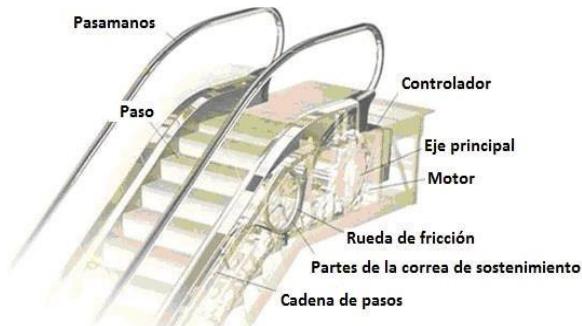




SISTEMAS DE CONTROL LABORATORIO



PROYECTO GLOBAL DE ASIGNATURA (PGA) Regulador de la velocidad de una escalera mecánica



Departamento de Ingeniería Telemática y Electrónica

25 de noviembre de 2024

Estela Mora Barba y Hugo Sánchez Rey

Pareja 4 – L05L06

ÍNDICE

ÍNDICE	2
ÍNDICE DE FIGURAS	4
ÍNDICE DE TABLAS	6
Indicaciones previas:	7
Fase 1: Modelado y caracterización de la planta en lazo abierto	8
Ejercicio 1.....	8
Ejercicio 2.....	14
Ejercicio 3.....	16
Ejercicio 4.....	22
Ejercicio 5.....	23
Fase 2: Diseño directo del regulador	27
Ejercicio 6.....	27
Ejercicio 7.....	29
Ejercicio 8.....	29
Ejercicio 9.....	30
Fase 3: Validación del diseño directo	31
Ejercicio 10.....	31
Ejercicio 11.....	34
Ejercicio 12.....	36
Ejercicio 13.....	38
Ejercicio 14.....	41
Fase 4: Diseño del regulador PID.....	42
Ejercicio 15.....	42
Ejercicio 16.....	44
Ejercicio 17.....	45
Ejercicio 18.....	46
Fase 5: Validación del regulador PID	47
Ejercicio 19.....	47
Ejercicio 20.....	50
Ejercicio 21.....	52
Ejercicio 22.....	54
Ejercicio 23.....	57
Anexo 1: Fase 1	58
Ejercicio 1: Obtener la FdeT de la planta en lazo abierto.	58
Ejercicio 2: Validar el modelo mediante Simulink.	59
Ejercicio 3: Modelación de la saturación y la zona muerta.....	60
Ejercicio 4: Equivalente discreto en régimen permanente y transitorio.	62
Ejercicio 5: Respuesta de la planta en régimen permanente para $\pm 1V$ con perturbación (10 personas = 700 kg).....	62
Anexo 2: Fase 2	64
Ejercicio 7: Dibujar el diagrama de bloques del sistema.	64
Ejercicio 8: Calcular la FdeT del sistema y caracterizar su salida a entradas escalón $\pm 1 \text{ m/s}$. ..	64
Ejercicio 9: Acción de control y velocidad de la escalera en régimen permanente.....	64
Anexo 3: Fase 3	66
Ejercicio 10: Simulación del sistema de control en Simulink.....	66

Ejercicio 11: Características de la respuesta para entrada +1 m/s.....	67
Ejercicio 12: Características de la respuesta para entrada -1 m/s.....	67
Ejercicio 13: Régimen permanente de la acción de control y la velocidad de la escalera.....	67
Anexo 4: Fase 4.....	69
Ejercicio 15: Calcular la FdeT del regulador PID.	69
Ejercicio 16: Calcular la FdeT del sistema y obtener los polos.....	69
Ejercicio 18: Acción de control y velocidad de la escalera en régimen permanente para entradas escalón ± 1 m/s.	70
Anexo 5: Fase 5.....	71
Ejercicio 19: Simulación del sistema de control en Simulink.....	71
Ejercicio 20: Características de la respuesta para entrada +1 m/s.....	72
Ejercicio 21: Características de la respuesta para entrada -1 m/s.....	73
Ejercicio 22: Régimen permanente de la acción de control y la velocidad de la escalera.....	73
Anexo 6: Funciones	75
representar.m:	75
salida.m:	76
comparacion.m:	77
escalera.m:	78
simulink.m:	79
Anexo 7: Código fuente del regulador Truxal.....	80
Anexo 8: Código fuente del regulador PID	81

ÍNDICE DE FIGURAS

Figura 1 Circuito SM	8
Figura 2 Motor CC	8
Figura 3 GUI Entrada 1V	9
Figura 4 GUI Entrada 2V	9
Figura 5 GUI Entrada 5V	10
Figura 6 GUI Entrada -1V	10
Figura 7 GUI Entrada -2V	10
Figura 8 GUI Entrada -5V	11
Figura 9. Entrada y salida de la planta con entradas escalón positivas.....	12
Figura 10. Entrada y salida de la planta con entradas escalón negativas	13
Figura 11. Modelo FdT de la planta	14
Figura 12. Comparación de la respuesta simulada de la planta y la respuesta real de la planta	15
Figura 13. GUI Entrada +10V	16
Figura 14. GUI Entrada -10V	16
Figura 15. Saturación positiva	17
Figura 16. Saturación negativa	18
Figura 17. GUI Entrada 0.19V	19
Figura 18. Zona muerta positiva.....	20
Figura 19. Zona muerta negativa	21
Figura 20. Modelo de la planta con Zona muerta y Saturación.....	21
Figura 21. Salida del modelo de la planta con zona muerta y saturación	22
Figura 22. Salida real, simulada y discreta a entrada escalón de 1V	23
Figura 23 GUI Entrada 1V y 10 personas	24
Figura 24 GUI Entrada -1V y 10 personas	24
Figura 25. Perturbación máxima con entrada escalón +1V	25
Figura 26. Perturbación máxima con entrada escalón -1V	25
Figura 27. Modelo en Simulink de la planta con perturbación	25
Figura 28. Salida del modelo de Simulink para las perturbaciones a entrada escalón $\pm 1V$	26
Figura 29. Diagrama de bloques del sistema (Truxal)	29
Figura 30. Modelo sistema con regulador Truxal sin alinealidades	31
Figura 31. Modelo sistema con regulador Truxal con alinealidades	31
Figura 32. Comparación de simulaciones Truxal	32
Figura 33. GUI Truxal 1 m/s	34
Figura 34. Comparación respuestas +1m/s Truxal.....	35
Figura 35. GUI Truxal -1 m/s	36
Figura 36. Comparación respuestas -1m/s Truxal	37
Figura 37 GUI LC Entrada 1 m/s con 5 Personas.....	38
Figura 38 GUI LC Entrada 1 m/s con 10 Personas.....	38
Figura 39 GUI LC Entrada -1 m/s con 5 Personas	39
Figura 40 GUI LC Entrada -1 m/s con 10 Personas	39
Figura 41. Comparación perturbaciones Truxal	40
Figura 42. Lugar de raíces de la cadena abierta resultante con un regulador P	43
Figura 43. Lugar de raíces de la cadena abierta resultante con un regulador I	43
Figura 44. Polos y ceros del sistema	45
Figura 45. Modelo sistema con regulador PID sin alinealidades	47
Figura 46. Modelo sistema con regulador PID con alinealidades	47
Figura 47. Comparación de simulaciones PID	48

Figura 48. GUI PID 1 m/s	50
Figura 49. Comparación respuestas +1m/s PID	51
Figura 50. GUI PID -1 m/s.....	52
Figura 51. Comparación respuestas -1m/s PID	53
Figura 52 GUI LC F5 Entrada 1 m/s y 5 personas	54
Figura 53 GUI LC F5 Entrada 1 m/s y 10 Personas	54
Figura 54 GUI LC F5 Entrada -1 m/s y 5 personas	55
Figura 55 GUI LC F5 Entrada -1 m/s y 10 personas	55
Figura 56. Comparación perturbaciones PID	56

ÍNDICE DE TABLAS

Tabla 1. Valores de Km y Tau a distintas entradas escalón	11
Tabla 2. Velocidad de la escalera y acción de control a entrada escalón ± 1 m/s con diferentes perturbaciones (Truxal).....	30
Tabla 3. Características Truxal	33
Tabla 4. Comparación Truxal +1 m/s.....	35
Tabla 5. Comparación Truxal -1 m/s.....	37
Tabla 6. Características perturbaciones Truxal.....	40
Tabla 7. Velocidad de la escalera y acción de control a entrada escalón ± 1 m/s con diferentes perturbaciones (PID).....	46
Tabla 8. Características PID	49
Tabla 9. Comparación PID +1 m/s.....	51
Tabla 10. Comparación PID -1 m/s.....	53
Tabla 11. Características perturbaciones PID	56

Indicaciones previas:

El código desarrollado en Matlab se puede encontrar por fases en los anexos de este documento.

Además, se han creado distintas funciones para simplificar el código y hacerlo más legible, que también se encuentran disponibles en el anexo.

Todo el proyecto se encuentra disponible en siguiente enlace:

https://github.com/Estelamb/PGA_Escaleras_Mecanicas

Fase 1: Modelado y caracterización de la planta en lazo abierto

Ejercicio 1

Capture y caracterice la salida de la planta en función de la tensión de entrada con un procedimiento análogo al empleado en la Práctica 2 y obtenga su función de transferencia (FdeT) en las unidades que considere más adecuadas para trabajar en el desarrollo de todo el diseño.

Antes que nada, debemos tener en cuenta el sistema que se nos está siendo presentado. El sistema de la planta relaciona una tensión $v_e(t)$ de acción con una velocidad angular $\omega_r(t)$. Este sistema es el sistema motorizado o SM, indicado en el enunciado compuesto por la siguiente estructura:

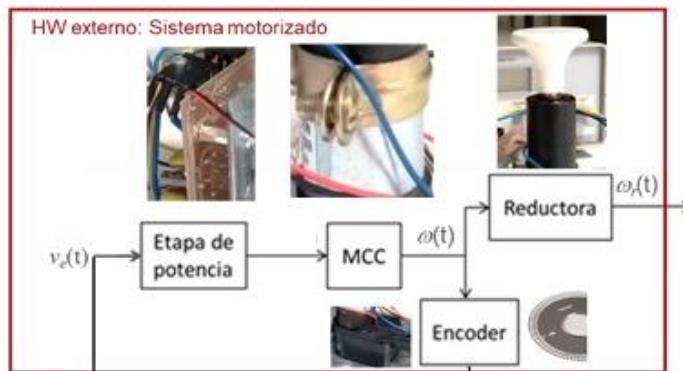


Figura 1 Circuito SM

Este sistema está compuesto por una etapa de potencia y un motor de corriente continua (MCC) y puede ser modelado de la siguiente manera:

1. Para un motor genérico de corriente continua que relaciona tensión con posición angular, cumple este modelo:

$$\text{Par generado: } K_1 i_a(t) \rightarrow K_1 I_a(s)$$

$$\text{Par resistente: } J \frac{d^2\theta(t)}{dt^2} \rightarrow J * s^2 * \theta(s)$$

$$\text{Tensión contraelectromotriz: } v_m(t) = K_2 \frac{d\theta(t)}{dt} \rightarrow K_2 s \theta(s)$$

$$\text{FdT: } G_p(s) = \frac{\theta(s)}{V_i(s)} = \frac{K_m}{s(\tau s + 1)}$$

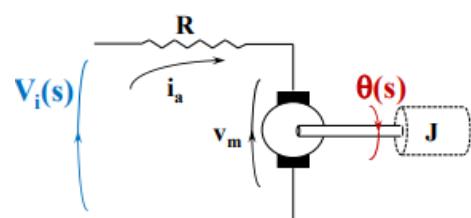


Figura 2 Motor CC

2. Estamos relacionando velocidad angular, para el mismo tipo de sistema, por lo que del sistema anterior podemos realizar el siguiente desarrollo:

$$\text{Velocidad angular: } w(t) = \frac{d\theta(t)}{dt} \rightarrow w(s) = s\theta(s)$$

$$FdeT: G_v(s) = \frac{w(s)}{V_i(s)} = \frac{K_m}{\tau s + 1} \left[\frac{m}{s * V} \right]$$

$$\text{Respuesta ante escalón: } w(t) = R K_m (1 - e^{-t/\tau})$$

Para poder caracterizar correctamente el sistema, se han realizado distintas mediciones en el entorno GUI proporcionado en el laboratorio. En este entorno se han realizado distintas mediciones para caracterizar el sistema y en cada una de esta se ha realizado las siguientes operaciones:

