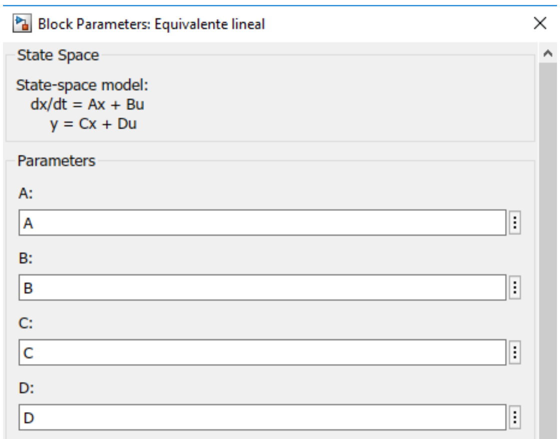
2. Las condiciones iniciales de las salidas del sistema no lineal S, R, C deben corresponder al punto de equilibrio.



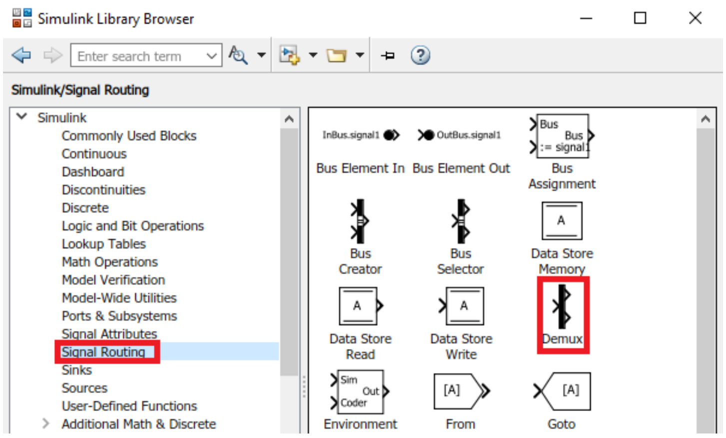
3. Adicione el bloque *State-Space* que hace parte de los bloques continuos de la librería de Simulink.



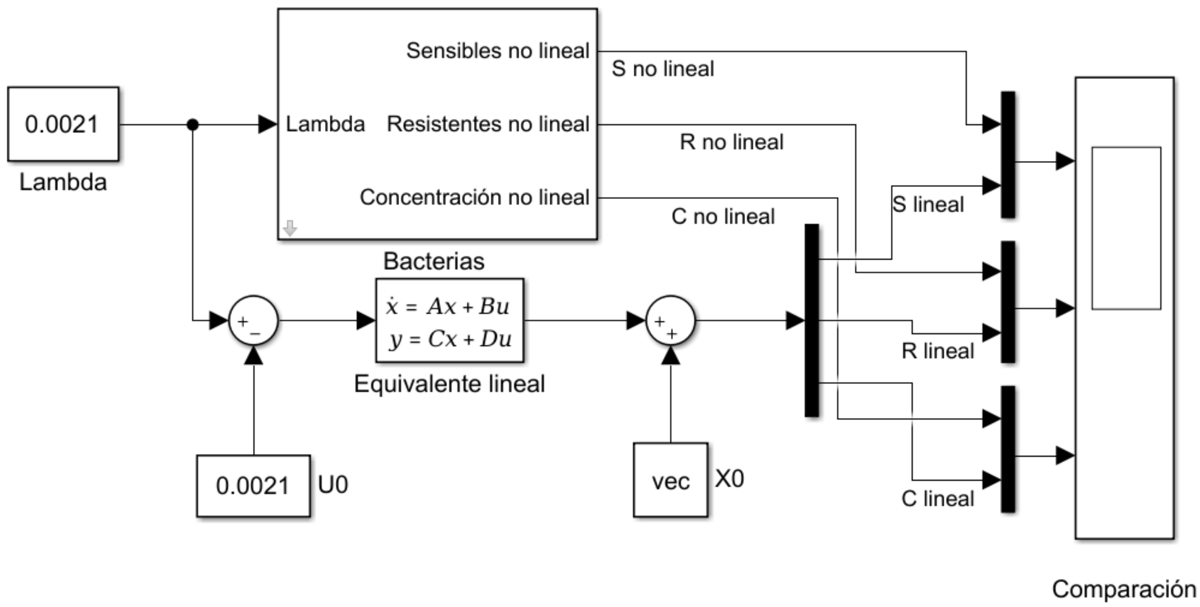
4. Ponga en los parámetros del bloque las matrices obtenidas mediante la función *linmod*.



