



GRADO EN INGENIERÍA EN TECNOLOGÍA  
INDUSTRIAL

# TRABAJO FIN DE GRADO

*PLATAFORMA EDUCACIONAL DE  
AUTOMÁTICA Y CONTROL*

**Alumno:** Lapuente de la Peña, Carlos

**Directora:** Cabanes Axpe, Itziar

**Curso:** 2018-2019

**Fecha:** 08/07/2019

## **Resumen**

El presente documento describe el desarrollo de una herramienta opensource programada con Mathematica® y pensada para dar soporte a alumnos de asignaturas donde se trate el análisis y el diseño de sistemas de control. La herramienta muestra el análisis temporal, la ubicación de los polos y ceros y el análisis frecuencial de un sistema realimentado definido por sus funciones de transferencia. La herramienta es capaz de facilitar la respuesta temporal, el lugar de las raíces y el diagrama de Bode del sistema, así como parámetros de su dinámica y estabilidad: tiempo de establecimiento; valores de la salida y el error en régimen estacionario; ubicación de polos y ceros; margen de ganancia; margen de fase; y frecuencias de ruptura. Está concebida para que no requiera de conocimiento previo para ser utilizada ya que cuenta con una interfaz clara y controles intuitivos.

## **Laburpena**

Dokumentu honek deskribatzen du Mathematica®-rekin programatutako opensource tresna baten garapen-prozesua. Tresna honen helburua zera da: ikasleei lagungarria izatea kontrol-sistemen analisisa eta diseinua jorratzen den ikasgaietan. Horretarako, sistema berrelikatu baten denbora-analisisa, poloen eta zeroen kokapena eta maiztasun-analisisa erakusteko gai da, sistemaren transferentzia-funtzioak jakinda. Halaber, denbora-erantzuna, erroen kokapena eta sistemaren Boderen diagrama aurkezten ditu, baita bere dinamika eta egonkortasunaren parametroak ere: ezarpen-denbora, irteera-balioak eta erregimen geldikorraren errorea, poloen eta zeroen kokapena, irabazi-marjina, fase-marjina eta haustura-maiztasunak. Tresna honek interfaze argia eta kontrol intuitiboak dauzka, edonork aurretiko esperientziarik gabe erraz erabili ahal izateko.

## **Abstract**

This document describes the development of an opensource tool programmed with Mathematica® and designed to support students in subjects dealing with the analysis and design of control systems. This tool shows the temporal analysis, the location of poles and zeros and the frequency analysis of a feedback system using its transfer functions. The program is capable of facilitating the temporal response, the root locus and the Bode diagram of the system, as well as its dynamics and stability parameters: settling time; steady-state values of output and error; location of poles and zeros; gain margin; phase margin; and cutoff frequencies. It is designed so that it does not require prior knowledge to be used, since it has a clear interface and intuitive controls.

### **Palabras clave**

herramienta software, Mathematica®, Automática y Control, análisis temporal, análisis frecuencial, lugar de las raíces

### **Hitz gakoak**

software tresna, Mathematica®, Automatika eta Kontrola, denbora-analisia, maiztasun-analisia, erroen kokapena

### **Key Words**

software tool, Mathematica®, Automatics and Control, temporal analysis, frequency analysis, root locus

## **Contenido**

|  |    |
|--|----|
| 1. Introducción .....  | 9  |
| 2. Contexto .....  | 10 |
| 3. Objetivos y alcance del proyecto .....                      | 11 |
| 3.1. Marco teórico.....  | 11 |
| 3.1.1 Conceptos básicos de sistemas realimentados .....        | 11 |
| 3.1.2 Análisis temporal .....                                  | 14 |
| 3.1.3 Análisis de la ubicación de los polos y ceros .....      | 14 |
| 3.1.4 Análisis frecuencial.....                                | 15 |
| 3.2. Objetivos .....   | 16 |
| 3.3. Alcance.....  | 16 |
| 4. Beneficios del proyecto.....                                | 17 |
| 5. Descripción de requerimientos.....                          | 18 |
| 5.1. Requerimientos de la interfaz y la entrada de datos ..... | 18 |
| 5.2. Requerimientos del análisis temporal .....                | 18 |
| 5.3. Requerimientos del lugar de las raíces .....              | 19 |
| 5.4. Requerimientos del análisis frecuencial .....             | 19 |
| 6. Análisis de alternativas .....                              | 20 |
| 6.1. Labview®.....   | 20 |
| 6.2. Mathematica® .....  | 20 |
| 6.3. Matlab®.....  | 21 |
| 6.4. Selección de la solución .....                            | 22 |
| 7. Descripción Mathematica® .....                              | 24 |
| 7.1. Introducción a Mathematica® .....                         | 24 |
| 7.2. Funciones relevantes Mathematica®.....                    | 26 |
| 7.2.1 Funciones de comparación y ciclos repetitivos .....      | 26 |
| 7.2.2 Funciones de manipulación de listas y expresiones.....   | 27 |
| 7.2.3 Funciones matemáticas.....                               | 27 |
| 7.2.4 Funciones para crear gráficas .....                      | 27 |
| 7.2.5 Funciones para crear controles .....                     | 28 |
| 7.2.6 Funciones para organizar interfaces.....                 | 29 |
| 8. Descripción de tareas .....                                 | 30 |
| 8.1. Paquetes de trabajo.....                                  | 30 |
| 8.2. Hitos .....   | 32 |
| 9. Algoritmos.....   | 33 |

|        |   |     |
|--------|---|-----|
| 9.1.   | Algoritmos del análisis temporal .....                                      | 33  |
| 9.1.1  | Cálculo de la salida, referencia y error.....                               | 33  |
| 9.1.2  | Cálculo del rango de la valores de las gráficas .....                       | 33  |
| 9.2.   | Algoritmos del análisis de la ubicación de los polos y ceros .....          | 35  |
| 9.2.1  | Cálculo de los polos en bucle abierto, polos en bucle cerrado y ceros... 35 |     |
| 9.2.2  | Cálculo del rango de la valores de la gráfica .....                         | 36  |
| 9.3.   | Algoritmos del análisis frecuencial .....                                   | 37  |
| 9.3.1  | Cálculo de la expresión del módulo .....                                    | 37  |
| 9.3.2  | Cálculo de la expresión de la fase .....                                    | 38  |
| 9.3.3  | Cálculo de la expresión del módulo asintótico .....                         | 38  |
| 9.3.4  | Cálculo del margen de ganancia y margen de fase .....                       | 41  |
| 9.3.5  | Cálculo del rango de la valores de las gráficas .....                       | 42  |
| 10.    | Descripción de los resultados.....  | 43  |
| 10.1.  | Resultados de la interfaz y la entrada de datos .....                       | 43  |
| 10.1.1 | Zona del sistema y entrada de datos .....                                   | 44  |
| 10.1.2 | Zona de los análisis .....  | 46  |
| 10.2.  | Resultados de los análisis.....   | 49  |
| 10.2.1 | Resultados del análisis temporal .....                                      | 49  |
| 10.2.2 | Resultados del análisis de polos y ceros .....                              | 51  |
| 10.2.3 | Resultados del análisis frecuencial .....                                   | 52  |
| 10.3.  | Ejemplo de aplicación .....   | 54  |
| 11.    | Descargo de gastos .....  | 56  |
| 12.    | Conclusiones.....   | 58  |
| 13.    | Bibliografía .....  | 59  |
| 14.    | Anexo I: Código.....  | 60  |
| 15.    | Anexo II: Manual de usuario.....  | 100 |

## Índice de figuras

|  |    |
|--|----|
| Figura 1. Diagrama de bloques de un sistema de control .....   | 12 |
| Figura 2. Izquierda: Salida de un sistema de segundo orden (azul) ante una referencia escalón (naranja). Derecha: Señal de error del mismo sistema (verde) .....               | 14 |
| Figura 3. Lugar de las raíces de un sistema de 2º orden .....  | 15 |
| Figura 4. Diagrama de Bode de un sistema de segundo orden: módulo y argumento. 15  |    |
| Figura 5. Logo de Labview® .....   | 20 |
| Figura 6. Logo de Mathematica®.....  | 21 |
| Figura 7. Logo de Matlab® .....  | 22 |
| Figura 8. Ejemplo de cuaderno de Mathematica® .....  | 24 |
| Figura 9. Ejemplos definición de funciones .....   | 26 |
| Figura 10. Ejemplo de varias gráficas combinadas .....   | 28 |
| Figura 11. Ejemplo de controles .....  | 29 |
| Figura 12. Diagrama de Gantt del proyecto .....  | 32 |
| Figura 13. Cálculo de las funciones del análisis temporal .....  | 33 |
| Figura 14. Cálculo del rango de la gráfica del análisis temporal.....  | 35 |
| Figura 15. Cálculo de los polos y ceros .....  | 36 |
| Figura 16. Cálculo del rango de la gráfica del análisis de la ubicación de polos y ceros .....   | 37 |
| Figura 17. Cálculo de la expresión del módulo del análisis frecuencial .....   | 37 |
| Figura 18. Cálculo de la expresión de la fase del análisis frecuencial .....   | 38 |
| Figura 19. Cálculo de la expresión del módulo asintótico del análisis frecuencial .....  | 40 |
| Figura 20. Cálculo del margen de ganancia y margen de fase.....  | 42 |
| Figura 21. Cálculo del rango de las gráficas del análisis frecuencial .....  | 42 |
| Figura 22. Interfaz de la herramienta software .....   | 43 |
| Figura 23. Tres ejemplos de texto explicativo .....  | 44 |
| Figura 24. Expresiones que cuentan con mensaje explicativo .....   | 44 |
| Figura 25. Zona del sistema.....   | 45 |
| Figura 26. Controles de las funciones de transferencia.....  | 45 |
| Figura 27. Controles de la ganancia del controlador .....  | 45 |
| Figura 28. Controles de la referencia .....  | 46 |
| Figura 29. Zona de los análisis .....  | 47 |
| Figura 30. Botones para mostrar y ocultar los análisis .....   | 47 |
| Figura 31. Campos de entrada de los ejes de los análisis .....   | 47 |
| Figura 32. Botón para reajustar los ejes de las gráficas de los análisis.....  | 48 |
| Figura 33. Ejemplo de zoom en una gráfica .....  | 48 |
| Figura 34. Cuadro de ayuda de la herramienta software.....   | 49 |
| Figura 35. Análisis temporal.....  | 50 |
| Figura 36. Botones para mostrar y ocultar las cuadrículas de las gráficas del análisis temporal.....   | 50 |
| Figura 37. Análisis de la ubicación de polos y ceros .....   | 51 |
| Figura 38. Ejemplo de discontinuidad del lugar de las raíces.....  | 52 |
| Figura 39. Análisis frecuencial .....  | 53 |
| Figura 40. Izquierda: Botones para mostrar y ocultar las cuadrículas de las gráficas del análisis frecuencial. Derecha: Botón para mostrar y ocultar el módulo asintótico..... | 54 |

Figura 41. Sistema realimentado utilizado en el ejemplo de aplicación ..... 54  
Figura 42. Análisis temporal con las ganancias del controlador mínima y máxima ..... 55

## **Índice de tablas**

|  |    |
|--|----|
| Tabla 1. Comparativa de los entornos de desarrollo considerados .....        | 23 |
| Tabla 2. Horas internas del proyecto .....                                   | 56 |
| Tabla 3. Amortizaciones del proyecto .....                                   | 56 |
| Tabla 4. Honorarios de los ingenieros y costes indirectos del proyecto ..... | 56 |
| Tabla 5. Descargo de gastos del proyecto .....                               | 57 |

# 1. Introducción

En este documento se presenta el desarrollo de una herramienta opensource programada con Mathematica® y pensada para dar soporte a alumnos de asignaturas donde se trate el Análisis y Diseño de Sistemas de Control.

La herramienta es capaz de facilitar el análisis temporal (respuesta temporal), la ubicación de los polos y ceros (denominado habitualmente como lugar de las raíces) y el análisis frecuencial (también denominado diagrama de Bode) de un sistema realimentado definido por sus funciones de transferencia. Estas funciones de transferencia representan la dinámica de cualquier sistema físico, y serán por tanto las entradas en esta herramienta de análisis y diseño de los sistemas de control.

La estructura de esta memoria es la siguiente. En primer lugar, se expone el contexto en el que surge el proyecto para posteriormente detallar los objetivos y alcance del mismo.

A continuación, se describen los beneficios que aporta el proyecto, así como los requerimientos que debe tener la herramienta software.

Seguidamente, se plantea un análisis de las diferentes alternativas para desarrollar la herramienta y se realiza la descripción del entorno seleccionado para ello.

En cuanto a la planificación, se describen las tareas llevadas a cabo en el desarrollo del proyecto.

Posteriormente, se describen los algoritmos empleados en la herramienta y los resultados obtenidos tras su desarrollo.

En el aspecto económico, se incluye el descargo de gastos del proyecto.

Finalmente, en el último apartado, se resaltan las principales conclusiones obtenidas a lo largo del proyecto.

## 2. Contexto

Este proyecto surge como un proyecto personal al cursar la asignatura “Automática y Control” del grado en Ingeniería en Tecnología Industrial de la Escuela de Ingeniería de Bilbao (EIB).

Durante ese periodo lectivo, el profesorado recomendó distintos programas para uso personal que servían de apoyo en el estudio de la asignatura. Estos programas resultaban útiles pero ofrecían resultados limitados y no permitían visualizar al mismo tiempo las diferentes acciones o análisis (es decir, introducir una sola vez el sistema a analizar y poder visualizar al mismo tiempo su respuesta temporal, la ubicación de sus polos y ceros y su diagrama de Bode). Así, por ejemplo, un software mostraba la respuesta temporal de sistemas realimentados de primer y segundo orden exclusivamente [1], mientras que otro mostraba los resultados de sistemas físicos concretos. [2]

Tras ver el potencial de algunos de esos programas, se decidió utilizarlos como punto de partida para crear una herramienta que aunase todos los potenciales y que diese apoyo a los alumnos de asignaturas relacionadas con la Automática y el Control, concretamente en la etapa de Análisis y Diseño de Sistemas de Control.

## **3. Objetivos y alcance del proyecto**

En este apartado se presentan los objetivos y el alcance del proyecto. Por tanto, en primer lugar se expone el marco teórico en el que se desarrolla el proyecto, que sirve para entender las características que debe tener la herramienta software a desarrollar. A continuación, se detallan los objetivos del proyecto y, finalmente, se expone el alcance del mismo.

### **3.1. Marco teórico**

En este apartado se presenta el marco teórico en el que se desarrolla el proyecto, el cual pretende aportar una mejor comprensión de las especificaciones que debe cumplir la herramienta, así como el contenido y la interfaz final de la misma. Para ello, se van a sintetizar conceptos relevantes de la teoría de Análisis de Sistemas Realimentados y de Diseño de Controladores.

Por tanto, este apartado comienza explicando conceptos básicos de sistemas realimentados, describiendo la estructura de un sistema de control y sus principales características. A continuación, se explica en qué consisten el análisis temporal, el análisis de la ubicación de polos y ceros y el análisis frecuencial de un sistema realimentado.

#### **3.1.1 Conceptos básicos de sistemas realimentados**

Los sistemas de control están muy presentes en nuestra vida diaria, ejemplos claros se pueden encontrar en el hogar (control de temperatura, sistema anti-robo o del teléfono móvil), en los sistemas de transporte (como el ABS o control de tracción de un automóvil o el control de crucero de un avión), en la industria (farmacéutica, máquina herramienta o de proceso) o en el control de tráfico de internet. Estos son solo algunos ejemplos ya que también juegan un papel importante en campos tan diferentes como la economía, la biología o la medicina.

En síntesis, los sistemas de control tienen como objetivo conseguir que un sistema (máquina, proceso o dispositivo) se comporte de una forma determinada automática, con la mínima intervención humana.

Estos sistemas de control pueden ser esencialmente de dos tipos: en bucle abierto o en bucle cerrado.

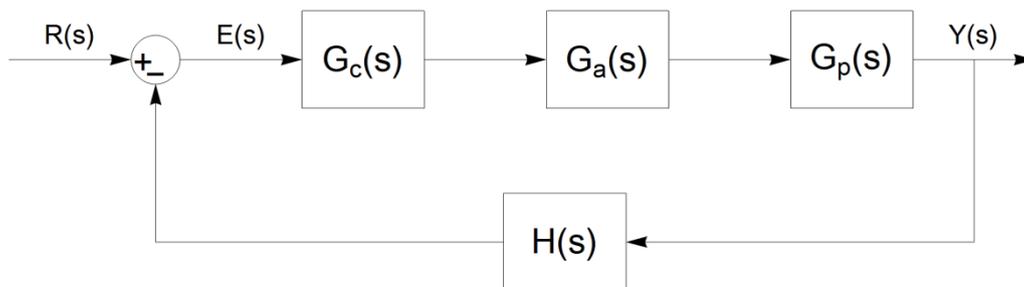
Los sistemas de control en bucle abierto son sistemas en los cuales no se mide la salida ni se realimenta para compararla con la entrada, por lo que la salida no tiene efecto sobre la acción de control. Así, a cada entrada de referencia le corresponde una condición de operación fija, esto es, una salida que ha sido previamente establecida. Como resultado de ello, la precisión del sistema depende de la calibración y de la influencia de las perturbaciones internas y externas. Por lo tanto, el control en bucle abierto solo se usa si se conoce la relación entre la entrada y la salida y si no hay perturbaciones.

Los sistemas de control en bucle cerrado se denominan también sistemas de control realimentados. En un sistema de control en bucle cerrado, se alimenta al controlador la señal de error de actuación, que es la diferencia entre la señal de entrada y la señal de realimentación (que puede ser la propia señal de salida o una función de la señal de salida y sus derivadas y/o integrales), con el fin de reducir el error y llevar la salida del sistema a un valor deseado.

Por lo tanto, en los sistemas de control realimentados se mide el comportamiento del sistema para corregirlo si no es el deseado. Cabe señalar que este tipo de procesos no son relativos exclusivamente a la tecnología desarrollada por el ser humano, sino que la realimentación es una característica de la vida. Todos los organismos comparten la habilidad de medir cómo están y realizar cambios si es necesario. La realimentación gobierna cómo crecen; cómo responden al estrés; o cómo regulan la temperatura corporal, la presión sanguínea o el nivel de colesterol. Es decir, el control no solo hace la vida más cómoda, sino que es esencial para la existencia de la misma.

Los sistemas de control realimentados típicamente están formados por cuatro tipos de elementos: sensores (para medir), computadores (para calcular y tomar decisiones), actuadores (para modificar) y el propio sistema o proceso. De esta forma, estos sistemas se pueden modelizar mediante un diagrama de bloques en el que cada uno de los elementos mencionados tiene una función de transferencia asociada.

En la Figura 1, se muestra el diagrama de bloques de un sistema de control realimentado típico. En ella se pueden ver las funciones de transferencia de los elementos del sistema:  $G_c(s)$  la del controlador,  $G_a(s)$  la del actuador,  $G_p(s)$  la del proceso y  $H(s)$  la del sensor, así como las señales o funciones más importantes del sistema:  $R(s)$ , la entrada o referencia del sistema;  $Y(s)$ , la salida del sistema; y  $E(s)$ , el error del sistema. [3]



**Figura 1. Diagrama de bloques de un sistema de control**

En un sistema de control lineal e invariante en el tiempo, las funciones de transferencia de los elementos del sistema y las funciones de las señales (salida, referencia, error, etc.) son cocientes de polinomios en  $s$ .

Las funciones de transferencia se obtienen al aplicar la herramienta matemática Transformada de Laplace a las ecuaciones integro-diferenciales obtenidas de aplicar las leyes físicas que gobiernan el proceso que se quiere controlar, como en el caso del proceso, o son especialmente diseñadas para lograr que las condiciones de la salida sean las deseadas, como en el caso del controlador.

Concretamente, el tipo de controlador más empleado en la industria es el PID (Proporcional, Integral y Derivativo). La función de transferencia de estos controladores tiene la siguiente estructura [3]:

$$G_c(s) = K_c \left( 1 + \frac{1}{T_i s} + T_d s \right)$$

donde  $K_c$  es la ganancia del controlador,  $T_i$  es el tiempo integral y  $T_d$  es el tiempo derivativo.

En los sistemas de control realimentados se define la función de transferencia en bucle cerrado ( $G_{BC}(s)$ ) como el cociente entre la salida y la referencia del sistema:

$$G_{BC}(s) = \frac{Y(s)}{R(s)} = \frac{G_c(s) G_a(s) G_p(s)}{1 + G_c(s) G_a(s) G_p(s)}$$

Aunque el sistema sea realimentado, es decir, en bucle cerrado, también se puede definir su función de transferencia en bucle abierto ( $G_{BA}(s)$ ) como el producto de las funciones de transferencia de sus elementos:

$$G_{BA}(s) = G_c(s) G_a(s) G_p(s) H(s)$$

Por otro lado, el error se define como la diferencia entre la señal de referencia o consigna  $R(s)$  (que impone el comportamiento deseado de un sistema) y la salida  $Y(s)$  (lo que realmente ocurre o el comportamiento real del sistema) después de pasar por el sensor  $H(s)$  (elemento encargado de medir dicha señal en función de la magnitud a medir):

$$E(s) = R(s) - Y(s) \cdot H(s)$$

A las raíces del polinomio del numerador de la función de transferencia en bucle cerrado se las conoce como ceros del sistema y a las del denominador, polos del sistema. Se puede demostrar que en un sistema realimentado, la ganancia del controlador afecta a la ubicación de los polos [3]. Y es la ubicación de los polos y ceros del sistema lo que impone su dinámica y estabilidad, por lo que se han desarrollado diferentes análisis que permiten estudiar estas características.

Cabe mencionar aquí lo que se entiende por estabilidad de un sistema. Existen diferentes definiciones de estabilidad como la de estabilidad BIBO (Bounde Input Bounde Output) y la de estabilidad asintótica, entre otras. Se dice que un sistema es estable BIBO si para una entrada acotada, su salida también es acotada. Por otro lado, un sistema es asintóticamente estable si para una entrada acotada la salida converge a un valor finito.

Los análisis que permiten estudiar la dinámica y estabilidad de un sistema son, entre otros, el análisis de la respuesta temporal (análisis temporal), el lugar de las raíces (análisis de la ubicación de los polos y ceros del sistema) y el diagrama de Bode (análisis temporal).

A continuación, se explica brevemente cada uno de estos análisis.

### 3.1.2 Análisis temporal

El análisis temporal de un sistema consiste en analizar cómo es su respuesta (salida) en el dominio del tiempo cuando a dicho sistema se le excita con una entrada (referencia). Para ello se miden y determinan algunas características de la salida del sistema (tiempo de establecimiento, tiempo de subida, sobreimpulso, etc.) cuando la referencia es una de las entradas patrón, como puede ser una entrada escalón, rampa o parábola. También resulta importante conocer el error del sistema. Se emplea para analizar la dinámica y la estabilidad del sistema.

Para realizar el análisis temporal de un sistema realimentado se representan la salida y la referencia a lo largo del tiempo en una misma gráfica y, en otra gráfica, el error a lo largo del tiempo. Además, se necesitan conocer los valores de parámetros relevantes de la dinámica del sistema: el tiempo de establecimiento (que es el tiempo que tarda la salida en alcanzar un determinado porcentaje de su valor final), el tiempo de subida, el sobreimpulso, etc. Estos parámetros pueden ser calculados teóricamente o medidos directamente en las gráficas.

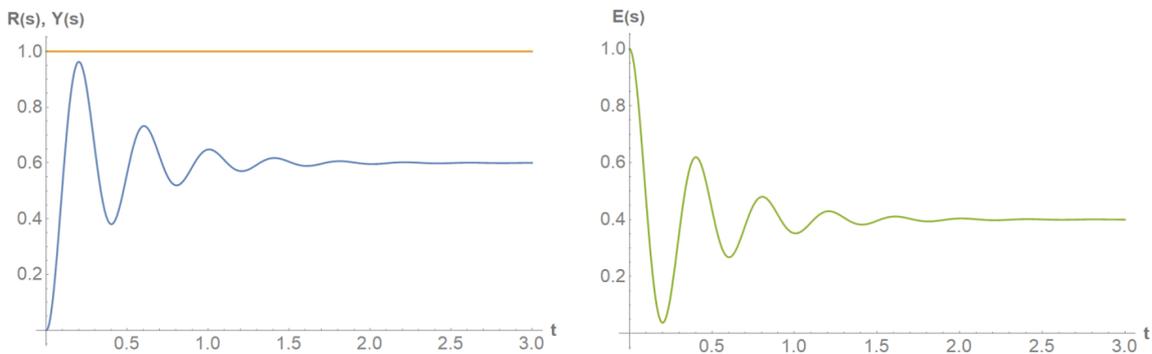


Figura 2. Izquierda: Salida de un sistema de segundo orden (azul) ante una referencia escalón (naranja). Derecha: Señal de error del mismo sistema (verde)

### 3.1.3 Análisis de la ubicación de los polos y ceros

Cuando se realiza el análisis de la ubicación de los polos y ceros de un sistema realimentado, se estudia dónde se encuentran los polos del sistema realimentado en el plano complejo  $s$ . Gracias a este análisis, se obtiene información acerca de cómo será la salida del sistema, si será estable o no, y para qué ganancia el controlador diseñado conseguirá obtener las características deseadas. Por tanto, se emplea para diseñar controladores y analizar la estabilidad del sistema.

Para realizar este análisis se construye la gráfica que muestra la trayectoria de los polos del sistema en bucle cerrado, es decir, cuando varía la ganancia del controlador, denominada comúnmente como "lugar de las raíces". En esta gráfica también se representa la ubicación de los polos y ceros del sistema.

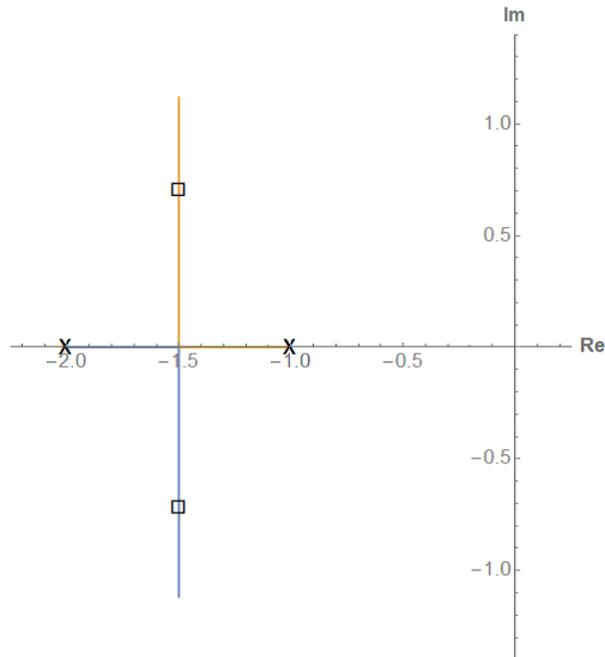


Figura 3. Lugar de las raíces de un sistema de 2º orden

### 3.1.4 Análisis frecuencial

Por último, el análisis frecuencial de un sistema consiste en determinar la amplificación y el desfase de la salida del sistema en bucle abierto cuando la entrada es una onda sinusoidal. Se emplea para conocer el comportamiento del sistema en un rango de frecuencias y analizar su estabilidad.

Para realizar el análisis frecuencial de un sistema se emplea el diagrama de Bode, que consta de dos gráficas. En la primera, se muestra el módulo de la función de transferencia en bucle abierto (expresada en decibelios) y en la otra se muestra su fase (expresada en grados). Ambas gráficas se representan con la frecuencia de la entrada en las abscisas (expresada en radianes/segundo). Además, resulta útil conocer los valores del margen de ganancia y el margen de fase, dos parámetros que dan información sobre la estabilidad del sistema; así como las frecuencias de ruptura del sistema.

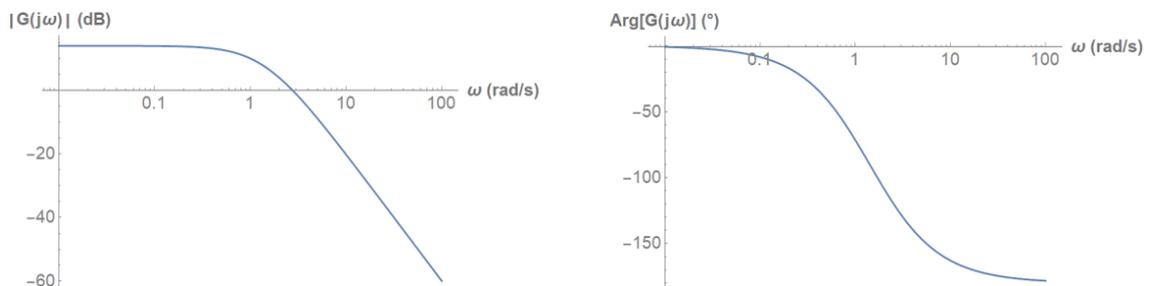


Figura 4. Diagrama de Bode de un sistema de segundo orden: módulo y argumento

### 3.2. Objetivos

El objetivo principal del proyecto es desarrollar una herramienta gratuita, opensource, y sencilla de usar que dé soporte a alumnos que cursen asignaturas relacionadas con la Automática y el Control, facilitando la etapa de análisis y diseño de sistemas de control.

Para lograr que la herramienta sea sencilla de usar y útil, deberá:

- Tener una interfaz clara con controles intuitivos.
- Facilitar simultáneamente tres análisis de un sistema realimentado: 1) el análisis temporal ante diferentes entradas patrón (escalón, rampa y parábola), 2) la evolución de los polos y ceros en bucle cerrado en función de la ganancia del controlador y, por último, 3) el diagrama de Bode. Para ello, se requerirán las funciones de transferencia de los elementos del sistema.
- Hacer estos análisis dinámicos, es decir, hacer que se actualicen de forma automática y simultánea (hasta donde lo permita la velocidad de computación) cuando se modifiquen los datos de entrada.
- Permitir visualizar simultáneamente el sistema y sus análisis, facilitando parámetros relevantes de la dinámica y estabilidad del sistema como el tiempo de establecimiento, el error en estado estacionario, los márgenes de ganancia y de fase, etc.

### 3.3. Alcance

La herramienta software a desarrollar será capaz de analizar exclusivamente sistemas realimentados en tiempo continuo que consten de alguno o todos de los siguientes elementos: controlador, actuador, planta y sensor.

Es alcance del proyecto modelizar sistemas físicos, pero sus datos de entrada siempre se deberán introducir como funciones de transferencia, no en otro formato.

No es alcance de este proyecto que la herramienta permita analizar sistemas en bucle abierto, sistemas en tiempo discreto ni sistemas con perturbaciones.

Por otro lado, la respuesta temporal del sistema solo podrá ser a una entrada escalón, rampa o parábola que comience en el instante inicial. La amplitud de esta entrada se podrá modificar pero no podrá ser variable a lo largo del tiempo.

## 4. Beneficios del proyecto

El principal beneficio del proyecto es docente, ya que el objetivo de la herramienta es dar soporte a alumnos que estén estudiando Análisis y Diseño de Sistemas de Control. Tal es el caso de los alumnos de tercer curso del grado en Ingeniería en Tecnología Industrial de la EIB, donde existe la materia obligatoria *Automática y Control* (6 ECTS), o el del Máster en Ingeniería de Control, Automatización y Robótica de la EIB, que incluye la asignatura *Control Automático* (4,5 ECTS) o el de cualquier otra titulación a nivel nacional del ámbito del Control.

Otro beneficio es técnico, puesto que la herramienta facilita los análisis temporal, de la ubicación de polos y ceros y frecuencial de un sistema realimentado sin requerir conocimiento previo para ser utilizada, a diferencia de entornos de desarrollo como Mathematica® o Matlab®, ya que cuenta con una interfaz clara e intuitiva. Además, los análisis se muestran junto al sistema y son dinámicos, lo que permite realizar una asociación más clara entre estos y las características del sistema como, por ejemplo, al ver el desplazamiento de los polos en el lugar de las raíces al variar la ganancia del controlador.

Finalmente, cabe señalar que la herramienta es gratuita y accesible para todo aquel que disponga del entorno Mathematica® y acceso a Internet. Además, es opensource y está construida de forma modular, lo que hace que sea fácilmente ampliable y modificable para cualquiera que conozca el lenguaje de programación Wolfram.

## **5. Descripción de requerimientos**

En este apartado se describen los requerimientos de la herramienta. En primer lugar se detallan las características que deberán tener de la interfaz y la entrada de datos. A continuación, se describen los requerimientos de cada uno de los análisis que realiza la herramienta.

### **5.1. Requerimientos de la interfaz y la entrada de datos**

La herramienta debe contar con una interfaz clara que cuente con controles intuitivos para que la herramienta sea fácil de usar sin conocimiento previo.

Debe ser posible definir un sistema realimentado que cuente con controlador, actuador, planta y/o sensor a través de sus funciones de transferencia.

La herramienta debe facilitar el análisis temporal, el análisis de la ubicación de los polos y ceros en el plano  $s$  y el análisis frecuencial del sistema. Además, estos análisis deben variar dinámicamente cuando se modifiquen los datos de entrada del sistema, es decir, deben actualizarse y visualizarse a la vez que se modifiquen los datos de entrada del sistema.

Estos análisis deben mostrarse a través de gráficas y valores de los parámetros de la dinámica y la estabilidad del sistema.

Las gráficas de los análisis deben mostrar los nombres de los ejes y las unidades de estos cuando las haya. También deben tener unos controles que permitan variar su rango (desplazarse y hacer zoom) de forma sencilla y cómoda para el usuario.

La entrada del sistema debe poder ser una función escalón, rampa o parábola de amplitud variable, a variar por el usuario.

Los controles en los que se deban introducir datos deben contar con una comprobación que asegure que los datos introducidos son válidos.

Los controles y expresiones deben disponer de un cuadro emergente con información que se muestre al situar el ratón sobre ellos para proporcionar una explicación de los mismos a modo de ayuda.

### **5.2. Requerimientos del análisis temporal**

El análisis temporal debe permitir conocer la evolución temporal de la salida, de la referencia y del error del sistema a través de sus representaciones gráficas. Además, deberá facilitar los parámetros cuantificables más importantes del análisis temporal, como es el tiempo de establecimiento y los valores de la salida y el error en régimen estacionario.

Para ello, debe constar de dos gráficas (la respuesta temporal), una encima de la otra, y una leyenda.

En una gráfica se debe visualizar la referencia y la salida del sistema a lo largo del tiempo y, en la otra, el error. Cada función tendrá asociado un color, lo que permitirá identificarlas mediante una leyenda.

Estas gráficas deben contar con unas guías móviles que permitan conocer los valores de las tres funciones en el instante de tiempo deseado. En la leyenda deben aparecer estos valores, gracias a los cuales se pueden medir los parámetros del sistema mencionados anteriormente.

### **5.3. Requerimientos del lugar de las raíces**

El análisis del lugar de las raíces debe permitir conocer la ubicación de los polos y ceros del sistema para un rango de valores de la ganancia del controlador.

Para ello, debe contar una gráfica (denominada el lugar de las raíces) y una leyenda.

En la gráfica se debe poder visualizar no solo las trayectorias de los polos en el plano complejo  $s$ , sino también su posición a medida que varíe el valor de la ganancia del controlador. También deben mostrarse los polos en bucle abierto (cuando la ganancia es 0) y los ceros del sistema.

La leyenda debe mostrar los valores de los polos en bucle abierto, polos en bucle cerrado y ceros del sistema.

### **5.4. Requerimientos del análisis frecuencial**

El análisis frecuencial debe permitir visualizar la amplificación y/o atenuación así como el desfase de la salida del sistema en bucle abierto y el valor de los siguientes parámetros relevantes del análisis en frecuencia: margen de ganancia, margen de fase y frecuencias de ruptura.

Para ello, debe contar con dos gráficas semilogarítmicas (denominadas el diagrama de Bode) y una leyenda.

En una gráfica se deberá visualizar el módulo de la función de transferencia en bucle abierto y, en la otra, la fase. También debe ser posible visualizar en ellas el margen de ganancia y el margen de fase mediante flechas. Además, en la gráfica del módulo debe ser posible visualizar la función de módulo asintótico.

La leyenda debe mostrar los valores del margen de ganancia y del margen de fase, así como las frecuencias de ruptura del sistema.

## 6. Análisis de alternativas

En este apartado se describen tres posibles entornos de desarrollo con los que crear la herramienta: Labview®, Matlab® y Mathematica®. En primer lugar, se describe cada uno de ellos y se señalan sus ventajas e inconvenientes. Al final, se hace una comparativa entre las tres alternativas y se selecciona la mejor para la creación de la herramienta software.

### 6.1. Labview®

Labview® es un entorno de desarrollo orientado al diseño y control de sistemas desarrollado por National Instruments®. Cuenta con muchos complementos en forma de módulos, toolkits, bibliotecas, y más, que extienden las capacidades de este entorno. [4]

El lenguaje utilizado en Labview® es gráfico, lo que hace que sea un lenguaje sencillo de utilizar con el que se pueden crear interfaces con facilidad.

Labview® no se emplea en ninguna asignatura del grado en Ingeniería en Tecnología Industrial de la EIB, por lo que los alumnos no están familiarizados con este entorno.

En cuanto a los requerimientos para su uso, Labview® necesita licencia. Con un precio de 49 € para estudiantes, esta es la más barata de las tres alternativas. Por ese precio se incluye Labview® en su edición de estudiantes junto a los módulos LabVIEW MathScript RT y LabVIEW Control Design and Simulation, necesarios para la creación de la herramienta software. Cabe señalar que los estudiantes de la EIB no reciben licencia de la universidad pero sí que pueden emplear este entorno en las aulas informáticas de la universidad.



Figura 5. Logo de Labview®

### 6.2. Mathematica®

Mathematica® es un entorno de desarrollo polivalente que abarca diferentes áreas como el cálculo, álgebra, redes neuronales, procesamiento de imágenes y ciencia de datos, entre otras. Fue desarrollado por WolframResearch®. [5]

El lenguaje de programación de Mathematica® es un lenguaje imperativo pensado para que sea sencillo y lógico de usar, por lo que sus funciones tienen nombres en inglés y una estructura común. El cálculo simbólico es su principal característica, lo cual resulta útil para la creación de la herramienta en este entorno ya que su lenguaje es imperativo y no visual. Además, cuenta con funciones que permiten crear con facilidad gráficas y controles, lo que permite crear interfaces.

Por otro lado, los estudiantes del grado en Ingeniería en Tecnología Industrial de la EIB emplean este entorno en tres asignaturas: *Álgebra*, de primer curso; y *Ampliación de Matemáticas* y *Ampliación de Ecuaciones Diferenciales*, de tercer curso.

Por último, Mathematica® también necesita licencia. Esta es la más cara de las tres alternativas ya que su precio es de 160 € en su edición para estudiantes. Por ese precio no se incluyen actualizaciones, aunque esto no sería un obstáculo para el desarrollo del proyecto. No obstante, por 240 € se ofrece la opción de adquirir Mathematica® con un año de actualizaciones. Cabe señalar que los estudiantes de la EIB reciben licencia de la universidad durante sus años de estudio y se encuentra instalado en los ordenadores en las aulas informáticas de la universidad.