En primer lugar, realizamos capturas del dispositivo a tensiones de entrada de $\pm 1V$, $\pm 2V$ y $\pm 5V$:

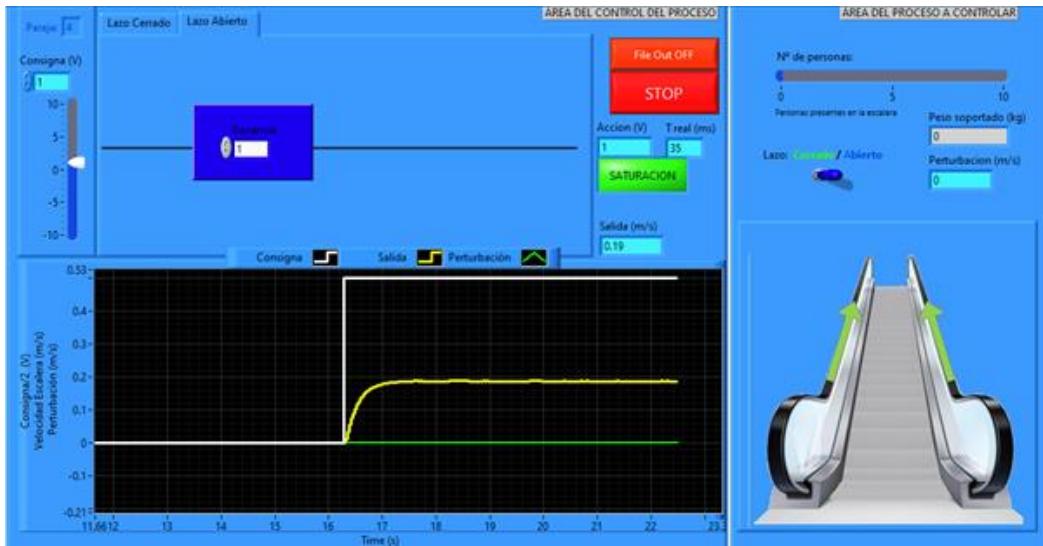


Figura 3 GUI Entrada 1V

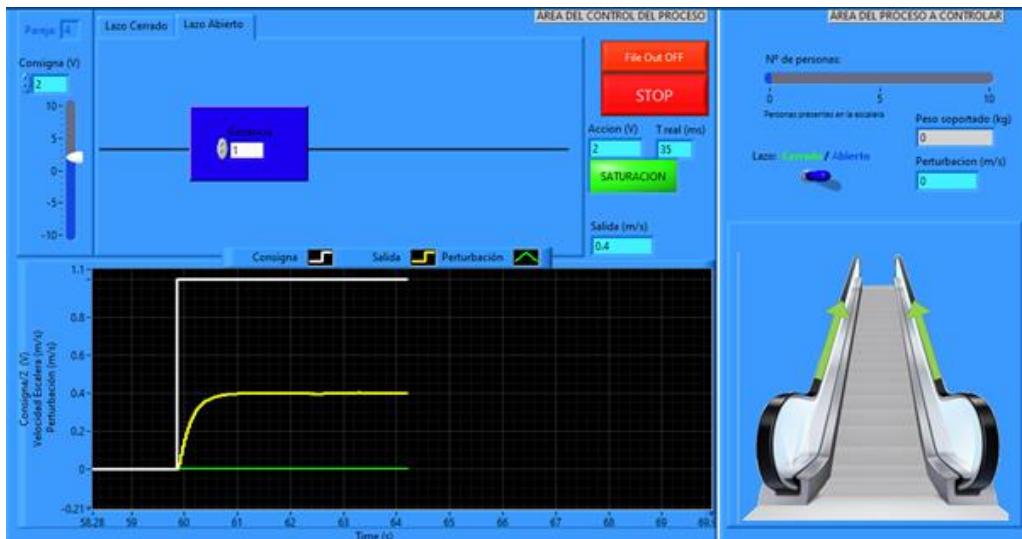


Figura 4 GUI Entrada 2V

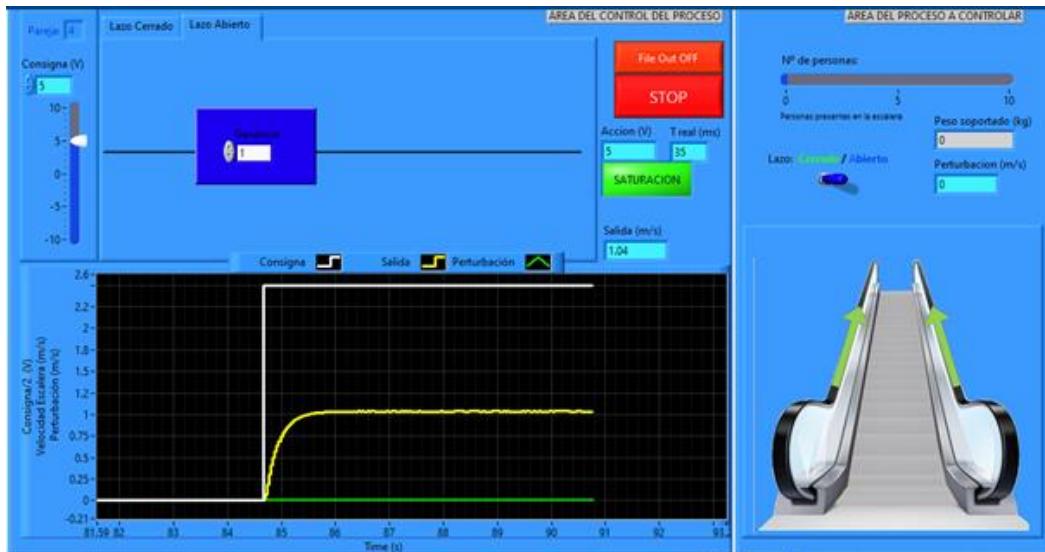


Figura 5 GUI Entrada 5V

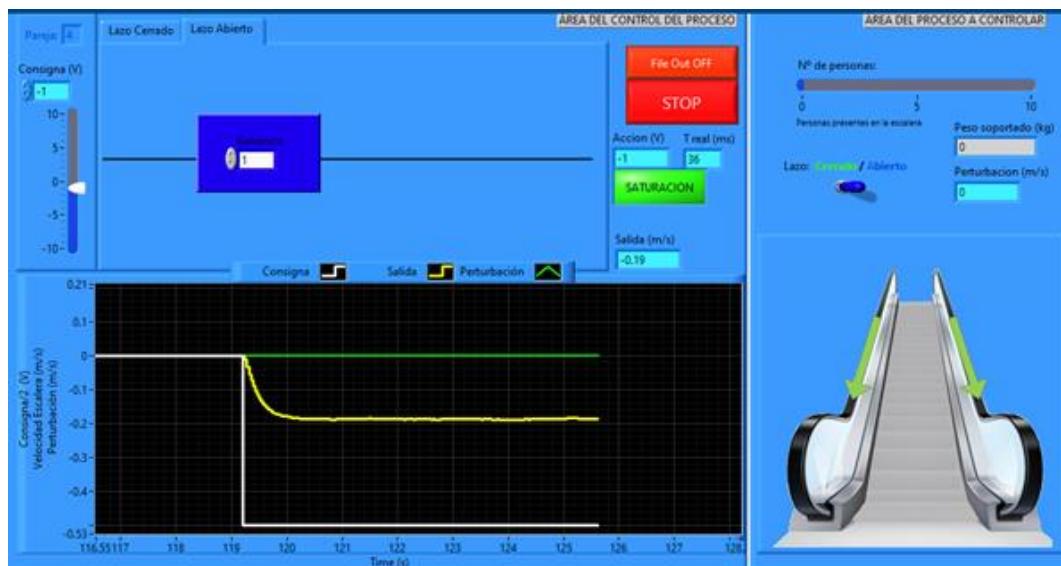


Figura 6 GUI Entrada -1V

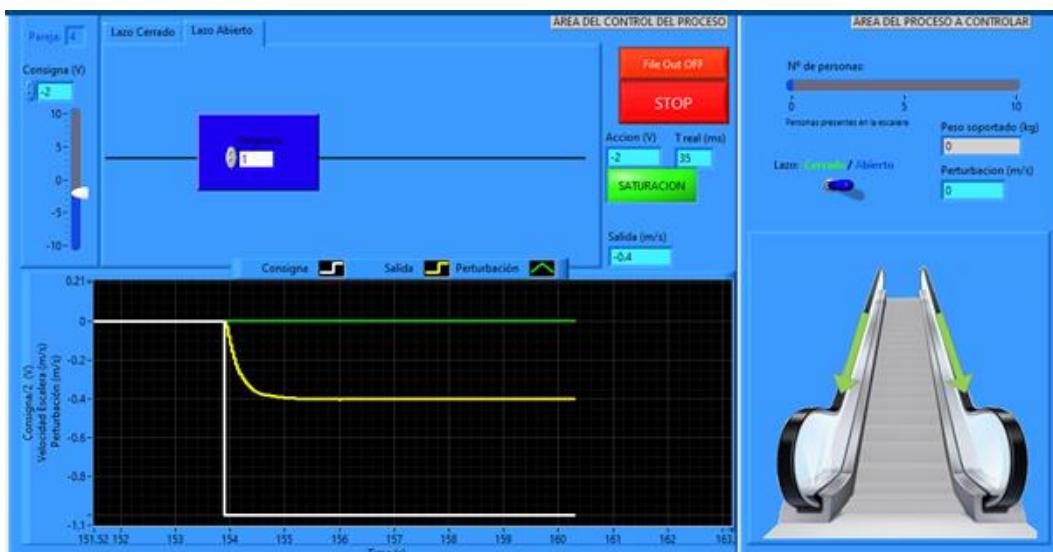


Figura 7 GUI Entrada -2V

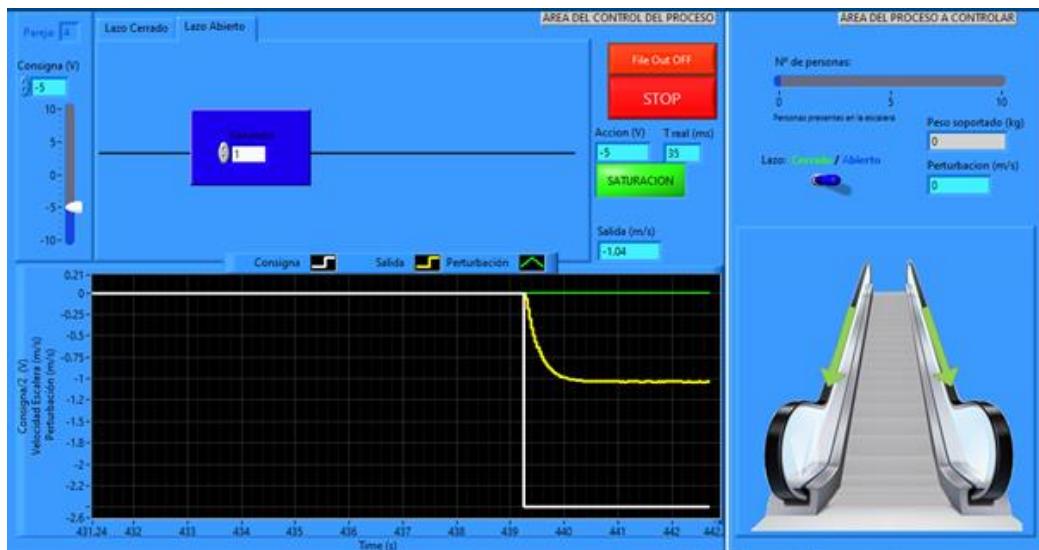


Figura 8 GUI Entrada -5V

De dichas capturas, obtenemos la K_m y la τ de cada salida de la siguiente manera:

1. Cargar el archivo .txt con los valores capturados.
2. Representar la entrada escalón y la respuesta de la planta en lazo abierto.
3. Recortar la señal de salida al instante en el que su valor es distinto de 0.
4. Ajustar la señal a 5 segundos.
5. **Hallar la K_m :** Mediante la media de las últimas muestras dividida por la tensión de la entrada escalón.

$$K_m = \frac{c(\infty)}{R} \left[\frac{V}{V} \right] \rightarrow R \equiv \text{Valor entrada escalón}$$

6. **Hallar la τ :** Instante del tiempo en el que la señal vale el 63.2% de la K_m .