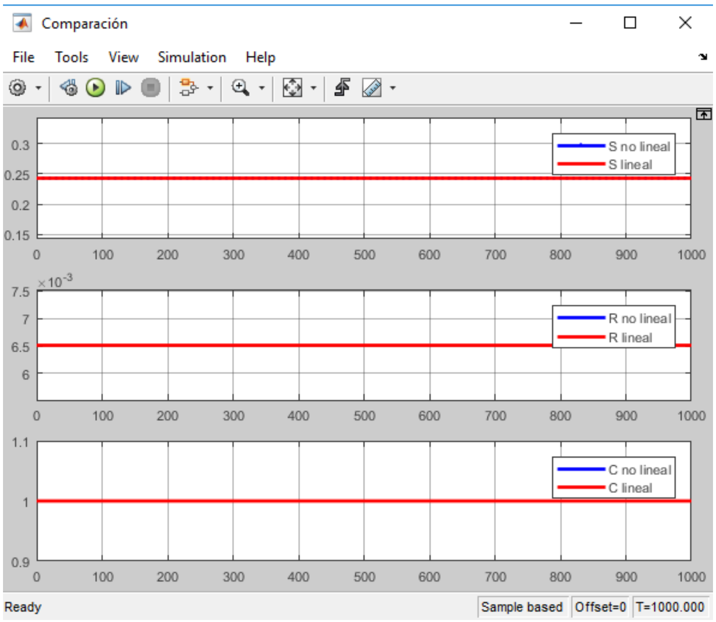
5. El modelo lineal asume que las condiciones del punto de operación representan las condiciones iniciales, siendo así es necesario realizar las siguientes modificaciones a las entradas y salidas:
   - Es necesario restar a la entrada el valor de u (Siendo en este caso Λ) en el punto de operación.
   - Es necesario sumar a la salida el valor de los estados de interés en el punto de operación. En nuestro caso podemos hacerlo a través del vector *vec* ya que nuestras salidas son los estados del sistema.
6. Las salidas del modelo lineal se dan en forma matricial, use el bloque *Demux* de la librería de Simulink para separar las señales.



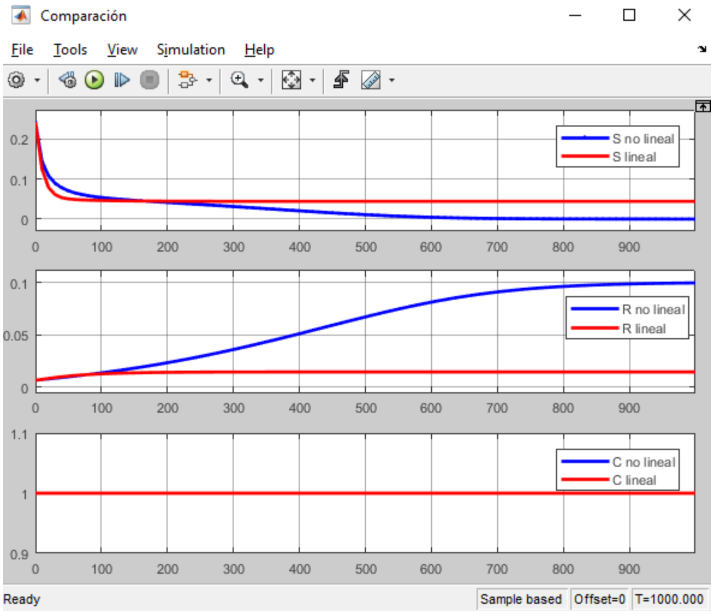
Finalmente debería obtener un diagrama de Simulink de este tipo:



Ahora podemos realizar comparaciones entre ambos modelos modificando los valores de entrada (Λ). Si realizamos una simulación con un valor de $\Lambda = \Lambda_0$ debemos obtener como resultado una línea recta:



A medida que se aleje del punto de operación habrá una diferencia cada vez mayor entre el modelo lineal y el no lineal, por ejemplo, para un valor de $\Lambda = 0{,}0037$ tenemos



Ejercicio

¿Podría explicar por qué la gráfica de C continúa siendo una línea recta?[a]

[a]La respuesta se ha dado en la Figura 11

2.5. Estabilidad y controlabilidad del sistema en el punto de operación

Por la definición de estabilidad para sistemas lineales tratada en la sección 1.4, podemos determinar si nuestro punto de operación es estable o inestable, para ello, calculamos los autovalores de la matriz $\mathbf{A}$ con la función *eig*. Recuerde que el punto será estable sólo si todos los autovalores tienen parte real negativa.

```
>> poles=eig(A)

poles =

   -0.0975
   -0.0153
   -0.0083

fx >>
```

Este resultado es consecuente con todos los resultados que hemos obtenido hasta el momento. **Un sistema inestable** se alejaría del punto de equilibrio ante cualquier cambio en las condiciones iniciales o en la entrada y por ende, habría sido muy difícil o imposible trazar la curva de linealización. Lo anterior no implica que un sistema no lineal con puntos críticos inestables no sea linealizable pero sí sugiere que el rango de linealidad será equivalente al punto crítico.

Ejemplo 2.1: Linealización en un punto crítico inestable

Recuerde del Ejemplo 1.4 que la ecuación del péndulo simple puede darse en la forma de ecuaciones de estado

$$\begin{cases} \dot{x_1} = x_2 \\ \dot{x_2} = \ddot{\theta} = -\dfrac{fx_2}{m} - \dfrac{g \operatorname{sen} x_1}{l} + \dfrac{u}{ml} \end{cases}$$

Siendo

$$\begin{cases} x_1 = \theta \\ x_2 = \dot{\theta} \\ u = F_{ext} \end{cases}$$

Calculando los puntos críticos con $u = 0, m = 3, l = 2, f = 0{,}4$ y $g = 9{,}8$ mediante *vpasolve* nos damos cuenta de que existe una solución trivial $x_1 = 0$, $x_2 = 0$ y una solución no trivial $x_1 = \pi$, $x_2 = 0$. Si decidimos linealizar en la solución no trivial obtenemos la siguiente matriz **A**

$$\begin{pmatrix} 0 & 1 \\ 4{,}9 & -0{,}1333 \end{pmatrix}$$

Cuyos autovalores son 2,1479 y $-2{,}2813$, por lo cual es un punto crítico inestable. Evidentemente $x_1 = \pi$, $x_2 = 0$ nos habla de un péndulo invertido y cualquier perturbación lo desplazará hacia otro punto crítico que sea estable.

En el caso de los puntos críticos inestables, nos llama principalmente la atención la idea de un control. Piense en el Ejemplo 2.1, ¿existe la posibilidad de mantener el sistema en el punto crítico aunque sea inestable a pesar de las perturbaciones? La respuesta es sí y además, puede ser bastante útil, por ejemplo, los scooter eléctricos que se autobalancean pueden entenderse como péndulos invertidos y su correcto funcionamiento depende de mantener al sistema en un punto de equilibrio inestable, de lo contrario ¡el usuario se caería!

Si bien no se han tratado suficientes temas para comprender a fondo el proceso de control, habiendo llegado a este punto de la guía nos encontramos a pocos pasos de poder aplicar un control lineal sobre un sistema no lineal de interés. Nuestro sistema de control será K y se encargará de realimentar la entrada de nuestro sistema para que sea estable o se estabilice con mayor velocidad. Para ello emplearemos la función *place*[4] de la forma $K = place(A, B, [autovaloresdeseados])$. Entre más negativos sean los autovalores deseados, el sistema convergerá con mayor velocidad, aunque para ello se requiere

[4]Click para abrir la documentación

un mayor esfuerzo de control.

Lamentablemente nuestro sistema de bacterias sensibles y resistentes **no es controlable** (Puede comprobarlo al tratar de hallar k). Lo anterior señala que no es posible mantener todos los estados del sistema en determinado punto a través de una realimentación de las entradas (como ya se había mencionado, $\dot{c}$ es insensible a Λ).