Figura 6. Logo de Mathematica®

### 6.3. Matlab®

Matlab® es un entorno de desarrollo orientado al cómputo numérico mediante el uso de matrices y optimizado para el análisis iterativo. También cuenta con muchas extensiones, denominadas toolboxes, que integran funciones en áreas específicas como sistemas de control, análisis de señales, machine learning, etc. Fue desarrollado por MathWorks®. [6]

El lenguaje de programación de este entorno es imperativo, al igual que el de Mathematica®. Su principal enfoque es el cálculo numérico pero también soporta el cálculo simbólico. No obstante, al no ser esta su principal prestación, esta característica, que resulta importante para la creación de la herramienta software, es más complicada de usar que en Mathematica®.

Sin embargo, Matlab® ofrece unas alternativas al lenguaje simbólico que pueden ser empleadas en la creación de la herramienta: Simulink® y Control System Designer. Ambas alternativas permiten un análisis dinámico de sistemas de control.

Simulink® es una toolbox de Matlab® que consiste en un entorno de desarrollo en el que se utiliza un lenguaje visual. Si bien esta es una característica útil, visualizar gráficas en este entorno resulta poco cómodo.

Control System Designer es una función de Control System Toolbox, un paquete de funciones para Matlab® y Simulink®. Al ejecutarla, se abre un entorno con múltiples análisis y que ofrece muchas posibilidades a la hora de realizar el diseño de sistemas de control. [7] La desventaja es que son tantas las opciones que ofrece que resulta complicado de aprender a utilizar y muchas de las características están fuera del ámbito de uso de los estudiantes de grado, estando orientadas a fines profesionales.

Por otro lado, los estudiantes del grado en Ingeniería en Tecnología Industrial de la EIB emplean Matlab® en las asignaturas *Ampliación de Métodos Numéricos* y *Control por Computador*, de tercer y cuarto curso, respectivamente.

Matlab®, al igual que Labview® y Mathematica®, necesita licencia. Esta tiene un precio de 69 € para estudiantes. Por este precio, se incluye Matlab® en su edición de estudiantes junto a Simulink® y Control System Toolbox, en el que se incluye la función Control System Designer, además de otras extensiones. Al igual que en el caso de Labview®, los estudiantes de la EIB no reciben licencia de la universidad pero sí que pueden acceder a Matlab® en las aulas informáticas de la universidad.



Figura 7. Logo de Matlab®

#### 6.4. Selección de la solución

A continuación se describen los cuatro aspectos que se han tenido en cuenta para seleccionar el entorno de desarrollo con el que elaborar la herramienta.

- Facilidad de programación de la herramienta en el entorno. Viene determinada por las características de los lenguajes de los entornos expuestos en los anteriores apartados. Este aspecto es el que se ha considerado como más importante en la comparación.
- Conocimiento previo del entorno por parte del desarrollador. Es también muy importante ya que, junto al anterior, determina en gran medida la complejidad de la creación de la herramienta
- Acceso al entorno por parte de los estudiantes de la EIB. En este aspecto se tiene en cuenta si los estudiantes de la EIB reciben licencia por parte de la universidad. Es bastante importante, ya que determina el alcance que tendrá la herramienta en la universidad.
- Precio de la licencia. Este aspecto se ha considerado como el menos importante ya que muchas universidades ofrecen los entornos de desarrollo considerados a sus alumnos, ya sea en forma de licencia para ordenadores personal o instalándolos en los equipos de la universidad, por lo que en muchos casos no sería necesario adquirirlo.

A cada uno de estos cuatro aspectos se le ha asignado un porcentaje que refleja su importancia en la valoración final. Con ellos se ha elaborado la Tabla 1 para realizar la comparación de los entornos descritos. Se ha puntuado cada aspecto con un número del 0 al 5 y finalmente se ha calculado la puntuación global de cada entorno realizando la media ponderada de las puntuaciones.

|                    | <b>Facilidad programación (35%)</b> | <b>Conocimiento previo (30%)</b> | <b>Acceso estudiantes EIB (20%)</b> | <b>Precio (15%)</b> | <b>Valoración global</b> |
|--------------------|-------------------------------------|----------------------------------|-------------------------------------|---------------------|--------------------------|
| <b>Labview</b>     | 5                                   | 1                                | 2                                   | 5                   | 3,2                      |
| <b>Mathematica</b> | 4                                   | 3                                | 5                                   | 2                   | 3,6                      |
| <b>Matlab</b>      | 2                                   | 2                                | 2                                   | 4                   | 2,3                      |

**Tabla 1. Comparativa de los entornos de desarrollo considerados**

Como se puede ver, la alternativa con mayor puntuación es Mathematica®, por lo que ha sido el entorno de desarrollo seleccionado para crear la herramienta.

## 7. Descripción Mathematica®

En este apartado se realiza una descripción de Mathematica®, el entorno de desarrollo empleado para la creación de la herramienta. Comienza con una introducción al entorno y posteriormente se exponen las funciones más relevantes para la creación de la herramienta.

### 7.1. Introducción a Mathematica®

El entorno de Mathematica® se divide en dos partes: el *Kernel*, que realiza las operaciones, y el *Front End*, que muestra la información del sistema, es decir, la interfaz. Esta interfaz consiste en archivos denominados cuadernos (*Notebooks*) organizados en celdas. Las celdas pueden ser básicamente de dos tipos: texto (*Title*, *Text...*) o entrada (*Input*). Las celdas de texto permiten poner títulos y párrafos para organizar el cuaderno y las de entrada son celdas ejecutables en las que escribir el código. [8]

Cada cuaderno puede tener varias celdas de entrada para ejecutar diferentes partes de un programa o realizar distintos cálculos de forma independiente. El resultado del código ejecutado se muestra por defecto debajo de este, en una nueva celda de salida (*Output*). Por defecto, en el margen derecho del cuaderno se muestran los corchetes que delimitan cada celda.

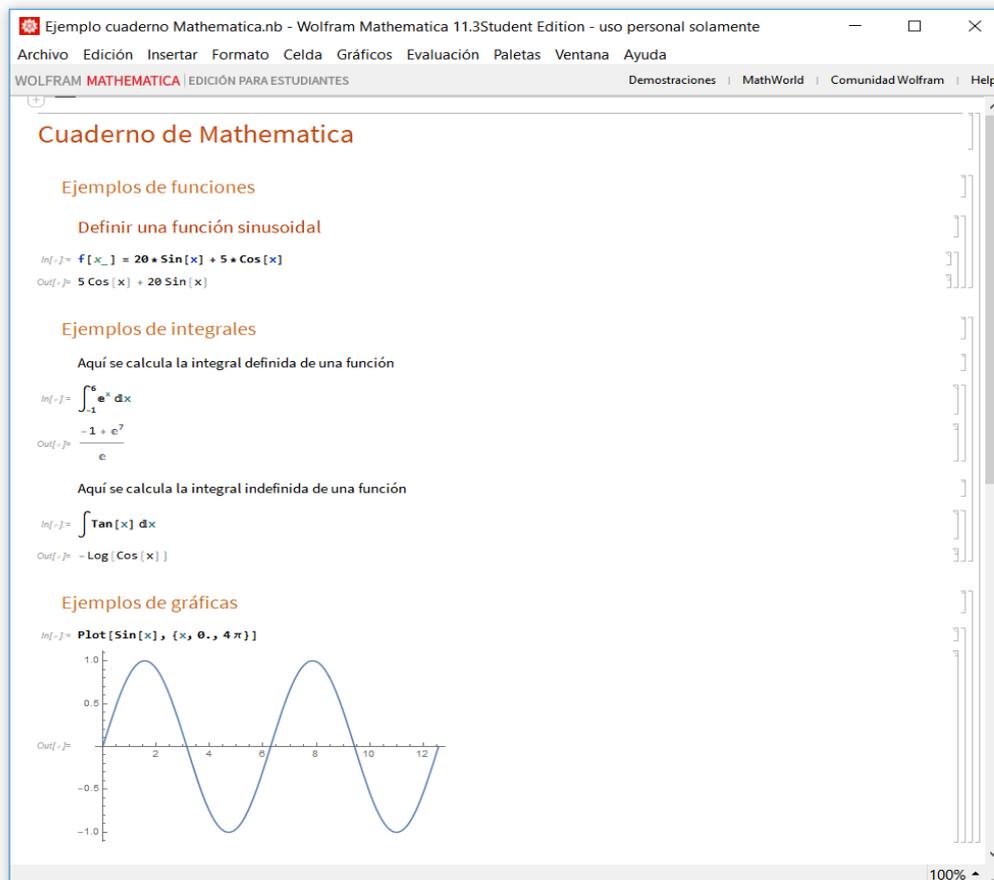


Figura 8. Ejemplo de cuaderno de Mathematica®

En cuanto al lenguaje de programación, Mathematica® tiene un lenguaje imperativo y principalmente simbólico, aunque también integra el cálculo numérico. Como se ha señalado antes, está diseñado para que sea sencillo y lógico de usar, por lo que sus funciones tienen nombres en inglés y una estructura común. Además, integra una gran variedad de símbolos que permiten representar operaciones como integrales y raíces cuadradas de forma natural.

El lenguaje de programación de Mathematica® integra diversos tipos de datos: cualquier tipo de número (real, entero, complejo...), cadenas de caracteres, arrays (denominados listas), expresiones simbólicas (ecuaciones, polinomios, funciones...), etc. Una ventaja de Mathematica® es que las variables pueden almacenar no solo estos tipos de datos, sino también objetos como gráficas y controles. Además, se puede asignar a cada variable cualquier tipo de dato, incluso si no era el que inicialmente guardaba.

Otra característica útil de Mathematica® es que permite, mediante la función *Dynamic*, hacer que las expresiones, como por ejemplo las ecuaciones y las gráficas, se actualicen de forma automática y simultánea cuando se modifiquen las variables especificadas, es decir, se logra que las expresiones sean dinámicas.

Las variables en este entorno por defecto son globales. Para localizarlas, se pueden emplear las funciones *Module*, *DynamicModule* o *Block*, especificando en ellas las variables locales y escribiendo dentro el código a ejecutar. Las diferencias entre las tres funciones residen, entre otras cosas, en la forma en la que localizan las variables.

Por un lado, *Module* y *DynamicModule* generan nombres únicos para las variables a localizar cada vez que se ejecuta el cuerpo de la función. *Module* genera estos nombres únicos para la sesión actual del *Kernel* mientras que *DynamicModule* lo hace para la del *Front End*.

Por otro lado, *Block* usa el nombre de las variables especificadas como si fueran locales. Debido a esto, resulta conveniente usar la función *Block* en funciones que vayan a ser llamadas varias veces para hacer un uso más eficiente de la memoria. [8]

Por último, para definir variables locales cuyo valor no vaya a cambiar, es decir, para definir constantes, se puede emplear la función *With*.

Cabe señalar que cualquier nombre que no corresponda a una función o constante integrada en Mathematica®, o a una variable definida por el usuario que tenga asignado un valor, Mathematica® lo interpreta como un símbolo que puede ser utilizado en una expresión simbólica.

Por otro lado, Mathematica® permite definir funciones de diferentes formas, ya sean funciones matemáticas o subfunciones semejantes a las de otros lenguajes de programación. La forma más habitual de definir una función es indicando su nombre con sus variables entre corchetes, pero también se puede crear una función sin especificar los nombres de las variables de la misma. Además, si se ponen dos puntos antes del igual en la definición de la función, la expresión de la función permanecerá sin evaluar, evaluándose cada vez que se llame a la función. En la imagen se define una función matemática definida con corchetes y, a continuación, una subfunción que

permite calcular la media de dos números definida de dos formas: especificando sus variables y sin hacerlo.

$$f[x_] = \text{Sin}[x] + 5 * x^2;$$
$$\text{MediaAritmetica}[a_, b_] := \frac{a + b}{2};$$
$$\text{MediaAritmetica} := \left( \frac{\#1 + \#2}{2} \right) \&;$$

Figura 9. Ejemplos definición de funciones

Mathematica® también cuenta con una biblioteca muy amplia de funciones, desde funciones matemáticas hasta funciones que permiten crear gráficas y controles. Esto, sumado a la característica comentada de crear expresiones dinámicas, hace que crear interfaces resulte sencillo en este entorno.

También cabe señalar que muchas de las funciones que ofrece Mathematica® cuentan con opciones que permiten personalizarlas añadiendo especificaciones. Por ejemplo, se puede modificar el tamaño y los colores de los ejes de una gráfica o añadirle un título, entre muchas otras.

## 7.2. Funciones relevantes Mathematica®

A continuación se presentan las funciones de mayor interés para la creación de la herramienta software. Se han dividido en seis grupos: funciones de comparación y ciclos repetitivos; funciones de manipulación de listas y expresiones; funciones matemáticas; funciones para crear gráficas; funciones para crear controles; y funciones para organizar interfaces.

### 7.2.1 Funciones de comparación y ciclos repetitivos

Mathematica® cuenta con funciones básicas de programación como las de comparación y ciclos repetitivos.

Entre las funciones de comparación se encuentran *If*, que representa la clásica estructura if-then-else, y *Which*, que representa una estructura anidada if-then-else en la que se van probando sentencias en orden hasta que una resulta verdadera. La única diferencia que tiene la función *If* de Mathematica® con respecto a estructura clásica *if-else* es que la expresión else se encuentra integrada dentro de la función *If*.

En cuanto a los ciclos repetitivos, Mathematica® cuenta, entre otras, con la función *Do*. Esta función permite crear bucles de un número conocido de pasos, como el bucle for. La única diferencia que tiene la función *Do* con la estructura for es que la comparación y el incremento de la variable iterativa no se especifican directamente, sino a través del primer valor y último valor de la variable iterativa junto con el paso.

### 7.2.2 Funciones de manipulación de listas y expresiones

Mathematica® también cuenta con una variedad de funciones de manipulación y creación de listas (arrays) y expresiones.

Entre las funciones sobre listas, se encuentran *Table* y *Length*, que sirven para crear listas de una longitud predefinida y para conocer la longitud de una lista, respectivamente. Por otro lado, se encuentran las funciones para manipular listas. Las más destacables son: *Join*, *Append*, *Prepend*, *Take*, *Flatten*, *Partition* y *Sort*. Las primeras tres permiten unir listas, añadir elementos al final de una lista y añadir elementos al principio de una lista, respectivamente. *Take* permite extraer elementos de una lista. *Flatten* unifica los niveles de una lista que contenga sublistas en una sola lista y *Partition* hace la función inversa, subdividiendo una lista en otra que contenga sublistas. Por último, *Sort* permite ordenar listas de acuerdo a una condición especificada.

En cuanto a las funciones para manipular expresiones, se encuentran las funciones *Simplify*, *Numerator*, *Denominator*, *FactorList* y *CoefficientList*. La primera permite realizar simplificaciones en una expresión, extrayendo factores comunes y agrupando términos para lograr lo que Mathematica® considera una expresión más sencilla. Las otras cuatro funciones permiten extraer partes de una expresión: el numerador, el denominador, la lista de los factores de un polinomio junto a su multiplicidad y la lista de coeficientes de un polinomio, respectivamente.

### 7.2.3 Funciones matemáticas

Mathematica® permite realizar operaciones matemáticas básicas como la raíz cuadrada (*Sqrt*), el logaritmo natural (*Log*) o el valor absoluto (*Abs*), entre muchas otras.

Entre otras funciones matemáticas se destacan también las funciones *Floor*, *Ceil* y *Round*. Estas permiten redondear un número a su entero inmediatamente inferior o superior; o al entero más cercano, respectivamente.

Mathematica® también permite realizar otras operaciones más complejas como calcular límites, con la función *Limit*, o transformadas y transformadas inversas de Laplace, con *LaplaceTransform* e *InverseLaplaceTransform*, respectivamente.

También se pueden resolver ecuaciones con la función *NSolve*. Además, Mathematica® permite especificar características de parámetros o variables de las ecuaciones empleando la función *Assuming*.

### 7.2.4 Funciones para crear gráficas

Entre las funciones para crear gráficas se encuentran *Plot* y dos de sus variantes: *LogLinearPlot* y *ParametricPlot*. Estas funciones permiten crear gráficas a partir de expresiones simbólicas. La primera muestra la gráfica de una función, la segunda muestra la gráfica semilogarítmica de una función y la tercera muestra la gráfica de una función paramétrica. Los colores, tamaños y demás características tanto de las funciones como de los ejes en los que se representan pueden ser modificados por el usuario añadiendo opciones a estas funciones.

Por otro lado, la función *Graphics* permite crear rectángulos, círculos, flechas y otras figuras. También permite mostrar texto y controles. Todas estas figuras y expresiones pueden ser modificadas fácilmente, cambiando sus dimensiones, proporciones, colores, opacidad, etc.

Para combinar gráficas de forma automática existe la función *Show*, que permite mostrar sobre los mismos ejes combinaciones de las funciones descritas en este apartado.

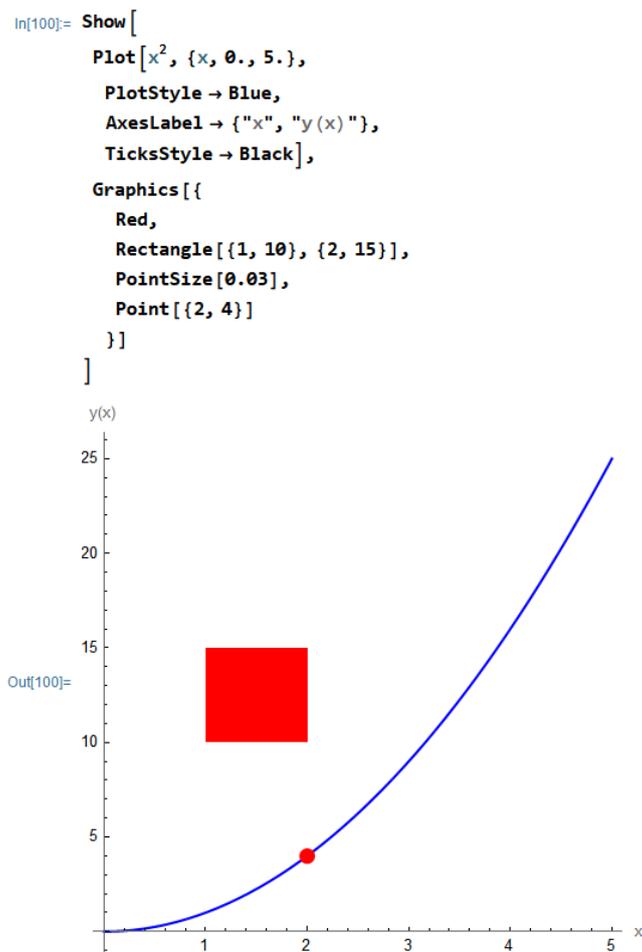


Figura 10. Ejemplo de varias gráficas combinadas

### 7.2.5 Funciones para crear controles

En cuanto a la creación de controles, Mathematica® cuenta con varias funciones para ello. Entre ellas, se destacan tres: *InputField*, *Slider* y *Button*. La primera permite crear un campo de entrada donde introducir cualquier tipo de expresión (número, cadena de caracteres, expresión simbólica...); la segunda, un deslizador para modificar el valor de una variable de forma continua; y la tercera, crear botones que realicen una acción deseada.

Por otro lado, también se puede lograr que determinadas teclas o acciones con el ratón realicen una acción determinada gracias a la función *EventHandler*. En este

ámbito también se puede incluir la función *Locator*, que crea puntos que pueden ser movidos con el ratón.

Por último, cabe destacar en este apartado la función *Tooltip*, que permite añadir a una expresión un cuadro de texto emergente que se muestre al situar el ratón sobre esta.

```
In[1]:= x = 0.5;  
  
InputField[Dynamic[x]]  
  
Slider[Dynamic[x], {0, 1}]  
  
Button["Imprimir valor de x", Print[x]]  
  
Tooltip["Una expresión", "Mensaje emergente"]
```

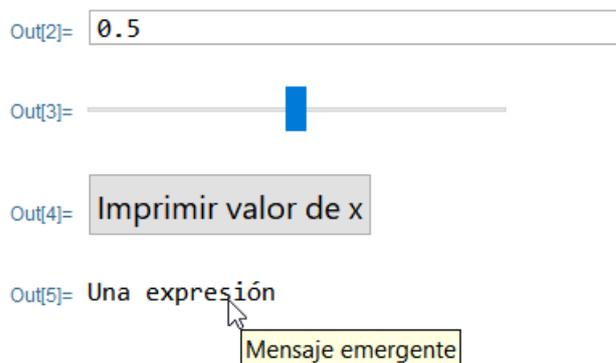


Figura 11. Ejemplo de controles

### 7.2.6 Funciones para organizar interfaces

Para organizar los elementos de una interfaz se puede emplear la función *Grid*, que crea una tabla que permite disponer distintos elementos en columnas y filas.

Además, para que el resultado del código no se muestre en el mismo documento, debajo del código escrito, se puede emplear la función *CreateDocument*. Con esta función, se puede crear un nuevo documento donde mostrar el resultado.

## 8. Descripción de tareas

En este apartado se describe el plan de trabajo del proyecto. En primer lugar, se presentan los paquetes de trabajo del proyecto, describiendo las tareas en las que se dividió cada uno. A continuación, se detallan los hitos alcanzados a lo largo del desarrollo del proyecto, que reflejaron los puntos de control del mismo.

### 8.1. Paquetes de trabajo

En este apartado se describen los paquetes de trabajo en los que se dividió el desarrollo del proyecto. En cada paquete de trabajo se expone una descripción de lo que se realizó durante el mismo junto con sus datos más relevantes: responsable y recursos humanos, carga de trabajo, recursos técnicos y duración estimada. Al final del apartado se muestra un diagrama de Gantt, realizado mediante el programa MS-Project®, que permite obtener una vista global del plan de trabajo.

#### **P.T.1 Diseño**

En este paquete de trabajo se realizó el diseño de la herramienta software. Se dividió en dos tareas: el diseño de la interfaz y el diseño de los algoritmos.

En la primera tarea se diseñó la interfaz preliminar de la herramienta, planificando los espacios necesarios para el sistema, es decir, la entrada de datos, y para cada uno de sus análisis.

En la segunda tarea se diseñaron los algoritmos que permiten obtener los análisis del sistema a partir de los datos de entrada.

**Responsable y recursos humanos:** Ingeniero Senior, Ingeniero Junior.

**Carga de trabajo:** 10 horas de diseño.

**Duración estimada:** 8 días naturales (6 días hábiles).

#### **P.T.2 Desarrollo**

En este paquete de trabajo se llevó a cabo la programación de la herramienta implementando la interfaz y los algoritmos diseñados en el paquete de trabajo anterior. Se dividió en seis tareas, cinco de ellas de desarrollo: programación de la entrada de datos, de la respuesta temporal, del lugar de las raíces, del diagrama de Bode y de la interfaz; y una última de validación de la herramienta.

En las primeras cinco tareas se creó la herramienta, programando la entrada de datos y los análisis e implementándolos en la interfaz. Estas tareas tuvieron una duración de 73 días, ocupando la mayor parte del tiempo del paquete de trabajo.

La última tarea corresponde a la validación de la herramienta probándola con diferentes sistemas de control. Se realizó la corrección de los errores encontrados hasta comprobar que los análisis que mostraba eran correctos y fiables.

**Responsable y recursos humanos:** Ingeniero Senior, Ingeniero Junior.

**Carga de trabajo:** 150 horas de programación y 10 horas de validación.

**Recursos técnicos:** 160 horas de ordenador y 160 horas de licencia.

**Duración estimada:** 112 días naturales (80 días hábiles).

### **P.T.3 Optimización**

En este paquete de trabajo se optimizó la herramienta. Se dividió de forma análoga al paquete de trabajo anterior, en cinco tareas de optimización: optimización de rendimiento, de la respuesta temporal, del lugar de las raíces, del diagrama de Bode y de la interfaz; y en una última, igual que antes, de validación de la herramienta.

En las primeras cinco tareas se optimizó la herramienta en diversos aspectos. En la primera tarea, se mejoró el tiempo de computación modificando la estructura del código. Para ello, se siguieron los consejos de programación que ofrece Mathematica® a través de su documentación online [8]. En las siguientes tres, se mejoraron los tiempos de computación de los tres análisis. En la última tarea, se optimizó la interfaz para que pueda ser vista correctamente en ordenadores de diferentes tamaños y se realizaron cambios para lograr un estilo más profesional y que fuese más atractiva. Estas tareas tuvieron una duración de 41 días, ocupando la mayor parte del tiempo del paquete de trabajo.

En la última tarea se volvió a validar la herramienta probándola con los mismos sistemas de la fase de validación anterior y con otros nuevos. Se hizo para comprobar que los análisis mostrados seguían siendo correctos después de haber realizado cambios en el código.

**Responsable y recursos humanos:** Ingeniero Senior, Ingeniero Junior.

**Carga de trabajo:** 80 horas de programación y 10 horas de validación.

**Recursos técnicos:** 90 horas de ordenador y 90 horas de licencia.

**Duración estimada:** 59 días naturales (43 días hábiles).

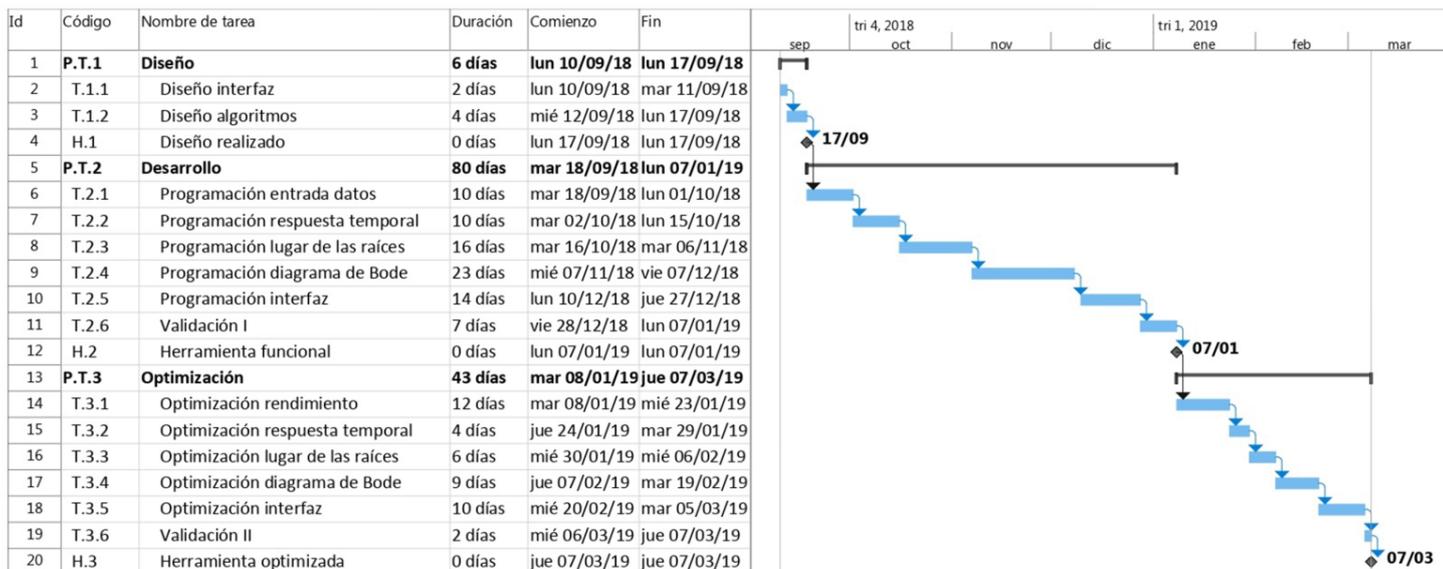


Figura 12. Diagrama de Gantt del proyecto

## 8.2. Hitos

En este apartado se detallan los hitos alcanzados en el desarrollo del proyecto. Son tres los hitos, uno al final de cada paquete de trabajo.

### H.1 Diseño realizado

Este hito está al final del primer paquete de trabajo. Refleja que se realizó el diseño preliminar completo de la herramienta.

### H.2 Herramienta funcional

Este hito está al final del segundo paquete de trabajo. Implica que se logró programar la herramienta y se probó para detectar y corregir los errores, logrando el primer diseño funcional de la misma.

### H.3 Herramienta optimizada

Este hito está al final del tercer paquete de trabajo. Refleja el final del proceso de desarrollo de la herramienta, en el que la herramienta, además de ser funcional, está optimizada, logrando mostrar los análisis del sistema de forma más rápida y teniendo una interfaz más profesional y atractiva.

## 9. Algoritmos

En este apartado se describen los algoritmos empleados en cada uno de los análisis. La estructura de los algoritmos de cada uno de los análisis es la misma: calcular las funciones que se tienen que representar, calcular parámetros si los hubiera y calcular el rango de valores en el que se mostrarán las gráficas. En primer lugar, se describen los algoritmos del análisis temporal; después, los del análisis de los polos y ceros y; finalmente, los del análisis frecuencial.

### 9.1. Algoritmos del análisis temporal

El análisis temporal del sistema consta de dos gráficas (la respuesta temporal) y una leyenda. En la primera de las gráficas se representan la referencia y la salida del sistema y en la otra se representa el error, ambas respecto al tiempo. En la leyenda se muestra el valor de las funciones en un instante variable, a variar por el usuario, lo que permite medir parámetros de la dinámica y la estabilidad del sistema como los valores de la salida y el error en estado estacionario y el tiempo de establecimiento, entre otros.

Tanto para obtener las gráficas como los valores de la leyenda se necesita calcular las expresiones de las funciones de la salida, la referencia y el error en el tiempo. También se necesita calcular el rango de valores en los que se representan las funciones.

#### 9.1.1 Cálculo de la salida, referencia y error

Para calcular las expresiones de la salida, la referencia y el error en el tiempo en primer lugar se calculan sus expresiones en  $s$ . La expresión de la referencia es conocida ya que es un dato de entrada. Las expresiones de la salida y el error se calculan a partir de las funciones de transferencia de los elementos del sistema y de la referencia.

Calculando la transformada inversa de Laplace de las expresiones en  $s$  de estas funciones se obtienen sus expresiones en el tiempo. Para ello, se emplea la función *InverseLaplaceTransform*.

$$\begin{aligned} \text{Salida}[t\_] &= \text{InverseLaplaceTransform}\left[\frac{Kc Gc Ga Gp}{1. + Kc Gc Ga Gp H} * Ru, s, t\right]; \\ \text{Referencia}[t\_] &= \text{InverseLaplaceTransform}[Ru, s, t]; \\ \text{Error}[t\_] &= \text{InverseLaplaceTransform}\left[\frac{1.}{1. + Kc Gc Ga Gp H} * Ru, s, t\right]; \end{aligned}$$

Figura 13. Cálculo de las funciones del análisis temporal

#### 9.1.2 Cálculo del rango de los valores de las gráficas

Para calcular el rango de valores representativos de las funciones se calcula por un lado el valor máximo de las abscisas, que es igual en las dos gráficas y, por otro, los

valores máximos y mínimos de las ordenadas de cada gráfica. Cabe señalar que el valor mínimo de las abscisas siempre será 0 en ambas gráficas, ya que es donde se ha definido el comienzo de la referencia.

Para conocer el máximo valor de abscisas, se calcula en primer lugar el valor final de la salida. Para hacerlo, se hace uso del teorema del valor final y se calcula

$$\lim_{s \rightarrow 0} (s \cdot Y(s))$$

ya que este límite es más sencillo y, por lo tanto, requiere de menor tiempo de computación que

$$\lim_{t \rightarrow \infty} y(t)$$

Si el valor del límite es un número finito, se procede a calcular valores de la salida, con un paso de 0.1, desde el valor 0.00 hasta que el error absoluto entre la salida y el valor del límite sea inferior a 0.00000005 o hasta alcanzar el valor máximo de las abscisas fijado en la herramienta, que es 20.00. Este valor de error está escogido ad hoc mediante ensayos de prueba y error con diversas funciones para garantizar que se muestra una parte representativa de las funciones.

Si, por el contrario, el límite es infinito o indeterminado, se calculan directamente los valores de la salida desde 0.00 hasta 20.00 con incrementos de 0.1.

Con el valor de la longitud de la lista donde se han guardado los valores de la salida se calcula el valor máximo de las abscisas. Este valor es igual a la longitud de la lista menos uno, multiplicada por el paso anterior, es decir, 0.1.

A continuación, se calculan los valores del error desde 0.00 hasta el valor máximo de abscisas calculado anteriormente.

En cuanto a la referencia, si es una función escalón, se calcula su valor únicamente en un punto ya que este no varía. Si es una función rampa o parábola, se calculan sus valores desde 0.00 hasta 20.00 con un paso de 0.1. En ambos casos, los valores se guardan en la misma lista que la de los valores de la salida ya que ambas funciones se dibujan en la misma gráfica.

Por último, mediante la función *MinMax*, se calculan el mínimo y el máximo de la lista con los valores de la salida y referencia y los de la lista con los valores del error para conocer los valores extremos de las ordenadas de cada gráfica.

```

ValorFinal = Limit[ $\frac{Kc Gc Ga Gp}{1. + Kc Gc Ga Gp H} * R u * s, s \rightarrow 0.$ ];
If[(ValorFinal  $\notin$  Reals) || (ValorFinal === Indeterminate),
ValoresSalRef = {If[R === 1. / s,
Chop@Referencia[1.],
(*Else*)
Table[Chop@Referencia[i], {i, 0., tAbsMax, 0.1}]
]},
Table[Chop@Salida[i], {i, 0., tAbsMax, 0.1}],
0.},
(*Else*)
ValoresSalRef = {If[R === 1. / s,
Chop@Referencia[1.],
(*Else*)
Table[Chop@Referencia[i], {i, 0., tAbsMax, 0.1}]
]},
If[Abs[ValorFinal - Chop@Salida[0.]] < 0.00000005,
Table[Chop@Salida[i], {i, 0., tAbsMax, 0.1}],
(*Else*)
Join[
{ValorActual = Chop@Salida[0.],
Reap[Do[
ValorAntiguo = ValorActual;
ValorActual = Chop@Salida[i];
If[Abs[ValorFinal - ValorActual] > 0.00000005 && Quiet@ (Abs[ValorActual - ValorAntiguo] / Abs[ValorActual]) > 0.000005,
Sow@ValorActual,
(*Else*)
Break[]
],
{i, 0.1, tAbsMax, 0.1}]]][2, 1]]
],
0.});
];
tMaximo = (Length[ValoresSalRef[[2]]] - 1) * 0.1;
ValoresError = Chop@{Table[Error[i], {i, 0., tMaximo, 0.1}], 0.};
RangoAutomaticoRT = {{0., tMaximo}, Round[MinMax[ValoresSalRef], RTPaso], Round[MinMax[ValoresError], RTPaso]};
{{tmin, tmax}, {SalRefmin, SalRefmax}, {Errormin, Errormax}} = RangoAutomaticoRT;

```

Figura 14. Cálculo del rango de la gráfica del análisis temporal

## 9.2. Algoritmos del análisis de la ubicación de los polos y ceros

El análisis de la ubicación de los polos y ceros del sistema consta de una gráfica (el lugar de las raíces) y una leyenda. En la gráfica se representan las trayectorias de los polos del sistema cuando varía la ganancia del controlador, así como los polos en bucle abierto (cuando la ganancia es 0), los polos en bucle cerrado (móviles en función de la ganancia) y los ceros del sistema. En la leyenda se pueden visualizar los valores de estos puntos, lo que permite conocer el valor de su posición.

Tanto para obtener la gráfica como los valores de la leyenda se necesita calcular las expresiones de los polos en función de la ganancia del controlador y las de los ceros del sistema. Además, para la representación de la gráfica se deben separar las expresiones de la parte real y la parte imaginaria de los polos y ceros. También se necesita calcular el rango de valores en el que mostrar la gráfica.

### 9.2.1 Cálculo de los polos en bucle abierto, polos en bucle cerrado y ceros

Para obtener los polos del sistema se resuelve la ecuación del denominador de la función de transferencia en bucle abierto igualada a 0, es decir, la ecuación

característica, con la ganancia del controlador  $k$  como parámetro y la variable compleja  $s$  como incógnita. Para ello, se emplea la función *NSolve*.

A continuación, se calculan la parte real e imaginaria de cada polo con las funciones *Re* e *Im*, respectivamente, para poder representarlos como puntos. Estos son los polos en bucle cerrado, ya que dependen de la ganancia del controlador. Haciendo una copia de estos puntos y dándole a la ganancia del controlador el valor 0, se obtienen los polos en bucle abierto.

Para calcular los ceros se procede de forma similar. En primer lugar, se resuelve la ecuación del numerador de la función de transferencia en bucle abierto igualada a 0 con la variable compleja  $s$  como incógnita, empleando la función *NSolve*. A continuación, se calculan la parte real e imaginaria de cada cero, de nuevo con las funciones *Re* e *Im*, para poder representarlos como puntos.

Para obtener las trayectorias de los polos se tiene en cuenta que estas son las funciones paramétricas de los polos en bucle cerrado obtenidas a partir de extraer la parte real y la imaginaria y cuyo parámetro es la ganancia del controlador, es decir, que las expresiones a utilizar son iguales a las de los polos en bucle cerrado calculadas antes.

```
(*.....Cálculo polos (raíces denominador).....*)
SolucionPolos = s /. NSolve[1 + k Gc Ga Gp H = 0., s];
nPolos = Length[SolucionPolos];
PolosYCeros = Table[{Re[SolucionPolos[[i]], Im[SolucionPolos[[i]]], {i, nPolos}}; (*convertir los polos a puntos*)
PolosYCeros = Join[PolosYCeros /. k -> 0., PolosYCeros]; (*añadir los puntos (polos) iniciales (en BA, cuando k = 0)*)
(*.....Cálculo ceros (raíces numerador).....*)
SolucionCeros = s /. NSolve[Gc Ga Gp = 0., s]; (*cálculo de los ceros*)
If[SolucionCeros === s,
  SolucionCeros = {}
];
nCeros = Length[SolucionCeros];
PolosYCeros = Join[PolosYCeros, Table[{Re[SolucionCeros[[i]], Im[SolucionCeros[[i]]], {i, nCeros}}]; (*convertir los ceros a puntos*)
(*.....Trayectorias de los polos (i.e., lugar de las raíces).....*)
EcParametricas = Table[{PolosYCeros[[i, 1]], PolosYCeros[[i, 2]], {i, nPolos + 1, 2 nPolos}};
(*.....Polos y ceros.....*)
MarcadoresPolosBA = Flatten[Table[{ColoresBase[[i]], Text[SMarcadoresLR["x"], PolosYCeros[[i]]], {i, nPolos}}];
MarcadoresPolosBC = Flatten[Table[{ColoresBaseClaros[[i - nPolos]], Text[SMarcadoresLR["□"], PolosYCeros[[i]] /. k -> Hold[Kc]], {i, nPolos + 1, 2 nPolos}}];
MarcadoresCeros = Flatten[Table[{ColoresBase[[i - nPolos]], Text[SMarcadoresLR["o"], PolosYCeros[[i]]], {i, 2 nPolos + 1, 2 nPolos + nCeros}}];
(*.....Leyenda.....*)
Leyenda = {
  Table[{SleyendasGraficasColor["x", ColoresBase[[i]], Item[SleyendasGraficas@Round[SolucionPolos[[i]] /. k -> 0., DecimalesLR], Alignment -> Left}}, {i, nPolos}},
  Table[{SleyendasGraficasColor["□", ColoresBaseClaros[[i]], Item[SleyendasGraficas@Round[SolucionPolos[[i]] /. k -> Hold[Kc], DecimalesLR], Alignment -> Left}}, {i, nPolos}},
  Table[{SleyendasGraficasColor["o", ColoresBase[[i + nPolos]], Item[SleyendasGraficas@Round[SolucionCeros[[i]], DecimalesLR], Alignment -> Left}}, {i, nCeros}}];
If[nCeros != 0,
  Leyenda[[3]] = Join[Leyenda[[3]], Table[{"", ""}, {i, nPolos - nCeros}]];
Leyenda = Partition[Flatten[Thread[{Leyenda[[1]], Leyenda[[2]], Leyenda[[3]]}], 6],
  (*Else*)
  Leyenda = Take[Leyenda, 2];
  Leyenda = Partition[Flatten[Thread[{Leyenda[[1]], Leyenda[[2]]}], 4]
];
```

Figura 15. Cálculo de los polos y ceros

## 9.2.2 Cálculo del rango de los valores de la gráfica

Para calcular el rango de valores en el que se mostrará la gráfica se tiene en cuenta que los polos en bucle cerrado empiezan sus trayectorias en los polos en bucle abierto y acaban en los ceros o en el infinito, por lo que los valores de estos puntos serán los extremos del rango en el que se representará la gráfica.