$$w(t)|_{t=\tau} = RK_m(1 - e^{-1}) = 0.632RK_m$$

Estos han sido los resultados:

Entrada escalón	K_m [(m/s)/V]	τ [s]
1V	0.1920	0.2625
-1V	0.1925	0.2625
2V	0.2030	0.2625
-2V	0.2023	0.2625
5V	0.2079	0.2625
-5V	0.2087	0.2625

Tabla 1. Valores de K_m y τ a distintas entradas escalón

En la siguiente figura, se pueden observar las **entradas (azul)** y **salidas (naranja)** de la planta con entradas escalón **positivas**:

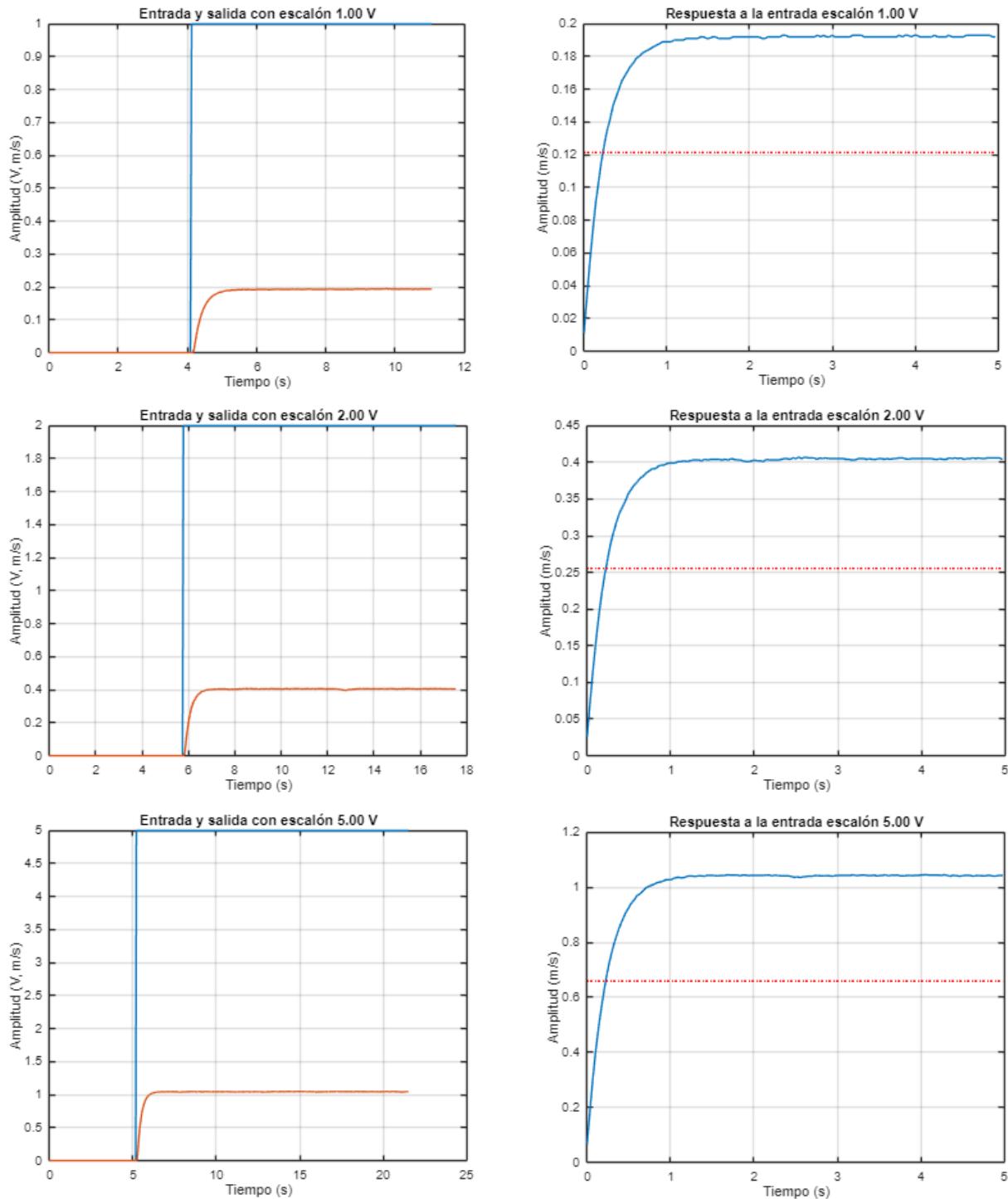


Figura 9. Entrada y salida de la planta con entradas escalón positivas

En la siguiente figura, se pueden observar las **entradas (azul)** y **salidas (naranja)** de la planta con entradas escalón **negativas**:

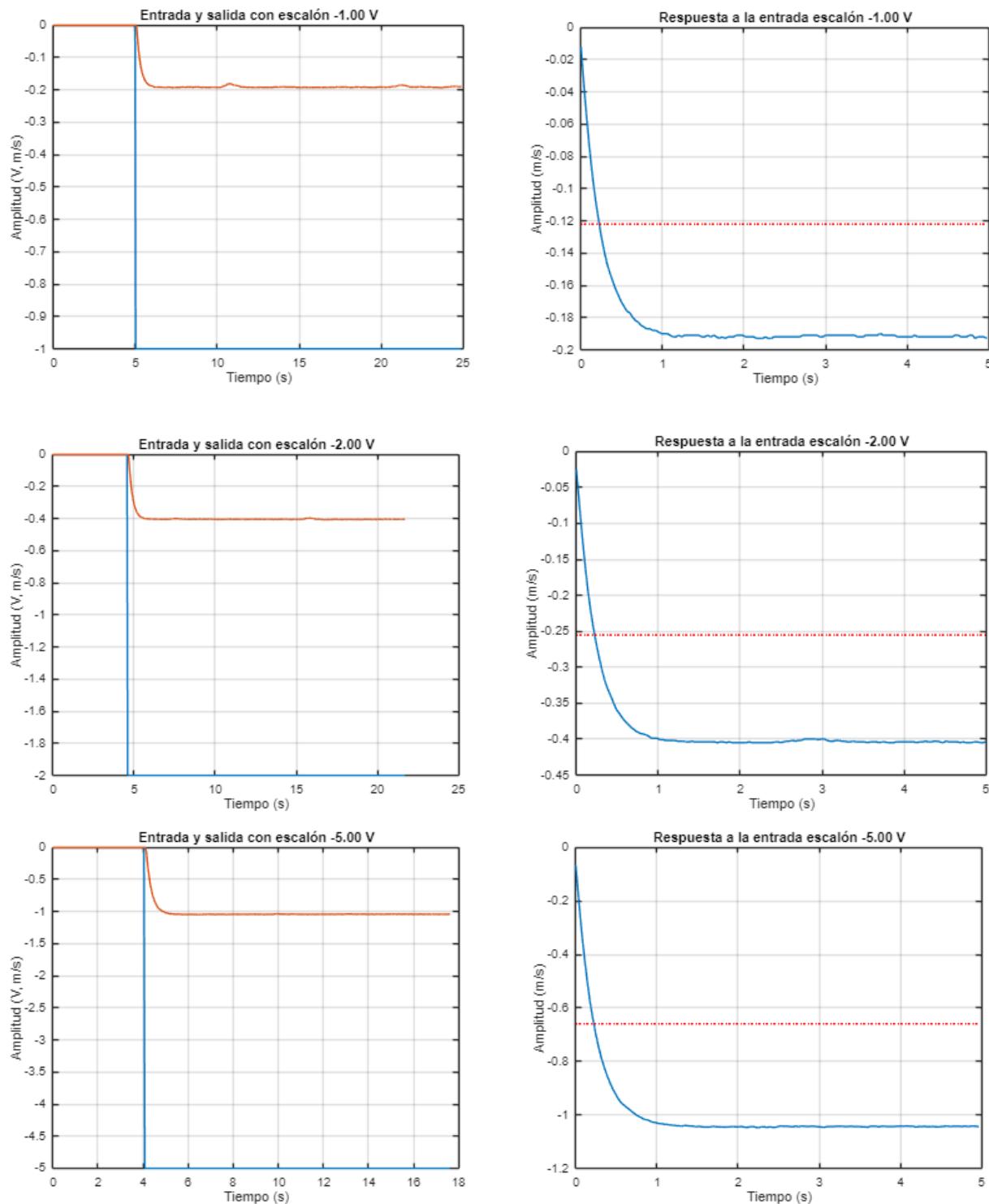


Figura 10. Entrada y salida de la planta con entradas escalón negativas

Después, para tener un valor más preciso de la K_m y la τ de la planta, hacemos la media de todas las K_m y τ obtenidas para cada entrada escalón, resultando en:

$$K_m = 0.2011 \left[\frac{m/s}{V} \right]$$

$$\tau = 0.2625 [s]$$

Finalmente, la **Función de Transferencia de la planta** es:

$$G(s) = \frac{K_m}{\tau * s + 1} = \frac{0.2011}{0.2625s + 1} \left[\frac{m/s}{V} \right]$$

De esta FdeT, también se pueden obtener diferentes valores de interés:

- La planta tiene un polo en $p = -3.8095$
- El tiempo de establecimiento de la planta es: $t_s = 0.7875 [s]$

Ejercicio 2

Corrobore la validez del modelo mediante su simulación en Matlab/Simulink, como también se hizo en la Práctica 2:

Para corroborar la validez del modelo obtenido en el apartado anterior, se ha realizado el siguiente modelo de simulación en Simulink:

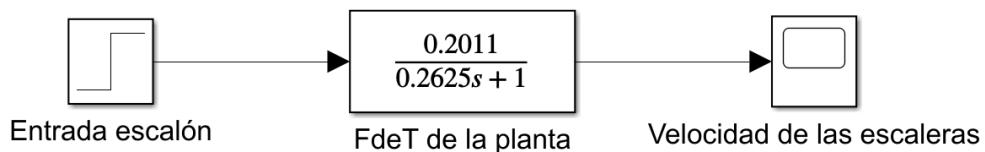


Figura 11. Modelo FdeT de la planta

En las siguientes figuras se puede observar la **salida de la simulación (azul)** y la **salida real (naranja)**:

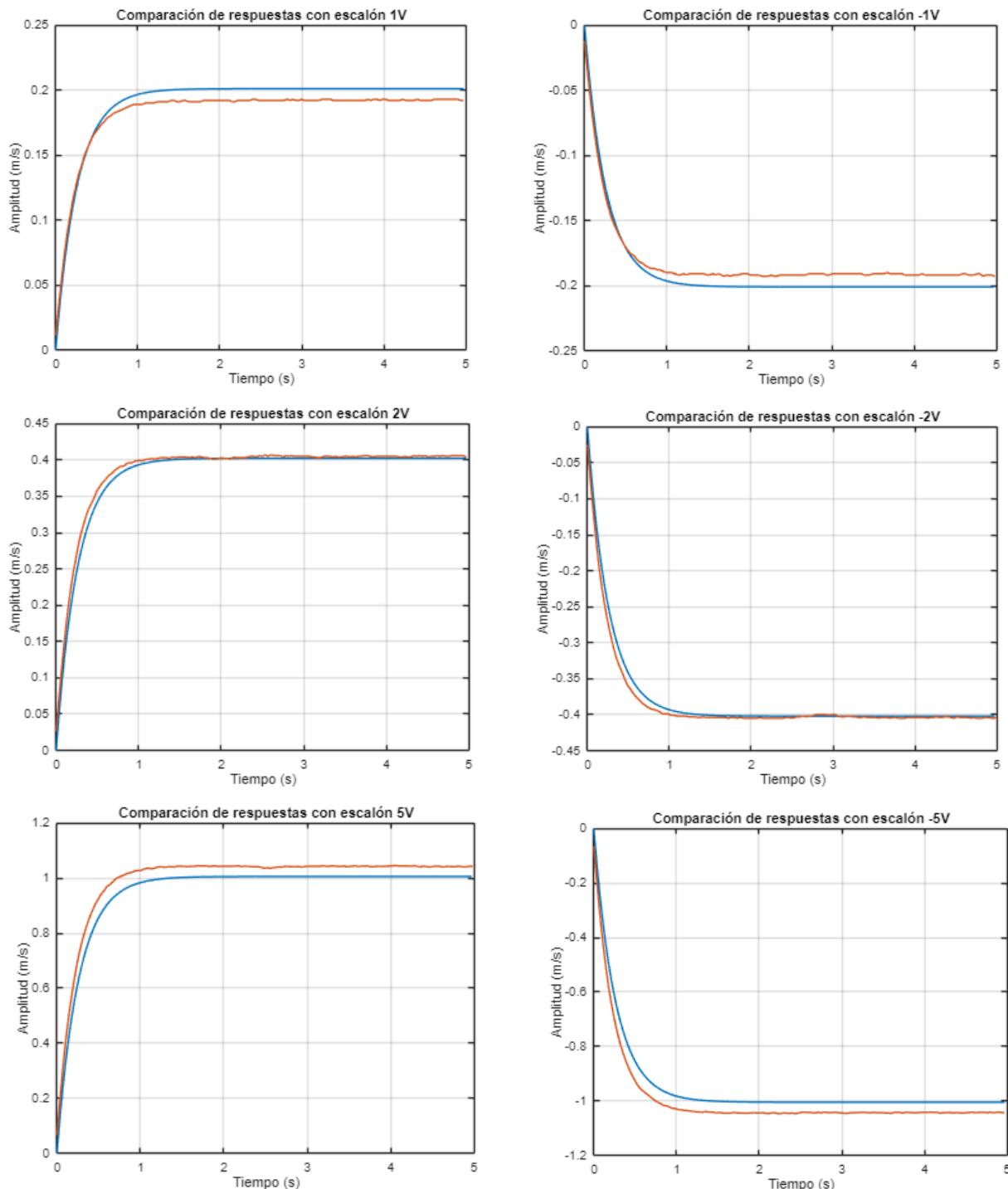


Figura 12. Comparación de la respuesta simulada de la planta y la respuesta real de la planta