Ejemplo 2.2: Control lineal a un sistema no lineal

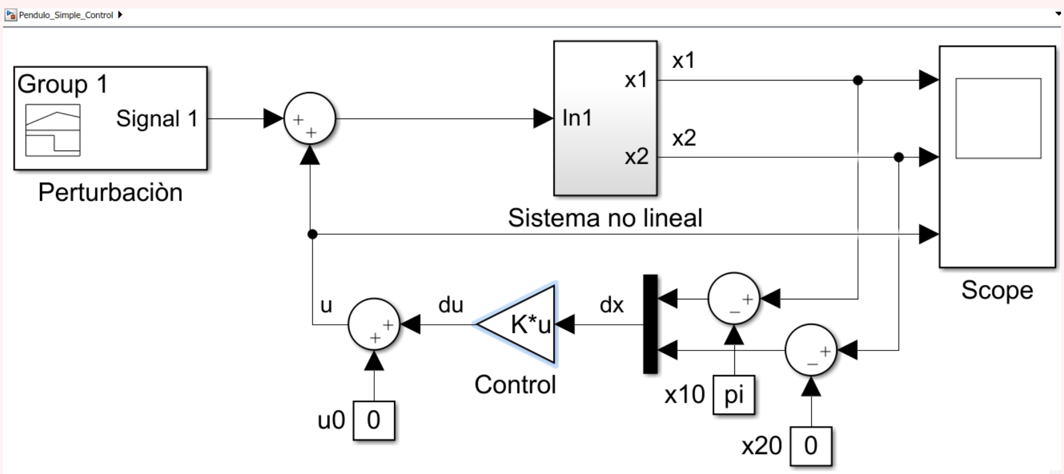
Retomando lo expuesto en el Ejemplo 2.1 podemos diseñar nuestro control K de la siguiente manera siendo $A1$ y $B2$ las matrices **A** y **B** respectivamente

```
>> K=place(A2,B2,[-1,-2])

K =

   41.4000   17.2000

fx >>
```

Ahora debemos preparar el modelo en Simulink para aplicar el control:

- Adicionamos un bloque *Gain* de la librería de Simulink y establecemos su valor en $-K$. Este será nuestro bloque de control.
- La entrada del bloque de control debe ser la diferencia entre las salidas del sistema y las condiciones deseadas.
- A la salida del bloque de control debemos adicionarle el valor de las entradas en el punto de operación.
- La salida resultante de la suma anterior debe ser adicionada a la entrada del sistema.

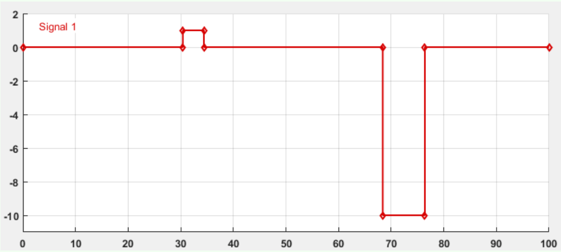
Siguiendo los pasos anteriores debemos obtener un modelo como el siguiente



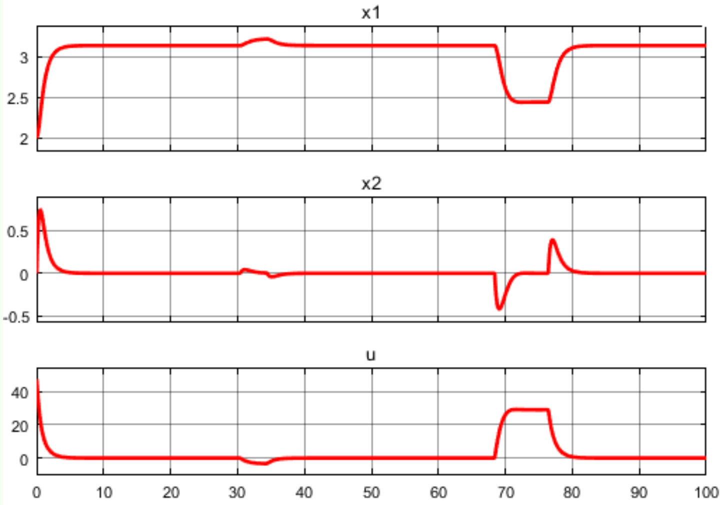
Donde el bloque *Perturbacion* es una señal de entrada aleatoria.