Los polos en bucle abierto son los polos en bucle cerrado cuando la ganancia del controlador vale 0 y ya han sido calculados, al igual que los ceros. En cuanto a los

polos en bucle cerrado que acaban en el infinito, se toma su valor máximo representable en la gráfica, que se corresponde con el valor máximo que puede tomar la ganancia, esto es, 100.00.

Aplicando la función *MinMax* a las partes reales de estos valores (polos en bucle abierto, ceros y polos en bucle cerrado cuando la ganancia del controlador toma el valor máximo) se obtienen los valores mínimo y máximo de las abscisas. Procediendo de igual forma con las parte imaginarias de los valores mencionados se obtienen los valores mínimo y máximo de las ordenadas.

```
ValoresLR = Chop@Join[Table[SolucionPolos /. k -> i, {i, {0., KcMax}}], SolucionCeros];
RangoAutomaticoLR = Round[{MinMax[Re@ValoresLR], MinMax[Im@ValoresLR]}, LRPaso];
```

Figura 16. Cálculo del rango de la gráfica del análisis de la ubicación de polos y ceros

### 9.3. Algoritmos del análisis frecuencial

El análisis frecuencial del sistema consta de dos gráficas (el diagrama de Bode) y una leyenda. En la primera de las gráficas se representa el módulo de la función de transferencia en bucle abierto (en decibelios) y en la otra se representa su fase (en grados), ambas respecto a la frecuencia de la entrada (en radianes/segundo). En la leyenda se muestran los valores del margen de fase y del margen de ganancia, así como las frecuencias de ruptura del sistema.

Tanto para obtener las gráficas como los valores de la leyenda se necesita calcular las expresiones del módulo y la fase de la función de transferencia en bucle abierto. Por otro lado, también es necesario calcular la expresión del módulo asintótico y el rango de valores en el que se representarán las funciones.

#### 9.3.1 Cálculo de la expresión del módulo

Para obtener la expresión del módulo, en primer lugar se sustituye la variable compleja  $s$  de la función de transferencia en bucle abierto por  $j\omega$ , siendo  $j$  la unidad imaginaria y  $\omega$  la frecuencia de la entrada en radianes/segundo. A continuación, se calcula su módulo en decibelios, es decir, veinte veces el logaritmo en base diez del módulo de la función de transferencia en bucle abierto.

Esta expresión se simplifica teniendo en cuenta que  $\omega$  es real y mayor que cero para mejorar la velocidad de los cálculos.

```
(*.....Función módulo.....*)
FuncionModulo = Kc Gc Ga Gp H;
FuncionModulo = FuncionModulo /. s -> i \omega;
FuncionModulo = ComplexExpand[FuncionModulo];
FuncionModulo = Assuming[\omega \in Reals && \omega > 0., Simplify[20. * Log10[Abs[FuncionModulo]]]];
```

Figura 17. Cálculo de la expresión del módulo del análisis frecuencial

### 9.3.2 Cálculo de la expresión de la fase

Para obtener la expresión de la fase de la función de transferencia en bucle abierto se necesita conocer el argumento de cada uno de los factores del numerador y denominador de esta función cuando se sustituye en ella la variable compleja  $s$  por  $j\omega$ . Los términos del numerador de la función aportarán un argumento positivo y los del denominador, negativo.

Para hacer esto, se extraen en primer lugar los factores del numerador de la función de transferencia en bucle abierto empleando las funciones *FactorList*, que extrae los coeficientes junto con su multiplicidad, y *Numerator*, que extrae el numerador de la función. A continuación, se sustituye la variable compleja  $s$  por  $j\omega$  en cada uno de los factores. Se procede de igual con la forma con el denominador de la función, empleando en este caso la función *Denominator* en lugar de *Numerator*. Mediante un ciclo repetitivo, se suma el argumento de cada uno de los factores del numerador, calculado como el arcotangente del cociente entre la parte imaginaria y la real, multiplicado por el número de repeticiones del factor. Procediendo de forma análoga, se restan a esta expresión los argumentos de los factores del denominador. Finalmente, la expresión de la fase se convierte de radianes a grados empleando la constante integrada en Mathematica® *Degree*.

```
(*.....Función argumento.....*)
Numerador = FactorList[Numerator[FuncionModulo]];
Numerador = Numerador /. s -> i w;
Denominador = FactorList[Denominator[FuncionModulo]];
Denominador = Denominador /. s -> i w;
Do[FuncionArgumento = FuncionArgumento + Numerador[[i, 2]] * ArcTan[Re[Numerador[[i, 1]]], Im[Numerador[[i, 1]]]], {i, Length[Numerador]}];
Do[FuncionArgumento = FuncionArgumento - Denominador[[i, 2]] * ArcTan[Re[Denominador[[i, 1]]], Im[Denominador[[i, 1]]]], {i, Length[Denominador]}];
FuncionArgumento = FuncionArgumento / Degree;
```

Figura 18. Cálculo de la expresión de la fase del análisis frecuencial

Cabe señalar que la expresión de la fase de la función de transferencia en bucle abierto se calcula de esta forma (sumando y restando los factores de su numerador y denominador) y no obteniendo el arcotangente del cociente entre la parte imaginaria y real de toda la función debido a la discontinuidad de la función arcotangente a lo largo del semieje real negativo del plano complejo.

### 9.3.3 Cálculo de la expresión del módulo asintótico

Para calcular la expresión del módulo asintótico se tienen en cuenta los cambios de pendiente que causan los polos y ceros en el origen y los simples y dobles, y que estos cambios ocurren desde el origen en el primer caso (en los del origen) y en las frecuencias de ruptura en los dos últimos (en los simples y dobles). Cabe señalar que este algoritmo está limitado a polos y ceros simples y dobles, por lo que no es capaz de representar polos y ceros de orden superior.

La expresión de módulo asintótico se representa como una sucesión de segmentos de los que se debe calcular sus extremos, así como las abscisas asociadas a estos, teniendo en cuenta que estos puntos dependen de los polos y ceros del sistema. Para ello, en primer lugar se deben extraer los factores del numerador y denominador de la función de transferencia del sistema en bucle abierto junto con su multiplicidad, esto es, los polos y ceros. Para extraer los factores se emplea la función *FactorList*.

A continuación, se comprueba si hay polos o ceros en el origen. Si los hay, se guarda el valor que tiene la pendiente en el primer tramo de la gráfica, es decir, hasta la primera frecuencia de ruptura. Este valor es el número de polos multiplicado por -20 o el número de ceros multiplicado por 20. Si no hay polos ni ceros en el origen, el valor de la pendiente es 0, el valor con el que se inicializa la variable.

El siguiente paso es calcular las frecuencias de ruptura de cada uno de los polos y ceros y el cambio de pendiente asociado a cada una de ellas, identificando si el factor es de primer o segundo orden. Esto último se hace comprobando el número de coeficientes que tiene cada factor, los cuales se pueden obtener con la función *CoefficientList*. Esta función se aplica primero a los factores del numerador (los ceros) y, a continuación, a los del denominador (los polos).

Si el número de coeficientes es dos, significa que es un polo o cero simple y, por lo tanto, su frecuencia de ruptura será el valor de la constante de tiempo. Este valor es igual al coeficiente del término independiente entre el del término en  $s$ . El cambio de pendiente en este caso será la multiplicidad del factor por 20 o -20, dependiendo de si es un cero o un polo, respectivamente. Si el número de coeficientes es tres, significa que es un polo o cero doble y la frecuencia de ruptura será la frecuencia natural, es decir, la raíz cuadrada del término independiente. El cambio de pendiente en este otro caso será la multiplicidad del factor por 40 o -40, dependiendo de si es un cero o un polo, respectivamente.

A continuación, se ordenan estas frecuencias de menor a mayor y se añaden la mínima y la máxima, que son 0.001 y 1000.00, respectivamente. Al hacerlo de esta forma (primero ordenándolas y después añadiendo la mínima y máxima) hay una limitación ya que se está suponiendo que ninguna frecuencia de ruptura del sistema es inferior a la mínima ni superior a la máxima. No obstante, dado que estas tienen valores muy extremos, el caso de que una frecuencia del sistema se encuentra fuera de esos límites es improbable.

El siguiente paso es calcular los valores de las ordenadas correspondientes a las frecuencias de ruptura. El primer valor es sencillamente el valor de la expresión de módulo evaluada a la frecuencia mínima definida en el sistema, es decir, 0.0001.

El segundo valor depende de los polos y ceros en el origen. Si la función de transferencia en bucle abierto contaba con alguno, el segundo punto se calcula como la pendiente asociada a sus polos o ceros calculada anteriormente multiplicada por la diferencia entre la primera frecuencia de ruptura y la frecuencia mínima. Si no había polos o ceros en el origen, el segundo punto es igual al primero.

El resto de puntos dependen de los polos y ceros, por lo que se calculan mediante un ciclo repetitivo. En cada iteración, se calcula la nueva pendiente sumándole la pendiente asociada al polo o cero de la iteración. A continuación, se calcula el valor de las ordenadas como el producto entre la pendiente y la diferencia entre la siguiente frecuencia de ruptura y la frecuencia de ruptura del polo o cero de la iteración.

Sumando estos valores acumulativamente desde el primero, si hay polos o ceros en el origen, o desde el segundo, si no polos ni ceros en el origen, se obtienen finalmente las ordenadas correspondientes. De esta forma, ya se tienen los puntos necesarios para representar la expresión del módulo asintótico.

```
(*.....Coordenadas x.....*)
Funcion = Kc Gc Ga Gp H;
Numerador = FactorList[Numerator[Funcion]];
Denominador = FactorList[Denominator[Funcion]];
(*.....Quitar factor común.....*)
Numerador = Numerador[[2 ;;]];
Denominador = Denominador[[2 ;;]];
(*.....Quitar ceros/polos origen.....*)
If[Rationalize[Denominador[[1, 1]]] === s,
  Pendiente = -Denominador[[1, 2]] * 20.;
  Denominador = Denominador[[2 ;;]],
  (*Else*)
  If[Length[Numerador] > 1,
    If[Rationalize[Numerador[[1, 1]]] === s,
      Pendiente = Numerador[[1, 2]] * 20.;
      Numerador = Numerador[[2 ;;]]
    ]
  ]
];
(*.....Frecuencias de ruptura.....*)
Do[Coefficientes = CoefficientList[Numerador[[i, 1]], s];
  If[Length[Coefficientes] == 3,
    FrecuenciasRuptura = Append[FrecuenciasRuptura, {Sqrt[Coefficientes[[1]], 40. * Numerador[[i, 2]]}],
    (*Else*)
    FrecuenciasRuptura = Append[FrecuenciasRuptura, {CoefficientList[Coefficientes[[1]], s][[1]], 20. * Numerador[[i, 2]]}],
    {i, Length[Numerador]}];
Do[Coefficientes = CoefficientList[Denominador[[i, 1]], s];
  If[Length[Coefficientes] == 3,
    FrecuenciasRuptura = Append[FrecuenciasRuptura, {Sqrt[Coefficientes[[1]], -40. * Denominador[[i, 2]]}],
    (*Else*)
    FrecuenciasRuptura = Append[FrecuenciasRuptura, {CoefficientList[Coefficientes[[1]], s][[1]], -20. * Denominador[[i, 2]]}],
    {i, Length[Denominador]}];
(*.....Ordenar de menor a mayor.....*)
FrecuenciasRuptura = Sort[FrecuenciasRuptura, #1[[1]] < #2[[1]] &];
(*añadimos la frecuencia mínima y la máxima del código*)
(*NOTA: asumimos que ninguna frecuencia del sistema es menor que la mínima ni mayor que la máxima del código*)
FrecuenciasRuptura = Join[{{1. / wAbsMax, 0.}}, FrecuenciasRuptura, {{wAbsMax, 0.}}];
(*.....Coordenadas y.....*)
(*.....Primer punto.....*)
CoordenadasY = {FuncionModulo /. w -> FrecuenciasRuptura[[1, 1]]};
(*.....Segundo punto.....*)
(*NOTA: depende de los ceros/polos en el origen*)
If[Pendiente != 0.,
  CoordenadasY = Append[CoordenadasY, Pendiente * (Log10[FrecuenciasRuptura[[2, 1]]] - Log10[FrecuenciasRuptura[[1, 1]]])],
  (*Else*)
  CoordenadasY = Append[CoordenadasY, CoordenadasY[[1]]];
];
(*.....Resto de puntos.....*)
Do[
  Pendiente = Pendiente + FrecuenciasRuptura[[i - 1, 2]];
  CoordenadasY = Append[CoordenadasY, Pendiente * (Log10[FrecuenciasRuptura[[i, 1]]] - Log10[FrecuenciasRuptura[[i - 1, 1]]])],
  {i, 3, Length[FrecuenciasRuptura]}];
(*.....Sumar valores acumulativamente.....*)
If[CoordenadasY[[1]] != CoordenadasY[[2]],
  (*si hay ceros/polos en el origen, hay pendiente no nula y hay que sumar los puntos desde el primero*)
  CoordenadasY = FoldList[Plus, CoordenadasY[[1]], CoordenadasY[[2 ;;]]],
  (*Else*)
  (*si no hay ceros/polos en el origen, hay pendiente nula y hay que sumar los puntos desde el segundo, que es igual que el primero*)
  CoordenadasY = Prepend[FoldList[Plus, CoordenadasY[[2]], CoordenadasY[[3 ;;]]], CoordenadasY[[1]]];
];
```

Figura 19. Cálculo de la expresión del módulo asintótico del análisis frecuencial

### 9.3.4 Cálculo del margen de ganancia y margen de fase

Para calcular los márgenes de ganancia y de fase se necesita conocer el punto en el que la expresión de la fase corta a la recta de  $-180$  grados y en el que la expresión del módulo corta a las abscisas, respectivamente.

Para calcular ambos parámetros, el primer paso es conocer las partes real e imaginaria de la función de transferencia en bucle abierto una vez sustituida la variable compleja  $s$  por  $j\omega$ .

Para calcular el margen de ganancia, se calcula en primer lugar el punto de corte entre la expresión de la fase y la recta de  $-180$  grados. Para ello, en este caso se utiliza como expresión de la fase el arcotangente del cociente entre la parte imaginaria y la parte real, en radianes. A esta expresión se le resta  $\pi$  y se iguala la expresión a  $0$ . Esta ecuación se simplifica a que la parte imaginaria sea igual a  $0$  y parte real sea menor que  $0$ .

$$\text{Arctan}\left(\frac{\text{Re}[G(j\omega)]}{\text{Im}[G(j\omega)]}\right) = -\pi \Rightarrow$$

$$\text{Im}[G(j\pi)] = 0, \quad \text{Re}[G(j\pi)] < 0$$

Se resuelve esta ecuación con la función *NSolve*. Si existe solución, el margen de ganancia es el valor de la expresión del módulo cambiada de signo evaluada en el punto de corte calculado. Si no existe solución, el margen de ganancia es infinito.

Para calcular el margen de fase, se calcula en primer lugar el punto de corte entre la expresión del módulo y las abscisas. Para ello, se tiene en cuenta que la expresión del módulo es igual a  $20$  veces el logaritmo en base diez de la raíz cuadrada de la suma de los cuadrados de las partes real e imaginaria. Esta expresión se iguala a  $0$  y la ecuación resultante se simplifica a que la suma de los cuadrados de las partes real e imaginaria igual a  $1$ .

$$20 \cdot \log \sqrt{\text{Re}[G(j\omega)]^2 + \text{Im}[G(j\omega)]^2} = 0 \Rightarrow$$

$$\text{Re}[G(j\omega)]^2 + \text{Im}[G(j\omega)]^2 = 1$$

Se resuelve esta ecuación con la función *NSolve*. Si existe solución, el margen de ganancia es  $180$  más el valor de la expresión de la fase evaluada en el punto de corte calculado, todo ello en grados. Si no existe solución, el margen de fase es  $180$  grados.

```

(*.....Componentes de la función.....*)
Componentes = Assuming[ $\omega \in \text{Reals}$  &&  $\omega > 0.$ , Simplify[{Re[FuncionModulo], Im[FuncionModulo]}]];
(*esta operación hace que los cálculos sean un poco más rápidos*)
Componentes = Rationalize[Componentes];
(*.....Margen de ganancia.....*)
CorteFase = NSolve[Componentes[[2]] == 0. && Componentes[[1]] > 0. &&  $\omega > 0.$ ,  $\omega$ , Reals];
If[CorteFase != {},
  CorteFase =  $\omega$  /. CorteFase[[1]];
  MargenDeGanancia = Round[FuncionModulo /.  $\omega \rightarrow$  CorteFase, DecimalesBode];
  PuntosFlechaMG = {{Log[CorteFase], MargenDeGanancia}, {Log[CorteFase], 0}}];
(*.....Margen de fase.....*)
CorteModulo = NSolve[Componentes[[1]]2 + Componentes[[2]]2 == 1. &&  $\omega > 0.$ ,  $\omega$ , Reals];
If[CorteModulo != {},
  CorteModulo =  $\omega$  /. CorteModulo[[1]];
  MargenDeFase = Round[FuncionArgumento /.  $\omega \rightarrow$  CorteModulo, DecimalesBode];
  PuntosFlechaMF = {{Log[CorteModulo], -180.}, {Log[CorteModulo], MargenDeFase}}];

```

Figura 20. Cálculo del margen de ganancia y margen de fase

### 9.3.5 Cálculo del rango de la valores de las gráficas

Para calcular el rango de las gráficas se calculan los valores de las funciones de las gráficas de módulo y fase en los puntos correspondientes a las potencias de 10 desde la frecuencia mínima, que es 0,0001, hasta la máxima, que es 1000.00. A continuación, se calculan el máximo y el mínimo de cada una de las listas de valores con la función *MinMax*.

Los valores extremos de la frecuencia siempre serán el mínimo y el máximo ya que están en el rango de valores habituales de los diagramas de Bode y no se ha considerado que sea necesario calcular otros valores.

```

(*.....Rango automático.....*)
ValoresModulo = Chop@Table[FuncionModulo /.  $\omega \rightarrow$  i, {i, {1. /  $\omega$ AbsMax, 1. /  $\omega$ AbsMax * 10., 1. /  $\omega$ AbsMax * 100., 1. /  $\omega$ AbsMax * 1000.,  $\omega$ AbsMax / 10.,  $\omega$ AbsMax / 100.,  $\omega$ AbsMax}}];
If[MargenDeGanancia != - $\infty$ ,
  ValoresModulo = Join[ValoresModulo, {0.}]
];
RangoAutomaticoModulo = Ceiling[MinMax[ValoresModulo], BodePaso];
ValoresArgumento = Chop@Table[FuncionArgumento /.  $\omega \rightarrow$  i, {i, {1. /  $\omega$ AbsMax, 1. /  $\omega$ AbsMax * 10., 1. /  $\omega$ AbsMax * 100., 1. /  $\omega$ AbsMax * 1000.,  $\omega$ AbsMax / 10.,  $\omega$ AbsMax / 100.,  $\omega$ AbsMax}}];
If[MargenDeFase != 0.,
  ValoresArgumento = Join[ValoresArgumento, {-180., 0.}]
];
RangoAutomaticoArgumento = Ceiling[MinMax[ValoresArgumento], BodePaso];
{ $\omega$ min,  $\omega$ max} = {1. /  $\omega$ AbsMax,  $\omega$ AbsMax};
{Modulomin, Modulomax} = RangoAutomaticoModulo;
{Argumentomin, Argumentomax} = RangoAutomaticoArgumento;

```

Figura 21. Cálculo del rango de las gráficas del análisis frecuencial

## 10. Descripción de los resultados

En este apartado se describen los resultados obtenidos en el proyecto. En primer lugar, se explica la interfaz y la forma de introducir los datos de entrada de la herramienta software. A continuación, se detallan los resultados obtenidos en cada uno de los análisis (temporal, de la ubicación de polos y ceros y frecuencial). Finalmente se expone un ejemplo de aplicación de la herramienta software, diseñando un controlador PID a partir de unas especificaciones dadas.

### 10.1. Resultados de la interfaz y la entrada de datos

En este apartado se explican los resultados referentes a la interfaz y la entrada de datos, que son la referencia y las funciones de transferencia de los elementos del sistema.

La herramienta se muestra en una ventana que tiene las mismas características y funciones que cualquier ventana de Windows®, excepto que su tamaño es fijo. También tiene se puede visualizar la herramienta a pantalla completa pulsando la tecla p. Pulsando la tecla Esc se sale del modo pantalla completa. Mostrar la herramienta a pantalla completa permite visualizar las gráficas más grandes.

La interfaz se ha estructurado de forma que se puedan visualizar simultáneamente el sistema y sus análisis. Se divide en dos zonas separadas por un divisor horizontal: una que permite definir el sistema y otra que muestra los análisis del mismo.

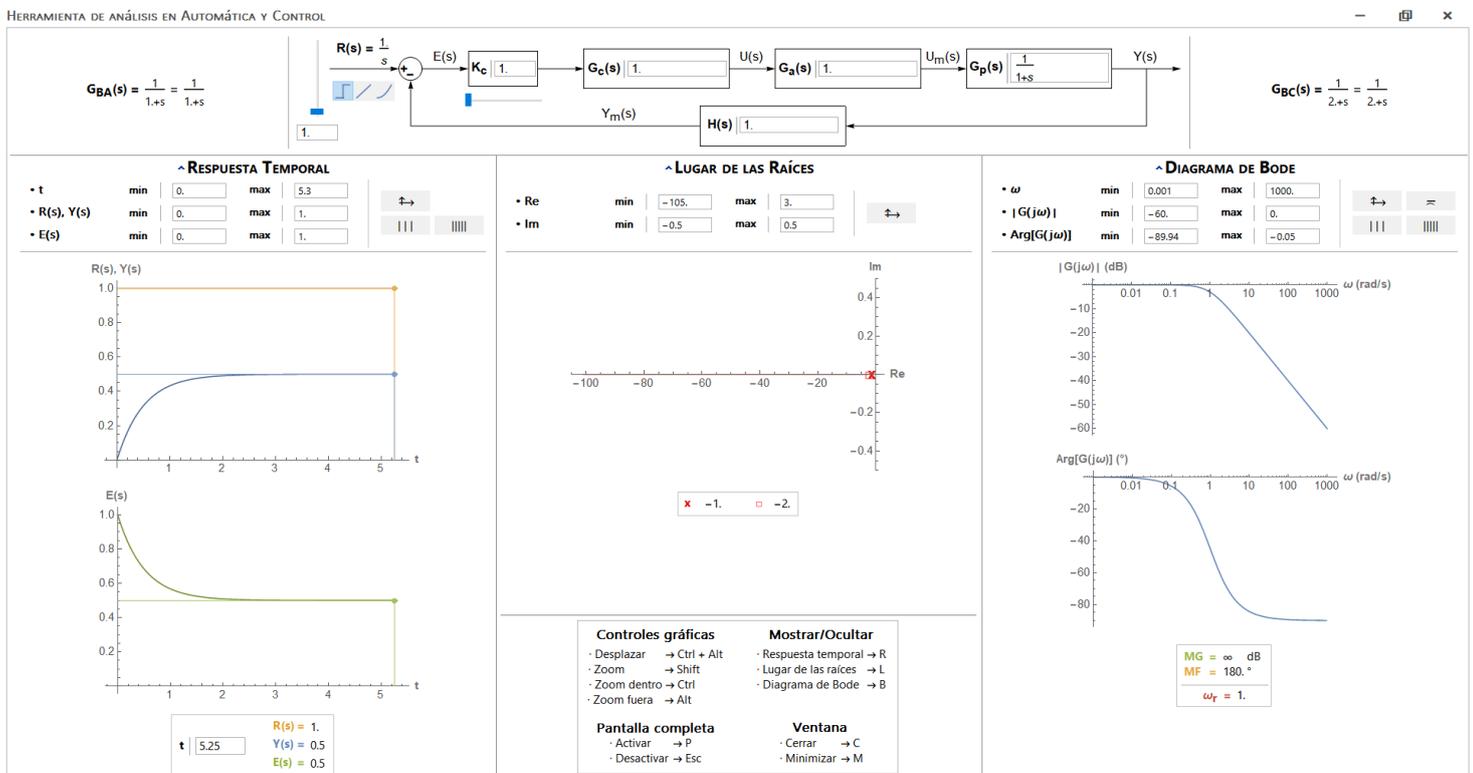


Figura 22. Interfaz de la herramienta software

Por otro lado, cabe señalar que al situar el ratón sobre algunos botones y abreviaturas aparece un texto explicativo, como se puede ver en la Figura 23.

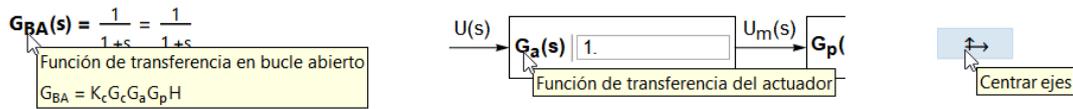


Figura 23. Tres ejemplos de texto explicativo

En la Figura 24 están señaladas las expresiones que cuentan con esta característica.

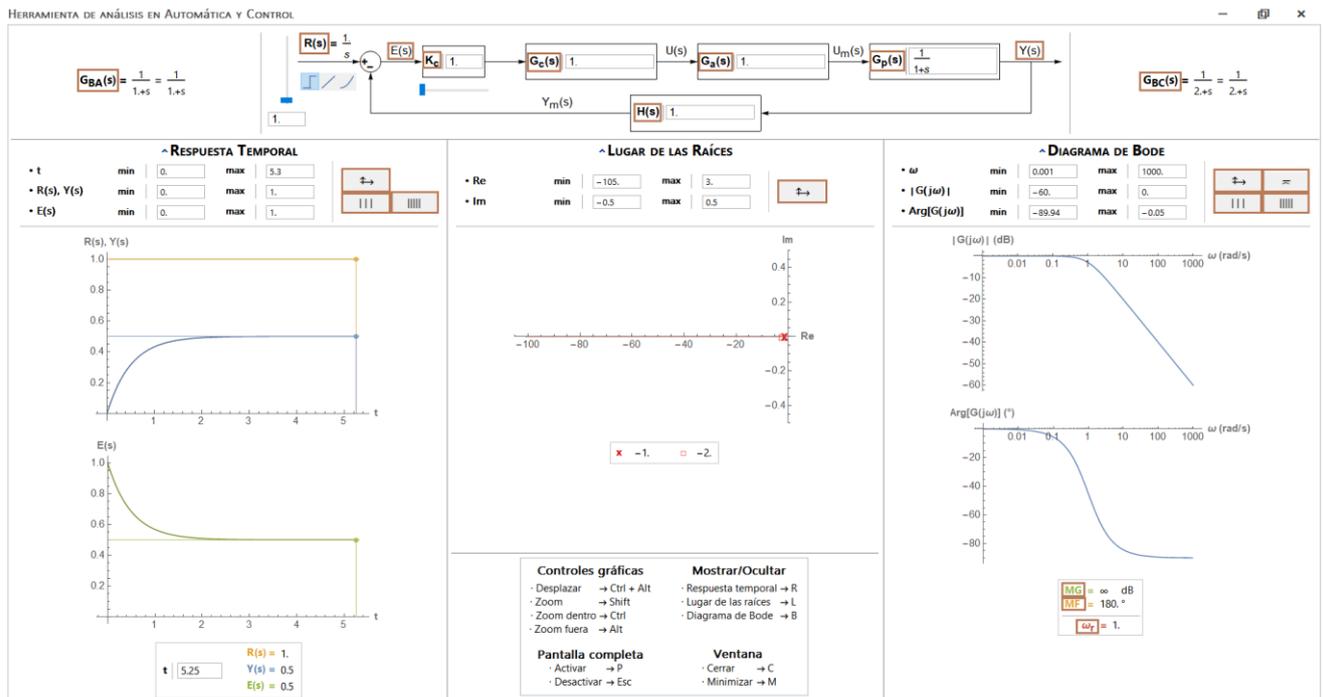


Figura 24. Expresiones que cuentan con mensaje explicativo

A continuación, se detallan los resultados obtenidos de cada una de las zonas de la herramienta. En primer lugar se explican los resultados de la zona del sistema, incluyendo la forma de introducir los datos de entrada. A continuación, se explican los resultados de la zona de los análisis, señalando las características comunes a los tres análisis.

### 10.1.1 Zona del sistema y entrada de datos

La zona en la que se define el sistema se encuentra en la parte superior. Se divide en tres partes: la expresión de la función de transferencia en bucle abierto, el diagrama de bloques del sistema y la expresión de la función de transferencia en bucle cerrado.

En el centro de esta zona se encuentra el diagrama de bloques del sistema en el que se introducen los datos de entrada, es decir, las funciones de transferencia de los elementos del sistema (controlador, actuador, planta y sensor) y la función de la referencia.

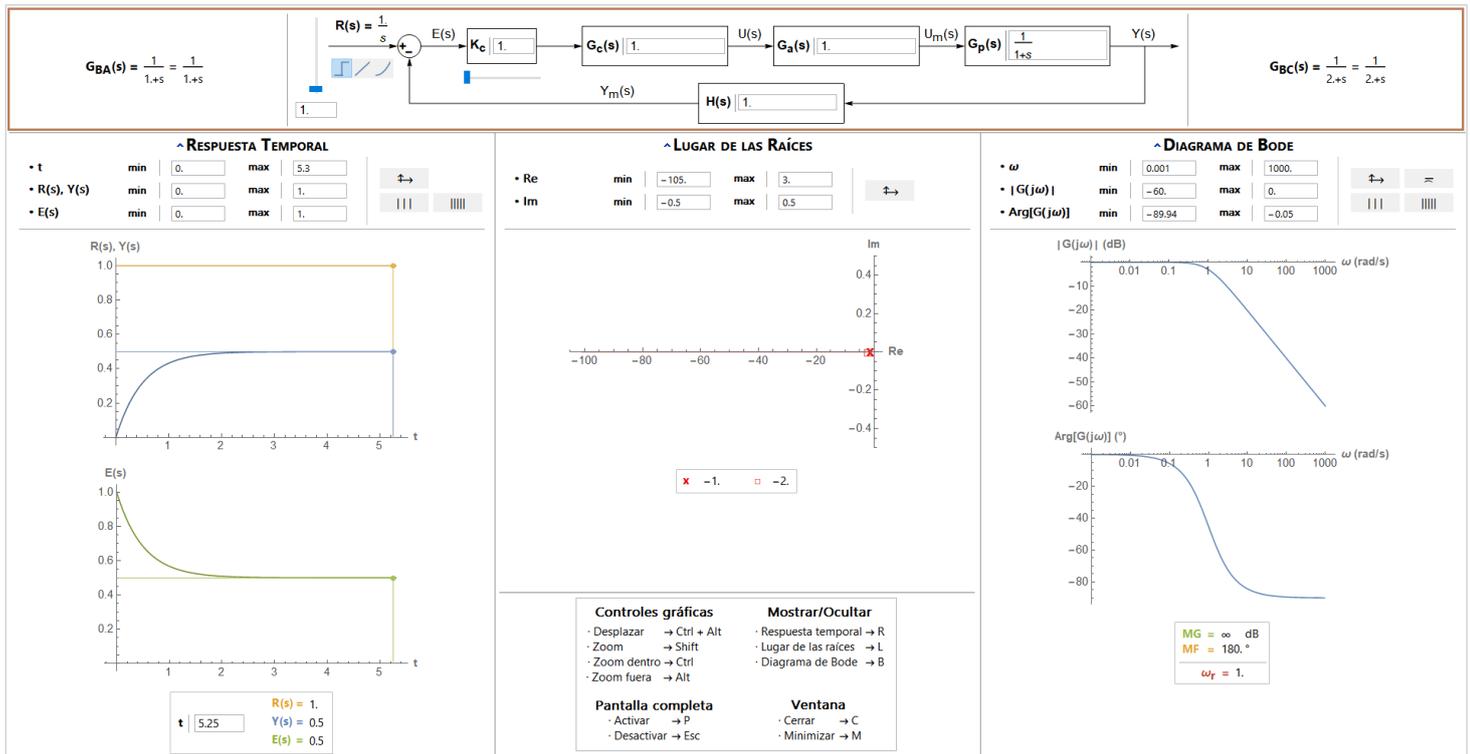


Figura 25. Zona del sistema

Los controles de las funciones de transferencia son campos de entrada, de forma que se pueden introducir directamente sus expresiones como cocientes de dos polinomios en  $s$ . Cuentan con una comprobación para que no se puedan introducir funciones inválidas o con errores de sintaxis, como por ejemplo introducir una variable que no sea  $s$  o no cerrar un paréntesis.

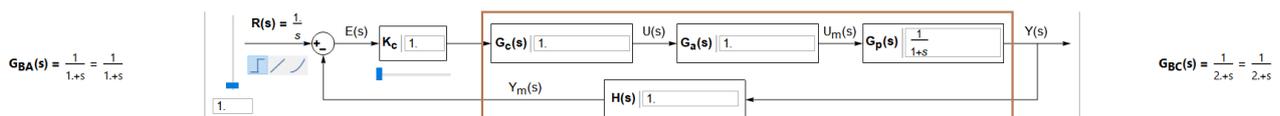


Figura 26. Controles de las funciones de transferencia

Cabe señalar que la función de transferencia del controlador está dividida en un campo de entrada con deslizador para la ganancia y un campo de entrada para el resto de la función del controlador. Esto es así para permitir ver cómo varían los análisis dinámicamente cuando varía la ganancia, lo cual se puede hacer de forma continua gracias al deslizador. El campo de entrada de la ganancia también cuenta con comprobación, por lo que solo acepta números reales con hasta dos decimales, en un rango de valores de 0.10 a 100.00, que es el rango del deslizador.

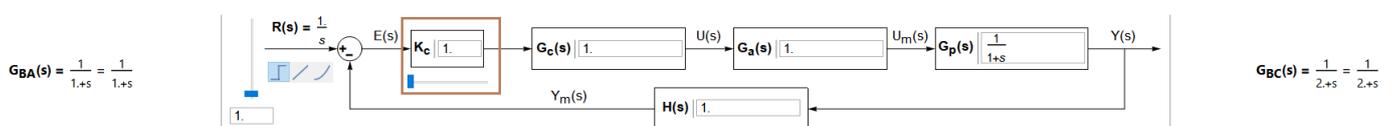


Figura 27. Controles de la ganancia del controlador

En cuanto a los controles para manipular la referencia, estos se dividen en dos. Por un lado, están los botones de selección que permiten seleccionar el tipo de función: escalón, rampa o parábola. Por otro lado, está el campo de entrada con deslizador para variar la amplitud de la entrada. El campo de entrada cuenta con una comprobación para que solo se puedan introducir números reales con hasta dos decimales, en un rango de valores de 0.10 a 100.00, que es el rango del deslizador.

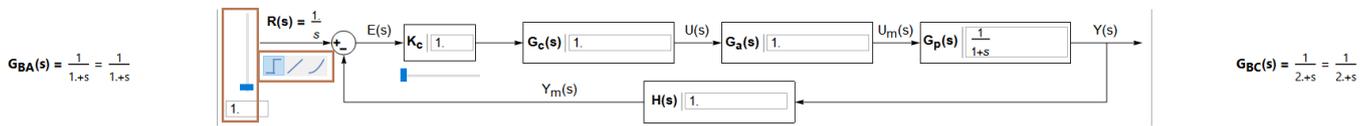


Figura 28. Controles de la referencia

Los controles se encuentran dentro del diagrama de bloques para que resulten más claros e intuitivos. Además, en el diagrama de bloques se muestran los nombres de todas las funciones, así como el valor de la referencia, que varía dinámicamente con los cambios de la misma.

A ambos lados del diagrama de bloques se muestran las funciones de transferencia del sistema en bucle abierto y bucle cerrado, separadas por divisores horizontales. Estas funciones se muestran tanto en su forma expandida como factorizada.

### 10.1.2 Zona de los análisis

La zona en la que se muestran los análisis se encuentra en la parte inferior, debajo de la zona del sistema. En esta zona se pueden visualizar los tres análisis separados por divisores verticales: el análisis temporal (Respuesta Temporal), el análisis de la ubicación de polos y ceros (Lugar de las Raíces) y el análisis frecuencial (Diagrama de Bode). También se encuentra en esta zona un cuadro de ayuda con las combinaciones de teclas que realizan funciones.

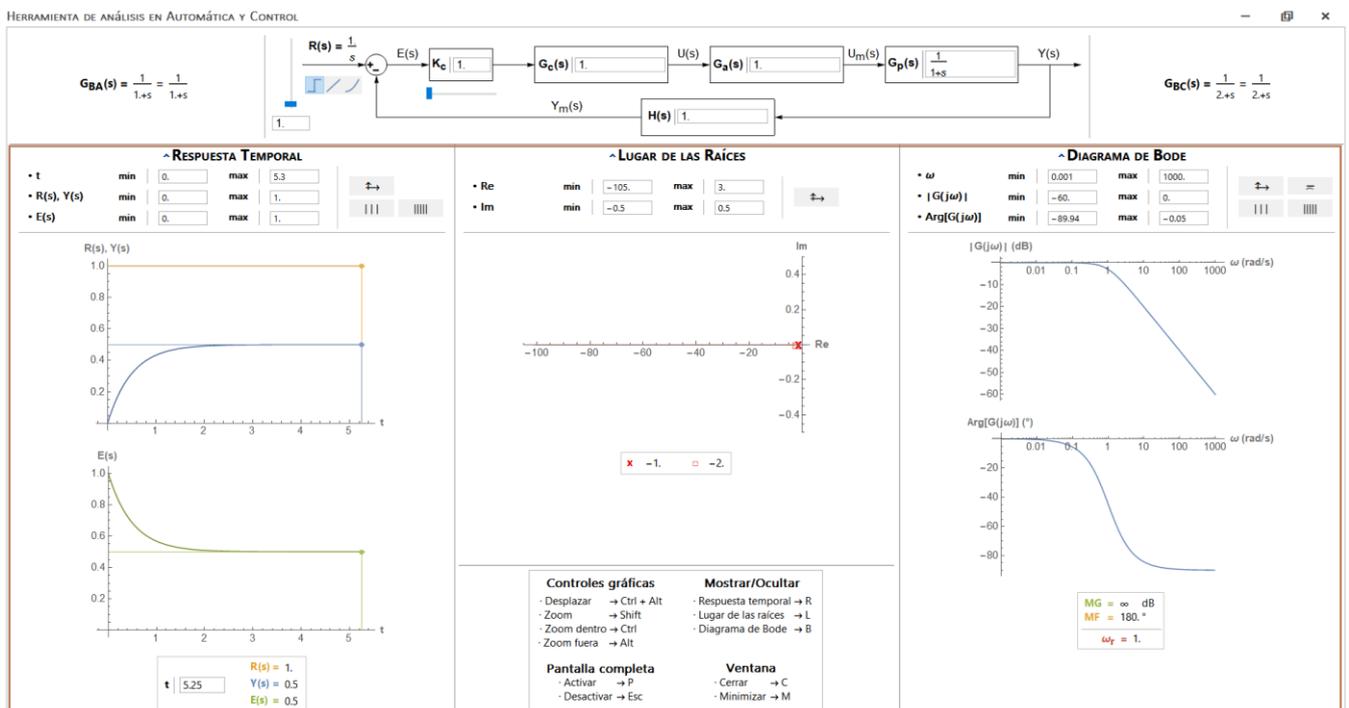


Figura 29. Zona de los análisis

Los análisis constan de una o dos gráficas, dependiendo del caso, y una leyenda con información adicional; y cada uno ocupa un tercio del ancho de la pantalla. A continuación, se detallan las características comunes a los tres análisis.