Como se puede observar, las señales no coinciden exactamente dado que la K_m la hemos obtenido con la media de todas. Por esto mismo, para entradas de $\pm 1V$, el valor absoluto de la salida real es menor que el de la salida simulada, mientras que para $\pm 5V$ es mayor y para $\pm 2V$ prácticamente coinciden con lo esperado.

Ejercicio 3

Proponga e implemente los siguientes experimentos para caracterizar las alinealidades de la planta: 1) para modelar la saturación excite el SM con grandes tensiones y 2) para modelar la zona muerta excite el SM con pequeñas tensiones. Debe de obtener los valores adecuados con sus unidades de la saturación y zona muerta, explicando el procedimiento de obtención, y añadir al modelo de Simulink los bloques de las alinealidades caracterizadas convenientemente colocadas y con coherencia de unidades.

Para poder caracterizar correctamente el sistema, se han realizado distintas mediciones en el entorno GUI proporcionado en el laboratorio. En este entorno se han realizado distintas mediciones para caracterizar la zona muerta y la saturación.

- En lo referente a la **saturación**:

Se han realizado pruebas para valores positivos en la GUI hasta los $\pm 20V$ y se ha representado en Matlab los datos obtenidos de los ficheros generados para $\pm 10V$ y $\pm 20V$:

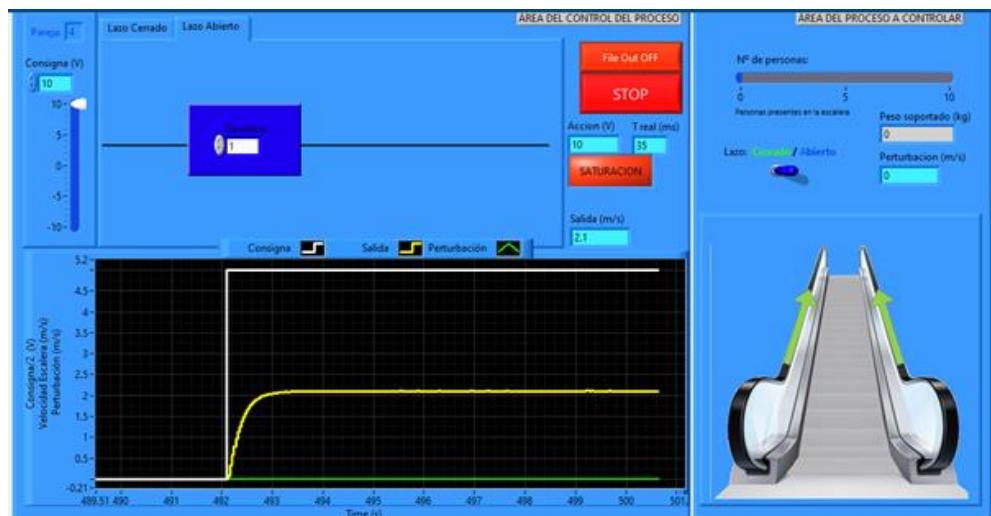


Figura 13. GUI Entrada $+10V$

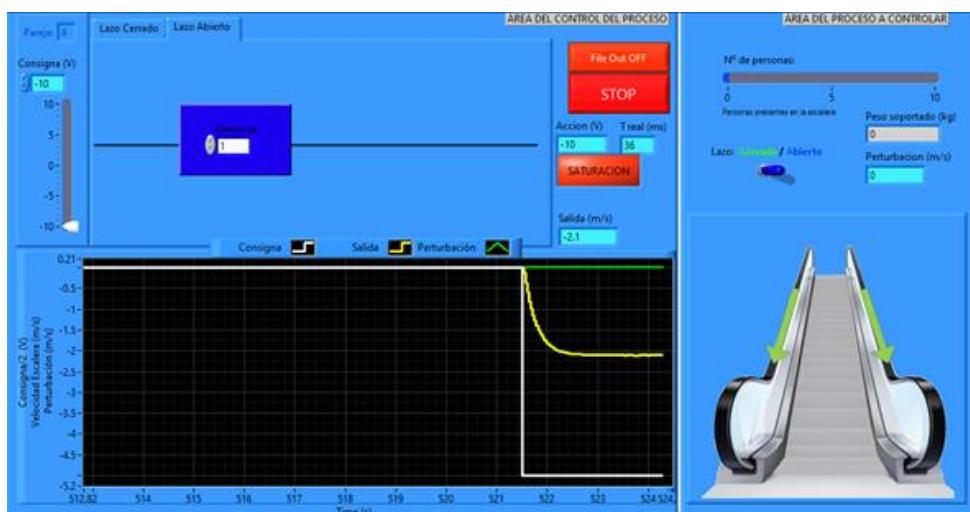


Figura 14. GUI Entrada $-10V$

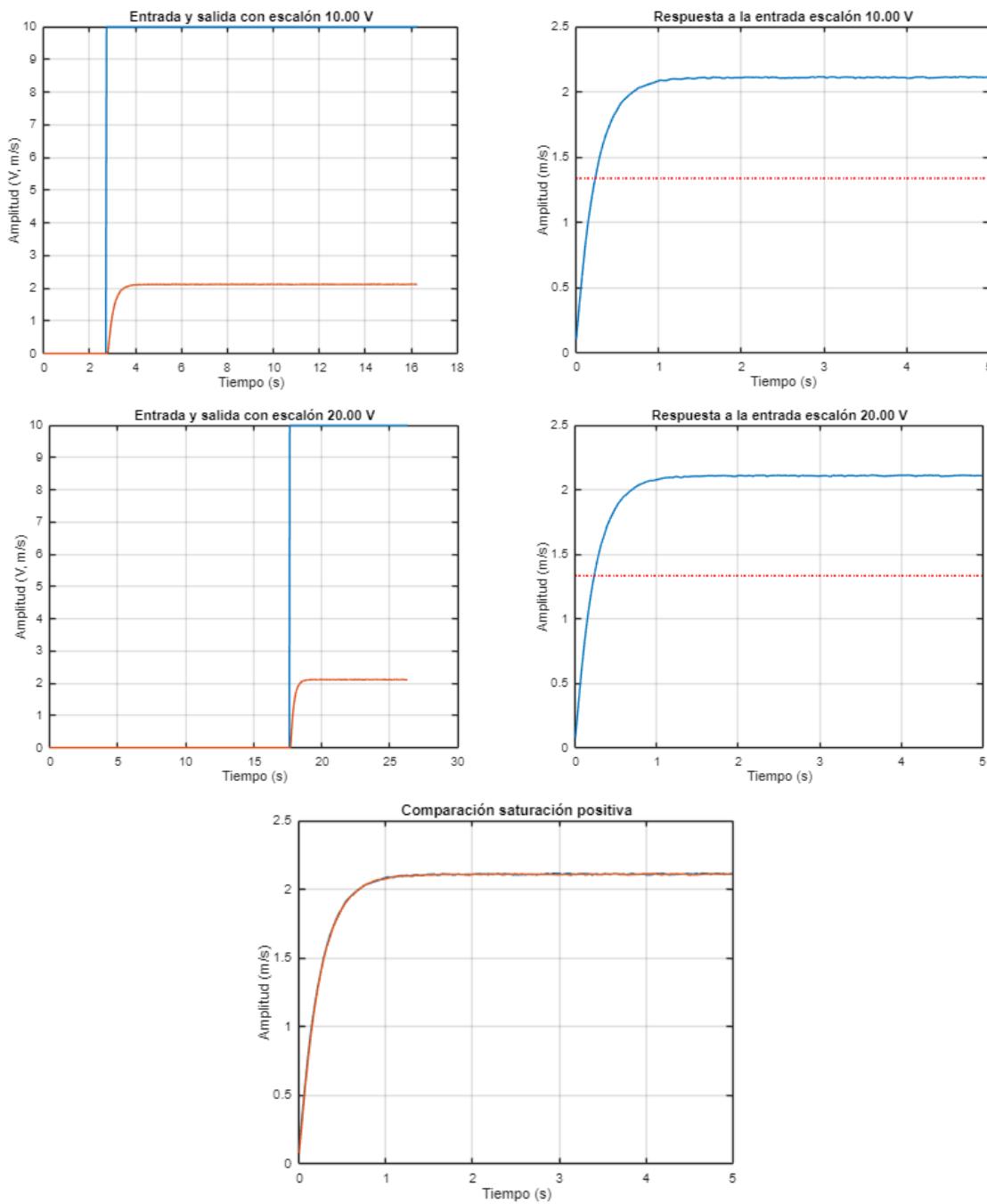


Figura 15. Saturación positiva

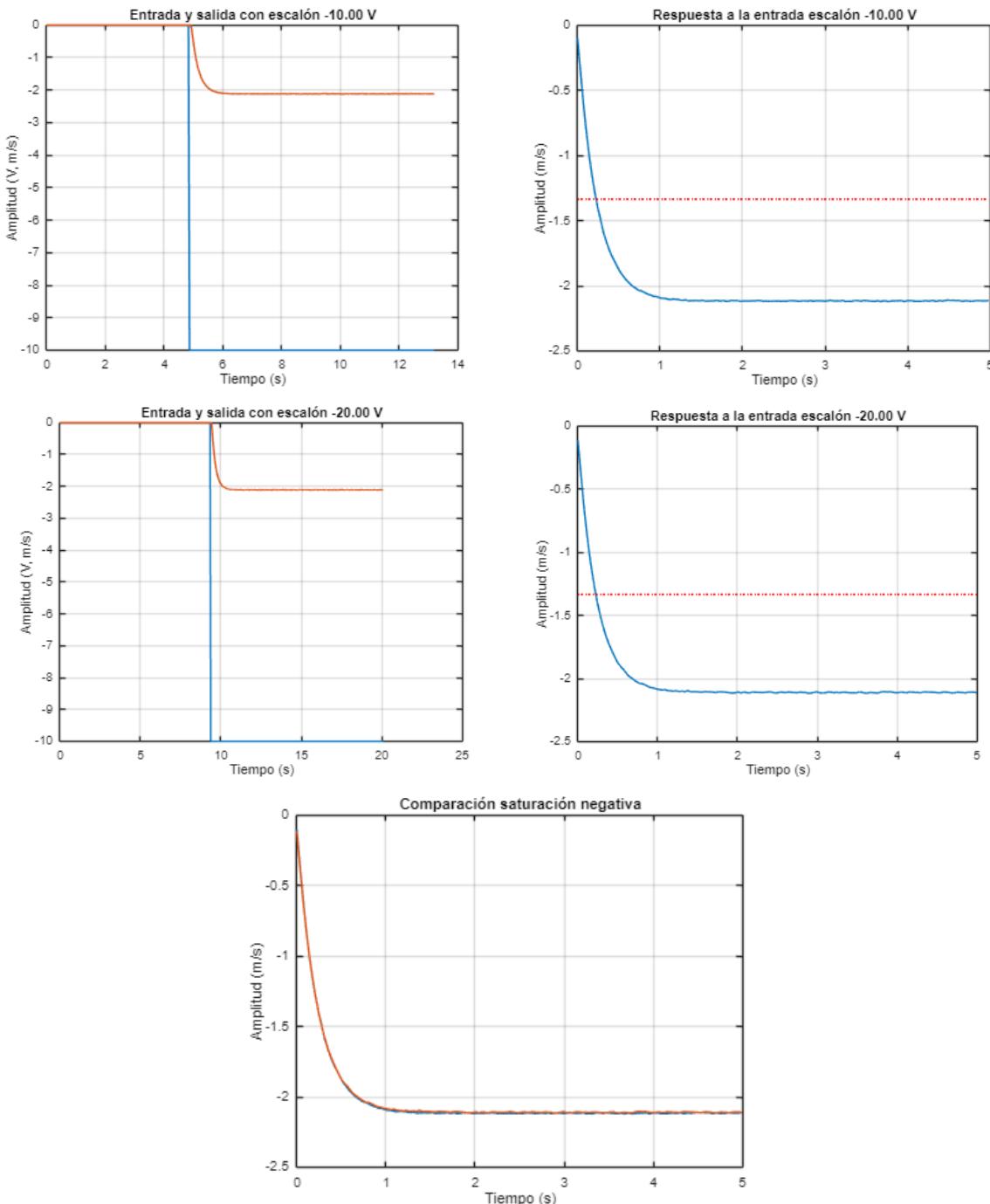


Figura 16. Saturación negativa

Como se puede observar para una entrada mayor que 10V, el valor final permanece siendo el mismo, de en torno a 2.1098 m/s. Pasa lo mismo para entradas menores de -10V (Valor final aproximado de -2.1123 m/s).