Validación del control para el Ejemplo 2.2

Para la señal de entrada aleatoria

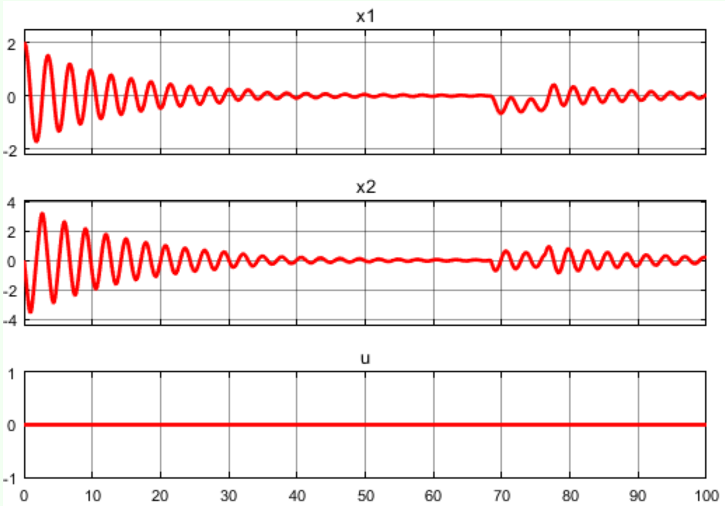


Y las condiciones iniciales $x_1 = 2$, $x_2 = 0$ tenemos las siguientes salidas



Entiéndase la gráfica de u como el esfuerzo de control.

El mismo sistema en ausencia de control, se comportaría de la siguiente manera



3. Linealización con Linear Analysis Tool (Simulink®)

Linear Analysis tool es una herramienta muy útil que podemos emplear para linealizar rápidamente con un conocimiento *a priori*. Podemos emplearla de la siguiente manera:

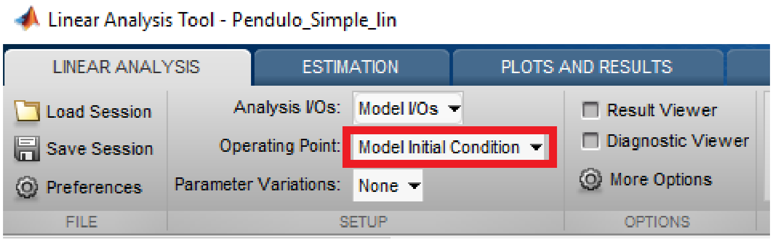
1. Es recomendable simular nuestro modelo con unos parámetros y entradas tales que converja. Para ello puede emplear la función *vpasolve* o simplemente usar un método de tanteo cambiando los valores del diagrama en simulink. El objetivo de lo anterior es ubicar un punto de operación, sin embargo, si lo desea, el Toolbox puede sugerirle puntos de operación.

2. Reemplace las salidas y entradas del sistema con bloques *out* e *in* respectivamente.

3. Damos click derecho sobre las señales de salida y seleccionamos $LinearAnalysisPoints- > Open - loopOutput$, en el caso de las entradas seleccionamos $Open - loopInput$ (Se hace así en caso de que el sistema no tenga realimentación).