La primera característica común es que a la derecha de cada uno de los nombres de los análisis se encuentra un botón, representado por una flecha azul, que permite mostrar y ocultar el análisis en cuestión. Esto resulta útil para aumentar la velocidad de los cálculos, de forma que si uno o dos de los análisis son irrelevantes para el estudio que se esté haciendo, se pueden ocultar y de esta forma los cambios que se realicen se verán reflejados con mayor rapidez en los análisis que sí se muestran. También se pueden usar las teclas r, l y b para mostrar y ocultar el análisis temporal, de la ubicación de polos y ceros y frecuencial, respectivamente.



Figura 30. Botones para mostrar y ocultar los análisis

Otra característica común a todos los análisis es el tipo de controles que permiten modificar el rango de las gráficas de cada uno: los campos de entrada de los valores máximo y mínimo de los ejes, el botón para centrar los ejes y las combinaciones de teclas y ratón que permiten desplazarse y hacer zoom.

Los campos de entrada para los valores máximo y mínimo de los ejes permiten ajustar con precisión el rango de las gráficas. Se encuentran en la parte superior de cada uno de los análisis. La primera fila corresponde en todos los casos a las abscisas y la siguiente o las siguientes filas (dependiendo de si hay una o dos gráficas), a las ordenadas. El número de decimales así como el rango de valores para cada uno de ellos varían en cada uno de los tres análisis, por lo que estos valores se detallan más adelante.

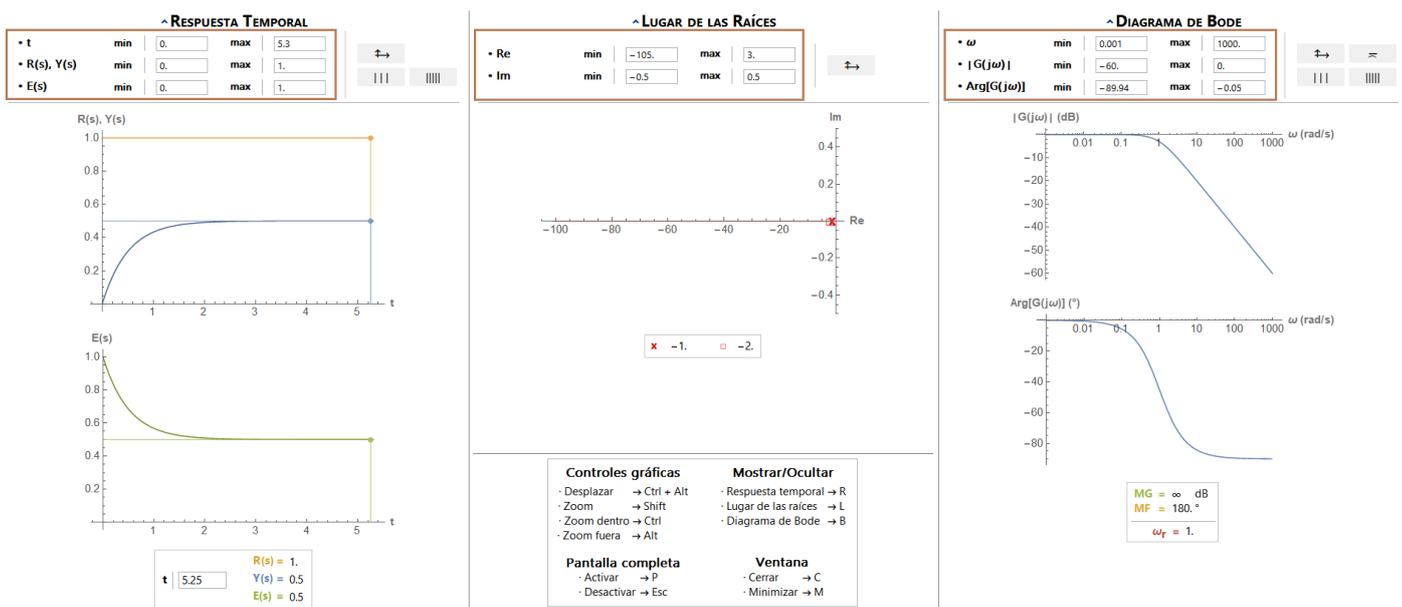


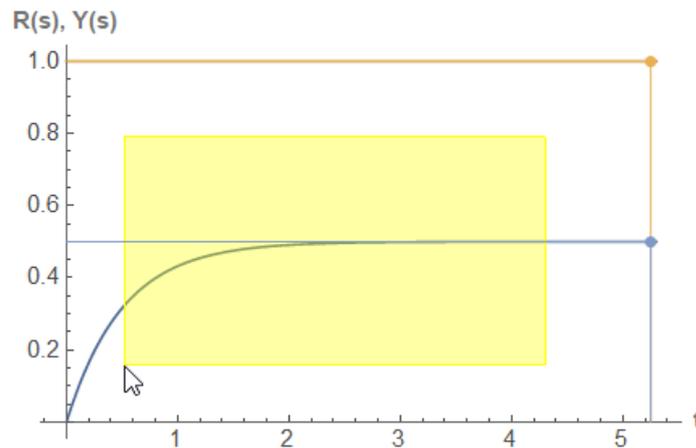
Figura 31. Campos de entrada de los ejes de los análisis

Por otro lado, el botón para centrar los ejes permite reajustar automáticamente el rango de los ejes de las gráficas a sus valores adecuados para que sea visible toda la información relevante. Este botón se encuentra a la derecha de los campos de entrada que sirven para modificar los valores máximo y mínimo de los ejes.



**Figura 32. Botón para reajustar los ejes de las gráficas de los análisis**

Por último, la herramienta también permite variar el rango con el ratón y una combinación de teclas para desplazarse por las gráficas y hacer zoom de forma cómoda para el usuario. Las combinaciones son las siguientes. Por un lado, manteniendo pulsadas las teclas Ctrl y Alt y arrastrando el ratón por las gráficas, se puede hacer un desplazamiento por ellas. Por otro lado, al mantener la tecla Shift pulsada y arrastrar el ratón por las gráficas, se muestra un rectángulo sobre el que se hará zoom cuando se suelte el ratón, siempre que la tecla Shift siga presionada. Por último, si se presionan las teclas Ctrl o Alt y se hace click sobre las gráficas, se realiza un zoom hacia adentro o hacia fuera, respectivamente, alrededor del punto en el que se hizo click.



**Figura 33. Ejemplo de zoom en una gráfica**

Además de los análisis, la herramienta cuenta con un cuadro de ayuda donde están señaladas las teclas que realizan funciones. Este cuadro se sitúa debajo del lugar de las raíces, en el medio de la interfaz, pegado al borde inferior y separado mediante un divisor horizontal.

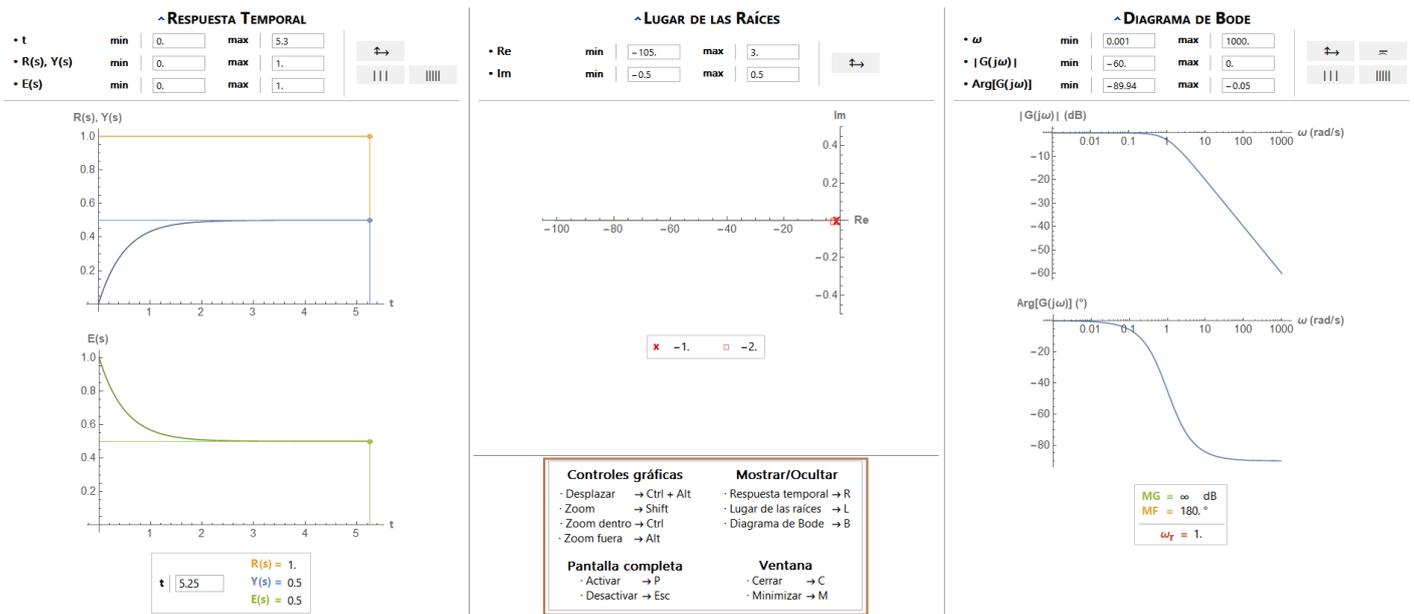


Figura 34. Cuadro de ayuda de la herramienta software

## 10.2. Resultados de los análisis

En este apartado se explican los resultados obtenidos relativos a los análisis del sistema que realiza la herramienta. Comienza presentando los análisis para luego analizar cada uno de ellos en apartados separados.

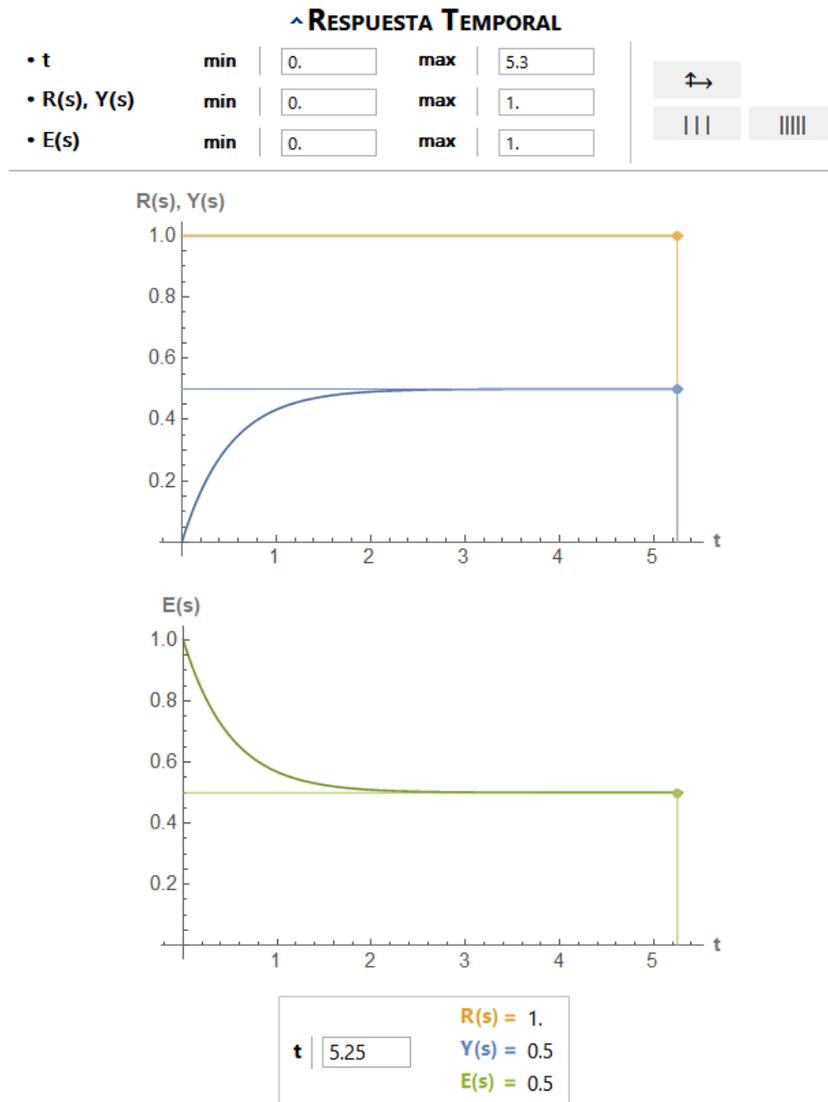
Como ya se ha comentado, los análisis del sistema mostrados por la herramienta son tres: el análisis temporal, el análisis de la ubicación de polos y ceros y el análisis frecuencial. Estos análisis son dinámicos, por lo que cualquier cambio realizado en los datos de entrada los hace variar de forma automática y simultánea (hasta donde lo permite la velocidad de computación). Esto permite hacer una asociación entre la expresión de las funciones de transferencia del sistema y sus características.

Todas las gráficas muestran los nombres de las variables representadas en cada eje junto a sus unidades, si las hubiera.

Por otro lado, cabe señalar que aunque Mathematica® integra funciones que permiten mostrar directamente las gráficas de los análisis (respuesta temporal, lugar de las raíces y diagrama de Bode), estas no se han empleado para aportar más flexibilidad al código de la herramienta.

### 10.2.1 Resultados del análisis temporal

El análisis temporal del sistema se muestra a la izquierda y consta de dos gráficas (la respuesta temporal) y una leyenda.



**Figura 35. Análisis temporal**

En la primera de las gráficas se representan la referencia y la salida del sistema y en la otra se representa el error, ambas respecto al tiempo. Cada una de las funciones tiene asociada un color para mayor claridad.

Los campos de entrada para modificar el rango de estas gráficas admiten todos 2 decimales como máximo. Los valores máximo y mínimo de las abscisas son 0.00 y 20.00, respectivamente; y los valores máximo y mínimo de las ordenadas, -110.00 y 110.00, respectivamente.

Este análisis también cuenta con dos botones situados junto al botón de centrar los ejes. Estos botones permiten mostrar y ocultar una cuadrícula de líneas en la gráfica. El primero, que tiene tres líneas verticales, permite mostrar líneas en los valores principales y el segundo, en todos los valores.



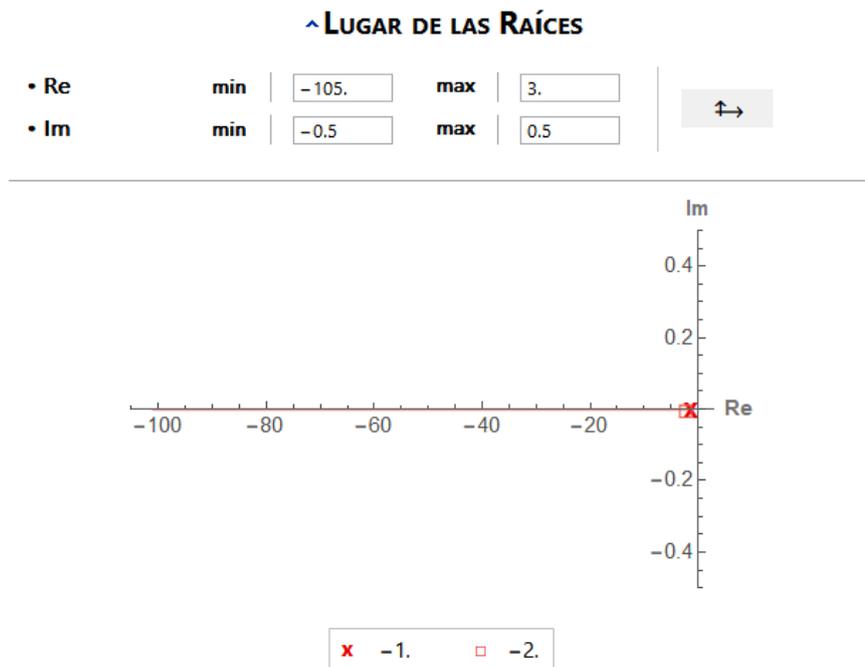
**Figura 36. Botones para mostrar y ocultar las cuadrículas de las gráficas del análisis temporal**

Además, las gráficas de este análisis cuentan con unas guías representadas por puntos. Estas guías se pueden mover haciendo click sobre ellas y arrastrándolas a lo largo de la función en la que se encuentran para variar el instante de tiempo en el que medir los valores de las funciones, mostrándose todos estos valores en la leyenda.

En la leyenda, como se ha mencionado, se muestra el valor del instante de tiempo seleccionado con las guías, así como los valores de las funciones ese instante, lo que permite medir los parámetros cuantificables más importantes del sistema: tiempo de establecimiento; valores de la salida y el error en régimen estacionario; sobreimpulso; etc. El valor del instante de tiempo es modificable no solo a través de las guías sino también mediante un campo de entrada en la leyenda. El campo de entrada admite 4 decimales, siendo el mismo número de decimales con los que se muestran los valores de las funciones, y el rango de valores válidos es el rango de las abscisas.

### 10.2.2 Resultados del análisis de polos y ceros

El análisis de polos y ceros del sistema se muestra en el centro y consta de una gráfica (el lugar de las raíces) y una leyenda.



**Figura 37. Análisis de la ubicación de polos y ceros**

En la gráfica se representan las trayectorias de los polos del sistema cuando varía la ganancia del controlador, así como los polos en bucle abierto (cuando la ganancia es 0), los polos en bucle cerrado (para la ganancia fijada por el usuario) y los ceros del sistema. Los polos en bucle abierto se representan por cruces; los polos en bucle cerrado, por cuadrados; y los ceros, por círculos. Cada polo y cada cero tiene un color asociado y, en el caso de los polos, la trayectoria correspondiente a cada uno tiene el mismo color que el polo.

Los campos de entrada para modificar el rango de estas gráficas admiten todos 2 decimales como máximo. Los valores máximo y mínimo de las abscisas y las ordenadas son -110.00 y 110.00, respectivamente.

La leyenda muestra los valores de los polos y ceros del sistema. La utilidad de esta leyenda es conocer de un vistazo los valores de los ceros y polos en bucle abierto, así como el valor de los polos en bucle cerrado para la ganancia del controlador fijada por el usuario, variando de forma dinámica cuando es modificada.

Por último, cabe señalar que el lugar de las raíces presenta discontinuidades en los colores utilizados para las trayectorias de cada polo en algunos sistemas con número impar de polos. Esto se debe a la discontinuidad de las raíces impares en el cuerpo de los números complejos. Se intentaron varias soluciones pero finalmente se optó por dejarlo así debido a que ninguna de las soluciones solucionaba completamente el problema.

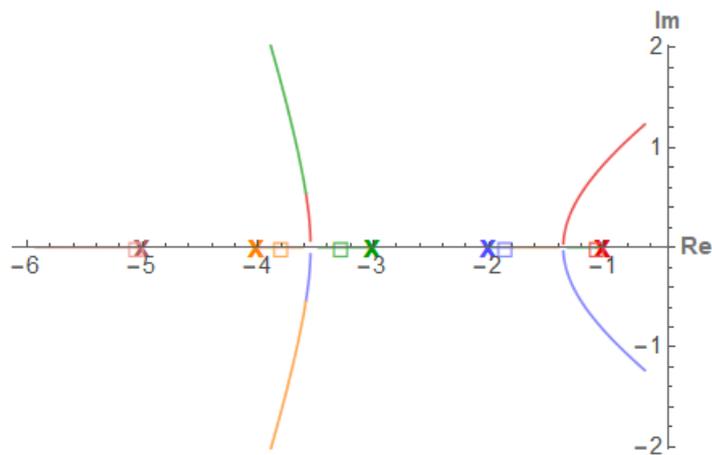
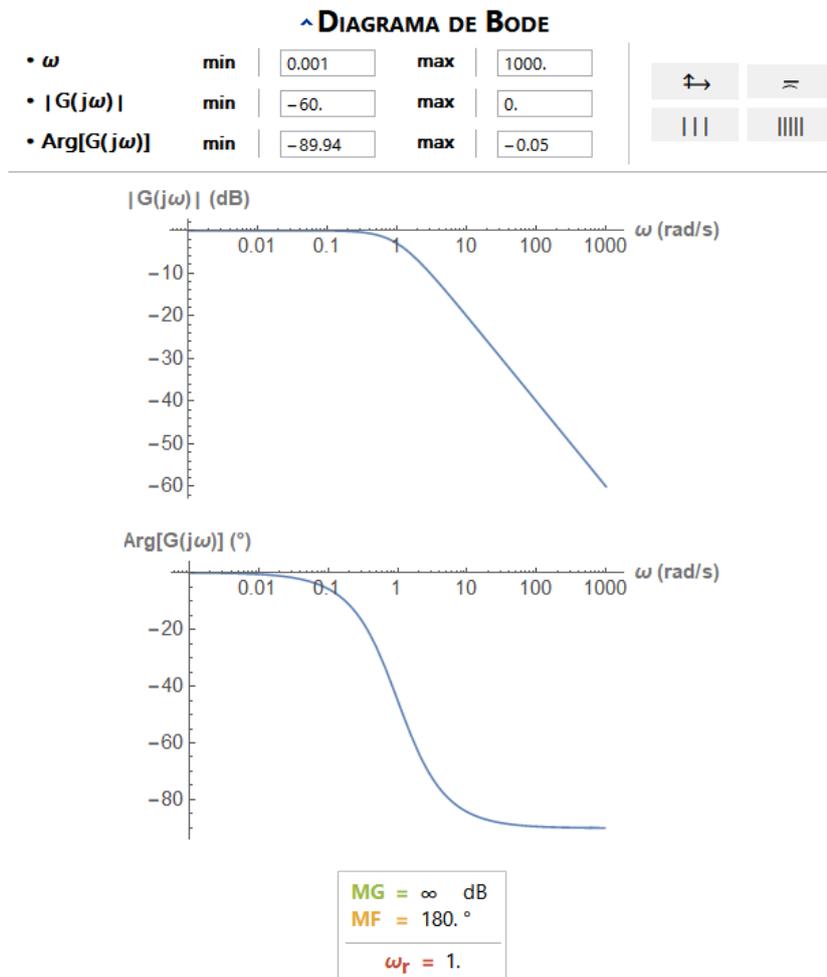


Figura 38. Ejemplo de discontinuidad del lugar de las raíces

### 10.2.3 Resultados del análisis frecuencial

El análisis frecuencial del sistema se muestra a la derecha y consta de dos gráficas (el diagrama de Bode) y una leyenda.



**Figura 39. Análisis frecuencial**

En la primera de las gráficas se representa el módulo de la función de transferencia en bucle abierto y, en la otra, su fase, ambas respecto a la frecuencia. En las gráficas también se representan los márgenes de ganancia y fase mediante flechas de colores distintos. Además, en la gráfica del módulo es posible visualizar el módulo asintótico.

Los campos de entrada para modificar las ordenadas de estas gráficas admiten 2 decimales como máximo. Los valores máximo y mínimo de las ordenadas de la gráfica del módulo son -200.00 y 200.00, respectivamente; y los de las ordenadas de la gráfica de la fase, -360.00 y 360.00. En cuanto a las abscisas, el número de decimales con los que se pueden introducir los valores máximo y mínimo varía dependiendo del valor que tomen. Si el valor es mayor o igual a 1, no admiten ningún decimal. En el resto de casos, el número de decimales admitidos es igual a una unidad mayor que el valor absoluto del exponente del 10 cuando el número está escrito en notación científica. Es decir, para valores entre 0.001 y 0.009 se admiten 3 decimales; entre 0.01 y 0.09, dos decimales; etc.

Este análisis, al igual que el análisis temporal, cuenta con dos botones situados junto al botón de centrar los ejes. Estos botones permiten mostrar y ocultar una cuadrícula de líneas en la gráfica. El primero, que tiene tres líneas verticales, permite mostrar

líneas en los valores principales y el segundo, en todos los valores. Además, este análisis cuenta con un cuarto botón que permite mostrar y ocultar el módulo asintótico.



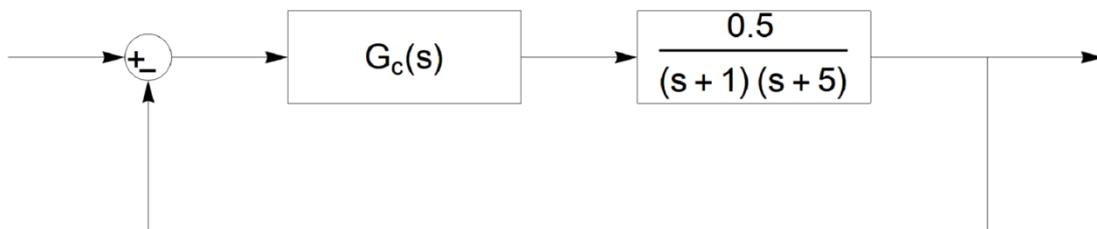
**Figura 40.** Izquierda: Botones para mostrar y ocultar las cuadrículas de las gráficas del análisis frecuencial. Derecha: Botón para mostrar y ocultar el módulo asintótico

En la leyenda se muestran los valores del margen de ganancia y del margen de fase con los mismos colores que sus flechas asociadas, así como las frecuencias de ruptura del sistema en color rojo. Todos estos valores se muestran con un máximo de 4 decimales.

### 10.3. Ejemplo de aplicación

En este apartado se presenta un ejemplo de aplicación de la herramienta software desarrollada en este proyecto. En este ejemplo se va a diseñar un controlador a partir de unas especificaciones dadas.

Sea el siguiente sistema realimentado, donde  $G_c(s)$  es la función de transferencia del controlador que se quiere diseñar.



**Figura 41.** Sistema realimentado utilizado en el ejemplo de aplicación

Las características que debe cumplir este sistema son:

- Error nulo ante una entrada escalón
- Tiempo de establecimiento (según el criterio del 2%) menor o igual a 2 segundos.
- Sobreimpulso menor o igual al 4,3%.

Lo primero que hay que notar es que como el sistema no tiene polos en el origen, no puede tener error nulo ante una entrada de tipo escalón. Por este motivo, no será suficiente con un controlador tipo P y se probará con un controlador tipo PI, ya que este introduce un polo en el origen.

$$G_c(s) = K_c \left( 1 + \frac{1}{T_i s} \right)$$

Se escoge el valor 1 para el tiempo integral ( $T_i$ ) ya que de esta forma se cancela el polo dominante del sistema.

Introduciendo estas funciones de transferencia en la herramienta software se puede comprobar si existe algún valor de la ganancia del controlador para el que se cumpla la condición del tiempo de establecimiento.

Modificando gradualmente el valor de la ganancia del controlador se observa que para valores comprendidos entre 13.85 y 24.5, aproximadamente, se cumplen todas las especificaciones impuestas.

En la siguiente figura se puede ver el análisis temporal del sistema para estos dos valores de la ganancia del controlador.

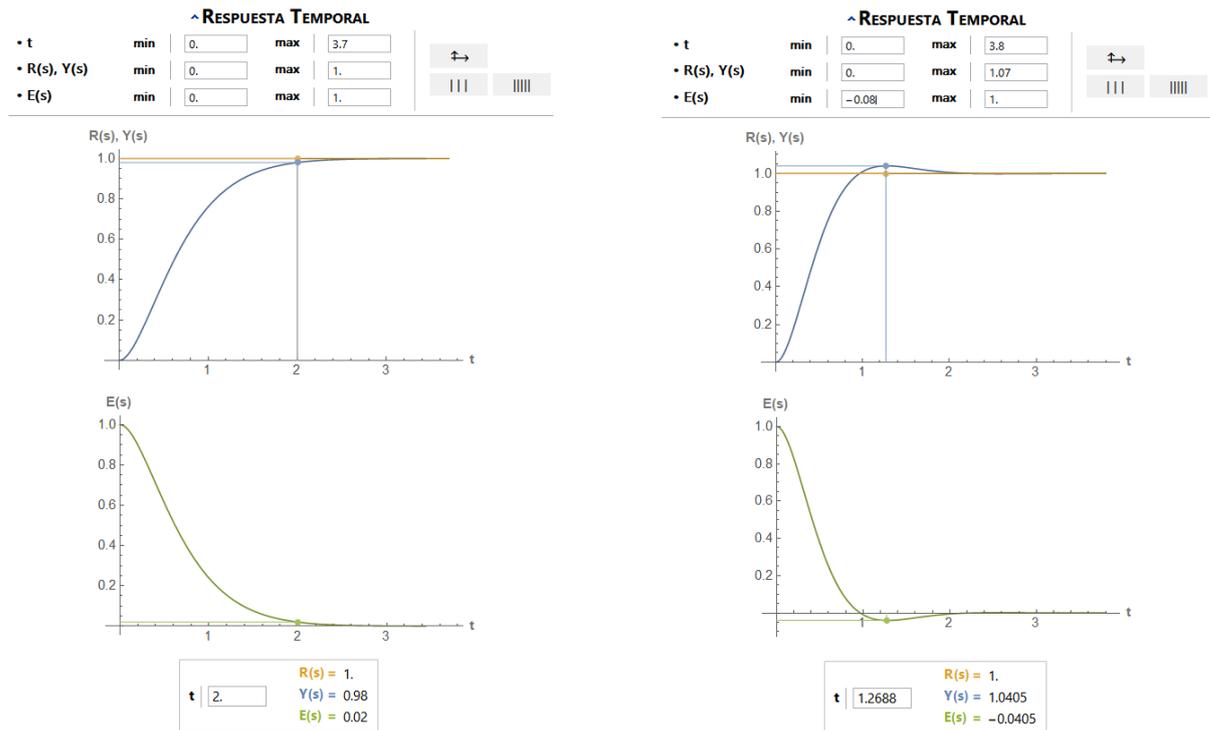


Figura 42. Análisis temporal con las ganancias del controlador mínima y máxima

Calculando los valores de forma teórica, se obtiene un rango de valores de 12.5 a 25, muy cercano a los valores reales.

De esta forma, se ha logrado diseñar el controlador empleando la herramienta software, así como comprobar los resultados obtenidos de forma teórica.

## 11. Descargo de gastos

En este apartado se describen los gastos originados en el desarrollo del proyecto. Comienza presentando los datos utilizados para obtener los gastos. Seguidamente, se da una visión completa de los mismos.

En la Tabla 2 se pueden ver los tres tipos de trabajos realizados en el proyecto, diseño, programación y validación, junto con el número de ingenieros senior y junior que los realizaron y el número de horas invertidas en cada uno.

| Horas internas | Nº Ingeniero Senior | Nº Ingeniero Junior | Duración |       |
|----------------|---------------------|---------------------|----------|-------|
| Diseño         | 1                   | 1                   | 10       | horas |
| Programación   | 0                   | 1                   | 230      | horas |
| Validación     | 0                   | 1                   | 20       | horas |

Tabla 2. Horas internas del proyecto

En la Tabla 3 se muestran las amortizaciones del proyecto, que son el ordenador y la licencia de Mathematica®, junto a su precio y vida útil. De estos dos datos se obtiene el coste unitario para ser utilizado posteriormente. La vida útil del ordenador se ha estimado en 5 años con un uso medio de 2 horas al día. La licencia de Mathematica® tiene una validez de 1 año y se ha estimado un uso medio de 2 horas al día.

| Amortizaciones       | Coste   | Vida útil |       | Coste unitario |
|----------------------|---------|-----------|-------|----------------|
| Ordenador            | 1.200 € | 3650      | horas | 0,33 €         |
| Licencia Mathematica | 160 €   | 730       | horas | 0,22 €         |

Tabla 3. Amortizaciones del proyecto

En la Tabla 4 se muestran los honorarios de los ingenieros senior y junior, fijados en 45 €/h y 30 €/h, respectivamente, así como los costes indirectos del proyecto, que se han estimado en un 5% de los costes directos.

|                        |    |                    |
|------------------------|----|--------------------|
| Coste Ingeniero Senior | 45 | €/hora             |
| Coste Ingeniero Junior | 30 | €/hora             |
| Costes indirectos      | 5% | de costes directos |

Tabla 4. Honorarios de los ingenieros y costes indirectos del proyecto

Con los datos de las tres tablas anteriores se ha elaborado el descargo de gastos del proyecto, obteniendo un coste total del proyecto de 8.806,34 €, siendo el coste directo del mismo de 8.386,99 €.

Como se puede observar en la Tabla 5, los gastos del proyecto proceden prácticamente en su totalidad de las horas de trabajo del ingeniero junior, siendo la mayor parte de ellas horas de programación.

Los siguientes gastos considerables, aunque un orden de magnitud menor que el anterior, son las horas de diseño del ingeniero senior. Un valor similar alcanzan los costes indirectos.

Por último, se puede ver que la amortización del ordenador y la licencia de Mathematica® apenas han supuesto un gasto en el proyecto, pero se ha decidido incluir estos gastos para reflejar todos los gastos derivados del desarrollo del proyecto.

| Concepto               | Nº unidades | Coste unitario | Coste             |
|------------------------|-------------|----------------|-------------------|
| <b>Horas internas</b>  |             |                | <b>8.250 €</b>    |
| Ingeniero Senior       | 10 h        | 45 €           | 450 €             |
| Ingeniero Junior       | 260 h       | 30 €           | 7.800 €           |
| <b>Amortizaciones</b>  |             |                | <b>136,99 €</b>   |
| Ordenador              | 250 h       | 0,33 €         | 82,19 €           |
| Licencia Mathematica   | 250 h       | 0,22 €         | 54,79 €           |
| <b>COSTES DIRECTOS</b> |             |                | <b>8.386,99 €</b> |
| Costes Indirectos      | 5%          |                | 419,35 €          |
| <b>TOTAL</b>           |             |                | <b>8.806,34 €</b> |

Tabla 5. Descargo de gastos del proyecto

## 12. Conclusiones

Empleando el entorno Mathematica®, se ha logrado crear una herramienta software que permite dar soporte a alumnos de asignaturas donde se trate el Análisis y Diseño de Sistemas de Control.

Para ello, la herramienta es capaz de facilitar tres análisis de un sistema realimentado definido por sus funciones de transferencia:

- Análisis temporal (respuesta temporal)
- Análisis de la ubicación de los polos y ceros (denominado habitualmente como lugar de las raíces)
- Análisis frecuencial (también denominado diagrama de Bode)

Estos análisis se muestran a través de gráficas y valores de los parámetros de la dinámica y la estabilidad del sistema.

La herramienta está concebida para que no requiera conocimiento previo para ser utilizada. Para ello, emplea una interfaz clara con controles intuitivos donde el sistema se muestra junto a sus análisis. Además, estos últimos varían automática y simultáneamente con las modificaciones de los datos de entrada. Estos dos hechos permiten realizar una asociación entre los parámetros del sistema y sus características de estabilidad y dinámica.

De esta forma, la herramienta ofrece una alternativa gratuita, open source y sencilla de usar a entornos profesionales como Matlab®, Labview® o el propio Mathematica®.

## 13. Bibliografía

La bibliografía consultada a lo largo del proyecto es la siguiente:

- [1] Departamento de Ingeniería de Sistemas y Automática, Cuaderno de seminarios y prácticas, Sección de Publicaciones de la EIB, 2015.
- [2] J. L. Guzmán, R. Costa, M. Berenguel y S. Dormido, Control Automático con herramientas interactivas, Pearson, 2012.
- [3] K. Ogata, Ingeniería de Control Moderna, 2010.
- [4] «Labview®,» [En línea]. Available: <https://www.ni.com/es-es/shop/labview.html>.
- [5] «Mathematica®,» [En línea]. Available: <http://www.wolfram.com/mathematica/>.
- [6] «Matlab®,» [En línea]. Available: <https://www.mathworks.com/products/matlab.html>.
- [7] «Control System Toolbox,» [En línea]. Available: <https://www.mathworks.com/products/control.html>.
- [8] «Documentación de Mathematica®,» [En línea]. Available: <https://reference.wolfram.com/language/>.