- En lo referente a la **zona muerta**:

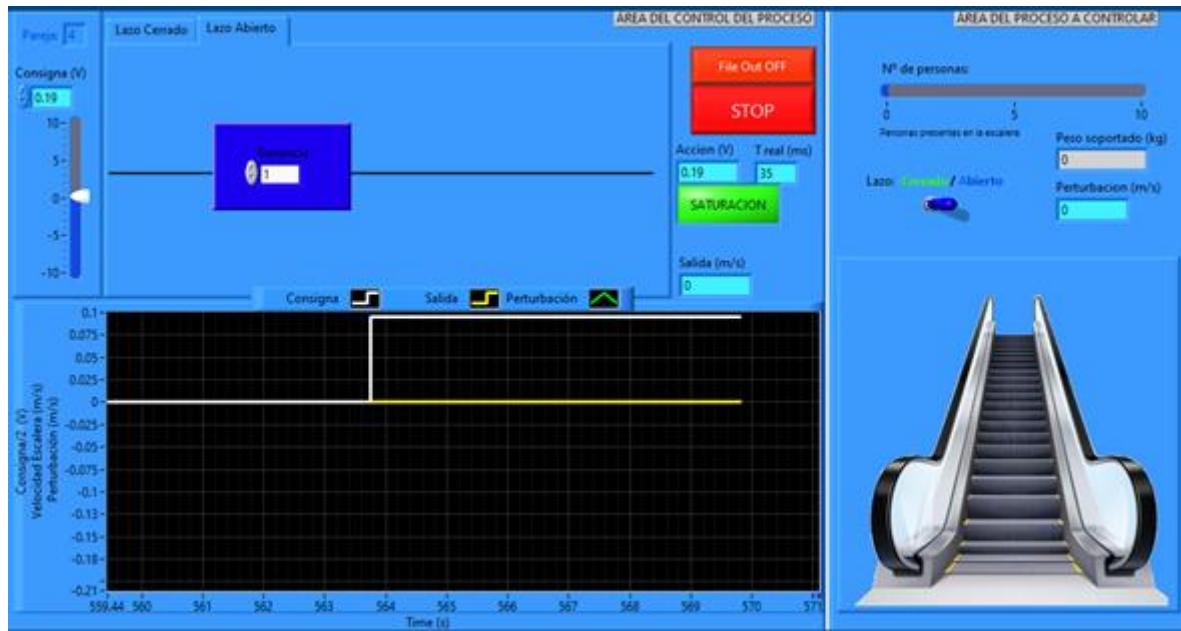


Figura 17. GUI Entrada 0.19V

Para la **zona positiva**, probando valores pequeños de entrada, se observa que:

- **Para 0.19 V:** no se obtiene salida.
- **Para 0.20 V:** se obtiene durante un par de segundos un valor de salida diferente a 0 m/s, pero después vuelve al 0.
- **Para 0.21 V:** se obtiene durante más segundos un valor de salida diferente a 0 m/s, pero después vuelve al 0.
- **Para 0.22 V:** se estabiliza la salida, por lo que, la **zona muerta positiva va de entradas escalón de 0 V a entradas escalón de 0.21 V.**

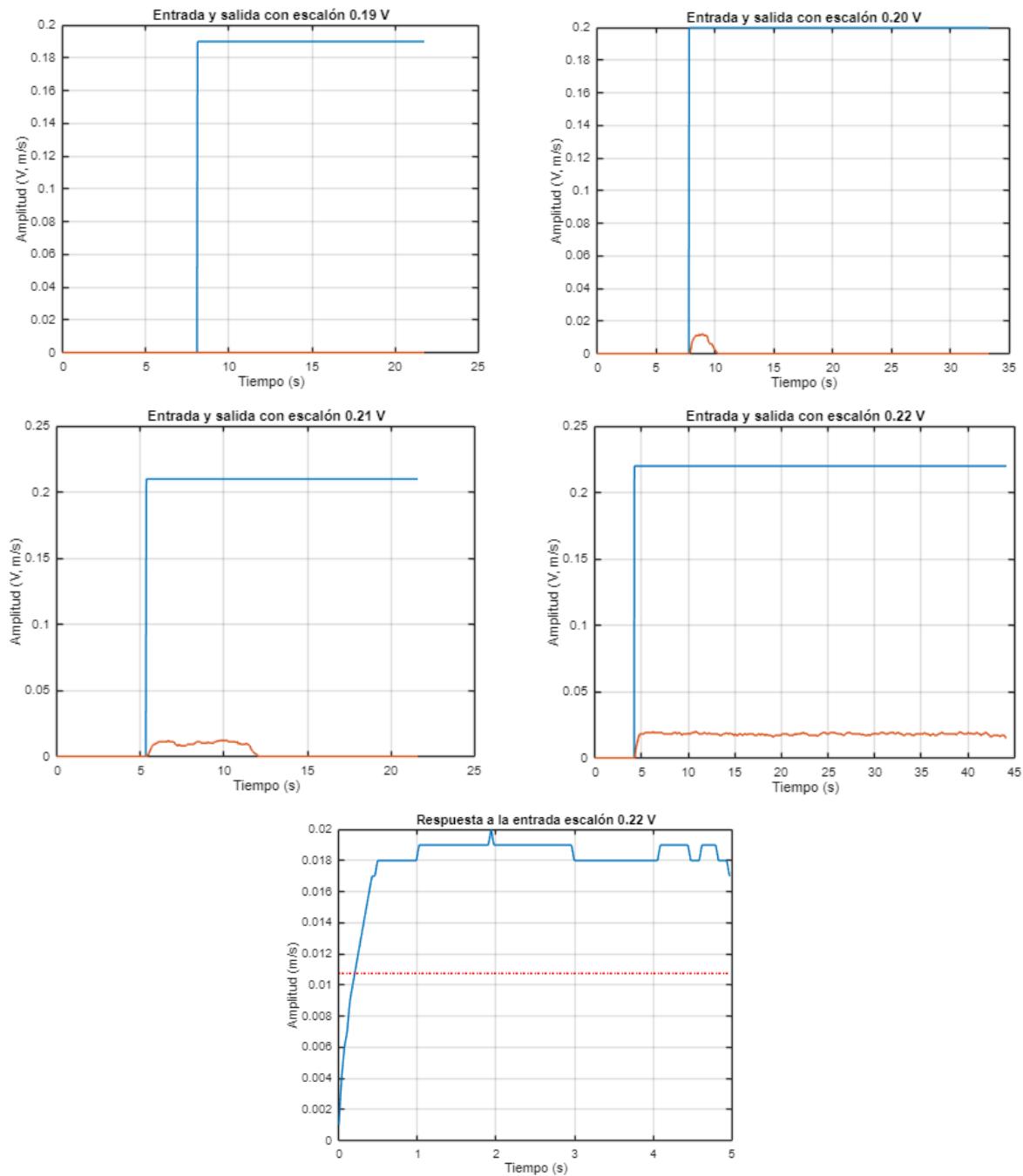


Figura 18. Zona muerta positiva

Para la **zona muerta negativa**, realizamos el mismo proceso:

- **Para -0.19 V:** no se obtiene salida.
- **Para -0.20 V:** se obtiene durante un par de segundos un valor de salida diferente a 0 m/s, pero después vuelve al 0.
- **Para -0.21 V:** se estabiliza la salida, por lo que, la **zona muerta negativa va de entradas escalón de 0 V a entradas escalón de -0.20 V**.

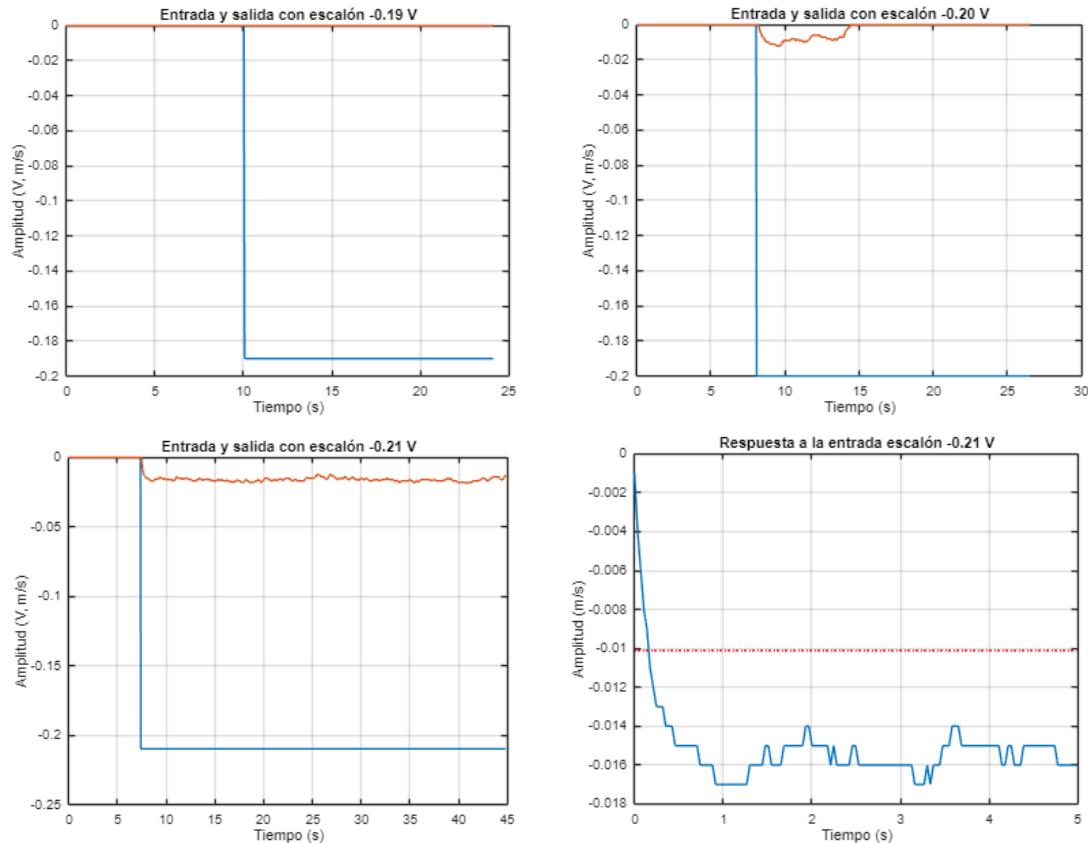


Figura 19. Zona muerta negativa

En conclusión, la zona muerta se produce cuando la entrada escalón se encuentra entre los valores -0.20V y 0.21V inclusives mientras que la saturación se produce cuando la señal de salida es mayor que 2.1098 m/s o menor que -2.1123 m/s.