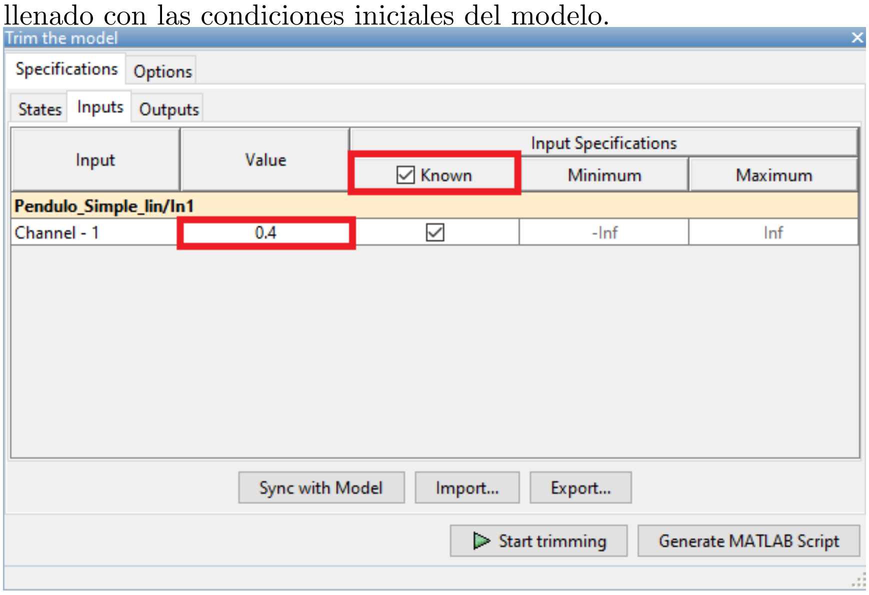
4. En la barra de tareas de Simulink seleccionamos $Analysis- > ControlDesign- > LinearAnalysis$

5. En el menú de $LinearAnalysis$ Hacemos click en el checkbox de $OperatingPoint$.

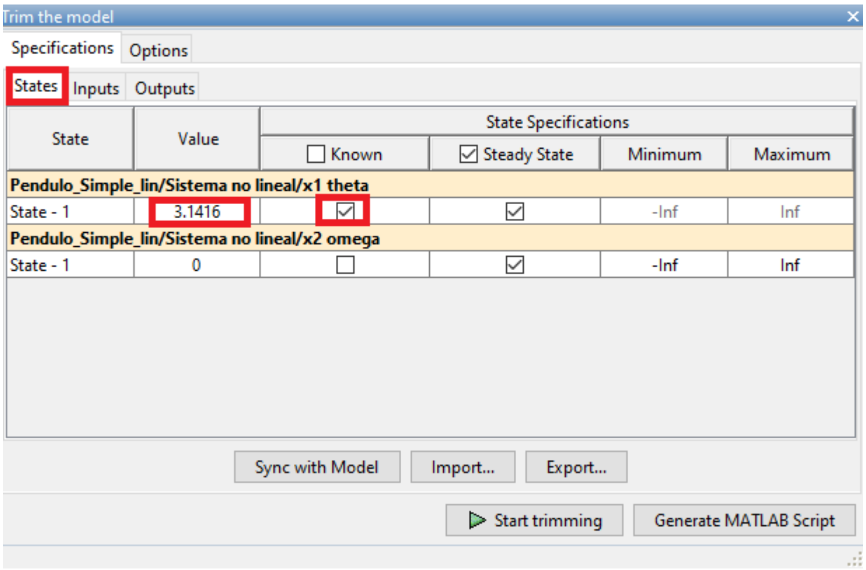


6. Procedemos dependiendo de la forma como queramos definir nuestro punto de operación:

   - En caso de que ya se tenga claro un punto de operación, entonces la entrada del sistema debe ser un bloque constante con el valor del punto de operación. Tenga en cuenta que la señal de entrada debe estar reportada mediante $Open - loopInput$. También es necesario que el modelo a linealizar tenga como condiciones iniciales a los valores de convergencia en el punto de operación. Si lo anterior se cumple puede saltar al siguiente numeral.
   - En caso de que quiera elegir un punto de operación pero aún no sepa los valores a los que convergen los estados, elija la opción $TrimModel$, en la pestaña de $inputs$ proporcione el valor del punto de operación que desea y seleccione $Known$. Las demás pestañas se habrán llenado con las condiciones iniciales del modelo.