## **14. Anexo I: Código**

En este anexo se incluye el código completo de la herramienta software desarrollada en este proyecto.





```

CONTROL=====
=====*)
(*-----Referencia y
Sistema-----*)
MostrarRefySist = True,
u = 1.,
R = 1. / s,
Kc = 1.,
Gc = 1.,
Ga = 1.,
Gp = 1 / (s + 1),
H = 1.,
(*-----Respuesta
temporal-----*)
LineasGridRT = None,
tmax,
tmin,
SalRefmin,
SalRefmax,
Errormin,
Errormax,
(*-----Lugar de las
raíces-----*)
Remin,
Remax,
Immin,
Immax,
(*-----Diagrama de
Bode-----*)
VisualizarModuloAsintotico = False,
LineasGridBode = None,
ωmin,
ωmax,
Modulomin,
Modulomax,
Argumentomin,
Argumentomax,
(*=====
INTERNAS=====
=====*)
PosicionInicial,
RangoInicial,
RangoProvisional,
PuntoZoom1,
PuntoZoom2,
(*-----Referencia y
Sistema-----*)
GBAFactorizada,
GBCFactorizada,
(*-----Respuesta
temporal-----*)
Salida,
Referencia,
Error,
Guiat,
Guiasalida,
Guiareferencia,
Guierror,
RangoAutomaticoRT,
HaciendoZoomSalRef = False,
HaciendoZoomError = False,
(*-----Lugar de las
raíces-----*)

```

```

EcParametricas,
MarcadoresPolosBA,
MarcadoresPolosBC,
MarcadoresCeros,
Leyenda,
Colores,
RangoAutomaticoLR,
HaciendoZoomLR = False,
(*-----Diagrama de
  Bode-----*)
FuncionModulo,
FuncionArgumento,
MargenDeGanancia,
MargenDeFase,
PuntosFlechaMG,
PuntosFlechaMF,
BodeAsintotico,
PuntosFrecuenciasRuptura,
LeyendaFrecuenciasRuptura,
RangoAutomaticoModulo,
RangoAutomaticoArgumento,
HaciendoZoomModulo = False,
HaciendoZoomArgumento = False,
(*=====
  FUNCIONES=====
  =====*)
(*-----Estilos de
  texto-----*)
SNegrita,
SApartado,
SSubApartado,
SControlesRango,
SApartadosAyuda,
STextoAyuda,
SNombresEjes,
SGbaGbc,
SGbaGbcPeq,
STextoDB,
SLeyendasGraficas,
SLeyendasGraficasColor,
SMarcadoresLR,
(*-----Cálculos-----
  -----*)
AjustarPuntos,
ActualizarPuntosGuia,
ActualizarSistema,
ActualizarRT,
ActualizarLR,
ActualizarBode,
(*-----Controles-----
  -----
  -----*)
ControlRango,
ControlLogRango,
(*-----Sistema-----
  -----
  -----*)
DiagramaDeBloques,
(*-----Respuesta
  temporal-----*)
RespuestaTemporalSistRef,
RespuestaTemporalError,
RespuestaTemporalValores,

```



```
SMarcadoresLR = Style[#1, Bold, FontSize → TamMarcadoresLR,
  FontProperties → {"ScreenResolution" → ResolucionPantalla}] &;
```

```
(*=====
CÁLCULOS=====
====*)
```

```
(*-----Ajustar puntos en los zooms
de las gráficas-----*)
```

```
SetAttributes[AjustarPuntos, HoldAll];
AjustarPuntos[VarX_, VarY_, XMin_, XMax_, YMin_, YMax_] := (
  If[VarX < XMin,
    VarX = XMin,
    (*Else*)
    If[VarX > XMax,
      VarX = XMax];
  If[VarY < YMin,
    VarY = YMin,
    (*Else*)
    If[VarY > YMax,
      VarY = YMax]];
];
```

```
(*-----Actualizar los puntos
de las guías-----*)
```

```
ActualizarPuntosGuia[Valor_] :=
  If[Valor ≤ 0,
    Guiat = 0.;
    GuiaSalida = Round[Salida[Guiat], DecimalesRT];
    GuiaReferencia = Round[Referencia[Guiat], DecimalesRT];
    GuiaError = Round[Error[Guiat], DecimalesRT],
    (*Else*)
    If[Valor ≥ tAbsMax,
      Guiat = tAbsMax;
      GuiaSalida = Round[Salida[Guiat], DecimalesRT];
      GuiaReferencia = Round[Referencia[Guiat], DecimalesRT];
      GuiaError = Round[Error[Guiat], DecimalesRT],
      (*Else*)
      Guiat = Round[Valor, DecimalesRT];
      GuiaSalida = Round[Salida[Guiat], DecimalesRT];
      GuiaReferencia = Round[Referencia[Guiat], DecimalesRT];
      GuiaError = Round[Error[Guiat], DecimalesRT]
    ]
  ];
```

```
(*-----Cálculos
Sistema-----*)
```

```
ActualizarSistema[] := Block[
  {Funcion},
  Funcion = Simplify[Kc Gc Ga Gp H];
  GBFAfactorizada =
    N@Rationalize[Factor[Numerator@Funcion] / Factor[Denominator@Funcion]];
  Funcion = Simplify[ $\frac{Kc Gc Ga Gp}{1. + Kc Gc Ga Gp H}$ ];
  GBCFactorizada =
    N@Rationalize[Factor[Numerator@Funcion] / Factor[Denominator@Funcion]];
  GBCFactorizada = N@Rationalize@GBCFactorizada;
];
```

```
(*-----Cálculos Respuesta
temporal-----*)
```

```
ActualizarRT[] := Block[
  {ValoresSalRef,
  ValoresError,
```

```

i,
ValorActual,
ValorAntiguo,
ValorFinal,
tMaximo},
Salida[t_] = InverseLaplaceTransform[ $\frac{Kc Gc Ga Gp}{1. + Kc Gc Ga Gp H} * Ru, s, t$ ];
Referencia[t_] = InverseLaplaceTransform[Ru, s, t];
Error[t_] = InverseLaplaceTransform[ $\frac{1.}{1. + Kc Gc Ga Gp H} * Ru, s, t$ ];
(*.....Rango automático.....*)
ValorFinal = Limit[ $\frac{Kc Gc Ga Gp}{1. + Kc Gc Ga Gp H} * Ru * s, s \rightarrow 0.$ ];
If[(ValorFinal  $\notin$  Reals) || (ValorFinal === Indeterminate),
ValoresSalRef = {If[R === 1./s,
Chop@Referencia[1.],
(*Else*)
Table[Chop@Referencia[i], {i, 0., tAbsMax, 0.1}
]},
Table[Chop@Salida[i], {i, 0., tAbsMax, 0.1}],
0.},
(*Else*)
ValoresSalRef = {If[R === 1./s,
Chop@Referencia[1.],
(*Else*)
Table[Chop@Referencia[i], {i, 0., tAbsMax, 0.1}
]},
If[Abs[ValorFinal - Chop@Salida[0.]] < 0.00000005,
Table[Chop@Salida[i], {i, 0., tAbsMax, 0.1}],
(*Else*)
Join[
{ValorActual = Chop@Salida[0.]},
Reap[Do[
ValorAntiguo = ValorActual;
ValorActual = Chop@Salida[i];
If[Abs[ValorFinal - ValorActual] > 0.00000005 &&
Quiet@(Abs[ValorActual - ValorAntiguo] / Abs[ValorActual]) > 0.000005,
Sow@ValorActual,
(*Else*)
Break[]
],
{i, 0.1, tAbsMax, 0.1}]]][2, 1]]
],
0.};
];
tMaximo = (Length[ValoresSalRef][2] - 1) * 0.1;
ValoresError = Chop@{Table[Error[i], {i, 0., tMaximo, 0.1}], 0.};
RangoAutomaticoRT = {{0., tMaximo},
Round[MinMax[ValoresSalRef], RTPaso], Round[MinMax[ValoresError], RTPaso]};
{{tmin, tmax}, {SalRefmin, SalRefmax}, {Errormin, Errormax}} = RangoAutomaticoRT;
(*.....Puntos guía.....*)
Guiat = tMaximo - 5 * RTPaso;
GuiaSalida = Round[Salida[Guiat], DecimalesRT];
GuiaReferencia = Round[Referencia[Guiat], DecimalesRT];
GuiaError = Round[Error[Guiat], DecimalesRT];
];
(*-----Cálculos Lugar de las raíces-----*)
ActualizarLR[] := Block[

```

```

{Expresiones,
 kRuptura,
 OrdenPolosCeros,
 SolucionPolos, (*para la leyenda ya que están como un número complejo*)
 nPolos,
 SolucionCeros, (*para la leyenda ya que están como un número complejo*)
 nCeros,
 PolosYCeros, (*para la representación de los polos y ceros;
 se guardan en formato de punto, i.e., {Re[],Im[]}*)
 ValoresLR},
(*.....Cálculo polos (raíces denominador).....*)
SolucionPolos = s /. NSolve[1 + k Gc Ga Gp H == 0., s];
nPolos = Length[SolucionPolos];
(*.....Ordenar polos.....*)
OrdenPolosCeros = Ordering[SolucionPolos /. k -> 0., nPolos, (Re[#1] > Re[#2]) &];
SolucionPolos = SolucionPolos[[OrdenPolosCeros]];
(*.....Corrección de algunos sistemas.....*)
If[nPolos == 3,
 Expresiones = Cases[SolucionPolos, _1/3, Infinity];
 Expresiones = DeleteDuplicates[Expresiones];
 Quiet[kRuptura = k /. Assuming[
   k > 0. && Element[k, Reals], Solve[Expresiones == 0. && k > 0., k, Reals]]];
 If[kRuptura != k,
  kRuptura = kRuptura[[1]];
  SolucionPolos = Piecewise[
   { {#, k < kRuptura},
     {# /.
      { x_ ^ (1/3) => x ^ (1/3) * (Cos[4. π/3.] + i * Sin[4. π/3.]), y_ / x_ ^ (1/3) =>
        y / (x ^ (1/3) * (Cos[4. π/3.] + i * Sin[4. π/3.])), k ≥ kRuptura }
   ] & /@ SolucionPolos
 ];
];
If[EvenQ@nPolos && nPolos > 2,
 PolosYCeros = Table[{Re[SolucionPolos[[i]], (-1)i * Abs@Im[SolucionPolos[[i]]],
  {i, nPolos}], (*convertir los polos a puntos*)
 (*Else*)
 PolosYCeros = Table[{Re[SolucionPolos[[i]], Im[SolucionPolos[[i]]], {i, nPolos}}]
 (*convertir los polos a puntos*)
 ];
PolosYCeros = Join[PolosYCeros /. k -> 0., PolosYCeros];
(*añadir los puntos (polos) iniciales (en BA, cuando k = 0)*)
(*.....Cálculo ceros (raíces numerador).....*)
SolucionCeros = s /. NSolve[Gc Ga Gp == 0., s];
(*cálculo de los ceros*)
If[SolucionCeros === s,
 SolucionCeros = {}
 ];
nCeros = Length[SolucionCeros];
If[nCeros ≠ 0,
 (*.....Ordenar ceros.....*)
 OrdenPolosCeros = Ordering[SolucionCeros, nCeros, (Re[#1] > Re[#2]) &];
 SolucionCeros = SolucionCeros[[OrdenPolosCeros]];
 PolosYCeros = Join[PolosYCeros, Table[{Re[SolucionCeros[[i]],
  Im[SolucionCeros[[i]]], {i, nCeros}}] (*convertir los ceros a puntos*)
 ];
(*.....Trayectorias de los polos (i.e., lugar de las raíces).....*)
EcParametricas =
 Table[{PolosYCeros[[i, 1]], PolosYCeros[[i, 2]], {i, nPolos + 1, 2 nPolos}}];
(*.....Polos y ceros.....*)
MarcadoresPolosBA = Flatten[
 Table[{ColoresBase[[i]], Text[SMarcadoresLR["x"], PolosYCeros[[i]]], {i, nPolos}}];

```

```

MarcadoresPolosBC = Flatten[Table[{ColoresBaseClaros[[i - nPolos]],
  Text[SMarcadoresLR["□"], PolosYCeros[[i]] /. k → Hold[Kc]]}, {i,
  nPolos + 1, 2 nPolos}]];
MarcadoresCeros = Flatten[Table[{ColoresBase[[i - nPolos]], Text[
  SMarcadoresLR["o"], PolosYCeros[[i]]}, {i, 2 nPolos + 1, 2 nPolos + nCeros}]];
(*.....Leyenda.....*)
Leyenda = {
  Table[{SLEYendasGraficasColor["x", ColoresBase[[i]]], Item[SLEYendasGraficas@Round[
    SolucionPolos[[i]] /. k → 0., DecimalesLR], Alignment → Left]}, {i, nPolos}},
  Table[{SLEYendasGraficasColor["□", ColoresBaseClaros[[i]]],
    Item[SLEYendasGraficas@Round[SolucionPolos[[i]] /. k → Hold[Kc], DecimalesLR],
    Alignment → Left]}, {i, nPolos}},
  Table[{SLEYendasGraficasColor["o", ColoresBase[[i + nPolos]]],
    Item[SLEYendasGraficas@Round[SolucionCeros[[i]], DecimalesLR],
    Alignment → Left]}, {i, nCeros}]];
If[nCeros ≠ 0,
  Leyenda[[3]] = Join[Leyenda[[3]], Table[{"", ""}, {i, nPolos - nCeros}]];
Leyenda = Partition[Flatten[Thread[{Leyenda[[1]], Leyenda[[2]], Leyenda[[3]]}], 6],
(*Else*)
Leyenda = Take[Leyenda, 2];
Leyenda = Partition[Flatten[Thread[{Leyenda[[1]], Leyenda[[2]]}], 4]
];
(*.....Colores plot.....*)
Colores = Take[ColoresBaseClaros, nPolos];
(*.....Rango automático.....*)
ValoresLR =
  Chop@Join[Table[SolucionPolos /. k → i, {i, {0., KcMax}}], SolucionCeros];
RangoAutomaticoLR = Round[{MinMax[Re@ValoresLR], MinMax[Im@ValoresLR]}, LRPaso];
(*__por si son iguales__*)
If[RangoAutomaticoLR[[1, 1]] == RangoAutomaticoLR[[1, 2]],
  RangoAutomaticoLR[[1, 1]] = RangoAutomaticoLR[[1, 1]] - 0.5;
  RangoAutomaticoLR[[1, 2]] = RangoAutomaticoLR[[1, 2]] + 0.5;
];
If[RangoAutomaticoLR[[2, 1]] == RangoAutomaticoLR[[2, 2]],
  RangoAutomaticoLR[[2, 1]] = RangoAutomaticoLR[[2, 1]] - 0.5;
  RangoAutomaticoLR[[2, 2]] = RangoAutomaticoLR[[2, 2]] + 0.5;
];
(*__para que se vean mejor los marcadores__*)
ValoresLR = RangoAutomaticoLR[[1, 2]] - RangoAutomaticoLR[[1, 1]];
RangoAutomaticoLR[[1, 1]] = RangoAutomaticoLR[[1, 1]] - 0.04 * ValoresLR;
RangoAutomaticoLR[[1, 2]] = RangoAutomaticoLR[[1, 2]] + 0.04 * ValoresLR;
{{Remin, Remax}, {Immin, Immax}} = RangoAutomaticoLR;
];

(*-----Cálculos Diagrama
de Bode-----*)
ActualizarBode[] := Block[
  {Funcion,
  Numerador,
  Denominador,
  Componentes,
  CorteFase, (*frecuencia de cruce de fase*)
  CorteModulo, (*frecuencia de cruce de ganancia*)
  ValoresModulo,
  ValoresArgumento,
  Coeficientes,
  Pendiente = 0.,
  FrecuenciasRuptura = {}, (*frecuencia,
  1er/2o orden (20 o 40), repeticiones (simple, doble, triple...),
  del numerador/denominador (1 o -1), e.g. {2.06, 20, 1, -1}*),
  CoordenadasY},
  (*-----Diagrama de

```

```

Bode-----*)
(*.....Inicializar variables.....*)
FuncionArgumento = 0.;
MargenDeGanancia = -∞; (*cuando se muestra se le pone un -*)
MargenDeFase = 0.; (*cuando se muestra se le suma 180*)
PuntosFlechaMG = {{}};
PuntosFlechaMF = {{}};
(*.....Función módulo 1.....*)
FuncionModulo = Kc Gc Ga Gp H;
(*.....Función argumento.....*)
Numerador = FactorList[Numerator[FuncionModulo]];
Numerador = Numerador /. s → i ω;
Denominador = FactorList[Denominator[FuncionModulo]];
Denominador = Denominador /. s → i ω;
Do[FuncionArgumento = FuncionArgumento + Numerador[[i, 2]] *
    ArcTan[Re[Numerador[[i, 1]], Im[Numerador[[i, 1]]], {i, Length[Numerador]}]];
Do[FuncionArgumento = FuncionArgumento - Denominador[[i, 2]] * ArcTan[
    Re[Denominador[[i, 1]], Im[Denominador[[i, 1]]], {i, Length[Denominador]}]];
FuncionArgumento = FuncionArgumento / Degree;
(*.....Función módulo 2.....*)
FuncionModulo = FuncionModulo /. s → i ω;
FuncionModulo = ComplexExpand[FuncionModulo];
(*.....Componentes de la función.....*)
Componentes =
    Assuming[ω ∈ Reals && ω > 0., Simplify[{Re[FuncionModulo], Im[FuncionModulo]}]];
(*esta operación hace que los cálculos sean un poco más rápidos*)
Componentes = Rationalize[Componentes];
(*.....Función módulo 3.....*)
FuncionModulo =
    Assuming[ω ∈ Reals && ω > 0., Simplify[20. * Log10[Abs[FuncionModulo]]]];
(*.....Margen de ganancia.....*)
(*NOTA: ArcTan[Componentes[[1]], Componentes[[2]]] == -π ⇒
    Componentes[[2]] == 0 && Componentes[[1]] < 0*)
CorteFase = NSolve[Componentes[[2]] == 0. && Componentes[[1]] > 0. && ω > 0., ω, Reals];
If[CorteFase != {},
    CorteFase = ω /. CorteFase[[1]];
    MargenDeGanancia = Round[FuncionModulo /. ω → CorteFase, DecimalesBode];
    PuntosFlechaMG = {{Log[CorteFase], MargenDeGanancia}, {Log[CorteFase], 0}}];
(*.....Margen de fase.....*)
(*NOTA: 20Log10[(Componentes[[1]]^2 + Componentes[[2]]^2)^1/2] == 0 ⇒
    Componentes[[1]]^2 + Componentes[[2]]^2 == 1*)
CorteModulo = NSolve[Componentes[[1]]^2 + Componentes[[2]]^2 == 1. && ω > 0., ω, Reals];
If[CorteModulo != {},
    CorteModulo = ω /. CorteModulo[[1]];
    MargenDeFase = Round[FuncionArgumento /. ω → CorteModulo, DecimalesBode];
    PuntosFlechaMF = {{Log[CorteModulo], -180.}, {Log[CorteModulo], MargenDeFase}}];
(*.....Rango automático.....*)
ValoresModulo =
    Chop@Table[FuncionModulo /. ω → i, {i, {1./ωAbsMax, 1./ωAbsMax * 10., 1./ωAbsMax *
        100., 1./ωAbsMax * 1000., ωAbsMax/10., ωAbsMax/100., ωAbsMax}}];
If[MargenDeGanancia ≠ -∞,
    ValoresModulo = Join[ValoresModulo, {0.}]
];
RangoAutomaticoModulo = Ceiling[MinMax[ValoresModulo], BodePaso];
ValoresArgumento = Chop@Table[FuncionArgumento /. ω → i,
    {i, {1./ωAbsMax, 1./ωAbsMax * 10., 1./ωAbsMax * 100.,
        1./ωAbsMax * 1000., ωAbsMax/10., ωAbsMax/100., ωAbsMax}}];
If[MargenDeFase ≠ 0.,
    ValoresArgumento = Join[ValoresArgumento, {-180., 0.}]
];
RangoAutomaticoArgumento = Ceiling[MinMax[ValoresArgumento], BodePaso];
{ωmin, ωmax} = {1./ωAbsMax, ωAbsMax};

```

```

{Modulomin, Modulomax} = RangoAutomaticoModulo;
{Argumentomin, Argumentomax} = RangoAutomaticoArgumento;
(*-----Diagrama de Bode
asintótico-----*)
(*.....Coordenadas
          X.....*)
Funcion = Kc Gc Ga Gp H;
Numerador = FactorList[Numerator[Funcion]];
Denominador = FactorList[Denominator[Funcion]];
(*.....Quitar factor común.....*)
Numerador = Numerador[[2 ;;]];
Denominador = Denominador[[2 ;;]];
(*.....Quitar ceros/polos origen.....*)
If[Rationalize[Denominador[[1, 1]]] === s,
  Pendiente = -Denominador[[1, 2]] * 20.;
  Denominador = Denominador[[2 ;;]];
  (*Else*)
  If[Length[Numerador] > 1,
    If[Rationalize[Numerador[[1, 1]]] === s,
      Pendiente = Numerador[[1, 2]] * 20.;
      Numerador = Numerador[[2 ;;]];
    ]
  ]
];
(*.....Frecuencias de ruptura.....*)
Do[Coefficientes = CoefficientList[Numerador[[i, 1]], s];
  If[Length[Coefficientes] == 3,
    FrecuenciasRuptura =
      Append[FrecuenciasRuptura, { $\sqrt{\text{Coefficients}[[1]}}$ , 40. * Numerador[[i, 2]]}],
    (*Else*)
    FrecuenciasRuptura = Append[FrecuenciasRuptura,
      {Coefficientes[[1]] / Coefficients[[2]], 20. * Numerador[[i, 2]]}],
    {i, Length[Numerador]}];
Do[Coefficientes = CoefficientList[Denominator[[i, 1]], s];
  If[Length[Coefficientes] == 3,
    FrecuenciasRuptura =
      Append[FrecuenciasRuptura, { $\sqrt{\text{Coefficients}[[1]}}$ , -40. * Denominator[[i, 2]]}],
    (*Else*)
    FrecuenciasRuptura = Append[FrecuenciasRuptura,
      {Coefficientes[[1]] / Coefficients[[2]], -20. * Denominator[[i, 2]]}],
    {i, Length[Denominator]}];
(*.....Ordenar de menor a mayor.....*)
FrecuenciasRuptura = Sort[FrecuenciasRuptura, #1[[1]] < #2[[1]] &];
(*añadimos la frecuencia mínima y la máxima del código*)
(*NOTA: asumimos que ninguna frecuencia del sistema
es menor que la mínima ni mayor que la máxima del código*)
FrecuenciasRuptura = Join[{1. /  $\omega_{AbsMax}$ , 0.}, FrecuenciasRuptura,
  {{ $\omega_{AbsMax}$ , 0.}}];
(*.....Coordenadas
          Y.....*)
(*.....Primer punto.....*)
CoordenadasY = {FuncionModulo /.  $\omega \rightarrow \text{FrecuenciasRuptura}[[1, 1]]$ };
(*.....Segundo punto.....*)
(*NOTA: depende de los ceros/polos en el origen*)
If[Pendiente  $\neq$  0.,
  CoordenadasY = Append[CoordenadasY, Pendiente *
    (Log10[FrecuenciasRuptura[[2, 1]]] - Log10[FrecuenciasRuptura[[1, 1]]]),
  (*Else*)
  CoordenadasY = Append[CoordenadasY, CoordenadasY[[1]]]
];
(*.....Resto de puntos.....*)

```

```

Do[
  Pendiente = Pendiente + FrecuenciasRuptura[[i - 1, 2]];
  CoordenadasY = Append[CoordenadasY, Pendiente *
    (Log10[FrecuenciasRuptura[[i, 1]]] - Log10[FrecuenciasRuptura[[i - 1, 1]]]),
    {i, 3, Length[FrecuenciasRuptura]}];
(*.....Sumar valores acumulativamente.....*)
If[CoordenadasY[[1]] ≠ CoordenadasY[[2]],
  (*si hay ceros/polos en el origen,
  hay pendiente no nula y hay que sumar los puntos desde el primero*)
  CoordenadasY = FoldList[Plus, CoordenadasY[[1]], CoordenadasY[[2 ;;]],
  (*Else*)
  (*si no hay ceros/polos en el origen, hay pendiente nula y hay que
  sumar los puntos desde el segundo, que es igual que el primero*)
  CoordenadasY = Prepend[FoldList[Plus, CoordenadasY[[2]],
    CoordenadasY[[3 ;;]], CoordenadasY[[1]]
];
(*.....Extraer solo frecuencias de ruptura.....*)
FrecuenciasRuptura = Flatten[FrecuenciasRuptura];
(*NOTA: Hay que usar Log y no Log10 para los puntos por el tipo de Plot*)
FrecuenciasRuptura = N[Log[FrecuenciasRuptura[;; ; 2]]];
(*.....Valores devueltos.....*)
BodeAsintotico = Thread[{FrecuenciasRuptura, CoordenadasY}];
FrecuenciasRuptura = FrecuenciasRuptura[[2 ;; -2]];
PuntosFrecuenciasRuptura = Thread[{FrecuenciasRuptura, 0.}];
FrecuenciasRuptura = Round[e^FrecuenciasRuptura, DecimalesBode];
LeyendaFrecuenciasRuptura = Table[{Tooltip[SleyendasGraficasColor[" $\omega_r =$ ", ],
  "Frecuencias de ruptura", TooltipDelay → DelayTooltip],
  SleyendasGraficas@FrecuenciasRuptura[[i]], {i, Length[FrecuenciasRuptura]}];
If[Length[FrecuenciasRuptura] > 6,
  LeyendaFrecuenciasRuptura = Join[LeyendaFrecuenciasRuptura,
    Table[{"", ""}, {i, 12 - Length[FrecuenciasRuptura]}]];
  LeyendaFrecuenciasRuptura = Partition[LeyendaFrecuenciasRuptura, 6];
  LeyendaFrecuenciasRuptura = Partition[Flatten[
    Thread[{LeyendaFrecuenciasRuptura[[1]], LeyendaFrecuenciasRuptura[[2]]}], 4
];
];

(*=====
CONTROLES=====
=====*)
(*-----Controles
normales-----*)
(*NOTA: ΔVariable tiene significados distintos en la versión de
Unido (diferencia entre VariableMax y VariableMin) y la
de Zoom (diferencia entre el nuevo valor y el antiguo)*)
SetAttributes[ControlRango, HoldAll];
ControlRango[VariableMin_, VariableMax_, Minimo_, Maximo_, Paso_] :=
Grid[{
  {SControlesRango["min"],
    InputField[Dynamic[VariableMin, (
      If[# ≥ Minimo,
        If[# < VariableMax,
          VariableMin = Floor[#, Paso],
          (*Else*)
          VariableMin = VariableMax - Paso]]
    ) &], ImageSize → TamInputFieldRango, BaseStyle → {TamLetraControlesRango,
    FontProperties → {"ScreenResolution" → ResolucionPantalla}}],
    SControlesRango["max"],
    InputField[Dynamic[VariableMax, (
      If[# ≤ Maximo,
        If[# > VariableMin,

```

```

        VariableMax = Floor[#, Paso],
        (*Else*)
        VariableMax = VariableMin + Paso]]
    ) &], ImageSize → TamInputFieldRango, BaseStyle → {TamLetraControlesRango,
    FontProperties → {"ScreenResolution" → ResolucionPantalla}}]]}
}, Alignment → {Center, Center}, ItemSize →
  {{{Scaled@0.15, Scaled@0.3}}, Automatic},
Dividers → {{False, Lighter[Gray, 0.3], False, Lighter[Gray, 0.3], False}, False}];

(*-----Controles
  logarítmicos-----*)
SetAttributes[ControlLogRango, HoldAll];
ControlLogRango[VariableMin_, VariableMax_, Minimo_, Maximo_] :=
Grid[{
  {SControlesRango["min"],
  InputField[Dynamic[VariableMin, (
    If[# ≥ Minimo,
    If[# < VariableMax,
    VariableMin = Floor[#, If[# < 1., 10.^ (MantissaExponent[#][2] - 1), 1]],
    (*Else*)
    If[MantissaExponent[VariableMax][1] == 0.1, (*esto es para los casos
    límite (0.001, 0.01, 0.1 y 1), para que VariableMin no se haga 0*)
    VariableMin = VariableMax - If[VariableMax ≤ 1., 10.^
    (MantissaExponent[VariableMax][2] - 2), 1],
    (*Else*)
    VariableMin = VariableMax -
    If[VariableMax < 1., 10.^ (MantissaExponent[VariableMax][2] - 1), 1]
    ]
    ]]]
  ) &], ImageSize → TamInputFieldRango, BaseStyle → {TamLetraControlesRango,
  FontProperties → {"ScreenResolution" → ResolucionPantalla}}],
  SControlesRango["max"],
  InputField[Dynamic[VariableMax, (
    If[# ≤ Maximo,
    If[Floor[#, If[# < 1., 10.^ (MantissaExponent[#][2] - 1), 1]] >
    VariableMin, (*aquí la comparación es diferente porque por
    ejemplo 4.5 es mayor que 4, pero después de quitarle los
    decimales se quedaría en 4, es decir, igual que VariableMin*)
    VariableMax = Floor[#, If[# < 1., 10.^ (MantissaExponent[#][2] - 1), 1]],
    (*Else*)
    VariableMax = VariableMin +
    If[VariableMin < 1, 10.^ (MantissaExponent[VariableMin][2] - 1), 1]
    ]]]
  ) &], ImageSize → TamInputFieldRango, BaseStyle → {TamLetraControlesRango,
  FontProperties → {"ScreenResolution" → ResolucionPantalla}}]]}
}, Alignment → {Center, Center}, ItemSize →
  {{{Scaled@0.15, Scaled@0.3}}, Automatic},
Dividers → {{False, Lighter[Gray, 0.3], False, Lighter[Gray, 0.3], False}, False}];

(*=====
  SISTEMA=====
  =====*)
DiagramaDeBloques[] = With[
  {L = 2., (*longitud de las flechas*)
  b = 3., (*base del rectángulo pequeño*)
  h = 1.5, (*altura del rectángulo pequeño*)
  B = 6.28, (*base de los rectángulos grandes*)
  Hr = 1.75, (*altura de los rectángulos grandes*)
  r = 0.5, (*radio del círculo*)
  dFL = 0.5, (*distancia de las flechas a las letras*)

```

```

EscalonImagen = Tooltip[Plot[UnitStep[t], {t, -0.5, 0.5},
  AspectRatio → 1, Axes → False, ExclusionsStyle → Automatic, ImageSize →
  TamGraficasBotonesEntrada], "Escalón", TooltipDelay → DelayTooltip],
RampaImagen = Tooltip[Plot[t, {t, 0., 1.}, AspectRatio → 1., Axes → False,
  ImageSize → TamGraficasBotonesEntrada], "Rampa", TooltipDelay → DelayTooltip],
ParabolaImagen = Tooltip[Plot[t^2, {t, 0., 1.}, AspectRatio → 1.,
  Axes → False, ImageSize → TamGraficasBotonesEntrada],
  "Parábola", TooltipDelay → DelayTooltip]],
Deploy[Graphics[{
  (*-----Flechas-----*)
  (*-----Flechas-----*)
  Arrowheads[0.01],
  (*.....R(s).....*)
  Arrow[{{-0.5 L, 0.}, {L, 0.}}],
  (*.....R(s).....*)
  (*O → Kc*)
  Arrow[{{L + 2 r, 0.}, {2 L + 2 r, 0.}}],
  (*.....R(s).....*)
  (*Kc → Gc*)
  Arrow[{{2 L + 2 r + b, 0.}, {3 L + 2 r + b, 0.}}],
  (*.....R(s).....*)
  (*Gc → Ga*)
  Arrow[{{3 L + 2 r + b + B, 0.}, {4 L + 2 r + b + B, 0.}}],
  (*.....R(s).....*)
  (*Ga → Gp*)
  Arrow[{{4 L + 2 r + b + 2 B, 0.}, {5 L + 2 r + b + 2 B, 0.}}],
  (*.....R(s).....*)
  (*Y(s)*)
  Arrow[{{5 L + 2 r + b + 3 B, 0.}, {5 L + 1.5 L + 2 r + b + 3 B, 0.}}],
  (*.....R(s).....*)
  (*Gp → H*)
  Arrow[{{5 L + 0.75 L + 2 r + b + 3 B, 0.}, {5 L + 0.75 L + 2 r + b + 3 B, -1.25 L},
    {L + r + (r + 4 L + 0.5 L + 3 B + b) / 2. + 0.5 B, -1.25 L}}],
  (*.....R(s).....*)
  (*H → O*)
  Arrow[{{L + r + (r + 4 L + 0.5 L + 3 B + b) / 2. - 0.5 B, -1.25 L},
    {L + r, -1.25 L}, {L + r, -r}}],
  (*-----Círculo-----*)
  (*-----*)
  Circle[{L + r, 0.}, r],
  (*-----Rectángulos-----*)
  (*-----*)
  Opacity[0.],
  White,
  EdgeForm[Thin],
  (*.....Kc(s).....*)
  Rectangle[{2 L + 2 r, -0.5 h}, {2 L + 2 r + b, 0.5 h}],
  (*.....Gc(s).....*)
  Rectangle[{3 L + 2 r + b, -0.5 Hr}, {3 L + 2 r + b + B, 0.5 Hr}],
  (*.....Ga(s).....*)
  Rectangle[{4 L + 2 r + b + B, -0.5 Hr}, {4 L + 2 r + b + 2 B, 0.5 Hr}],
  (*.....Gp(s).....*)
  Rectangle[{5 L + 2 r + b + 2 B, -0.5 Hr}, {5 L + 2 r + b + 3 B, 0.5 Hr}],
  (*.....H(s).....*)
  Rectangle[{L + r + (r + 4 L + 0.5 L + 2 B + b) / 2., - (1.25 L + 0.5 Hr)},
    {L + r + (r + 4 L + L / 2. + 2 B + b) / 2. + B, - (1.25 L - 0.5 Hr)}],
  (*-----Texto y controles-----*)
  Opacity[1.],
  Black,
  FontWeight → Bold,

```

```

Text[SNegrita@STextoDB@"+", {L + r/2., 0}],
Text[SNegrita@STextoDB@"-", {L + r, -r/2.}],
FontWeight → Plain,
(*.....R(s).....*)
Inset[
Grid[{
  {VerticalSlider[Dynamic[u, (u = #;
    ActualizarRT[]) &], {UPaso,
    UMax, UPaso}, ImageSize → {Automatic, TamSliderEntrada}]},
  {InputField[Dynamic[u, (
    If[# ≥ UMax,
      u = UMax,
      (*Else*)
      If[# ≤ UPaso,
        u = UPaso,
        (*Else*)
        u = Floor[#, UPaso]
      ]];
    ActualizarRT[];
    SelectionMove[EvaluationNotebook[], All, Cell]
  ) &], FieldSize →
    TamInputFieldSistPeq, BaseStyle → {TamLetraInputFieldSist,
    FontTracking → "Narrow", FormatType → StandardForm,
    FontProperties → {"ScreenResolution" → ResolucionPantalla}}]}],
  Alignment → {Center, Center}, Spacings → {0., {0., 1., 0.}},
  {-1.5, -1}],
Text[Row[{Tooltip[SNegrita@STextoDB@"R(s) = ", "Referencia", TooltipDelay →
  DelayTooltip], STextoDB@Dynamic[R * u]}], {0.25 L, 1.5 * dFL}],
Inset[SetterBar[Dynamic[R, (R = #;
  ActualizarRT[]) &], {1./s → EscalonImagen,
  1./s2 → RampaImagen, 2./s3 → ParabolaImagen}], {0.5, -1}],
(*.....E(s).....*)
Tooltip[Text[STextoDB@"E(s)", {L + 0.5 L + 2 r, dFL}],
"Error", TooltipDelay → DelayTooltip],
(*.....Kc(s).....*)
Inset[
Grid[{
  {Tooltip[SNegrita@STextoDB@"Kc",
  "Ganancia del controlador", TooltipDelay → DelayTooltip],
  InputField[Dynamic[Kc, (
    If[# ≥ KcMax,
      Kc = KcMax,
      (*Else*)
      If[# ≤ KcPaso,
        Kc = KcPaso,
        (*Else*)
        Kc = Round[#, KcPaso]
      ]];
    ActualizarSistema[];
    If[MostrarRTExt, ActualizarRT[]];
    If[MostrarBodeExt, ActualizarBode[]];
    SelectionMove[EvaluationNotebook[], All, Cell]
  ) &], FieldSize →
    TamInputFieldSistPeq, BaseStyle → {TamLetraInputFieldSist,
    FontTracking → "Narrow", FormatType → StandardForm,
    FontProperties → {"ScreenResolution" → ResolucionPantalla}}]}],
  Alignment → {Center, Center},
  ItemSize → {Automatic, 1.9},
  Dividers → {{False, Gray, False}, False}],
  {2 L + 2 r + 0.52 b, -0.02}],

```

```

Inset[Slider[Dynamic[Kc, (Kc = #;
  ActualizarSistema[];
  If[MostrarRTEExt, ActualizarRT[]];
  If[MostrarBodeExt, ActualizarBode[]] &],
  {KcPaso, KcMax, KcPaso}, ImageSize → TamSliderEntrada],
{2 L + 2 r + 0.52 b, -1.4}],
(*.....Gc(s).....*)
Inset[
Grid[{
  {Tooltip[SNegrita@STextoDB@"Gc(s)",
    "Función de transferencia del controlador", TooltipDelay → DelayTooltip],
  InputField[Dynamic[Gc, (
    If[(Variables[Numerator@#] == {s} || NumericQ[Numerator@#]) &&
      (Variables[Denominator@#] == {s} || NumericQ[Denominator@#]),
      Gc = #;
      ActualizarSistema[];
      If[MostrarRTEExt, ActualizarRT[]];
      If[MostrarBodeExt, ActualizarBode[]];
      If[MostrarLREExt, ActualizarLR[]];
      SelectionMove[EvaluationNotebook[], All, Cell] &],
    FieldSize → TamInputFieldSist, BaseStyle → {TamLetraInputFieldSist,
      FontTracking → "Narrow", FormatType → StandardForm,
      FontProperties → {"ScreenResolution" → ResolucionPantalla}}]
  },
  Alignment → {Center, Center},
  ItemSize → {Automatic, 2.4},
  Dividers → {{False, Gray, False}, False}],
{3 L + 2 r + b + 0.51 B, -0.03}],
(*.....U(s).....*)
Text[STextoDB@"U(s)", {3 L + 0.5 L + 2 r + b + B, dFL}],
(*.....Ga(s).....*)
Inset[
Grid[{
  {Tooltip[SNegrita@STextoDB@"Ga(s)",
    "Función de transferencia del actuador", TooltipDelay → DelayTooltip],
  InputField[Dynamic[Ga, (
    If[(Variables[Numerator@#] == {s} || NumericQ[Numerator@#]) &&
      (Variables[Denominator@#] == {s} || NumericQ[Denominator@#]),
      Ga = #;
      ActualizarSistema[];
      If[MostrarRTEExt, ActualizarRT[]];
      If[MostrarBodeExt, ActualizarBode[]];
      If[MostrarLREExt, ActualizarLR[]];
      SelectionMove[EvaluationNotebook[], All, Cell] &],
    FieldSize → TamInputFieldSist, BaseStyle → {TamLetraInputFieldSist,
      FontTracking → "Narrow", FormatType → StandardForm,
      FontProperties → {"ScreenResolution" → ResolucionPantalla}}]
  },
  Alignment → {Center, Center},
  ItemSize → {Automatic, 2.4},
  Dividers → {{False, Gray, False}, False}],
{4 L + 2 r + b + B + 0.51 B, -0.03}],
(*.....Um(s).....*)
Text[Style[STextoDB@"Um(s)",
  "SingleLetterItalics" → False], {4 L + 0.5 L + 2 r + b + 2 B, dFL}],
(*.....Gp(s).....*)
Inset[
Grid[{
  {Tooltip[SNegrita@STextoDB@"Gp(s)",
    "Función de transferencia de la planta", TooltipDelay → DelayTooltip],