En la siguiente figura se puede apreciar el modelado de la planta incluyendo la zona muerta y la saturación:

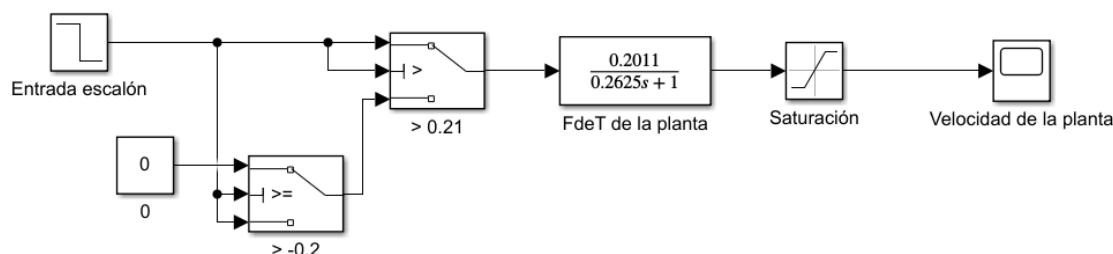


Figura 20. Modelo de la planta con Zona muerta y Saturación

Para entradas escalón de $\pm 0.2\text{V}$ (naranja), $\pm 1\text{V}$ (azul) y $\pm 20\text{V}$ (amarillo) al modelo de Simulink, se obtiene las salidas esperadas:

- **Para $\pm 0.2\text{V}$:** la salida debe valer 0 m/s ya que está en zona muerta.
- **Para $\pm 1\text{V}$:** la salida es la respuesta normal al escalón.
- **Para $\pm 20\text{V}$:** la salida está saturada a 2.1098 y -2.1123 m/s respectivamente.

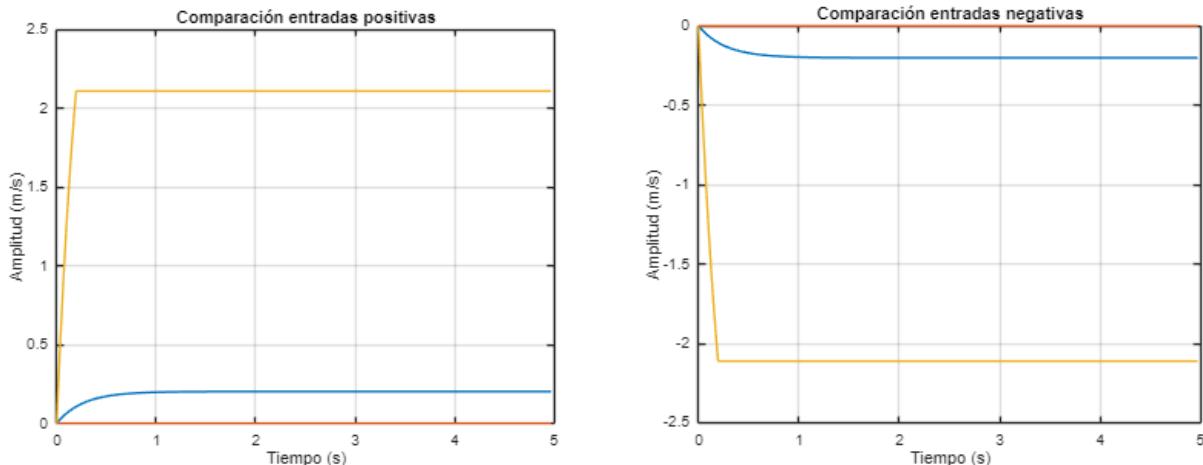


Figura 21. Salida del modelo de la planta con zona muerta y saturación

Ejercicio 4

Calcule el equivalente discreto de la planta y caracterice teóricamente su respuesta ante entrada escalón, tanto en régimen permanente como transitorio.

Para obtener el equivalente discreto de la planta partimos de la función de transferencia de la planta calculada anteriormente:

$$G(s) = \frac{0.2011}{0.2625s + 1} \left[\frac{m/s}{V} \right] \quad T_s = 35 [ms]$$

Aplicando la siguiente función en Matlab:

```
BoG = c2d(Gs, Ts, 'zoh');
```

Obtenemos el equivalente discreto de la planta:

$$B_o G(z) = \frac{K_{BG}}{z - p_1} = \frac{0.0251}{z - 0.8752} \left[\frac{m/s}{V} \right]$$

Para entrada escalón de 1V, obtenemos esta respuesta donde tenemos la **salida real** (Azul), la **salida simulada** (Amarillo) y la **salida del modelo discreto** (Naranja).

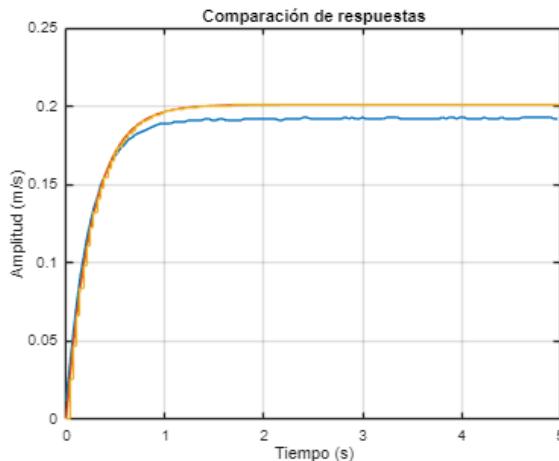


Figura 22. Salida real, simulada y discreta a entrada escalón de 1V

Para el régimen permanente:

Al ser la planta de tipo 0:

$$k_o[BoG] = \lim_{z \rightarrow 1} \frac{0.0251}{z - 0.8752} = 0.20112$$

Para el régimen transitorio:

Tenemos un polo real simple, por lo tanto, el sistema es sobreamortiguado:

$$p = 0.8752$$

$$k_s = \left\lceil \frac{\ln(0.05)}{\ln(|p|)} \right\rceil = \left\lceil \frac{-3}{\ln(|0.8752|)} \right\rceil = [22.505] = 23$$

$$t_s = k_s * T_s = 23 * 0.035 = 0.805 \text{ s}$$

Ejercicio 5

Caracterice la respuesta de la planta en régimen permanente para consignas de ± 1 V con perturbación: cuando el mayor número de personas posible está presente en la escalera. Justifique los resultados con ayuda de Simulink.

En la interfaz GUI hemos realizado unas pruebas para consignas +1 y -1V cuando el número de personas es el máximo posible (10 personas):

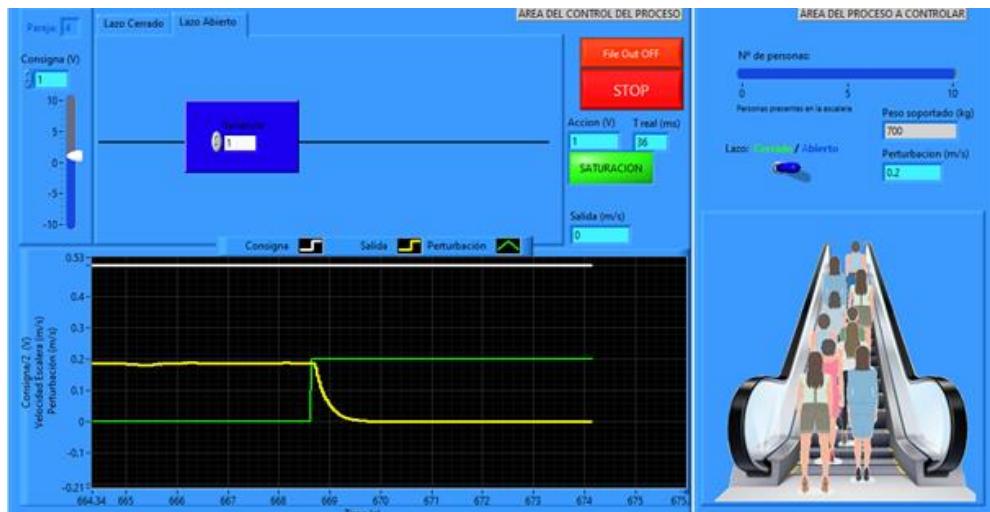


Figura 23 GUI Entrada 1V y 10 personas

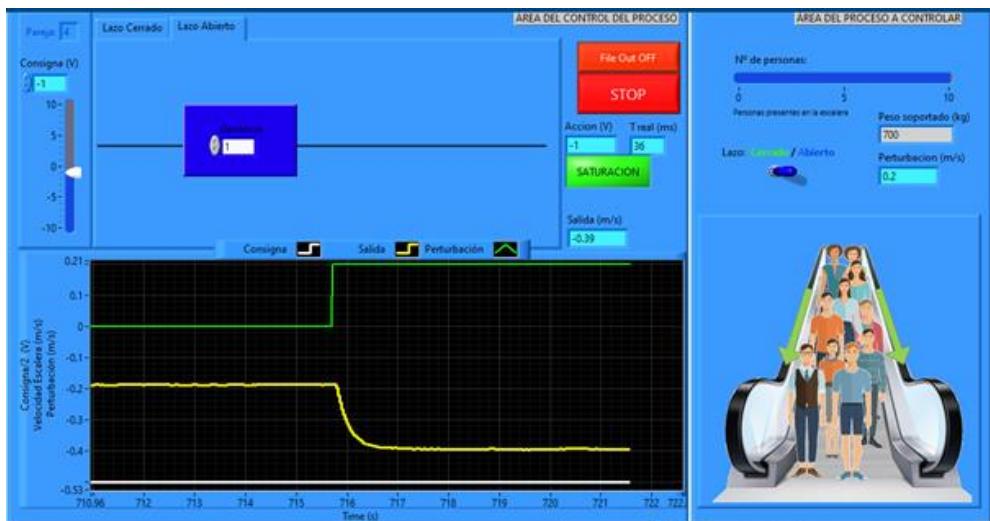


Figura 24 GUI Entrada -1V y 10 personas

En las siguientes capturas, podemos observar como la salida se mantiene normal hasta que se activa la perturbación:

- En el caso de la entrada escalón 1V, se puede ver que cuando se activa la perturbación, como esta vale 0.2V, se resta y la salida es 0V.
- En el caso de la entrada escalón -1V, se puede observar que cuando se activa la perturbación, esta se resta, por lo que la salida vale -0.4V.

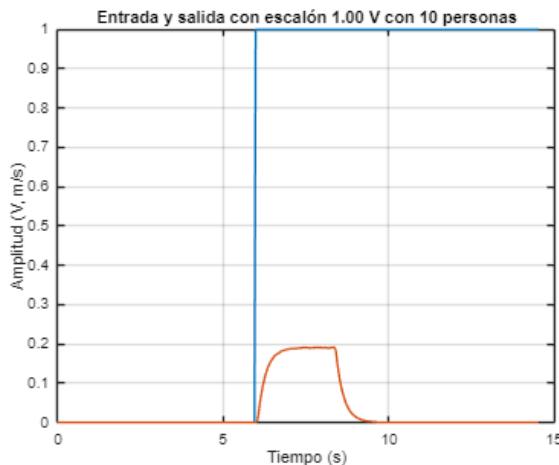


Figura 25. Perturbación máxima con entrada escalón +1V

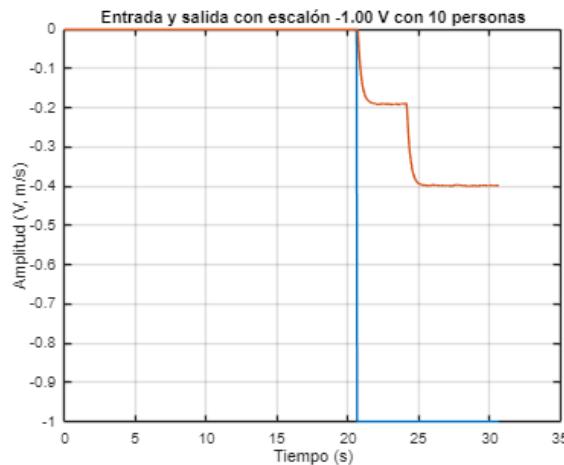


Figura 26. Perturbación máxima con entrada escalón -1V

Estos valores tienen sentido dado que cuando las escaleras suben (Entrada de tensión positiva), las personas producen una perturbación sustractiva y la escalera necesita más tensión para subir, mientras que cuando las escaleras bajan (Tensión negativa), el peso de las personas aumenta la velocidad de bajada.