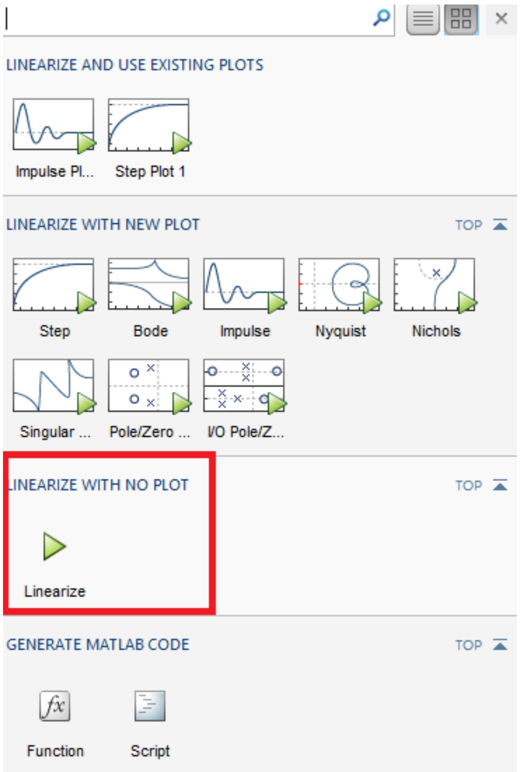
     

     Seleccione $Starttrimming$ para obtener las condiciones del punto de operación. Dichas condiciones se almacenarán en el Workspace del Toolbox y se seleccionarán automáticamente para iniciar el proceso de linealización.
   - En caso de que no tenga seleccionado un punto de operación, entonces puede jugar con las opciones de $TrimModel$. Por ejemplo, si desea que su punto de operación esté dado por el valor de una de las salidas, entonces puede especificarlo en la pestaña de $States$



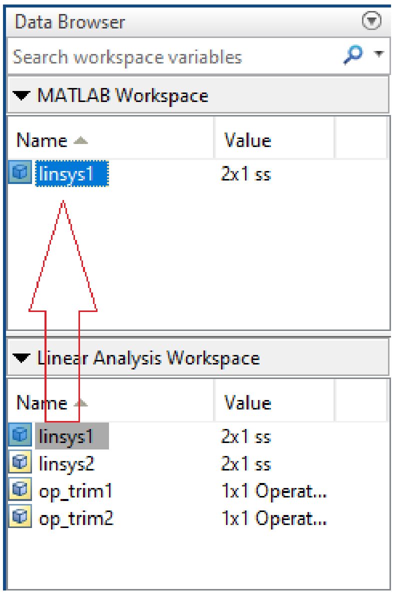
- Finalmente, si no tiene ninguna preferencia por algún punto de operación, ejecute inmediatamente $TrimModel$.

7. Ahora linealizamos nuestro modelo con alguna de las siguientes opciones



   Si escoge la opción del recuadro rojo, el modelo simplemente se linealizará. Las otras opciones permiten de antemano saber cuál será la respuesta del modelo lineal con respecto a cierto tipo de entradas en el rango de linealidad.

8. Los resultados del proceso de linealización se guardan en el *Workspace* del toolbox bajo los nombres de linsys. Si da doble click sobre ellos puede apreciar los valores de las matrices **A,B,C,D**. Tambien puede arrastrar los resultados al *Workspace* de Matlab para trabajar con ellos posteriormente.



4. Resumen: Pasos para linealizar

Los numerales de esta sección son hipervínculos, utilícelos para recordar cómo llevar a cabo el proceso de linealización.

1. Escriba las ecuaciones de su modelo e identifique si es no lineal.
2. Escriba las ecuaciones en forma de variables de estado.
3. Obtenga los puntos de equilibrio (o puntos críticos).
   a) Por simulación, a partir del valor en que se estabilizan los estados para una entrada determinada
   b) Matemáticamente, igualando a cero las ecuaciones.
   c) Numéricamente, con funciones como vpasolve de Matlab.
4. Si es posible, clasifique el tipo de cada punto de equilibrio.
5. Seleccione el punto de operación en el cual se va a linealizar el modelo.
6. Linealización del modelo no lineal
   a) Analíticamente, utilizando el Jacobiano.
   b) Numéricamente, a partir de Simulink y la función linmod.
   c) Numéricamente, a partir de otras herramientas.
7. Comparar en simulación la respuesta lineal con la respuesta no lineal cerca al punto de operación.
8. Análisis del proceso y conclusiones.

Referencias

[Esteva and Leyton, 2011] Esteva, L. and Leyton, J. R. (2011). Un modelo matemático sobre bacterias sensibles y resistentes a antibióticos. XIX(2):55–73.

[Zill and Wright, 2015] Zill, D. G. and Wright, W. S. (2015). *Ecuaciones Diferenciales con probleas con valor en la frontera.*