```

```

InputField[Dynamic[Gp, (
  If[(Variables[Numerator@#] == {s} || NumericQ[Numerator@#]) &&
    (Variables[Denominator@#] == {s} || NumericQ[Denominator@#]),
    Gp = #;
    ActualizarSistema[];
    If[MostrarRTEExt, ActualizarRT[]];
    If[MostrarBodeExt, ActualizarBode[]];
    If[MostrarLREExt, ActualizarLR[]];
    SelectionMove[EvaluationNotebook[], All, Cell] &],
  FieldSize → TamInputFieldSist, BaseStyle → {TamLetraInputFieldSist,
  FontTracking → "Narrow", FormatType → StandardForm,
  FontProperties → {"ScreenResolution" → ResolucionPantalla}}]
}],
Alignment → {Center, Center},
ItemSize → {Automatic, 2.4},
Dividers → {{False, Gray, False}, False}],
{5 L + 2 r + b + 2 B + 0.51 B, -0.03}],
(*.....Y(s).....*)
Tooltip[Text[STextoDB@"Y(s)", {5 L + 0.75 L + 2 r + b + 3 B, dFL}],
"Salida", TooltipDelay → DelayTooltip],
(*.....H(s).....*)
Inset[
Grid[{
  {Tooltip[SNegrita@STextoDB@"H(s)",
    "Función de transferencia del sensor", TooltipDelay → DelayTooltip],
  InputField[Dynamic[H, (
    If[(Variables[Numerator@#] == {s} || NumericQ[Numerator@#]) &&
      (Variables[Denominator@#] == {s} || NumericQ[Denominator@#]),
      H = #;
      ActualizarSistema[];
      If[MostrarRTEExt, ActualizarRT[]];
      If[MostrarBodeExt, ActualizarBode[]];
      If[MostrarLREExt, ActualizarLR[]];
      SelectionMove[EvaluationNotebook[], All, Cell] &],
    FieldSize → TamInputFieldSist, BaseStyle → {TamLetraInputFieldSist,
    FontTracking → "Narrow", FormatType → StandardForm,
    FontProperties → {"ScreenResolution" → ResolucionPantalla}}]
  }],
Alignment → {Center, Center},
ItemSize → {Automatic, 2.4},
Dividers → {{False, Gray, False}, False}],
{L + r + (r + 4 L + L / 2. + 2 B + b) / 2. + 0.51 B, -1.25 L}],
(*.....Ym(s).....*)
Text[Style[STextoDB@"Ym(s)", "SingleLetterItalics" → False],
{L / 2. + r + (r + 3 L + L / 2. + 2 B) / 2., -1.25 L + dFL}]
}],
PlotRange → {{-2.6, 36.}, {1.4, -3.5}},
ImageSize → TamDB]]
];

```

```

(*=====
RESPUESTA
TEMPORAL=====*)

```

```

RespuestaTemporalSistRef [] :=
EventHandler[
  Deploy[Dynamic[Show[
    Plot[
      {Button[Chop@Salida[t], ActualizarPuntosGuia[MousePosition["Graphics"][[1]]],
      Button[Chop@Referencia[t],

```

```

    ActualizarPuntosGuia [MousePosition["Graphics"] [[1]]]],
    {t, tmin, tmax},
    PlotRange → {{tmin, tmax}, {SalRefmin, SalRefmax}},
    PlotRangePadding → Scaled[0.04],
    AxesOrigin → If[SalRefmin > 0. && SalRefmax > 0.,
        {tmin, SalRefmin},
        (*Else*)
        If[SalRefmin < 0. && SalRefmax < 0.,
            {tmin, SalRefmax},
            (*Else*)
            {tmin, 0.}
        ]],
    PlotStyle → {Darker[■, 0.2], Darker[■, 0.2]},
    GridLines → LineasGridRT,
    Epilog → Reap[If[Guiat > tmax,
        If[(SalRefmin ≤ GuiaReferencia) && (GuiaReferencia ≤ SalRefmax),
            Sow@{Lighter[■, 0.2], Line[
                {{tmin, Dynamic@GuiaReferencia}, {tmax, Dynamic@GuiaReferencia}}]},
            (*Else*)
            Sow@{}];
        If[(SalRefmin ≤ GuiaSalida) && (GuiaSalida ≤ SalRefmax),
            Sow@{Lighter[■, 0.2],
                Line[{{tmin, Dynamic@GuiaSalida}, {tmax, Dynamic@GuiaSalida}}]},
            (*Else*)
            Sow@{}],
        (*Else*)
        If[(tmin ≤ Guiat) && (Guiat ≤ tmax),
            If[(SalRefmin > GuiaReferencia) && (SalRefmin < 0.),
                Sow@{Lighter[■, 0.2], Line[{{Dynamic@Guiat, SalRefmin},
                    {Dynamic@Guiat, Min[{SalRefmax, 0.}]}]}]},
                (*Else*)
                If[(GuiaReferencia > SalRefmax) && (SalRefmax > 0.),
                    Sow@{Lighter[■, 0.2], Line[{{Dynamic@Guiat, SalRefmax},
                        {Dynamic@Guiat, Max[{SalRefmin, 0.}]}]}]},
                    (*Else*)
                    If[(SalRefmin ≤ GuiaReferencia) && (GuiaReferencia ≤ SalRefmax),
                        Sow@{Lighter[■, 0.2], PointSize[TamPuntoGuias],
                            Point[Dynamic@{Guiat, GuiaReferencia}], Line[{{tmin,
                                Dynamic@GuiaReferencia}, Dynamic@{Guiat, GuiaReferencia},
                                    {Dynamic@Guiat, TakeLargestBy[{Max[{SalRefmin, 0.}],
                                        Min[{SalRefmax, 0.}]}], Abs, 1] [[1]]}}]},
                            (*Else*)
                            Sow@{}]]];
            If[(SalRefmin > GuiaSalida) && (SalRefmin < 0.),
                Sow@{Lighter[■, 0.2], Line[{{Dynamic@Guiat, SalRefmin},
                    {Dynamic@Guiat, TakeSmallest[{SalRefmax, 0.}, 1] [[1]]}}]},
                (*Else*)
                If[(GuiaSalida > SalRefmax) && (SalRefmax > 0.),
                    Sow@{Lighter[■, 0.2], Line[{{Dynamic@Guiat, SalRefmax},
                        {Dynamic@Guiat, TakeLargest[{SalRefmin, 0.}, 1] [[1]]}}]},
                    (*Else*)
                    If[(SalRefmin ≤ GuiaSalida) && (GuiaSalida ≤ SalRefmax),
                        Sow@{Lighter[■, 0.2],
                            PointSize[TamPuntoGuias], Point[Dynamic@{Guiat, GuiaSalida}],
                                Line[{{tmin, Dynamic@GuiaSalida}, Dynamic@{Guiat, GuiaSalida},
                                    {Dynamic@Guiat, TakeLargestBy[{TakeLargest[{SalRefmin, 0.}, 1] [[1]],
                                        TakeSmallest[{SalRefmax, 0.}, 1] [[1]], Abs, 1] [[1]]}}]},
                                (*Else*)
                                Sow@{}]]],
                        (*Else*)
                        Sow@{}]
                    ]
                ]
            ]
        ]
    ]

```

```

    ]][2, 1]
  ],
Graphics[ {
  Locator[Dynamic[{Guiat, GuiaReferencia},
    (ActualizarPuntosGuia[#[[1]]) &], Appearance → None],
  Locator[Dynamic[{Guiat, GuiaSalida}, (ActualizarPuntosGuia[#[[1]]) &],
    Appearance → None],
  If[HaciendoZoomSalRef,
    {Yellow,
     EdgeForm[Yellow],
     Opacity[0.35],
     Rectangle[PuntoZoom1, Dynamic[MousePosition["Graphics"]]]}],
  }],
ImagePadding → {{25., 15.}, {9., 27.}},
TicksStyle → Directive[TamLetraTicksEjes, Darker[Gray, 0.4],
  FontProperties → {"ScreenResolution" → ResolucionPantalla}],
AxesLabel → {SNombresEjes@"t", SNombresEjes@"R(s), Y(s)"},
AspectRatio → RelAspectoGraficas,
ImageSize → Dynamic@If[CurrentValue[EvaluationNotebook[], WindowSize] ===
  {Full, Full}, TamGraficasPantallaCompleta, TamGraficas]
], TrackedSymbols ⇒ {Salida, Referencia, Guiat, HaciendoZoomSalRef,
  LineasGridRT, tmin, tmax, SalRefmin, SalRefmax}],
{"MouseClicked", 1} ⇒
Which[
  (*.....Zoom dentro.....*)
CurrentValue["ControlKey"],
RangoProvisional =
  Floor[({{tmin, tmax}, {SalRefmin, SalRefmax}} - MousePosition["Graphics"]) *
    FraccionZoom + MousePosition["Graphics"], RTPaso];
AjustarPuntos[RangoProvisional[[1, 1]], RangoProvisional[[2, 1]],
  0., tAbsMax, -yAbsMax, yAbsMax];
AjustarPuntos[RangoProvisional[[1, 2]], RangoProvisional[[2, 2]],
  0., tAbsMax, -yAbsMax, yAbsMax];
{{tmin, tmax}, {SalRefmin, SalRefmax}} = RangoProvisional,
(*.....Zoom fuera.....*)
CurrentValue["AltKey"],
RangoProvisional =
  Floor[({{tmin, tmax}, {SalRefmin, SalRefmax}} - MousePosition["Graphics"]) /
    FraccionZoom + MousePosition["Graphics"], RTPaso];
AjustarPuntos[RangoProvisional[[1, 1]], RangoProvisional[[2, 1]],
  0., tAbsMax, -yAbsMax, yAbsMax];
AjustarPuntos[RangoProvisional[[1, 2]], RangoProvisional[[2, 2]],
  0., tAbsMax, -yAbsMax, yAbsMax];
{{tmin, tmax}, {SalRefmin, SalRefmax}} = RangoProvisional],
{"MouseDown", 1} ⇒
Which[
  (*.....Moverse 1.....*)
CurrentValue["ControlKey"] && CurrentValue["AltKey"],
PosicionInicial = MousePosition["GraphicsImageScaled"];
RangoInicial = {{tmin, tmax}, {SalRefmin, SalRefmax}},
(*.....Zoom rectángulo 1.....*)
CurrentValue["ShiftKey"],
PuntoZoom1 = Floor[MousePosition["Graphics"], RTPaso];
AjustarPuntos[PuntoZoom1[[1]], PuntoZoom1[[2]], 0., tAbsMax, -yAbsMax, yAbsMax];
HaciendoZoomSalRef = True],
(*.....Moverse 2.....*)
{"MouseDragged", 1} ⇒
If[CurrentValue["ControlKey"] && CurrentValue["AltKey"],
RangoProvisional =
  RangoInicial + Floor[(PosicionInicial - MousePosition["GraphicsImageScaled"]) *
    (RangoInicial[[All, 2]] - RangoInicial[[All, 1])], RTPaso];

```

```

If[RangoProvisional[[1, 1]] ≥ 0. && RangoProvisional[[1, 2]] ≤ tAbsMax,
  {tmin, tmax} = RangoProvisional[[1]];
If[RangoProvisional[[2, 1]] ≥ -yAbsMax && RangoProvisional[[2, 2]] ≤ yAbsMax,
  {SalRefmin, SalRefmax} = RangoProvisional[[2]]
],
(*.....Zoom rectángulo 2.....*)
{"MouseUp", 1} => (
  If[CurrentValue["ShiftKey"] && HaciendoZoomSalRef,
    PuntoZoom2 = Floor[MousePosition["Graphics"], RTPaso];
    If[PuntoZoom2 ≠ PuntoZoom1,
      AjustarPuntos[PuntoZoom2[[1]], PuntoZoom2[[2]], 0., tAbsMax, -yAbsMax, yAbsMax];
      {{tmin, tmax}, {SalRefmin, SalRefmax}} =
        Map[Sort, Thread[{{PuntoZoom1, PuntoZoom2}}] (*hace falta todo esto
          porque el rectángulo se puede hacer desde cualquier vértice*)
        ]
    ];
    HaciendoZoomSalRef = False)},
PassEventsDown → True
];

```

RespuestaTemporalError[] :=

```

EventHandler[
  Deploy[Dynamic[Show[
    Plot[Button[Chop@Error[t],
      ActualizarPuntosGuia[MousePosition["Graphics"][[1]], {t, tmin, tmax},
      PlotRange → {{tmin, tmax}, {Errormin, Errormax}},
      PlotRangePadding → Scaled[0.04],
      AxesOrigin → If[Errormin > 0 && Errormax > 0,
        {tmin, Errormin},
        (*Else*)
        If[Errormin < 0 && Errormax < 0,
          {tmin, Errormax},
          (*Else*)
          {tmin, 0}
        ]],
      PlotStyle → Darker[■, 0.2],
      GridLines → LineasGridRT,
      Epilog → If[Guiat > tmax,
        If[(Errormin ≤ GuiaError) && (GuiaError ≤ Errormax),
          {Lighter[■, 0.2], Line[
            {{tmin, Dynamic@GuiaError}, {tmax, Dynamic@GuiaError}}]],
          (*Else*)
          {}],
        (*Else*)
        If[(tmin ≤ Guiat) && (Guiat ≤ tmax),
          If[(Errormin > GuiaError) && (Errormin < 0),
            {Lighter[■, 0.2], Line[
              {Dynamic@Guiat, Errormin}, {Dynamic@Guiat, Min[{Errormax, 0}]}]],
            (*Else*)
            If[(GuiaError > Errormax) && (Errormax > 0),
              {Lighter[■, 0.2], Line[
                {Dynamic@Guiat, Errormax}, {Dynamic@Guiat, Max[{Errormin, 0}]}]],
              (*Else*)
              If[(Errormin ≤ GuiaError) && (GuiaError ≤ Errormax),
                {Lighter[■, 0.2], PointSize[TamPuntoGuias],
                  Point[Dynamic@{Guiat, GuiaError}], Line[{{tmin, Dynamic@GuiaError},
                    Dynamic@{Guiat, GuiaError}, {Dynamic@Guiat, TakeLargestBy[
                      {Max[{Errormin, 0}], Min[{Errormax, 0}], Abs, 1] [[1]]}}]],
                (*Else*)
                {}]]],
          (*Else*)
          {}]]],
    ]],
  ]],
];

```

```

    {}]
  ]
],
Graphics[ {
  Locator[Dynamic[{Guiat, GuiaError},
    (ActualizarPuntosGuia[#[[1]]) &], Appearance → None],
  If[HaciendoZoomError,
    {Yellow,
    EdgeForm[Yellow],
    Opacity[0.35],
    Rectangle[PuntoZoom1, Dynamic[MousePosition["Graphics"]]]}]
}],
ImagePadding → {{26., 15.}, {9., 27.}},
TicksStyle → Directive[TamLetraTicksEjes, Darker[Gray, 0.4],
  FontProperties → {"ScreenResolution" → ResolucionPantalla}],
AxesLabel → {SNombresEjes@"t", SNombresEjes@"E(s)"},
AspectRatio → RelAspectoGraficas,
ImageSize → Dynamic@If[CurrentValue[EvaluationNotebook[], WindowSize] ===
  {Full, Full}, TamGraficasPantallaCompleta, TamGraficas]
], TrackedSymbols ⇒ {Error, Guiat, LineasGridRT, HaciendoZoomError,
  tmin, tmax, Errormax, Errormin}]],
{"MouseClicked", 1} ⇒
Which[
  (*.....Zoom dentro.....*)
  CurrentValue["ControlKey"],
  RangoProvisional = Floor[({{tmin, tmax}, {Errormin, Errormax}} - MousePosition[
    "Graphics"]) * FraccionZoom + MousePosition["Graphics"], RTPaso];
  AjustarPuntos[RangoProvisional[[1, 1]], RangoProvisional[[2, 1]],
    0., tAbsMax, -yAbsMax, yAbsMax];
  AjustarPuntos[RangoProvisional[[1, 2]], RangoProvisional[[2, 2]],
    0., tAbsMax, -yAbsMax, yAbsMax];
  {{tmin, tmax}, {Errormin, Errormax}} = RangoProvisional,
  (*.....Zoom fuera.....*)
  CurrentValue["AltKey"],
  RangoProvisional = Floor[({{tmin, tmax}, {Errormin, Errormax}} - MousePosition[
    "Graphics"]) / FraccionZoom + MousePosition["Graphics"], RTPaso];
  AjustarPuntos[RangoProvisional[[1, 1]], RangoProvisional[[2, 1]],
    0., tAbsMax, -yAbsMax, yAbsMax];
  AjustarPuntos[RangoProvisional[[1, 2]], RangoProvisional[[2, 2]],
    0., tAbsMax, -yAbsMax, yAbsMax];
  {{tmin, tmax}, {Errormin, Errormax}} = RangoProvisional],
{"MouseDown", 1} ⇒
Which[
  (*.....Moveuse 1.....*)
  CurrentValue["ControlKey"] && CurrentValue["AltKey"],
  PosicionInicial = MousePosition["GraphicsImageScaled"];
  RangoInicial = {{tmin, tmax}, {Errormin, Errormax}},
  (*.....Zoom rectángulo 1.....*)
  CurrentValue["ShiftKey"],
  PuntoZoom1 = Floor[MousePosition["Graphics"], RTPaso];
  AjustarPuntos[PuntoZoom1[[1]], PuntoZoom1[[2]], 0., tAbsMax, -yAbsMax, yAbsMax];
  HaciendoZoomError = True],
(*.....Moveuse 2.....*)
{"MouseDragged", 1} ⇒
If[CurrentValue["ControlKey"] && CurrentValue["AltKey"],
  RangoProvisional =
  RangoInicial + Floor[(PosicionInicial - MousePosition["GraphicsImageScaled"]) *
    (RangoInicial[[All, 2]] - RangoInicial[[All, 1]]), RTPaso];
  If[RangoProvisional[[1, 1]] ≥ 0. && RangoProvisional[[1, 2]] ≤ tAbsMax,
    {tmin, tmax} = RangoProvisional[[1]];
  If[RangoProvisional[[2, 1]] ≥ -yAbsMax && RangoProvisional[[2, 2]] ≤ yAbsMax,

```

```

    {Errormin, Errormax} = RangoProvisional[[2]]
  ],
  (*.....Zoom rectángulo 2.....*)
  {"MouseDown", 1} => (
    If[CurrentValue["ShiftKey"] && HaciendoZoomError,
      PuntoZoom2 = Floor[MousePosition["Graphics"], RTPaso];
      If[PuntoZoom2 ≠ PuntoZoom1,
        AjustarPuntos[PuntoZoom2[[1]], PuntoZoom2[[2]], 0., tAbsMax, -yAbsMax, yAbsMax];
        {{tmin, tmax}, {Errormin, Errormax}} =
          Map[Sort, Thread[{{PuntoZoom1, PuntoZoom2}}]] (*hace falta todo esto
            porque el rectángulo se puede hacer desde cualquier vértice*)
      ]
    ];
    HaciendoZoomError = False)},
  PassEventsDown → True
];

```

RespuestaTemporalValores [] :=

```

Framed[
  Grid[{
    {Grid[{{SNegrita@SleyendasGraficas@"t",
      InputField[Dynamic[Guiat, (ActualizarPuntosGuia[#]) &],
        ImageSize → TamInputFieldGuias, BaseStyle → {TamLetraLeyendasGraficas,
          FontProperties → {"ScreenResolution" → ResolucionPantalla}}]},
      Alignment → {Left, Center}, Spacings → {{0., 1., 0.}, 0.},
      ItemSize → {{Scaled@0.05, Scaled@0.95}, Automatic},
      Dividers → {{False, Lighter[Gray], False}, False}],
      Tooltip[SleyendasGraficasColor["R(s) =", #], "Referencia",
        TooltipDelay → DelayTooltip],
      SleyendasGraficas@Dynamic[GuiaReferencia]},
    {SpanFromAbove,
      Tooltip[SleyendasGraficasColor["Y(s) =", #],
        "Salida", TooltipDelay → DelayTooltip],
      SleyendasGraficas@Dynamic[GuiaSalida]},
    {SpanFromAbove,
      Tooltip[SleyendasGraficasColor["E(s) =", #],
        "Error", TooltipDelay → DelayTooltip],
      SleyendasGraficas@Dynamic[GuiaError]}
  ], Alignment → {Left, Center},
  Spacings → {{0.6, 0., 0., 0.75}, 0.}, ItemSize →
    {{Scaled@0.2, Scaled@0.08, Automatic}, Scaled[0.072 * RelacionAspectoPantalla]},
  Dividers → {{Opacity[0.], {False}, Opacity[0.]}, False}],
  FrameStyle → {Thin, Lighter[Gray, 0.5]}
];

```

```

(*=====
  LUGAR DE LAS
  RAÍCES=====

```

```

LugarRaices [] :=
  Grid[{
    {EventHandler[
      Deploy[Dynamic[Show[
        ParametricPlot[
          EcParametricas, {k, 0., KcMax},
          (*MaxRecursion→15,
          PlotPoints→7,*)
          PlotRange → {{Remin, Remax}, {Immin, Immax}},
          AxesOrigin → If[Immin > 0 && Immax > 0,
            {If[Sign[Remin] ≠ Sign[Remax],
              0,
              (*Else*)

```

```

    TakeSmallestBy[{Remin, Remax}, Abs, 1][[1]]
  ], Immin},
  (*Else*)
  If[Immin < 0 && Immax < 0,
    {If[Sign[Remin] ≠ Sign[Remax],
      0,
      (*Else*)
      TakeSmallestBy[{Remin, Remax}, Abs, 1][[1]]
    ]}, Immax},
  (*Else*)
  {If[Sign[Remin] ≠ Sign[Remax],
    0,
    (*Else*)
    TakeSmallestBy[{Remin, Remax}, Abs, 1][[1]]
  ]}, 0}
  ]],
  PlotStyle → Colores
],
Graphics[{{MarcadoresPolosBA, Dynamic[ReleaseHold[MarcadoresPolosBC],
  TrackedSymbols ⇒ {Kc}], MarcadoresCeros},
  If[HaciendoZoomLR,
    {Yellow,
      EdgeForm[Yellow],
      Opacity[0.35],
      Rectangle[PuntoZoom1, Dynamic[MousePosition["Graphics"]]]}
  ]},
  TicksStyle → Directive[TamLetraTicksEjes, Darker[Gray,
    0.4], FontProperties → {"ScreenResolution" → ResolucionPantalla}],
  AxesLabel → {SNombresEjes@"Re", SNombresEjes@"Im"},
  AspectRatio → RelAspectoGraficas,
  ImageSize → Dynamic@If[CurrentValue[EvaluationNotebook[], WindowSize] ==
    {Full, Full}, TamGraficasPantallaCompleta, TamGraficas
  ], TrackedSymbols ⇒ {EcParametricas, MarcadoresPolosBA, MarcadoresPolosBC,
    MarcadoresCeros, HaciendoZoomLR, Remin, Remax, Immin, Immax}],
  {"MouseClicked", 1} ⇒
  Which[
    (*.....Zoom dentro.....*)
    CurrentValue["ControlKey"],
    RangoProvisional = Floor[({{Remin, Remax}, {Immin, Immax}} - MousePosition[
      "Graphics"]) * FraccionZoom + MousePosition["Graphics"], LRPaso];
    AjustarPuntos[RangoProvisional[[1, 1]], RangoProvisional[[2, 1]],
      -ReAbsMax, ReAbsMax, -ImAbsMax, ImAbsMax];
    AjustarPuntos[RangoProvisional[[1, 2]], RangoProvisional[[2, 2]],
      -ReAbsMax, ReAbsMax, -ImAbsMax, ImAbsMax];
    {{Remin, Remax}, {Immin, Immax}} = RangoProvisional,
    (*.....Zoom fuera.....*)
    CurrentValue["AltKey"],
    RangoProvisional = Floor[({{Remin, Remax}, {Immin, Immax}} - MousePosition[
      "Graphics"]) / FraccionZoom + MousePosition["Graphics"], LRPaso];
    AjustarPuntos[RangoProvisional[[1, 1]], RangoProvisional[[2, 1]],
      -ReAbsMax, ReAbsMax, -ImAbsMax, ImAbsMax];
    AjustarPuntos[RangoProvisional[[1, 2]], RangoProvisional[[2, 2]],
      -ReAbsMax, ReAbsMax, -ImAbsMax, ImAbsMax];
    {{Remin, Remax}, {Immin, Immax}} = RangoProvisional],
  {"MouseDown", 1} ⇒
  Which[
    (*.....Moverse 1.....*)
    CurrentValue["ControlKey"] && CurrentValue["AltKey"],
    PosicionInicial = MousePosition["GraphicsImageScaled"];
    RangoInicial = {{Remin, Remax}, {Immin, Immax}},
    (*.....Zoom rectángulo 1.....*)
    CurrentValue["ShiftKey"],

```

```

PuntoZoom1 = Floor [MousePosition ["Graphics"], LRPaso];
AjustarPuntos [PuntoZoom1[[1]],
  PuntoZoom1[[2]], -ReAbsMax, ReAbsMax, -ImAbsMax, ImAbsMax];
HaciendoZoomLR = True],
(*.....Move 2.....*)
{"MouseDown", 1} :=
If [CurrentValue ["ControlKey"] && CurrentValue ["AltKey"],
  RangoProvisional = RangoInicial + Floor [
    (PosicionInicial - MousePosition ["GraphicsImageScaled"]) *
    (RangoInicial[[All, 2]] - RangoInicial[[All, 1]]), LRPaso];
If [RangoProvisional[[1, 1]] ≥ -ReAbsMax && RangoProvisional[[1, 2]] ≤ ReAbsMax,
  {Remin, Remax} = RangoProvisional[[1]];
If [RangoProvisional[[2, 1]] ≥ -ImAbsMax && RangoProvisional[[2, 2]] ≤ ImAbsMax,
  {Immin, Immax} = RangoProvisional[[2]]
],
(*.....Zoom rectángulo 2.....*)
{"MouseUp", 1} := (
If [CurrentValue ["ShiftKey"] && HaciendoZoomLR,
  PuntoZoom2 = Floor [MousePosition ["Graphics"], LRPaso];
If [PuntoZoom2 ≠ PuntoZoom1,
  AjustarPuntos [PuntoZoom2[[1]],
    PuntoZoom2[[2]], -ReAbsMax, ReAbsMax, -ImAbsMax, ImAbsMax];
  {{Remin, Remax}, {Immin, Immax}} = Map [Sort, Thread [
    {PuntoZoom1, PuntoZoom2}]] (*hace falta todo esto porque
    el rectángulo se puede hacer desde cualquier vértice*)
  ]
];
HaciendoZoomLR = False)},
PassEventsDown → True
]],
{Framed [
  Dynamic [Grid [ReleaseHold [Leyenda],
    Alignment → {Center, Center}, Spacings → {{0.2, {1., 3.}, 0.7}, 0.}, ItemSize →
      {{Scaled@0.03, {Automatic}}, Scaled[0.06 * RelacionAspectoPantalla]},
    Dividers → {{False}, Opacity[0.]}, {{False}, Opacity[0.]}},
    TrackedSymbols → {Leyenda, Kc}],
    Alignment → {Center, Center}, FrameStyle → {Thin, Lighter[Gray, 0.5]}]
], Alignment → {Center, Top}, Spacings → {0., {0., 0.5, 0.}},
ItemSize →
  Dynamic@If [CurrentValue [EvaluationNotebook[], WindowSize] === {Full, Full},
    {Scaled@1., {Scaled[0.955 * RelacionAspectoPantalla],
      Scaled[0.52 * RelacionAspectoPantalla]}},
    (*Else*)
    {Scaled@1., {Scaled[0.875 * RelacionAspectoPantalla],
      Scaled[0.465 * RelacionAspectoPantalla]}},
  ]
];

(*=====
DIAGRAMA DE
BODE=====*)
BodeModulo[] :=
EventHandler [
  Deploy [Dynamic [Show [
    LogLinearPlot [
      FuncionModulo, {ω, ωmin, ωmax},
      PlotRange → {{ωmin, ωmax}, {Modulomin, Modulomax}},
      PlotRangePadding → Scaled[0.04],
      AxesOrigin → If [Modulomin > 0 && Modulomax > 0,
        {ωmin, Modulomin},
        (*Else*)
        If [Modulomin < 0 && Modulomax < 0,

```

```

        {ωmin, Modulomax},
        (*Else*)
        {ωmin, 0}
    ]],
    GridLines → LineasGridBode
],
Graphics[ {
    {Arrowheads[TamFlechasMgMf], Lighter@■, Tooltip[Arrow[PuntosFlechaMG],
        "Margen de ganancia", TooltipDelay → DelayTooltip]},
    If[VisualizarModuloAsintotico,
        {■, PointSize[1.3 * TamPuntoGuias], Point[PuntosFrecuenciasRuptura],
            Darker[Blue], Dashed, Line[BodeAsintotico]}],
    If[HaciendoZoomModulo,
        {Yellow,
            EdgeForm[Yellow],
            Opacity[0.35],
            Rectangle[{Log@PuntoZoom1[[1]],
                PuntoZoom1[[2]], Dynamic[MousePosition["Graphics"]]}]}
    ]],
    TicksStyle → Directive[TamLetraTicksEjes, Darker[Gray, 0.4],
        FontProperties → {"ScreenResolution" → ResolucionPantalla}],
    AxesLabel → {SNombresEjes@"ω (rad/s)", SNombresEjes@"|G(jω)| (dB)"},
    AspectRatio → RelAspectoGraficas,
    ImageSize → Dynamic@If[CurrentValue[EvaluationNotebook[], WindowSize] ===
        {Full, Full}, TamGraficasPantallaCompleta, TamGraficas]
], TrackedSymbols → {FuncionModulo, PuntosFlechaMG, VisualizarModuloAsintotico,
    PuntosFrecuenciasRuptura, BodeAsintotico, LineasGridBode,
    HaciendoZoomModulo, ωmin, ωmax, Modulomin, Modulomax}],
{"MouseClicked", 1} ⇒
Which[
    (*.....Zoom dentro.....*)
    CurrentValue["ControlKey"],
    RangoProvisional = ({Log[{ωmin, ωmax}], {Modulomin, Modulomax}} - MousePosition[
        "Graphics"]) * FraccionZoom + MousePosition["Graphics"];
    RangoProvisional[[1]] = e^RangoProvisional[[1]];
    RangoProvisional[[1, 1]] =
        Floor[RangoProvisional[[1, 1]], If[RangoProvisional[[1, 1]] < 1.,
            10.^-(Abs[MantissaExponent[RangoProvisional[[1, 1]]][[2]] + 1), 1.]];
    RangoProvisional[[1, 2]] = Floor[RangoProvisional[[1, 2]], If[RangoProvisional[[1, 2]] <
        1., 10.^-(Abs[MantissaExponent[RangoProvisional[[1, 2]]][[2]] + 1), 1.]];
    RangoProvisional[[2]] = Floor[RangoProvisional[[2]], BodePaso];
    AjustarPuntos[RangoProvisional[[1, 1]], RangoProvisional[[2, 1]],
        1./ωAbsMax, ωAbsMax, -ModuloAbsMax, ModuloAbsMax];
    AjustarPuntos[RangoProvisional[[1, 2]], RangoProvisional[[2, 2]],
        1./ωAbsMax, ωAbsMax, -ModuloAbsMax, ModuloAbsMax];
    {{ωmin, ωmax}, {Modulomin, Modulomax}} = RangoProvisional,
    (*.....Zoom fuera.....*)
    CurrentValue["AltKey"],
    RangoProvisional = ({Log[{ωmin, ωmax}], {Modulomin, Modulomax}} - MousePosition[
        "Graphics"]) / FraccionZoom + MousePosition["Graphics"];
    RangoProvisional[[1]] = e^RangoProvisional[[1]];
    RangoProvisional[[1, 1]] =
        Floor[RangoProvisional[[1, 1]], If[RangoProvisional[[1, 1]] < 1.,
            10.^-(Abs[MantissaExponent[RangoProvisional[[1, 1]]][[2]] + 1), 1.]];
    RangoProvisional[[1, 2]] = Floor[RangoProvisional[[1, 2]], If[RangoProvisional[[1, 2]] <
        1., 10.^-(Abs[MantissaExponent[RangoProvisional[[1, 2]]][[2]] + 1), 1.]];
    RangoProvisional[[2]] = Floor[RangoProvisional[[2]], BodePaso];
    AjustarPuntos[RangoProvisional[[1, 1]], RangoProvisional[[2, 1]],
        1./ωAbsMax, ωAbsMax, -ModuloAbsMax, ModuloAbsMax];
    AjustarPuntos[RangoProvisional[[1, 2]], RangoProvisional[[2, 2]],

```

```

1. /  $\omega$ AbsMax,  $\omega$ AbsMax, -ModuloAbsMax, ModuloAbsMax];
{{ $\omega$ min,  $\omega$ max}, {Modulomin, Modulomax}} = RangoProvisional],
{"MouseDown", 1} =>
Which[
(*.....Moverse 1.....*)
CurrentValue["ControlKey"] && CurrentValue["AltKey"],
PosicionInicial = MousePosition["GraphicsImageScaled"];
RangoInicial = {Log[{ $\omega$ min,  $\omega$ max}], {Modulomin, Modulomax}},
(*.....Zoom rectángulo 1.....*)
CurrentValue["ShiftKey"],
PuntoZoom1 = MousePosition["Graphics"];
PuntoZoom1[[1]] = e^PuntoZoom1[[1]];
PuntoZoom1[[1]] = Floor[PuntoZoom1[[1]], If[PuntoZoom1[[1]] < 1.,
10.^-(Abs[MantissaExponent[PuntoZoom1[[1]]][2]] + 1), 1.]];
PuntoZoom1[[2]] = Floor[PuntoZoom1[[2]], BodePaso];
AjustarPuntos[PuntoZoom1[[1]],
PuntoZoom1[[2]], 1. /  $\omega$ AbsMax,  $\omega$ AbsMax, -ModuloAbsMax, ModuloAbsMax];
HaciendoZoomModulo = True],
(*.....Moverse 2.....*)
{"MouseDragged", 1} =>
If[CurrentValue["ControlKey"] && CurrentValue["AltKey"],
RangoProvisional = RangoInicial + (PosicionInicial - MousePosition[
"GraphicsImageScaled"]) * (RangoInicial[[All, 2]] - RangoInicial[[All, 1]]);
RangoProvisional[[1]] = e^RangoProvisional[[1]];
RangoProvisional[[1, 1]] =
Floor[RangoProvisional[[1, 1]], If[RangoProvisional[[1, 1]] < 1.,
10.^-(Abs[MantissaExponent[RangoProvisional[[1, 1]]][2]] + 1), 1.]];
RangoProvisional[[1, 2]] = Floor[RangoProvisional[[1, 2]], If[RangoProvisional[[1, 2]] <
1., 10.^-(Abs[MantissaExponent[RangoProvisional[[1, 2]]][2]] + 1), 1.]];
RangoProvisional[[2]] = Floor[RangoProvisional[[2]], BodePaso];
If[RangoProvisional[[1, 1]]  $\geq$  1. /  $\omega$ AbsMax && RangoProvisional[[1, 2]]  $\leq$   $\omega$ AbsMax,
{ $\omega$ min,  $\omega$ max} = RangoProvisional[[1]]];
If[
RangoProvisional[[2, 1]]  $\geq$  -ModuloAbsMax && RangoProvisional[[2, 2]]  $\leq$  ModuloAbsMax,
{Modulomin, Modulomax} = RangoProvisional[[2]]
],
(*.....Zoom rectángulo 2.....*)
{"MouseUp", 1} => (
If[CurrentValue["ShiftKey"] && HaciendoZoomModulo,
PuntoZoom2 = MousePosition["Graphics"];
PuntoZoom2[[1]] = e^PuntoZoom2[[1]];
PuntoZoom2[[1]] = Floor[PuntoZoom2[[1]], If[PuntoZoom2[[1]] < 1.,
10.^-(Abs[MantissaExponent[PuntoZoom2[[1]]][2]] + 1), 1.]];
PuntoZoom2[[2]] = Floor[PuntoZoom2[[2]], BodePaso];
If[PuntoZoom2  $\neq$  PuntoZoom1,
AjustarPuntos[PuntoZoom2[[1]], PuntoZoom2[[2]],
1. /  $\omega$ AbsMax,  $\omega$ AbsMax, -ModuloAbsMax, ModuloAbsMax];
{{ $\omega$ min,  $\omega$ max}, {Modulomin, Modulomax}} = Map[Sort,
Thread[{PuntoZoom1, PuntoZoom2}]] (*hace falta todo esto
porque el rectángulo se puede hacer desde cualquier vértice*)
]
];
HaciendoZoomModulo = False)},
PassEventsDown  $\rightarrow$  True
];

BodeArgumento[] :=
EventHandler[Deploy[Dynamic[Show[
LogLinearPlot[
FuncionArgumento, { $\omega$ ,  $\omega$ min,  $\omega$ max},

```

```

PlotRange → {{ $\omega$ min,  $\omega$ max}, {Argumentomin, Argumentomax}},
PlotRangePadding → Scaled[0.04],
AxesOrigin → If[Argumentomin > 0 && Argumentomax > 0,
  { $\omega$ min, Argumentomin},
  (*Else*)
  If[Argumentomin < 0 && Argumentomax < 0,
    { $\omega$ min, Argumentomax},
    (*Else*)
    { $\omega$ min, 0}
  ]],
GridLines → LineasGridBode
],
Graphics[ {
  {Arrowheads[TamFlechasMgMf], Lighter@■, Tooltip[Arrow[PuntosFlechaMF],
    "Margen de fase", TooltipDelay → DelayTooltip], Gray, Dashed,
    Line[{ {Log[1./ $\omega$ AbsMax], -180.}, {Log[ $\omega$ AbsMax], -180.} } ]}],
  If[HaciendoZoomArgumento,
    {Yellow,
      EdgeForm[Yellow],
      Opacity[0.35],
      Rectangle[{Log@PuntoZoom1[[1]],
        PuntoZoom1[[2]], Dynamic[MousePosition["Graphics"]]}]}
  ]
},
TicksStyle → Directive[TamLetraTicksEjes, Darker[Gray, 0.4],
  FontProperties → {"ScreenResolution" → ResolucionPantalla}],
AxesLabel → {SNombresEjes@" $\omega$  (rad/s)", SNombresEjes@"Arg[G(j $\omega$ )] (°)"},
AspectRatio → RelAspectoGraficas,
ImageSize → Dynamic@If[CurrentValue[EvaluationNotebook[], WindowSize] ===
  {Full, Full}, TamGraficasPantallaCompleta, TamGraficas]
], TrackedSymbols → {FuncionArgumento, PuntosFlechaMF, LineasGridBode,
  HaciendoZoomArgumento,  $\omega$ min,  $\omega$ max, Argumentomin, Argumentomax}]],
{"MouseClicked", 1} →
Which[
  (*.....Zoom dentro.....*)
  CurrentValue["ControlKey"],
  RangoProvisional = ({Log[{ $\omega$ min,  $\omega$ max}], {Argumentomin, Argumentomax}} -
    MousePosition["Graphics"]) * FraccionZoom + MousePosition["Graphics"];
  RangoProvisional[[1]] = e^RangoProvisional[[1]];
  RangoProvisional[[1, 1]] =
    Floor[RangoProvisional[[1, 1], If[RangoProvisional[[1, 1]] < 1.,
      10.^-(Abs[MantissaExponent[RangoProvisional[[1, 1]]][[2]] + 1), 1.]];
  RangoProvisional[[1, 2]] = Floor[RangoProvisional[[1, 2], If[RangoProvisional[[1, 2]] <
    1., 10.^-(Abs[MantissaExponent[RangoProvisional[[1, 2]]][[2]] + 1), 1.]];
  RangoProvisional[[2]] = Floor[RangoProvisional[[2], BodePaso];
  AjustarPuntos[RangoProvisional[[1, 1]],
    RangoProvisional[[2, 1], 1./ $\omega$ AbsMax,  $\omega$ AbsMax, -360., 360.];
  AjustarPuntos[RangoProvisional[[1, 2]], RangoProvisional[[2, 2]],
    1./ $\omega$ AbsMax,  $\omega$ AbsMax, -360., 360.];
  {{ $\omega$ min,  $\omega$ max}, {Argumentomin, Argumentomax}} = RangoProvisional,
  (*.....Zoom fuera.....*)
  CurrentValue["AltKey"],
  RangoProvisional = ({Log[{ $\omega$ min,  $\omega$ max}], {Argumentomin, Argumentomax}} -
    MousePosition["Graphics"]) / FraccionZoom + MousePosition["Graphics"];
  RangoProvisional[[1]] = e^RangoProvisional[[1]];
  RangoProvisional[[1, 1]] =
    Floor[RangoProvisional[[1, 1], If[RangoProvisional[[1, 1]] < 1.,
      10.^-(Abs[MantissaExponent[RangoProvisional[[1, 1]]][[2]] + 1), 1.]];
  RangoProvisional[[1, 2]] = Floor[RangoProvisional[[1, 2], If[RangoProvisional[[1, 2]] <
    1., 10.^-(Abs[MantissaExponent[RangoProvisional[[1, 2]]][[2]] + 1), 1.]];
  RangoProvisional[[2]] = Floor[RangoProvisional[[2], BodePaso];
  AjustarPuntos[RangoProvisional[[1, 1]],