En Simulink, se ha realizado un modelado de la planta con la perturbación de la siguiente manera:

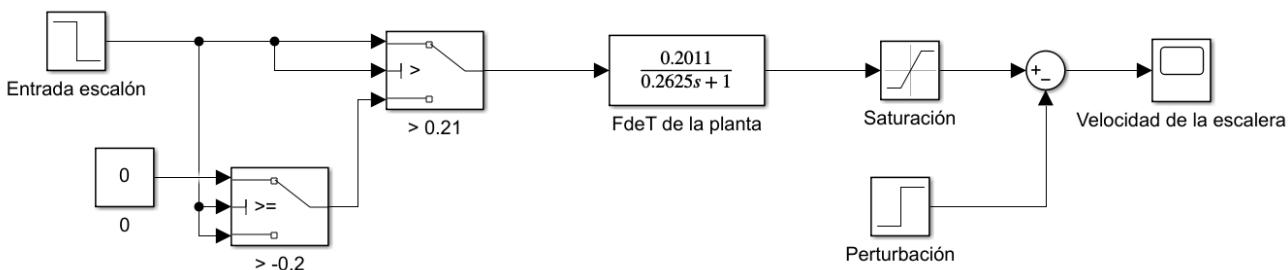


Figura 27. Modelo en Simulink de la planta con perturbación

Al ejecutar la simulación obtenemos las siguientes respuestas para perturbaciones con valor 0 m/s (azul), 0.1 m/s (naranja) y 0.2 m/s (amarillo):

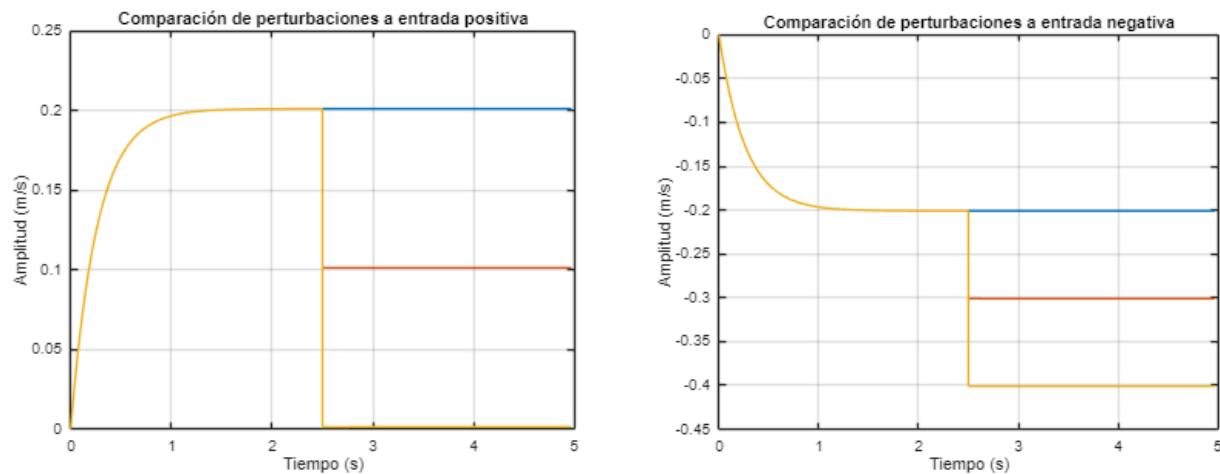


Figura 28. Salida del modelo de Simulink para las perturbaciones a entrada escalón $\pm 1V$

Para perturbaciones con valor 0 m/s (es decir ninguna persona), 0.1 m/s (5 personas) y 0.2 m/s (10 personas), se observa que para entradas escalón de $\pm 1V$, las salidas del modelo de Simulink son las deseadas.

Fase 2: Diseño directo del regulador

Se plantea en este punto la utilización de un regulador diseñado por el **método directo de Truxal** en el sistema realimentado para que la velocidad de la escalera adquiera su **régimen nominal de desplazamiento en 980 ms**, permitiendo un **error en régimen permanente del 3%** ante escalón.

Ejercicio 6

Calcule la FdeT del regulador con sus correspondientes unidades, mediante la aplicación detallada del método directo, el resultado debe presentarlo de la forma factorizada, polinomial y con potencias negativas en z. Presente la ecuación en diferencias del regulador.

Recordamos que tenemos los siguientes parámetros:

- **Del enunciado de la fase:**

$$t_s = 980 \text{ ms} = 0.98 \text{ s}$$

$$e_{rpp} = 3\% = 0.03$$

- **Del apartado anterior:**

$$B_o G(z) = \frac{0.0251}{z - 0.8752} \left[\frac{m/s}{V} \right]$$

Ahora podemos comenzar con el desarrollo del método directo de Truxal:

1. **Comprobar si es aplicable (Estabilidad y entrada escalón):**

- Todos los factores del EQD están dentro de la circunferencia unitaria:

$$p = 0.8752$$

El polo se encuentra dentro de la circunferencia unitaria y el modelo es estable, por lo que cumple con esta condición.

- Tiene entrada escalón:

Si, es una de las partes que se indican en la especificación del sistema.

El modelo cumple con las 2 condiciones, por lo tanto es **aplicable**.

2. **Planteamiento del modelo, aplicación de la especificación dinámica:**

$$M(z) = \frac{k_m}{z - p} \left[\frac{m/s}{m/s} \right]$$

$$k_s = \left[\frac{t_s}{T_s} \right] = \left[\frac{0.98}{0.035} \right] = 28$$

$$|p| = e^{-\frac{3}{k_s}} = e^{-\frac{3}{28}} = 0.8984$$

$$M(z) = \frac{k_m}{z - 0.8984}$$

3. Verificación de la causalidad:

$$gD_{BG} - gN_{BG} \leq gD_M - gN_M$$

$$1 - 0 \leq 1 - 0$$

Se cumple, por lo tanto, el regulador es **causal**.

4. Aplicación de la especificación estática: errp = 3%.

$$M(z) = R - e_{rpp} = \lim_{z \rightarrow 1} \frac{k_m}{z - 0.8984} = 1 - 0.03 = \frac{k_m}{1 - 0.8984} \rightarrow k_m = 0.0986 \left[\frac{m/s}{m/s} \right]$$

5. Modelo final:

$$M(z) = \frac{0.0986}{z - 0.8984} \left[\frac{m/s}{m/s} \right]$$

$$D_{BG} = z - 0.8752; N_{BG} = 0.0251$$

$$D_M = z - 0.8984; N_M = 0.0986$$

$$F(z) = \frac{N_F}{D_f} = \frac{M}{BG * (1 - M)} = \frac{N_M * D_{BG}}{N_{BG} * (D_M - N_M)} = \frac{0.0986z - 0.0863}{0.0251z - 0.0250} \left[\frac{V}{m/s} \right]$$

$$F(z) = \frac{A(z)}{E(z)} = \frac{3.9283 - 3.4382z^{-1}}{1 - 0.996z^{-1}} \left[\frac{V}{m/s} \right]$$

$$TF^{-1}\{A(z) * F_D(z)\} = TF^{-1}\{E(z) * F_N(z)\}$$

$$a[k] = 3.9283e[k] - 3.4382e[k-1] + 0.996a[k-1]$$

Ejercicio 7

Dibuje una topología (diagrama de bloques) del sistema de acuerdo con las unidades elegidas, que deberán aparecer indicadas en cada señal o secuencia del dibujo.

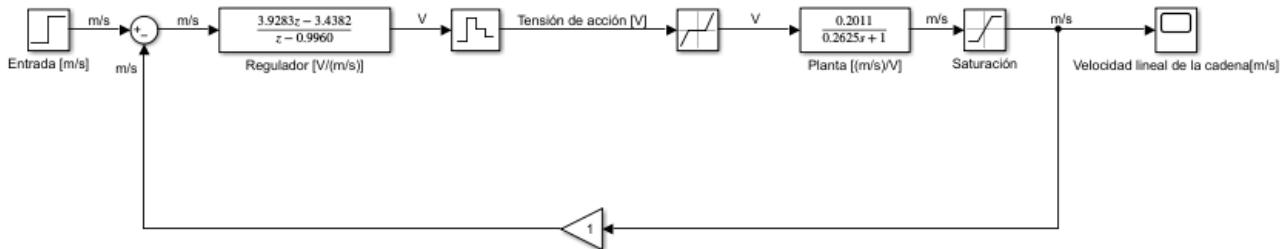


Figura 29. Diagrama de bloques del sistema (Truxal)

Ejercicio 8

Calcule la FdT del sistema y caracterice teóricamente su salida ante entradas escalón de $\pm 1 \text{ m/s}$.

Como se ha calculado en apartados anteriores, partimos de las siguientes ecuaciones:

$$F(z) = \frac{3.928z - 3.438}{z - 0.997} \left[\frac{V}{m/s} \right]$$

$$B_o G(z) = \frac{0.0251}{z - 0.8752} \left[\frac{m/s}{V} \right]$$

Por lo que, la FdT del sistema resulta ser la siguiente:

$$M(z) = \frac{C(z)}{R(z)} = \frac{CD(z)}{1 + CA(z)} = \frac{F(z)B_o G(z)}{1 + H(z)F(z)B_o G(z)} = \frac{0.0986}{z - 0.8984} \left[\frac{m/s}{m/s} \right]$$

Ya obtenida la FdT del sistema, podemos caracterizar de manera teórica la salida ante entradas escalón:

$$c(\infty) = R * \lim_{z \rightarrow 1} M(z) \left\{ \begin{array}{l} R = 1 \text{ m/s} \rightarrow c(\infty) = 1 * 0.9705 = 0.9705 \text{ m/s} \\ R = -1 \text{ m/s} \rightarrow c(\infty) = -1 * 0.9705 = -0.9705 \text{ m/s} \end{array} \right.$$

Ejercicio 9

Calcule en régimen permanente tanto la acción de control como la velocidad de la escalera subiendo y bajando en diferentes condiciones de carga de la escalera, para consignas de ± 1 m/s. Relacione los incrementos o decrementos de acción de control con el peso de las personas presentes en la escalera.

Para la realización de este apartado, se ha creado una función escalera en Matlab que realiza las siguientes operaciones:

$$k_0[M(z)_{\text{entrada}}] = R * \lim_{z \rightarrow 1} M(z)$$

$$k_0[M(z)_{\text{pert}}] = P * \lim_{z \rightarrow 1} \frac{1}{1 + F(z) * B_o G(z)}$$

$$k_0[B_o G(z)] = \lim_{z \rightarrow 1} B_o G(z)$$

$$c(\infty) = k_0[M(z)_{\text{entrada}}] + k_0[M(z)_{\text{pert}}]$$

$$a(\infty) = \frac{c(\infty) + P}{k_0[B_o G(z)]}$$

Entrada escalón [m/s]	Perturbación [m/s]	Acción de control [V]	Velocidad de la escalera [m/s]
1	0	4.8268	0.9705
-1	0	-4.8268	-0.9705
1	0.1	5.3095	0.9675
-1	0.1	-4.3441	-0.9734
1	0.2	5.7922	0.9646
-1	0.2	-3.8615	-0.9764

Tabla 2. Velocidad de la escalera y acción de control a entrada escalón ± 1 m/s con diferentes perturbaciones (Truxal)

Como se puede observar, cuantas más personas se suben a la escalera, mayor es el peso que ejercen:

- **En el caso de la subida:** la señal de acción aumenta dado que se necesita más energía para mover la escalera y que se mantenga constante la velocidad.
- **En el caso de la bajada:** cuando mayor es el peso, más facilidad tiene la escalera para bajar, por lo que la señal de control disminuye (en módulo) para que pueda mantenerse constante la velocidad.

Además, en este caso se especificó un error del 3% en régimen permanente, error que se puede observar en la velocidad de la escalera, ya que esta no llega a ser igual que la entrada escalón, sino que tiene un error de aproximadamente 3% en todos los casos.

Fase 3: Validación del diseño directo

En lazo cerrado, justifique que el regulador diseñado cumple con lo esperado, realizando las siguientes tareas:

Ejercicio 10

Simule en Simulink el sistema de control con y sin alinealidades. Mida las características de las respuestas (tanto el régimen permanente como el transitorio) ante entradas escalón de $\pm 1 \text{ m/s}$. Analice las diferencias de las medidas respecto de los cálculos teóricos.