```

```

RangoProvisional[[2, 1]], 1. /  $\omega$ AbsMax,  $\omega$ AbsMax, -360., 360.];
AjustarPuntos[RangoProvisional[[1, 2]], RangoProvisional[[2, 2]],
  1. /  $\omega$ AbsMax,  $\omega$ AbsMax, -360., 360.];
{{ $\omega$ min,  $\omega$ max}, {Argumentomin, Argumentomax}} = RangoProvisional],
{"MouseDown", 1} :=>
Which[
  (*.....Moverse 1.....*)
  CurrentValue["ControlKey"] && CurrentValue["AltKey"],
  PosicionInicial = MousePosition["GraphicsImageScaled"];
  RangoInicial = {Log[{ $\omega$ min,  $\omega$ max}], {Argumentomin, Argumentomax}},
  (*.....Zoom rectángulo 1.....*)
  CurrentValue["ShiftKey"],
  PuntoZoom1 = MousePosition["Graphics"];
  PuntoZoom1[[1]] = e^PuntoZoom1[[1]];
  PuntoZoom1[[1]] = Floor[PuntoZoom1[[1]], If[PuntoZoom1[[1]] < 1.,
    10.^-(Abs[MantissaExponent[PuntoZoom1[[1]][[2]]] + 1), 1.]];
  PuntoZoom1[[2]] = Floor[PuntoZoom1[[2]], BodePaso];
  AjustarPuntos[PuntoZoom1[[1]],
    PuntoZoom1[[2]], 1. /  $\omega$ AbsMax,  $\omega$ AbsMax, -360., 360.];
  HaciendoZoomArgumento = True],
  (*.....Moverse 2.....*)
{"MouseDragged", 1} :=>
If[CurrentValue["ControlKey"] && CurrentValue["AltKey"],
  RangoProvisional = RangoInicial + (PosicionInicial - MousePosition[
    "GraphicsImageScaled"]) * (RangoInicial[[All, 2]] - RangoInicial[[All, 1]]);
  RangoProvisional[[1]] = e^RangoProvisional[[1]];
  RangoProvisional[[1, 1]] =
    Floor[RangoProvisional[[1, 1]], If[RangoProvisional[[1, 1]] < 1.,
      10.^-(Abs[MantissaExponent[RangoProvisional[[1, 1]][[2]]] + 1), 1.]];
  RangoProvisional[[1, 2]] = Floor[RangoProvisional[[1, 2]], If[RangoProvisional[[1, 2]] <
    1., 10.^-(Abs[MantissaExponent[RangoProvisional[[1, 2]][[2]]] + 1), 1.]];
  RangoProvisional[[2]] = Floor[RangoProvisional[[2]], BodePaso];
  If[RangoProvisional[[1, 1]]  $\geq$  1. /  $\omega$ AbsMax && RangoProvisional[[1, 2]]  $\leq$   $\omega$ AbsMax,
    { $\omega$ min,  $\omega$ max} = RangoProvisional[[1]]];
  If[RangoProvisional[[2, 1]]  $\geq$  -360. && RangoProvisional[[2, 2]]  $\leq$  360.,
    {Argumentomin, Argumentomax} = RangoProvisional[[2]]
  ],
  (*.....Zoom rectángulo 2.....*)
{"MouseUp", 1} :=> (
  If[CurrentValue["ShiftKey"] && HaciendoZoomArgumento,
    PuntoZoom2 = MousePosition["Graphics"];
    PuntoZoom2[[1]] = e^PuntoZoom2[[1]];
    PuntoZoom2[[1]] = Floor[PuntoZoom2[[1]], If[PuntoZoom2[[1]] < 1.,
      10.^-(Abs[MantissaExponent[PuntoZoom2[[1]][[2]]] + 1), 1.]];
    PuntoZoom2[[2]] = Floor[PuntoZoom2[[2]], BodePaso];
    If[PuntoZoom2  $\neq$  PuntoZoom1,
      AjustarPuntos[PuntoZoom2[[1]],
        PuntoZoom2[[2]], 1. /  $\omega$ AbsMax,  $\omega$ AbsMax, -360., 360.];
      {{ $\omega$ min,  $\omega$ max}, {Argumentomin, Argumentomax}} = Map[Sort,
        Thread[{PuntoZoom1, PuntoZoom2}]] (*hace falta todo esto
        porque el rectángulo se puede hacer desde cualquier vértice*)
    ]
  ];
  HaciendoZoomArgumento = False)},
PassEventsDown  $\rightarrow$  True
];

BodeValores[] :=
Framed[Grid[{
  {Tooltip[SleyendasGraficasColor["MG = ", Lighter[■, 0.1]],

```



```

NumberForm[Denominator@GBAFactorizada, {8, DecimalesSist}] > 28),
Grid[{
  {Row[{Tooltip[SNegrita@SGbaGbc@"GBA(s) = ",
    "Función de transferencia en bucle abierto\nGBA = KcGcGaGpH",
    TooltipDelay → DelayTooltip], SGbaGbc@Dynamic@
    NumberForm[GBAFactorizada, {8, DecimalesSist}], " = "]}],
  {Row[{SGbaGbc@" = ", SGbaGbc@Dynamic@
    NumberForm[ExpandNumerator@ExpandDenominator@
      GBAFactorizada, {8, DecimalesSist}]}]}]
}, Alignment → {Left, Center}, Spacings → {0., {0., 0.5, 0.}}],
(*Else*)
Grid[{
  {Row[{Tooltip[SNegrita@SGbaGbc@"GBA(s) = ",
    "Función de transferencia en bucle abierto\nGBA = KcGcGaGpH",
    TooltipDelay → DelayTooltip], SGbaGbc@
    Dynamic@NumberForm[GBAFactorizada, {8, DecimalesSist}],
    SGbaGbc@" = ", SGbaGbc@Dynamic@NumberForm[ExpandNumerator@
      ExpandDenominator@GBAFactorizada, {8, DecimalesSist}]}]}]
}, Alignment → {Left, Center}, Spacings → {0., 0.}]
]
],
DiagramaDeBloques [],
Dynamic@
If[(StringLength@ToString@NumberForm[Numerator@GBCFactorizada,
  {8, DecimalesSist}] > 50) || (StringLength@ToString@
  NumberForm[Denominator@GBCFactorizada, {8, DecimalesSist}] > 50),
Grid[{
  {Row[{Tooltip[SNegrita@SGbaGbcPeq@"GBC(s) = ",
    "Función de transferencia en bucle cerrado\nGBC
    =  $\frac{K_c G_c G_a G_p}{1 + K_c G_c G_a G_p H} = \frac{G^*}{1 + G_{BA}} \backslash n *G$ 
    ≡ Función de transferencia de
    la\ncadena directa",
    TooltipDelay → DelayTooltip], SGbaGbcPeq@Dynamic@NumberForm[
      GBCFactorizada, {8, DecimalesSist}], " = "]}],
  {Row[{SGbaGbcPeq@" = ", SGbaGbcPeq@Dynamic@
    NumberForm[ExpandNumerator@ExpandDenominator@
      GBCFactorizada, {8, DecimalesSist}]}]}]
}, Alignment → {Left, Center}, Spacings → {0., {0., 0.5, 0.}}],
(*Else*)
If[(StringLength@ToString@NumberForm[Numerator@GBCFactorizada,
  {8, DecimalesSist}] > 28) || (StringLength@ToString@
  NumberForm[Denominator@GBCFactorizada, {8, DecimalesSist}] > 28),
Grid[{
  {Row[{Tooltip[SNegrita@SGbaGbc@"GBC(s) = ",
    "Función de transferencia en bucle cerrado\nGBC
    =  $\frac{K_c G_c G_a G_p}{1 + K_c G_c G_a G_p H} = \frac{G^*}{1 + G_{BA}} \backslash n *G$ 
    ≡ Función de transferencia de
    la\ncadena directa",
    TooltipDelay → DelayTooltip], SGbaGbc@Dynamic@NumberForm[
      GBCFactorizada, {8, DecimalesSist}], " = "]}],
  {Row[{SGbaGbc@" = ", SGbaGbc@Dynamic@
    NumberForm[ExpandNumerator@ExpandDenominator@
      GBCFactorizada, {8, DecimalesSist}]}]}]
}, Alignment → {Left, Center}, Spacings → {0., {0., 0.5, 0.}}],
(*Else*)
Grid[{
  {Row[{Tooltip[SNegrita@SGbaGbc@"GBC(s) = ",
    "Función de transferencia en bucle cerrado\nGBC

```

$$= \frac{K_c G_c G_a G_p}{1 + K_c G_c G_a G_p H} = \frac{G^*}{1 + G_{BA}} \backslash n *G$$

≡ Función de transferencia de la cadena directa", TooltipDelay → DelayTooltip],

SGbaGbc@Dynamic@NumberForm[GBCFactorizada, {8, DecimalesSist}],  
 SGbaGbc@" = ", SGbaGbc@Dynamic@NumberForm[ExpandNumerator@  
 ExpandDenominator@GBCFactorizada, {8, DecimalesSist}]]]

}, Alignment → {Left, Center}, Spacings → {0., 0.}]

]

}}, Alignment → {Center, Center}, ItemSize → {{Scaled[0.19],  
 Scaled[0.62], Scaled[0.19]}, Scaled[0.14 \* RelacionAspectoPantalla]},  
 Dividers → {{False, {Lighter[Gray, 0.2]}, False}, False}],  
 SpanFromLeft},

(\*=====

2 FILA: CONTROLES

GRÁFICAS=====

(\*-----Respuesta  
 temporal-----\*)

```
{Dynamic[Grid[{
  Row[{Opener[Dynamic[MostrarRTEExt]],
    SApartado["Respuesta Temporal"]}], SpanFromLeft},
  If[MostrarRTEExt,
    {Grid[{
      {SSubApartado[" t"], ControlRango[tmin, tmax, 0., tAbsMax, RTPaso]},
      {SSubApartado[" R(s), Y(s)"],
        ControlRango[SalRefmin, SalRefmax, -yAbsMax, yAbsMax, RTPaso]},
      {SSubApartado[" E(s)"], ControlRango[Errormin,
        Errormax, -yAbsMax, yAbsMax, RTPaso]}
    ], Alignment → {Left, Center, {{1, 2} → {Center, Center},
      {2, 2} → {Center, Center}, {3, 2} → {Center, Center}}},
    Spacings → {0., 0.}, ItemSize → {{Scaled@0.22, Scaled@0.72},
      Scaled[0.12 * RelacionAspectoPantalla]},
    Dividers → {{{False}, Lighter[Gray, 0.5]}, False}],
    Grid[{
      {Button[Overlay[{Pane[" →",
        BaselinePosition → Scaled[.25]], "↑"}, Alignment → -0.6],
        {{tmin, tmax}, {SalRefmin, SalRefmax}, {Errormin,
          Errormax}} = RangoAutomaticoRT,
        Appearance → EstiloBotones, ImageSize → TamBotonesRango,
        Tooltip → "Centrar ejes"],
        ""},
      {Button["| | |", If[LineasGridRT != Automatic,
        LineasGridRT = Automatic, LineasGridRT = None],
        Appearance → Dynamic[If[LineasGridRT == Automatic,
          {EstiloBotones, "Pressed"}, EstiloBotones]],
        ImageSize → TamBotonesRango, Tooltip → "Cuadrícula"],
      Button["| | | | |", If[LineasGridRT != All, LineasGridRT =
        All, LineasGridRT = None],
        Appearance → Dynamic[If[LineasGridRT == All,
          {EstiloBotones, "Pressed"}, EstiloBotones]],
        ImageSize → TamBotonesRango, Tooltip → "Cuadrícula fina"]}
    ], Alignment → {Center, Center}, Spacings → {0., 0.}, ItemSize →
    {Scaled@0.5, Scaled[0.42 * RelacionAspectoPantalla]}]
  },
  (*Else*)
  Nothing]
}, Alignment → {Center, Center, {{1, 1} → {Center, Bottom}}},
  Spacings → {0., 0.}, ItemSize → {{Scaled@0.74, Scaled@0.22}, {Scaled[0.08 *
    RelacionAspectoPantalla], Scaled[0.29 * RelacionAspectoPantalla]}},
  Dividers → {False, {False, False, Lighter[Gray, 0.3]}}},
  TrackedSymbols => {MostrarRTEExt}],
```

(\*-----Lugar de las raíces-----\*)

```
Dynamic[Grid[{
  Row[{Opener[Dynamic[MostrarLRExt]],
    SApartado["Lugar de las Raíces"]}], SpanFromLeft},
  If[MostrarLRExt,
    {Grid[{
      {SSubApartado[" Re"],
        ControlRango[Remin, Remax, -ReAbsMax, ReAbsMax, LRPaso]},
      {SSubApartado[" Im"], ControlRango[Immin, Immax,
        -ImAbsMax, ImAbsMax, LRPaso]}
    }, Alignment → {Left, Center, {{1, 2} → {Center, Center},
      {2, 2} → {Center, Center}}},
    Spacings → {0., 0.}, ItemSize → {{Scaled@0.22, Scaled@0.72},
      Scaled[0.12 * RelacionAspectoPantalla]},
    Dividers → {{{False}, Lighter[Gray, 0.5]}, False}},
    Grid[{
      {Button[Overlay[{Pane[" →",
        BaselinePosition → Scaled[.25]], "↑"], Alignment → -0.6},
        {{Remin, Remax}, {Immin, Immax}} = RangoAutomaticoLR,
        Appearance → EstiloBotones,
        ImageSize → TamBotonesRango, Tooltip → "Centrar ejes"],
        ""}
    }, Alignment → {Center, Center}, Spacings → {0., 0.}, ItemSize →
      {Scaled@0.5, Scaled[0.42 * RelacionAspectoPantalla]}]
    },
    (*Else*)
    Nothing
  ], Alignment → {Center, Center, {{1, 1} → {Center, Bottom}}},
  Spacings → {0., 0.}, ItemSize → {{Scaled@0.74, Scaled@0.22}, {Scaled[0.08 *
    RelacionAspectoPantalla], Scaled[0.29 * RelacionAspectoPantalla]}},
  Dividers → {False, {False, False, Lighter[Gray, 0.3]}}},
  TrackedSymbols => {MostrarLRExt}],
```

(\*-----Diagrama de Bode-----\*)

```
Dynamic[Grid[{
  Row[{Opener[Dynamic[MostrarBodeExt]],
    SApartado["Diagrama de Bode"]}], SpanFromLeft},
  If[MostrarBodeExt,
    {Grid[{
      {SSubApartado[" ω"], ControlLogRango[ωmin, ωmax, 1. / ωAbsMax, ωAbsMax]},
      {SSubApartado[" |G(jω)|"], ControlRango[Modulomin,
        Modulomax, -ModuloAbsMax, ModuloAbsMax, BodePaso]},
      {SSubApartado[" Arg[G(jω)]"], ControlRango[Argumentomin,
        Argumentomax, -360., 360., BodePaso]}
    }, Alignment → {Left, Center, {{1, 2} → {Center, Center},
      {2, 2} → {Center, Center}, {3, 2} → {Center, Center}}},
    Spacings → {0., 0.}, ItemSize → {{Scaled@0.22, Scaled@0.72},
      Scaled[0.12 * RelacionAspectoPantalla]},
    Dividers → {{{False}, Lighter[Gray, 0.5]}, False}},
    Grid[{
      {Button[Overlay[{Pane[" →",
        BaselinePosition → Scaled[.25]], "↑"], Alignment → -0.6},
        {ωmin, ωmax} = {1. / ωAbsMax, ωAbsMax};
        {Modulomin, Modulomax} = RangoAutomaticoModulo;
        {Argumentomin, Argumentomax} = RangoAutomaticoArgumento,
        Appearance → EstiloBotones,
        ImageSize → TamBotonesRango, Tooltip → "Centrar ejes"],
      Button["≍", VisualizarModuloAsintotico = ! VisualizarModuloAsintotico,
        Appearance → Dynamic[If[VisualizarModuloAsintotico,
          {EstiloBotones, "Pressed"}, EstiloBotones]],
        ImageSize → TamBotonesRango, Tooltip → "Módulo asintótico"}],
```

```

    {Button["| | |", If[LineasGridBode != Automatic,
        LineasGridBode = Automatic, LineasGridBode = None],
        Appearance → Dynamic[If[LineasGridBode === Automatic,
            {EstiloBotones, "Pressed"}, EstiloBotones]],
        ImageSize → TamBotonesRango, Tooltip → "Cuadrícula fina"],
    Button["| | | |", If[LineasGridBode != All, LineasGridBode =
        All, LineasGridBode = None],
        Appearance → Dynamic[If[LineasGridBode === All,
            {EstiloBotones, "Pressed"}, EstiloBotones]],
        ImageSize → TamBotonesRango, Tooltip → "Cuadrícula"]}
    }, Alignment → {Center, Center}, Spacings → {0., 0.}, ItemSize →
    {Scaled@0.5, Scaled[0.42 * RelacionAspectoPantalla]}
    },
    (*Else*)
    Nothing]
    }, Alignment → {Center, Center, {{1, 1} → {Center, Bottom}}},
    Spacings → {0., 0.}, ItemSize → {{Scaled@0.74, Scaled@0.22}, {Scaled[0.08 *
        RelacionAspectoPantalla], Scaled[0.29 * RelacionAspectoPantalla]}},
    Dividers → {False, {False, False, Lighter[Gray, 0.3]}}],
    TrackedSymbols => {MostrarBodeExt}]],
(*=====
    0.=====3 FILA:
        GRÁFICAS=====
(*-----Respuesta
temporal-----*)
{Dynamic[If[MostrarRTEExt,
    ActualizarRT[];
    (*está aquí para cuando se muestre con las teclas*)
    Grid[{
        {RespuestaTemporalSistRef[]},
        {RespuestaTemporalError[]},
        {RespuestaTemporalValores[]}
    ]},
    Alignment → {Center, Top, {{3, 1} → {Center, Bottom}}}, Spacings → {0., 0.},
    ItemSize → Dynamic@If[CurrentValue[EvaluationNotebook[],
        WindowSize] === {Full, Full},
        {Scaled@1., {{Scaled[0.95 * RelacionAspectoPantalla],
            Scaled[0.25 * RelacionAspectoPantalla]}},
        (*Else*)
        {Scaled@1., {{Scaled[0.88 * RelacionAspectoPantalla],
            Scaled[0.25 * RelacionAspectoPantalla]}}
    ]
    ],
    (*Else*)
    ""], TrackedSymbols => {MostrarRTEExt}],
(*-----Lugar de las
raíces-----*)
Dynamic[If[MostrarLREExt,
    ActualizarLR[];
    (*está aquí para cuando se muestre con las teclas*)
    LugarRaices[],
    (*Else*)
    ""], TrackedSymbols => {MostrarLREExt}],
(*-----Diagrama de
Bode-----*)
Dynamic[If[MostrarBodeExt,
    ActualizarBode[];
    (*está aquí para cuando se muestre con las teclas*)
    Grid[{
        {BodeModulo[]},
        {BodeArgumento[]},
        {BodeValores[]}
    ]
    ],
    (*Else*)
    ""], TrackedSymbols => {MostrarBodeExt}]]

```

```

    }, Alignment → {Center, Top}, Spacings → {0., 0.},
    ItemSize →
    Dynamic@If[CurrentValue[EvaluationNotebook[], WindowSize] === {Full, Full},
      {Scaled@1., {{Scaled[0.815 * RelacionAspectoPantalla]},
        Scaled[0.515 * RelacionAspectoPantalla]}},
      (*Else*)
      {Scaled@1., {{Scaled[0.745 * RelacionAspectoPantalla]},
        Scaled[0.515 * RelacionAspectoPantalla]}}
    ]
  ],
  (*Else*)
  ""], TrackedSymbols → {MostrarBodeExt}}],
(*=====
  4 FILA:
  AYUDA=====
{SpanFromAbove,
  Grid[{
    {Panel[Grid[{
      {SApartadosAyuda@"Controles gráficas", SApartadosAyuda@"Mostrar/Ocultar"},
      {STextoAyuda@"· Desplazar → Ctrl + Alt",
        STextoAyuda@"· Respuesta temporal → R"},
      {STextoAyuda@"· Zoom → Shift ",
        STextoAyuda@"· Lugar de las raíces → L"},
      {STextoAyuda@"· Zoom dentro → Ctrl ",
        STextoAyuda@"· Diagrama de Bode → B"},
      {STextoAyuda@"· Zoom fuera → Alt ", ""},
      {SApartadosAyuda@"Pantalla completa", SApartadosAyuda@"Ventana "},
      {STextoAyuda@"· Activar → P "},
      {STextoAyuda@"· Cerrar → C "},
      {STextoAyuda@"· Desactivar → Esc", STextoAyuda@"· Minimizar → M"}
    ]}, Alignment → {Center, Baseline, {{1, 1} → {Center, Top}, {1, 2} →
      {Center, Top}, {6, 1} → {Center, Bottom}, {6, 2} → {Center, Bottom}}},
    Spacings → {{0., 2., 0.}, 0.}, ItemSize → {{Scaled@0.325,
      Scaled@0.325}, {Scaled[0.075 * RelacionAspectoPantalla],
        Scaled[0.065 * RelacionAspectoPantalla]}, Scaled[
        0.12 * RelacionAspectoPantalla], Scaled[0.065 * RelacionAspectoPantalla],
        Scaled[0.065 * RelacionAspectoPantalla]}}},
    Alignment → {Center, Center}, Spacings → {0., 0.}]
  ], Alignment → {Center, Center}, Spacings → {0., 0.},
  ItemSize → {Scaled@0.98, Scaled[0.64 * RelacionAspectoPantalla]},
  Dividers → {False, {Darker[Gray, 0.1], False}}],
  SpanFromAbove}
}],
Alignment → {Center, Top, {{1, 1} → {Center, Center}, {4, 2} → {Center, Bottom}}},
Spacings → {0., 0.},
ItemSize →
  Dynamic@If[CurrentValue[EvaluationNotebook[], WindowSize] === {Full, Full},
    {Scaled[1./3.], {Scaled[0.16 * RelacionAspectoPantalla],
      Scaled[0.135 * RelacionAspectoPantalla], Scaled[
        0.508 * RelacionAspectoPantalla], Scaled[0.21 * RelacionAspectoPantalla]}},
    (*Else*)
    {Scaled[1./3.], {Scaled[0.16 * RelacionAspectoPantalla],
      Scaled[0.135 * RelacionAspectoPantalla], Scaled[
        0.459 * RelacionAspectoPantalla], Scaled[0.21 * RelacionAspectoPantalla]}}
  ],
  Dividers → {{False, {Gray}, False}, {False, Gray, {False}}}], (*Grid*)
  FrameMargins → 0.
] (*Panel*)
] (*DynamicModule*)
] (*Block*)
];

```

```
(*Desactivar avisos*)
```

```
Off[General::munfl];
```

```
Off[NSolve::ratnz];
```

```
(*//////////////////////////////////////  
//////////////////////////////////////  
//////////////////////////////////////*)  
(*//////////////////////////////////////  
CREAR  
VENTANA//////////////////////////////////////  
//////////////////////////////////////*)  
(*//////////////////////////////////////  
//////////////////////////////////////  
//////////////////////////////////////*)
```

```
(*
```

```
EscaldoVertical con tamaños
```

```
Table[
```

```
Floor[(CurrentValue["ScreenArea"][[1,2,2]]-3.)/CurrentValue["ScreenArea"][[1,1,2]],0.001]-  
0.004*i,{i,0,4}]
```

```
*)
```

```
With[
```

```
{(*=====
```

```
CONSTANTES=====
```

```
====*)
```

```
TamVentana =
```

```
{CurrentValue["ScreenArea"][[1, 1, 2]], CurrentValue["ScreenArea"][[1, 2, 2]]},
```

```
TamBarra = {Scaled[1.], 0.026 * CurrentValue["ScreenArea"][[1, 2, 2]]},
```

```
TamBotones = {Scaled[0.97], 0.026 * CurrentValue["ScreenArea"][[1, 2, 2]]},
```

```
MargenesCelda = {{5., 0.}, {0., 1.}},
```

```
TamLetraBarra = 12,
```

```
TamLetraBotones = 16,
```

```
RelAspecto = Round[(CurrentValue["ScreenArea"][[1, 2, 2]] - 5) /  
CurrentValue["ScreenArea"][[1, 1, 2]], 0.001],
```

```
Resolucion = CurrentValue["ScreenResolution"][[1]],
```

```
DynamicModule[
```

```
{(*=====
```

```
VARIABLES=====
```

```
====*)
```

```
MostrarRTExt = True,
```

```
MostrarLRExt = True,
```

```
MostrarBodeExt = True,
```

```
Margenes,
```

```
PosInicial,
```

```
MargenesIniciales,
```

```
ΔPosicion},
```

```
Margenes := {{-20, Automatic}, {Automatic, 20}};
```

```
CreateDocument[
```

```
{Grid[{
```

```
(*=====
```

```
FILA
```

```
1=====
```

```
{Grid[{
```

```
{(*-----Título-----  
-----*)
```

```
EventHandler[
```

```
Pane[Style["Herramienta de análisis en Automática y Control",
```

```
■, FontFamily → "Calibri Light",
```

```

    FontWeight → "SemiBold", FontVariations → {"CapsType" → "SmallCaps"},
    FontProperties → {"ScreenResolution" → Resolucion}],
TamBarra, Alignment → {Left, Center}, ContentPadding → False],
{"MouseDown" => (SelectionMove[EvaluationNotebook[], All, Cell];
  If[CurrentValue["MouseClickedCount"] == 2,
    If[CurrentValue[EvaluationNotebook[],
      WindowMargins] === {{0, Automatic}, {Automatic, 0}},
      SetOptions[EvaluationNotebook[], WindowMargins → Margenes],
      (*Else*)
    If[CurrentValue[EvaluationNotebook[], WindowSize] === {Full, Full},
      SetOptions[EvaluationNotebook[], WindowSize → {Automatic, Automatic}];
      SetOptions[EvaluationNotebook[], WindowSize → TamVentana];
      SetOptions[EvaluationNotebook[],
        WindowMargins → {{0, Automatic}, {Automatic, 0}},
        (*Else*)
      SetOptions[EvaluationNotebook[],
        WindowMargins → {{0, Automatic}, {Automatic, 0}}]
    ]],
    (*Else*)
    PosInicial = MousePosition[];
    MargenesIniciales = CurrentValue[EvaluationNotebook[], WindowMargins]
  ]),
"MouseDragged" => (ΔPosicion = MousePosition[] - PosInicial;
  SetOptions[EvaluationNotebook[], WindowMargins →
    (MargenesIniciales + {{ΔPosicion[[1]], 0}, {0, ΔPosicion[[2]]}})]),
"MouseUp" => If[(CurrentValue[EvaluationNotebook[], WindowMargins][[
  2, 2]] ≤ 0) && (MousePosition[][[2]] == 0),
  SetOptions[EvaluationNotebook[], WindowMargins →
    {{0, Automatic}, {Automatic, 0}}],
  (*Else*)
  If[CurrentValue[EvaluationNotebook[],
    WindowMargins] != {{0, Automatic}, {Automatic, 0}},
    Margenes = CurrentValue[EvaluationNotebook[], WindowMargins]
  ]
}],
(*-----Minimizar-----
-----*)
EventHandler[
  Deploy@Style[
    Pane["-", TamBotones, Alignment → {Center, Center}, ContentPadding → False],
    ■, FontFamily → "Calibri Light", FontWeight → "SemiBold", FontSize →
    TamLetraBotones, FontProperties → {"ScreenResolution" → Resolucion},
    Background → Dynamic@If[CurrentValue["MouseOver"] &&
      CurrentValue["MouseButtonTest"],
      Lighter[Gray, 0.6],
      (*Else*)
      If[CurrentValue["MouseOver"],
        Lighter[Gray, 0.8],
        (*Else*)
        None]
  ],
  {"MouseDown" => SelectionMove[EvaluationNotebook[], All, Cell],
  "MouseUp" => If[CurrentValue[
    "MouseOver"], FrontEndTokenExecute["WindowMiniaturize"]]}],
(*-----Maximizar-----
-----*)
EventHandler[
  Deploy@Dynamic@If[CurrentValue[EvaluationNotebook[],
    WindowMargins] === {{0, Automatic}, {Automatic, 0}},
  Style[
    Pane[Overlay[{"□", Pane[" □", {Automatic, 0.015 *

```

```

        CurrentValue["ScreenArea"][[1, 2, 2]], ContentPadding → False]],
        Alignment → {Center, Bottom}, ContentPadding → False], ImageSize →
        TamBotones, Alignment → {Center, Center}, ContentPadding → False],
        ■, FontFamily → "Calibri Light", FontWeight → "SemiBold", FontSize →
        TamLetraBotones, FontProperties → {"ScreenResolution" → Resolucion},
        Background → If[CurrentValue["MouseOver"] &&
        CurrentValue["MouseButtonTest"],
        Lighter[Gray, 0.6],
        (*Else*)
        If[CurrentValue["MouseOver"],
        Lighter[Gray, 0.8],
        (*Else*)
        None]
    ],
    (*Else*)
    Style[
    Pane["□", ImageSize → TamBotones,
    Alignment → {Center, Center}, ContentPadding → False],
    ■, FontFamily → "Calibri Light", FontWeight → "SemiBold", FontSize →
    TamLetraBotones, FontProperties → {"ScreenResolution" → Resolucion},
    Background → If[CurrentValue["MouseOver"] &&
    CurrentValue["MouseButtonTest"],
    Lighter[Gray, 0.6],
    (*Else*)
    If[CurrentValue["MouseOver"],
    Lighter[Gray, 0.8],
    (*Else*)
    None]
    ]
    ],
    {"MouseDown" ⇒ SelectionMove[EvaluationNotebook[], All, Cell],
    "MouseUp" ⇒ If[CurrentValue["MouseOver"],
    If[CurrentValue[EvaluationNotebook[],
    WindowMargins] === {{0, Automatic}, {Automatic, 0}},
    SetOptions[EvaluationNotebook[], WindowMargins → Margenes],
    (*Else*)
    If[CurrentValue[EvaluationNotebook[], WindowSize] === {Full, Full},
    SetOptions[EvaluationNotebook[], WindowSize → {Automatic, Automatic}];
    SetOptions[EvaluationNotebook[], WindowSize → TamVentana];
    SetOptions[EvaluationNotebook[],
    WindowMargins → {{0, Automatic}, {Automatic, 0}}],
    (*Else*)
    SetOptions[EvaluationNotebook[],
    WindowMargins → {{0, Automatic}, {Automatic, 0}}]
    ]]]}
    ],
    (*-----Cerrar-----
    -----*)
    EventHandler[
    Deploy@Style[
    Pane["×", ImageSize → TamBotones,
    Alignment → {Center, Center}, ContentPadding → False],
    ■, FontFamily → "Calibri Light", FontWeight → "SemiBold", FontSize →
    TamLetraBotones, FontProperties → {"ScreenResolution" → Resolucion},
    Background → Dynamic@If[CurrentValue["MouseOver"] &&
    CurrentValue["MouseButtonTest"],
    Lighter[■, 0.5],
    (*Else*)
    If[CurrentValue["MouseOver"],
    ■,
    (*Else*)

```

```

None]]
    ],
    {"MouseDown" => SelectionMove[EvaluationNotebook[], All, Cell],
     "MouseUp" => If[CurrentValue["MouseOver"], FrontEndTokenExecute["Close"]]}]}
(*Grid*)
},
Alignment -> {Left, Center, {{1, 2} -> {Center, Center},
 {1, 3} -> {Center, Center}, {1, 4} -> {Center, Center}}},
Spacings -> {0., 0.},
ItemSize -> {{Scaled[0.91], {Scaled[0.03]}}, Automatic}
]],
(*=====
  FILA
  2=====
{EventHandler [
  HerramientaAutomatica[MostrarRTEExt,
  MostrarLREExt, MostrarBodeExt, Resolucion, RelAspecto],
  "MouseDown" => SelectionMove[EvaluationNotebook[], All, Cell],
  PassEventsDown -> True
]}
}, Alignment -> {Center, Center, {{2, 1} -> {Center, Top}}}, Spacings -> {0., 0.},
ItemSize -> {Scaled[1.], {Scaled[0.03 * RelAspecto], Scaled[0.97 * RelAspecto]}}]
},
(*=====
  OPCIONES=====
=====*)
NotebookEventActions -> {
  "EscapeKeyDown" => If[CurrentValue[EvaluationNotebook[], WindowSize] === {Full, Full},
    SetOptions[EvaluationNotebook[], WindowSize -> {Automatic, Automatic}];
    SetOptions[EvaluationNotebook[], WindowSize -> TamVentana];
    SetOptions[EvaluationNotebook[],
      WindowMargins -> {{0, Automatic}, {Automatic, 0}}]],
  {"KeyDown", "m" } => FrontEndTokenExecute["WindowMiniaturize"],
  {"KeyDown", "M" } => FrontEndTokenExecute["WindowMiniaturize"],
  {"KeyDown", "c" } => FrontEndTokenExecute["Close"],
  {"KeyDown", "C" } => FrontEndTokenExecute["Close"],
  {"KeyDown", "p" } => (SetOptions[EvaluationNotebook[], WindowSize -> {Full, Full}];
    SetOptions[EvaluationNotebook[],
      WindowMargins -> {{5, Automatic}, {Automatic, 5}}]),
  {"KeyDown", "P" } => (SetOptions[EvaluationNotebook[], WindowSize -> {Full, Full}];
    SetOptions[EvaluationNotebook[],
      WindowMargins -> {{5, Automatic}, {Automatic, 5}}]),
  {"KeyDown", "r" } => (MostrarRTEExt = ! MostrarRTEExt),
  {"KeyDown", "R" } => (MostrarRTEExt = ! MostrarRTEExt),
  {"KeyDown", "l" } => (MostrarLREExt = ! MostrarLREExt),
  {"KeyDown", "L" } => (MostrarLREExt = ! MostrarLREExt),
  {"KeyDown", "b" } => (MostrarBodeExt = ! MostrarBodeExt),
  {"KeyDown", "B" } => (MostrarBodeExt = ! MostrarBodeExt),
  PassEventsDown -> True},
CellMargins -> MargenesCelda,
CellFrameMargins -> 0.,
ShowCellBracket -> False,
ShowCellLabel -> False,
"TrackCellChangeTimes" -> False,
WindowFrame -> "Frameless",
WindowElements -> {},
DockedCells -> Cell[^],
WindowFrameElements -> {"CloseBox"},
ScrollingOptions -> {"VerticalScrollRange" -> Fit},
WindowSize -> TamVentana,
WindowMargins -> {{0, Automatic}, {Automatic, 0}},

```

```
    Deployed → True,  
    Editable → False,  
    Saveable → False,  
    WindowTitle → "Herramienta de análisis en Automática y Control"  
  ] (*CreateDocument*)  
] (*DynamicModule*)  
];
```

## **15. Anexo II: Manual de usuario**

# 1. Introducción

En este anexo se presenta un manual de usuario que explica cómo usar la *Herramienta de análisis en Automática y Control*, una herramienta software programada en el entorno Mathematica® y pensada para dar soporte a alumnos de asignaturas donde se trate el Análisis y Diseño de Sistemas de Control.

La herramienta es capaz de mostrar el análisis temporal, el análisis de la ubicación de polos y ceros y el análisis frecuencial de un sistema realimentado definido por sus funciones de transferencia.

La estructura del documento es la siguiente. En primer lugar, se muestra cómo abrir la herramienta y se describe la interfaz. A continuación, se explica cómo definir el sistema realimentado, es decir, cómo introducir los datos de entrada. Finalmente, se detallan las características de cada uno de los tres análisis que realiza la herramienta.

## 2. Abrir el programa y descripción de la interfaz

En este apartado se explica en primer lugar cómo abrir la herramienta. A continuación, se describe la interfaz de la herramienta.

### 2.1. Abrir el programa

Para abrir el programa, se debe hacer doble click sobre su icono.



Figura 1. Icono de la herramienta software

Al hacerlo, se abre la siguiente ventana.



Figura 2. Ventana para abrir la herramienta software

Al hacer click en el botón "Abrir" se muestra la herramienta.

### 2.2. Descripción de la interfaz

La interfaz se divide en dos zonas separadas por un divisor horizontal: una que permite definir el sistema y otra que muestran los análisis del mismo.

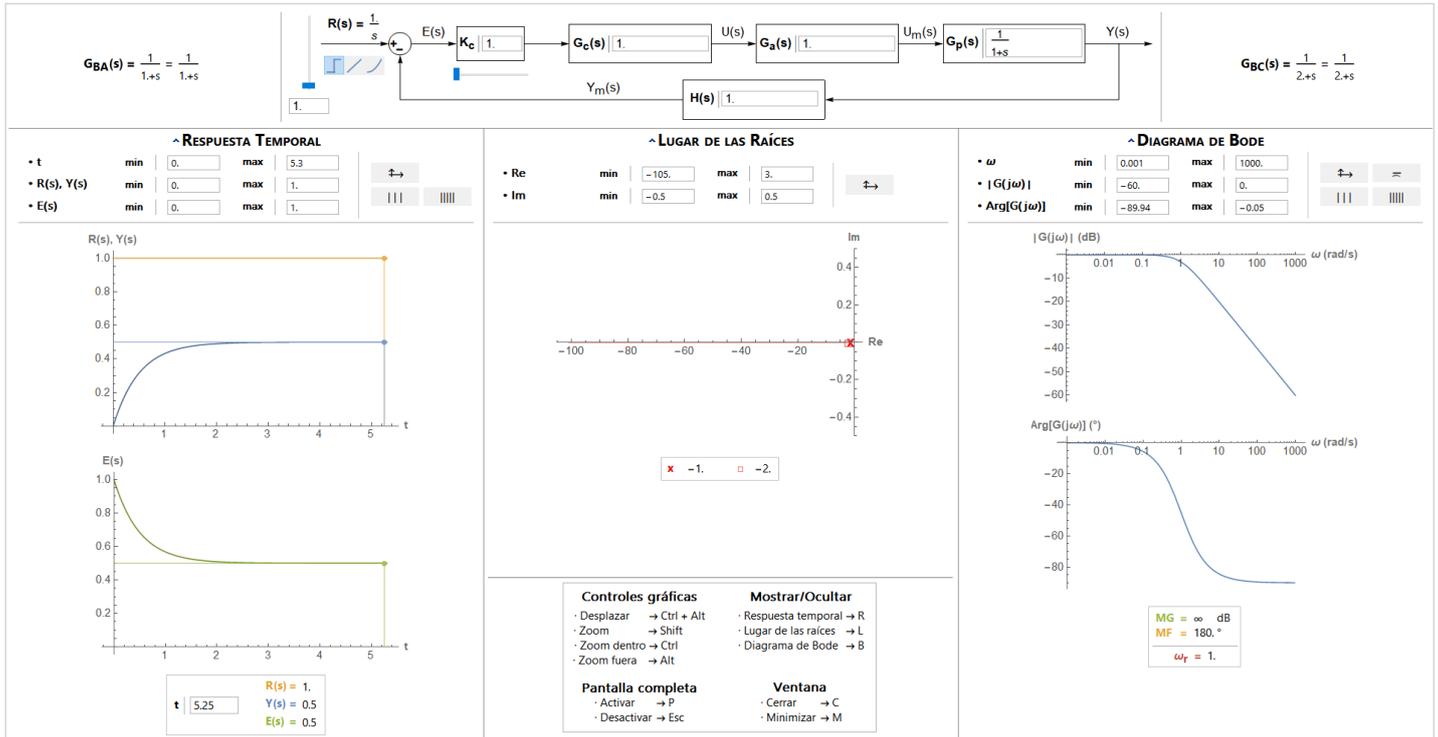


Figura 3. Interfaz de la herramienta software

La zona en la que se define el sistema se encuentra en la parte superior. En ella se encuentra el diagrama de bloques del sistema en el que se introducen los datos de entrada, es decir, las funciones de transferencia de los elementos del sistema y la función de la referencia. En esta zona también se muestran las funciones de transferencia del sistema en bucle abierto y bucle cerrado. Se encuentran a ambos lados del diagrama de bloques, separadas por divisores verticales.

La zona en la que se muestran los análisis del sistema se encuentra en la parte inferior, debajo de la del sistema. En esta zona se pueden ver los tres análisis (temporal, de polos y ceros y frecuencial) separados por divisores verticales. Las características de cada uno de ellos se detallan más adelante. En esta zona también se muestra un cuadro de ayuda con las combinaciones de teclas que tienen asignada una función.

La herramienta se muestra en una ventana que tiene las mismas características y funciones que cualquier ventana de Windows®, excepto que su tamaño es fijo. También tiene se puede visualizar la herramienta a pantalla completa pulsando la tecla p. Pulsando la tecla Esc se sale del modo pantalla completa. Mostrar la herramienta a pantalla completa permite visualizar las gráficas más grandes.

Por otro lado, al situar el ratón sobre algunos botones y abreviaturas aparece un texto explicativo, como se puede ver en la Figura 4.

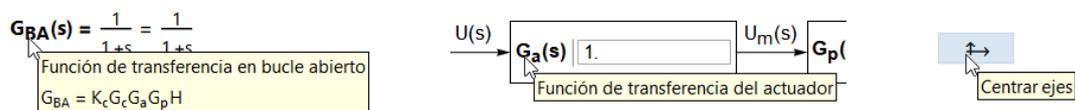


Figura 4. Tres ejemplos de texto explicativo

En la Figura 5 están señaladas las expresiones que cuentan con esta característica.

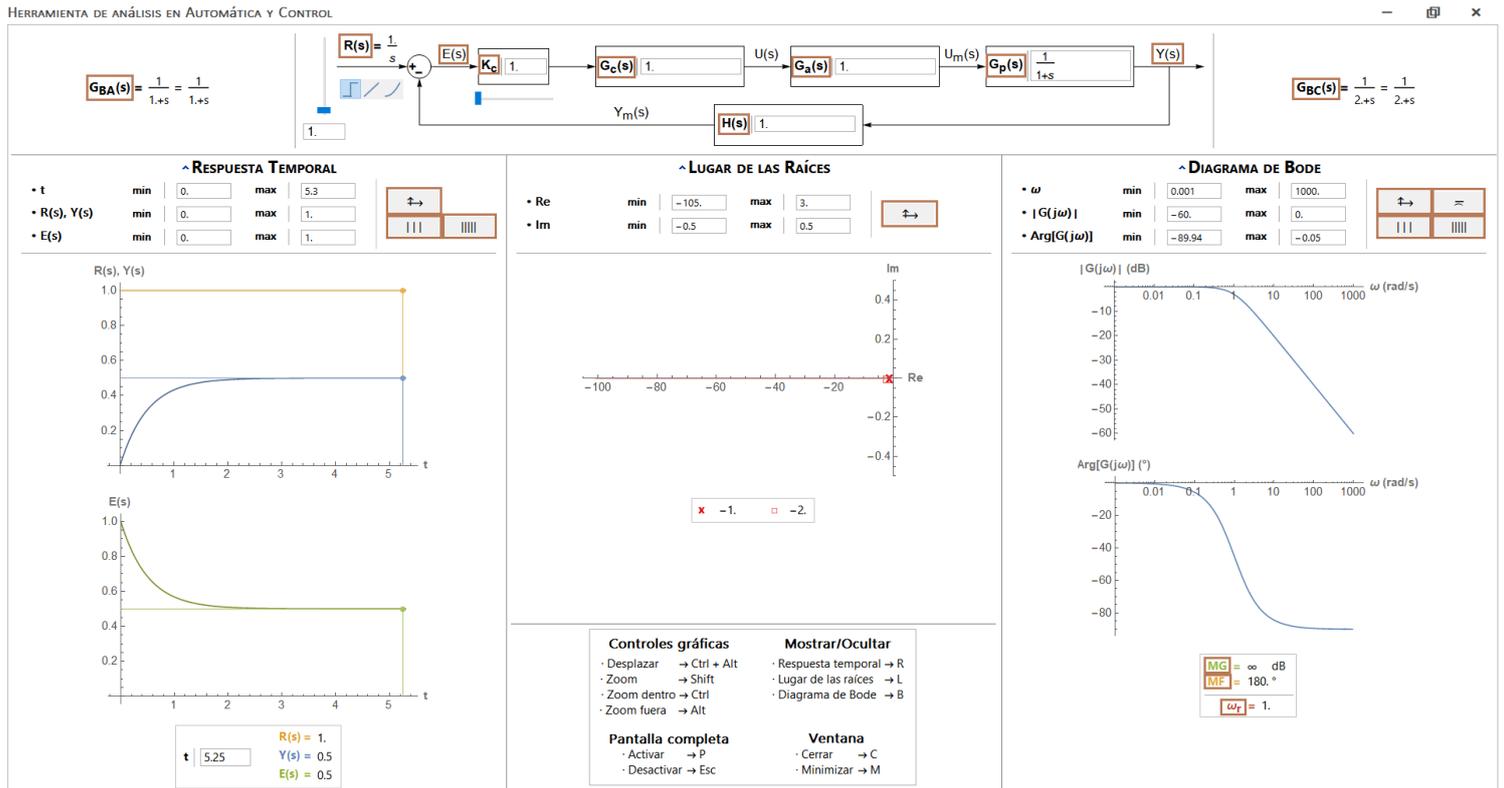


Figura 5. Expresiones que cuentan con mensaje explicativo

### 3. Definir el sistema

En este apartado se detallan las características de los controles que permiten introducir los datos de entrada, es decir, los datos del sistema. Comienza explicando los controles para definir la referencia y, a continuación, los de las funciones de transferencia. Estos controles se encuentran en la parte superior de la herramienta, dentro del diagrama de bloques.

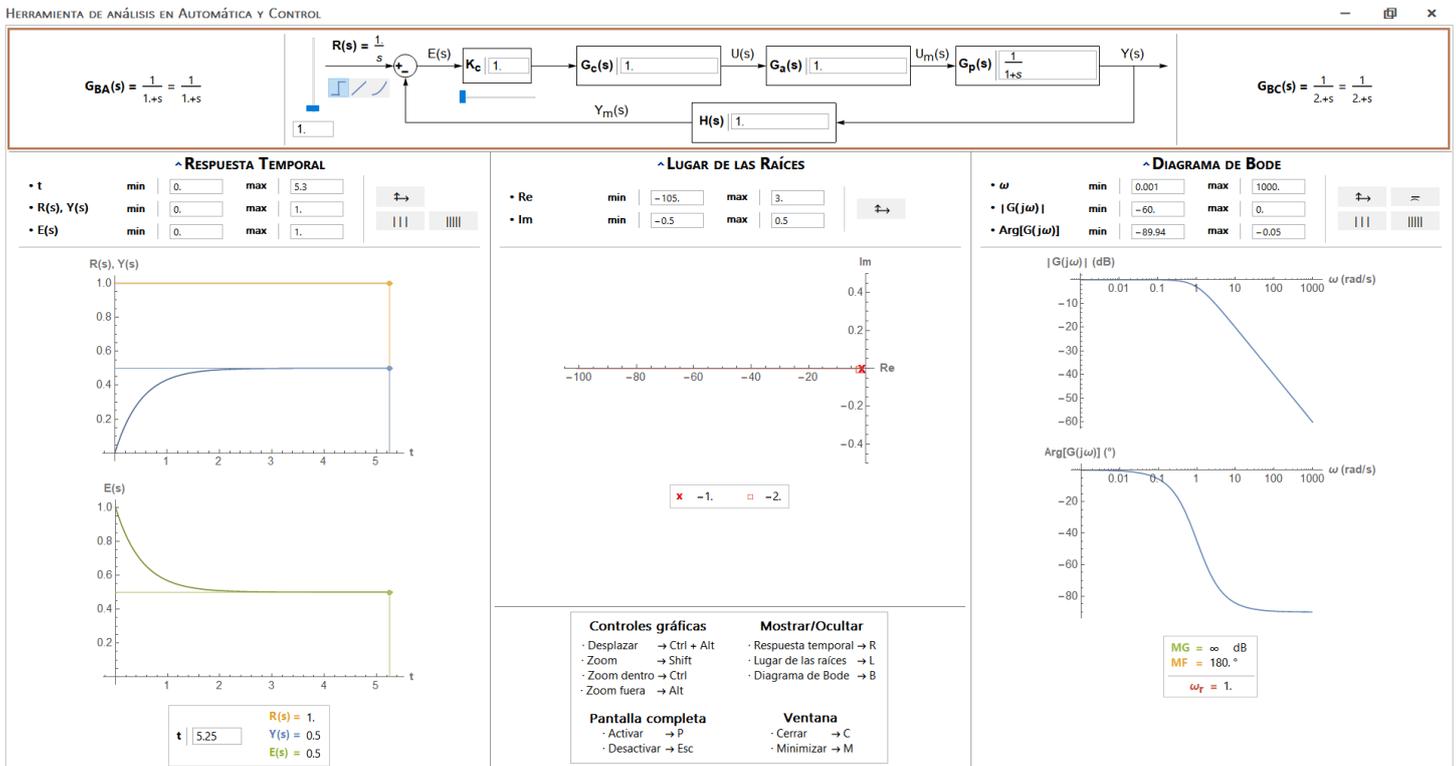


Figura 6. Controles para definir el sistema

#### 3.1. Definir la referencia

Los controles para definir la referencia \$R(s)\$ se encuentran en la parte izquierda del diagrama de bloques. Unos controles permiten ajustar el tipo de función y otros, su amplitud. Estos controles son:

1. Botones de selección. Permiten seleccionar el tipo de función de entre 3 opciones: escalón, rampa y parábola, de izquierda a derecha. Por defecto está seleccionada la entrada escalón.
2. Deslizador con campo de entrada. Permiten ajustar la amplitud de la función. El deslizador permite hacerlo de forma continua y el campo de entrada, discreta, para ajustar el valor con precisión. Por defecto su valor es 1.00, tiene un rango de 0.01 a 100.00 y admite 2 decimales.

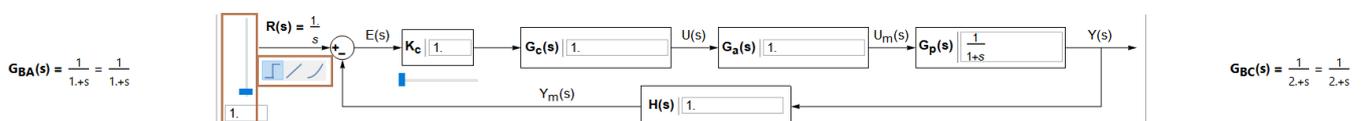


Figura 7. Controles para definir la referencia

La expresión de la referencia, que varía dinámicamente, se muestra al lado de los controles.

### 3.2. Definir las funciones de transferencia

Los controles para definir las funciones de transferencia se encuentran en la parte derecha del diagrama de bloques. Estos controles son campos de entrada para las funciones de transferencia del controlador, el actuador, la planta y el sensor y un deslizador con campo de entrada para la ganancia del controlador.

Los campos de entrada permiten introducir las funciones de transferencia de los elementos del sistema como constantes o cocientes de polinomios en  $s$ . Por defecto, la función de transferencia de la planta está fijada a  $1/(s+1)$  y, el resto, a 1.

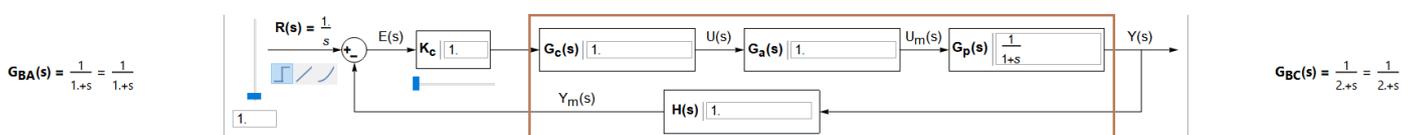


Figura 8. Controles para definir las funciones de transferencia

Cabe señalar que a la hora de introducir las funciones de transferencia, se pueden emplear las ayudas que ofrece Mathematica para escribir el numerador encima del denominador (pulsando Ctrl + / o empleando las paletas). Otra opción es introducirlas en una misma línea usando paréntesis y al acabar se cambiarán automáticamente a la otra forma, siempre que la expresión no sea muy grande. Estas dos formas de introducir las funciones de transferencia se pueden ver en la Figura 10.

$$\frac{1}{(s+1)(s+2)} \quad 1 / ((s+1)(s+2))$$

Figura 9. Dos formas de introducir las funciones de referencia

Cabe hacer una aclaración respecto a esto último: si la función de transferencia es demasiado extensa, el campo de entrada y su contenido puede cambiar de aspecto, como se puede ver en la Figura 10. Esto no altera el funcionamiento del programa. Si se desea evitar esto, se recomienda dividir la función de transferencia entre el controlador, el actuador y la planta. Aunque esto no sea estrictamente correcto en un sistema real, no influye en el análisis que se está haciendo del mismo.

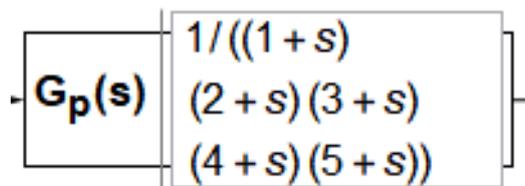


Figura 10. Ejemplo de función de transferencia demasiado extensa

El deslizador con campo de entrada permite ajustar la ganancia del controlador. La razón de que el controlador esté dividido en dos controles es que de esta forma se puede variar la ganancia de forma continua para ver cómo afecta su valor a los diferentes análisis. Por defecto su valor es 1.00 y tiene un rango de 0.01 a 100.00 y admite 2 decimales.

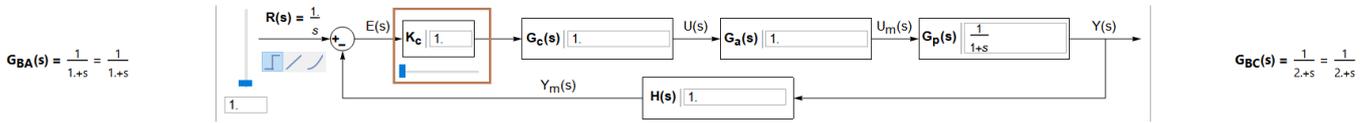


Figura 11. Controles para modificar la ganancia del controlador

## 4. Análisis del sistema

En este apartado se explican las características y utilidades de cada uno de los análisis del sistema que muestra la herramienta. Estos análisis son, de izquierda a derecha: el análisis temporal, el análisis de polos y ceros y el análisis frecuencial. Comienza explicando los controles que son comunes a los tres análisis: los que permiten mostrar y ocultar cada uno de los análisis y los que permiten variar el rango. A continuación, se describe cada uno de los análisis.

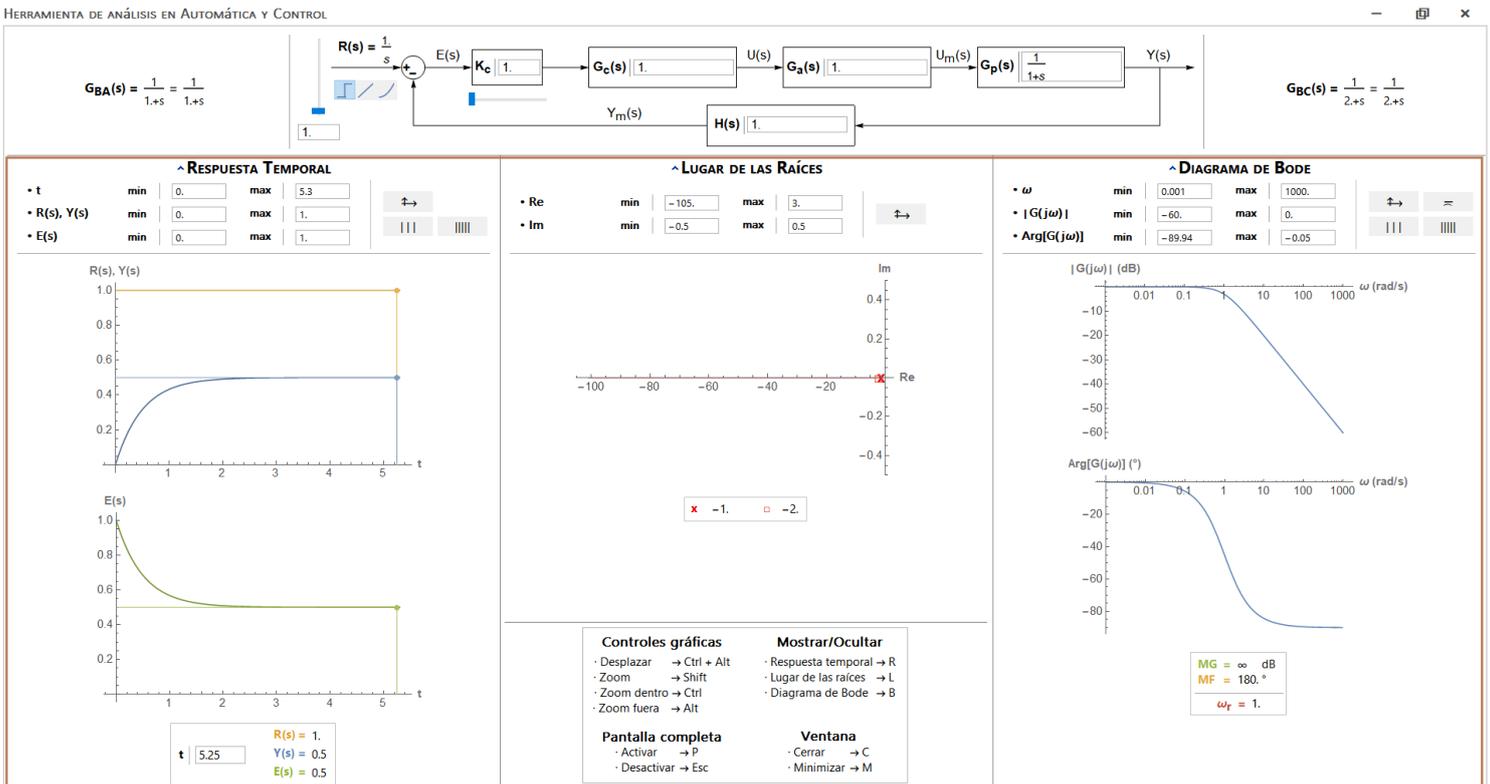


Figura 12. Análisis del sistema

### 4.1. Controles comunes a los tres análisis

Los controles comunes a los tres análisis son de dos tipos: los que permiten mostrar y ocultar cada uno de los análisis y los que permiten variar el rango.

Los controles que permiten mostrar y ocultar los análisis son botones, representados por flechas azules, que se encuentran a la derecha de cada uno de los nombres de los análisis. Ocultar los análisis que no sean de interés en cada situación resulta útil para aumentar la velocidad de los cálculos; de esta forma, los cambios que se realicen se verán reflejados con mayor rapidez en los análisis que sí se muestran. También se pueden usar las teclas r, l y b para mostrar y ocultar el análisis temporal, de polos y ceros y frecuencial, respectivamente.



Figura 13. Botones para mostrar y ocultar los análisis

Los controles que permiten variar el rango son tres: campos de entrada que permiten definir los valores de los ejes, un botón que permite centrar automáticamente los ejes y las combinaciones de teclas y ratón que permiten desplazarse y hacer zoom.

Los campos de entrada se encuentran en la parte superior de cada uno de los análisis. Permiten ajustar los valores máximo y mínimo de los ejes de las gráficas con precisión. La primera fila corresponde en todos los casos a las abscisas y la siguiente o las siguientes filas (dependiendo de si hay una o dos gráficas), a las ordenadas. El número de decimales así como el rango de valores para cada uno de ellos varían en cada uno de los tres análisis, por lo que estos valores se detallarán más adelante.

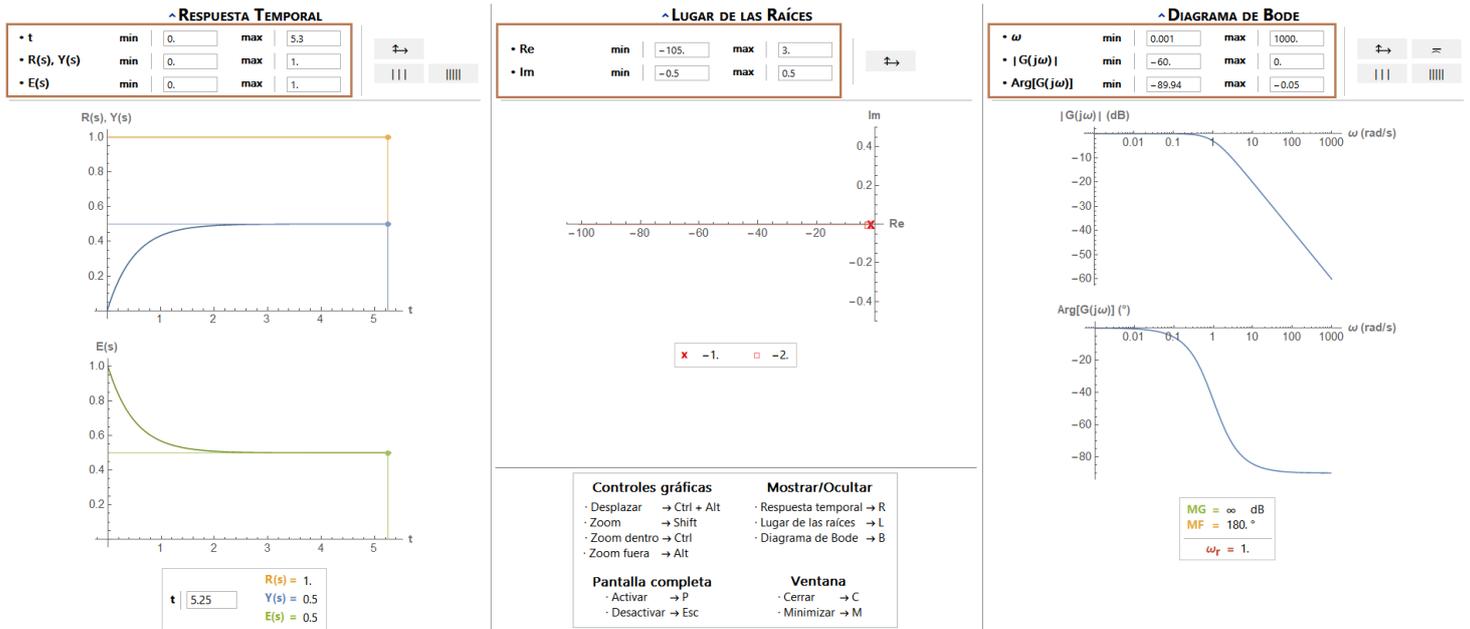


Figura 14. Campos de entrada de los ejes de los análisis

El botón para centrar los ejes permite reajustar automáticamente los ejes de la gráfica o las gráficas a sus valores adecuados para que sea visible toda la información relevante.



Figura 15. Botón para reajustar las gráficas de los análisis

Las combinaciones de teclas y ratón que permiten desplazarse y hacer zoom son las siguientes:

1. Manteniendo pulsadas las teclas Ctrl y Alt se puede hacer click y arrastrar el ratón por una gráfica para desplazarse por ella.
2. Manteniendo pulsada la tecla Shift se puede hacer click en un punto de la gráfica y arrastrar el ratón para crear un rectángulo sobre el que se hará zoom al soltar, siempre que la tecla Shift siga presionada.
3. Manteniendo pulsada la tecla Ctrl o Alt se puede hacer click sobre un punto de la gráfica para hacer zoom alrededor del punto en el que se hizo click hacia dentro o hacia fuera, respectivamente.

Estas y otras teclas que tienen asignadas una función aparecen en un cuadro de ayuda situado en la parte inferior central de la herramienta, debajo del lugar de las raíces, separado mediante un divisor horizontal.

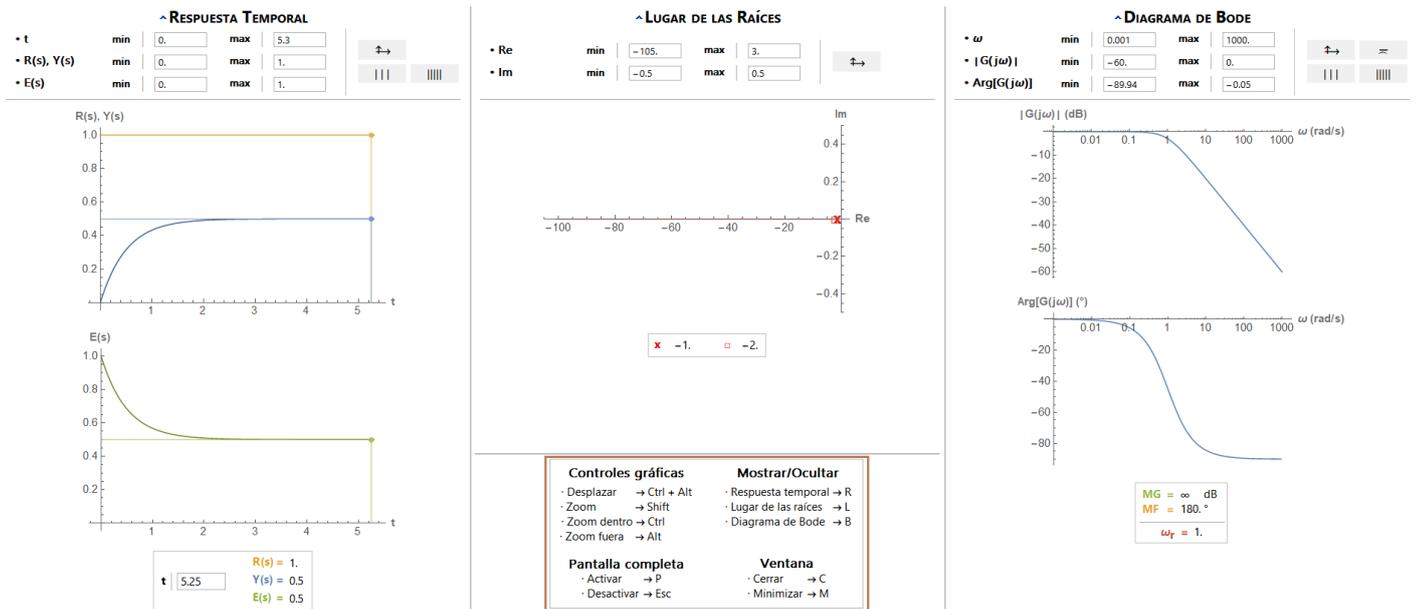


Figura 16. Cuadro de ayuda de la herramienta software

## 4.2. Análisis temporal

El análisis temporal del sistema se muestra a la izquierda y consta de dos gráficas (la respuesta temporal) y una leyenda.

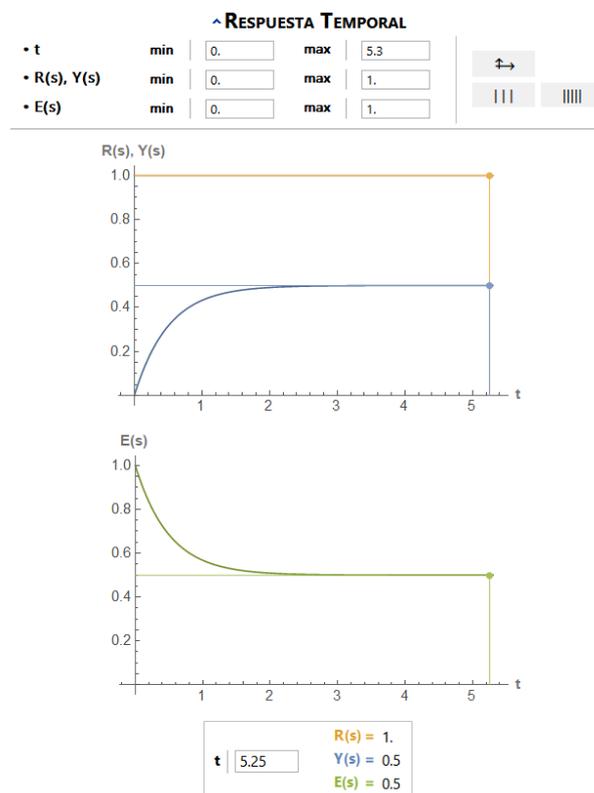


Figura 17. Análisis temporal

En la primera de las gráficas se representan la referencia y la salida del sistema y en la de abajo se representa el error, ambas respecto al tiempo. Cada una de las funciones tiene asociada un color para mayor claridad. En las gráficas también se muestran unas guías que se pueden mover arrastrando los puntos mediante los que se representan o haciendo click sobre las funciones.

En la leyenda se muestra el valor de estas guías, esto es, el instante de tiempo y el valor de las funciones ese instante, lo que permite medir los parámetros cuantificables más importantes del sistema: tiempo de establecimiento; y valores de la salida y el error en régimen estacionario. El valor del instante de tiempo es modificable no solo a través de las guías sino también mediante un campo de entrada en la leyenda. El campo de entrada admite 4 decimales, que es el mismo número de decimales con los que se muestran los valores de las funciones.

Los campos de entrada para modificar el rango de estas gráficas admiten todos 2 decimales como máximo. Los valores máximo y mínimo de las abscisas son 0.00 y 20.00, respectivamente; y los valores máximo y mínimo de las ordenadas de ambas gráficas son -110.00 y 110.00, respectivamente.

Los botones con líneas verticales que están junto al de centrar los ejes, a la derecha de los campos de entrada de los valores del rango, sirven para mostrar y ocultar un mallado en la gráfica. El primero, que tiene tres líneas verticales, permite mostrar líneas en los valores principales y el segundo, en todos los valores.



Figura 18. Botones para mostrar y ocultar las cuadrículas del análisis temporal

### 4.3. Análisis de los polos y ceros

El análisis de polos y ceros del sistema se muestra en el centro y consta de una gráfica (el lugar de las raíces) y una leyenda.

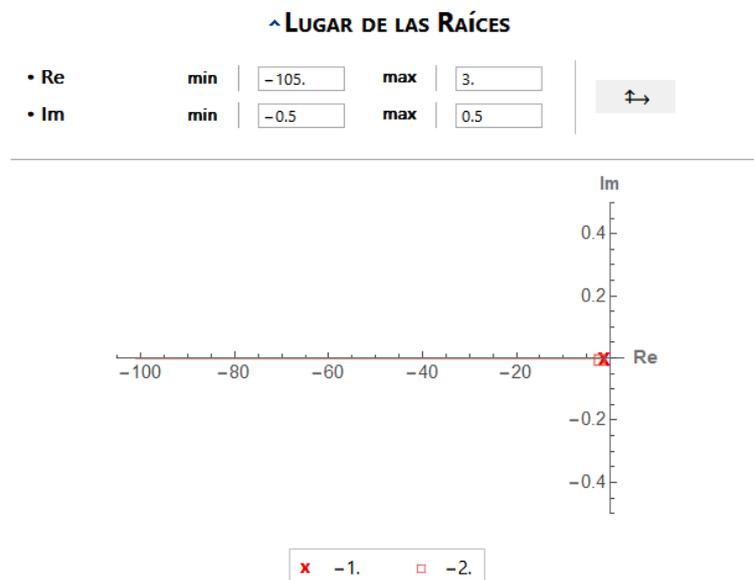


Figura 19. Análisis de la ubicación de polos y ceros

En la gráfica se representan las trayectorias de los polos del sistema cuando varía la ganancia del controlador, y también los polos en bucle abierto (cuando la ganancia es 0), los polos en bucle cerrado (para la ganancia fijada) y los ceros del sistema. Los polos en bucle abierto se representan por cruces ("x"), los polos en bucle cerrado por cuadrados ("□") y los ceros por círculos ("o"). Cada polo y cada cero tiene un color asociado y en el caso de los polos, la trayectoria correspondiente a cada uno tiene el mismo color que el polo.

La leyenda muestran los valores de los polos en bucle abierto, polos en bucle cerrado y ceros del sistema. La utilidad de esta leyenda es conocer de un vistazo los valores de los ceros y polos en bucle abierto, así como el valor de los polos en bucle cerrado para la ganancia del controlado especificada.

Los campos de entrada para modificar el rango de estas gráficas admiten todos 2 decimales como máximo. Los valores máximo y mínimo de las abscisas y las ordenadas son -110.00 y 110.00, respectivamente.

Por último, cabe señalar que el lugar de las raíces presenta discontinuidades en los colores utilizados en cada polo para algunos sistemas con número impar de polos. Esto se debe a limitaciones en la construcción del lugar de las raíces, aunque la forma del lugar de las raíces no se ve alterada.

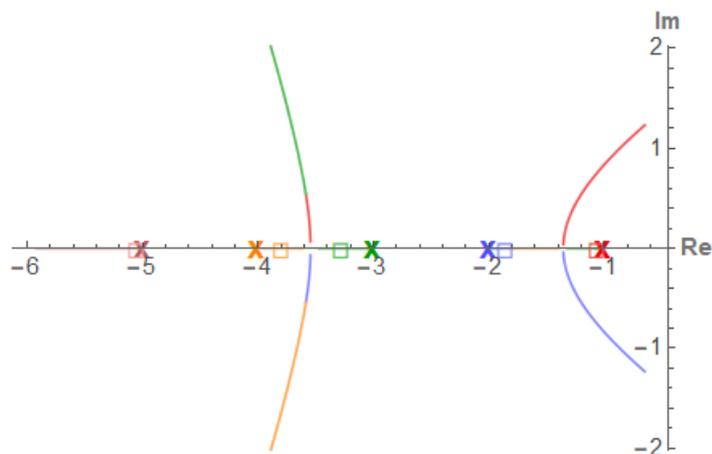
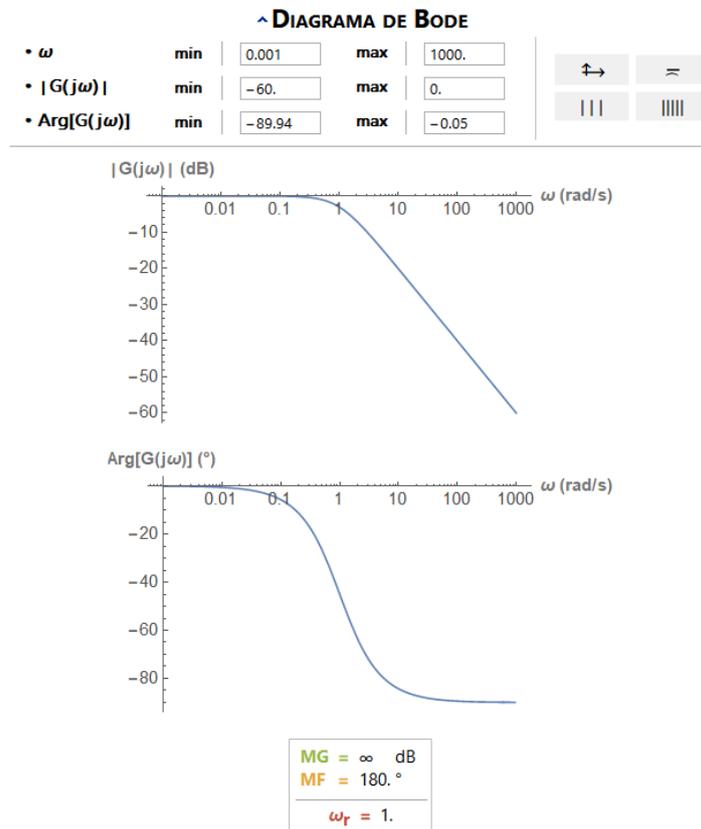


Figura 20. Ejemplo de discontinuidad del lugar de las raíces

#### 4.4. Análisis frecuencial

El tercer resultado es el diagrama de Bode del sistema, que muestra la amplificación del sistema ante respuestas sinusoidales de diferente frecuencia en la gráfica superior y el desfase en esas condiciones en la inferior.



**Figura 21. Análisis frecuencial**

El análisis frecuencial del sistema se muestra a la derecha y consta de dos gráficas (el diagrama de Bode) y una leyenda.

En la primera de las gráficas se representa el módulo de la función de transferencia en bucle abierto y, en la otra, su fase, ambas respecto a la frecuencia. En las gráficas también se representan los márgenes de ganancia y fase mediante flechas, y cada uno tiene asociado un color. Además, en la gráfica del módulo es posible visualizar el módulo asintótico.

En la leyenda se muestran los valores del margen de ganancia y del margen de fase, así como las frecuencias de ruptura del sistema. Todos estos valores se muestran con un máximo de 4 decimales.

Los campos de entrada para modificar las ordenadas de estas gráficas admiten 2 decimales como máximo. Los valores máximo y mínimo de las ordenadas de la gráfica del módulo son gráficas son -200.00 y 200.00, respectivamente; y los de las ordenadas de la gráfica de la fase, -360.00 y 360.00, respectivamente. En cuanto a las abscisas, el número de decimales con los que se pueden introducir los valores máximo y mínimo varía dependiendo del valor que tomen. Si el valor es mayor o igual a 1, no admiten ningún decimal. En el resto de casos, el número de decimales es igual a una unidad mayor que el valor absoluto del exponente del 10 cuando el número está escrito en notación científica. Es decir, que para valores entre 0.001 y 0.009, se admiten 3 decimales; entre 0.01 y 0.09, dos; etc.

Los botones con líneas verticales que están junto al de centrar los ejes, a la derecha de los campos de entrada de los valores del rango, sirven para mostrar y ocultar un mallado en la gráfica, al igual que en las gráficas del análisis temporal. El primero, que

tiene tres líneas verticales, permite mostrar líneas en los valores principales y el segundo, en todos los valores. El cuarto botón permite mostrar y ocultar el módulo asintótico.



**Figura 22. Izquierda: Botones para mostrar y ocultar las cuadrículas de las gráficas del análisis frecuencial. Derecha: Botón para mostrar y ocultar el módulo asintótico**