En la herramienta de Simulink, se han realizado los siguientes modelos para validar el regulador desarrollado en la fase anterior:

- **Modelo sin alinealidades:**

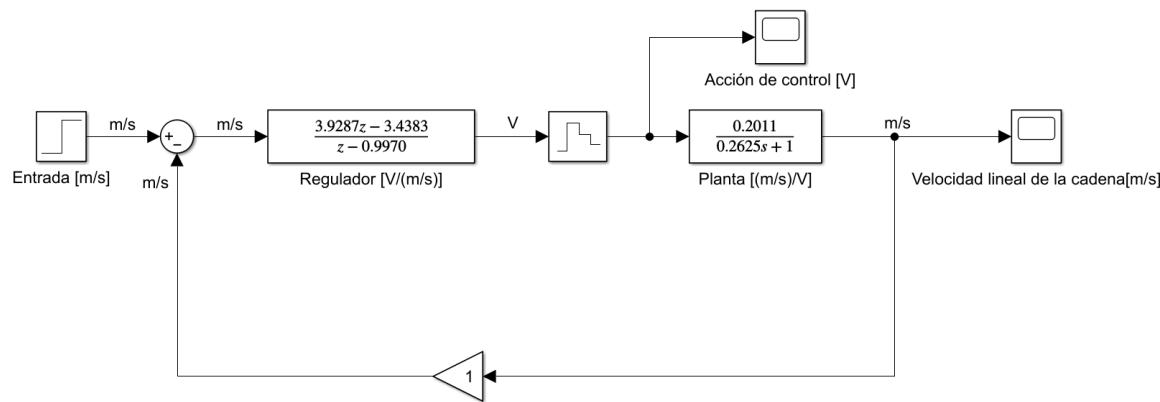


Figura 30. Modelo sistema con regulador Truxal sin alinealidades

- **Modelo con alinealidades:**

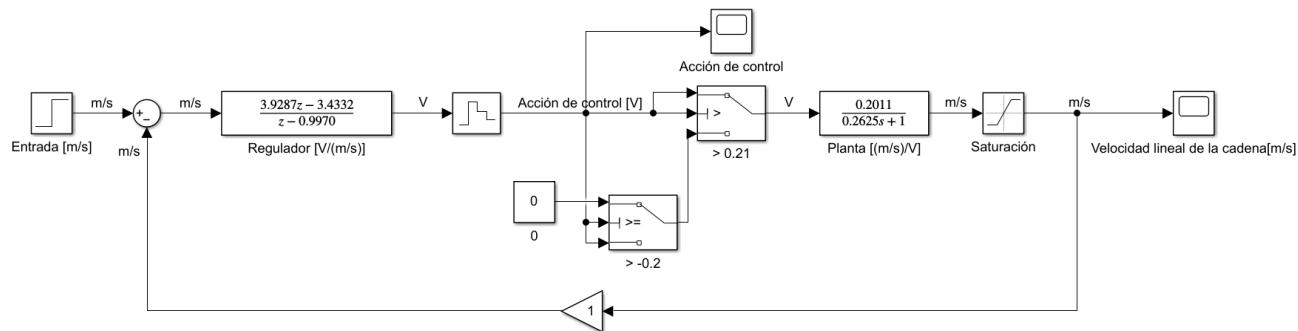


Figura 31. Modelo sistema con regulador Truxal con alinealidades

Simulando las mismas señales que en el primer ejercicio obtenemos las siguientes respuestas, de la entrada (azul) y la salida (naranja):

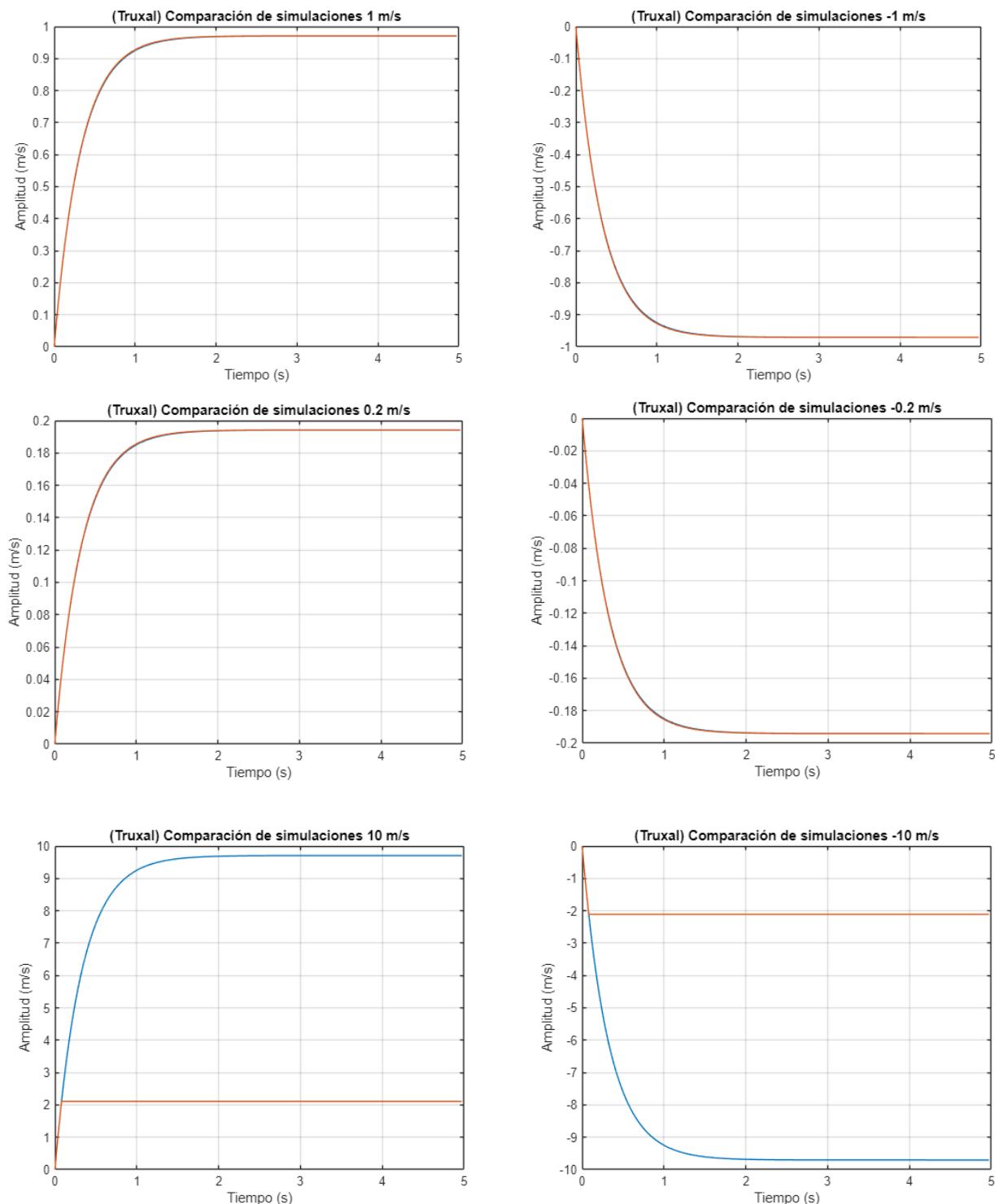


Figura 32. Comparación de simulaciones Truxal

Podemos observar que:

- **Para $\pm 1V$:** las dos señales coinciden.
- **Para $\pm 10V$:** las dos señales no coinciden. Esto se debe a que la saturación es respecto a la salida, por lo que sigue ocurriendo cuando la salida es mayor que 2.1098 m/s o menor que -2.1123 m/s .

- Para $\pm 0.2V$: se observa que las dos salidas coinciden. Esto se debe a que la zona muerta ha cambiado dado que la acción de control ya no coincide con la señal de entrada.

En la siguiente tabla se puede observar la comparación de las características de las respuestas (tanto en régimen permanente como en transitorio) de los cálculos teóricos y las simulaciones sin y con alinealidades para una entrada escalón de $\pm 1 \text{ m/s}$:

Entrada	Tiempo de establecimiento [s]	Salida (∞) [m/s]	Acción de control [V]
Teórica $+1 \text{ m/s}$	0.980	0.9705	4.8268
$+1 \text{ m/s}$ sin alinealidades	0.979	0.9705	4.8260
$+1 \text{ m/s}$ con alinealidades	0.979	0.9705	4.8260
Teórica -1 m/s	0.980	-0.9705	-4.8268
-1 m/s sin alinealidades	0.979	-0.9705	-4.8260
-1 m/s con alinealidades	0.979	-0.9705	-4.8260

Tabla 3. Características Truxal

Se observa que, para las simulaciones, los valores no cambian dado que para entrada $\pm 1 \text{ m/s}$ no afectan las alinealidades a la respuesta.

Ejercicio 11

Mediante el GUI, obtenga la captura de la respuesta del sistema para una consigna de 1 m/s (subiendo) y sin personas transportadas (sin carga). Adjunte una captura de pantalla de la ventana de gestión del sistema de control en la que se aprecie con la suficiente nitidez el código fuente del regulador. Mida las características de la respuesta y coteje los resultados con los de la simulación del ejercicio anterior, así como con los cálculos teóricos realizados en la fase de diseño del regulador.

Mediante la interfaz GUI, se ha obtenido la siguiente respuesta a la entrada de 1 m/s con fichero que hemos introducido (fichero disponible en el anexo del documento):

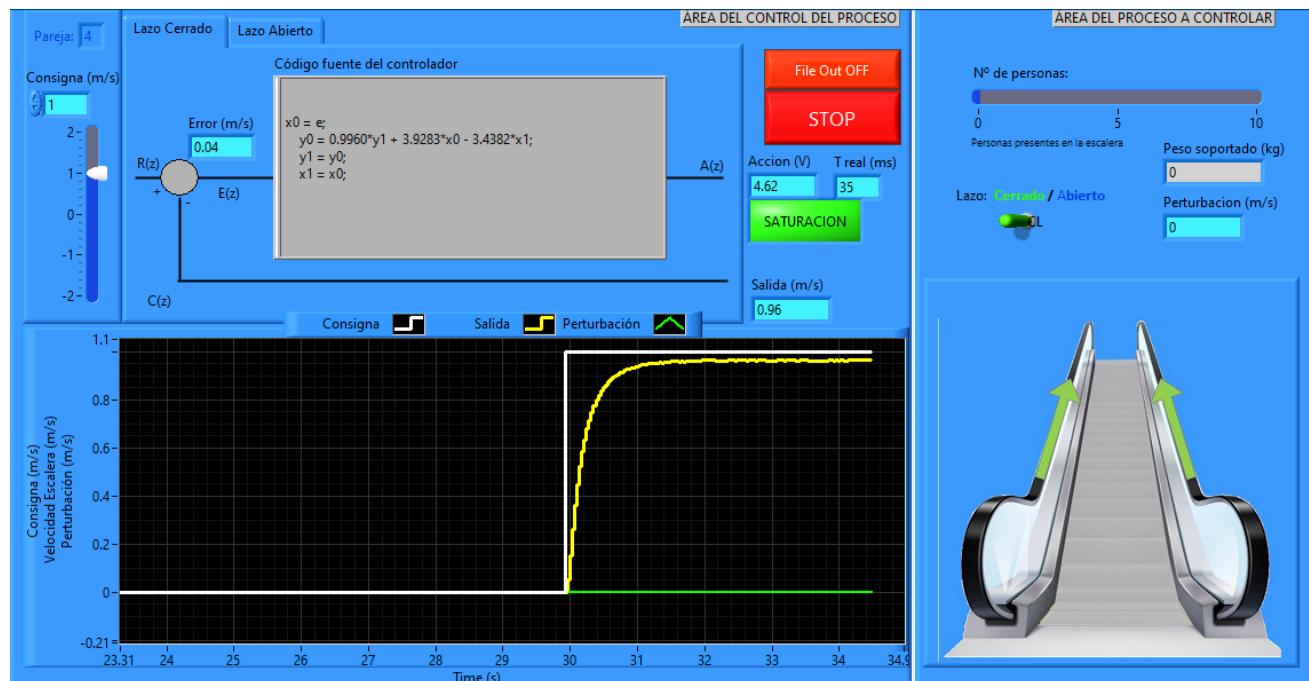


Figura 33. GUI Truxal 1 m/s

Guardando los ficheros con el contenido de esta captura, lo vamos a continuar mediante Matlab para mayor detalle:

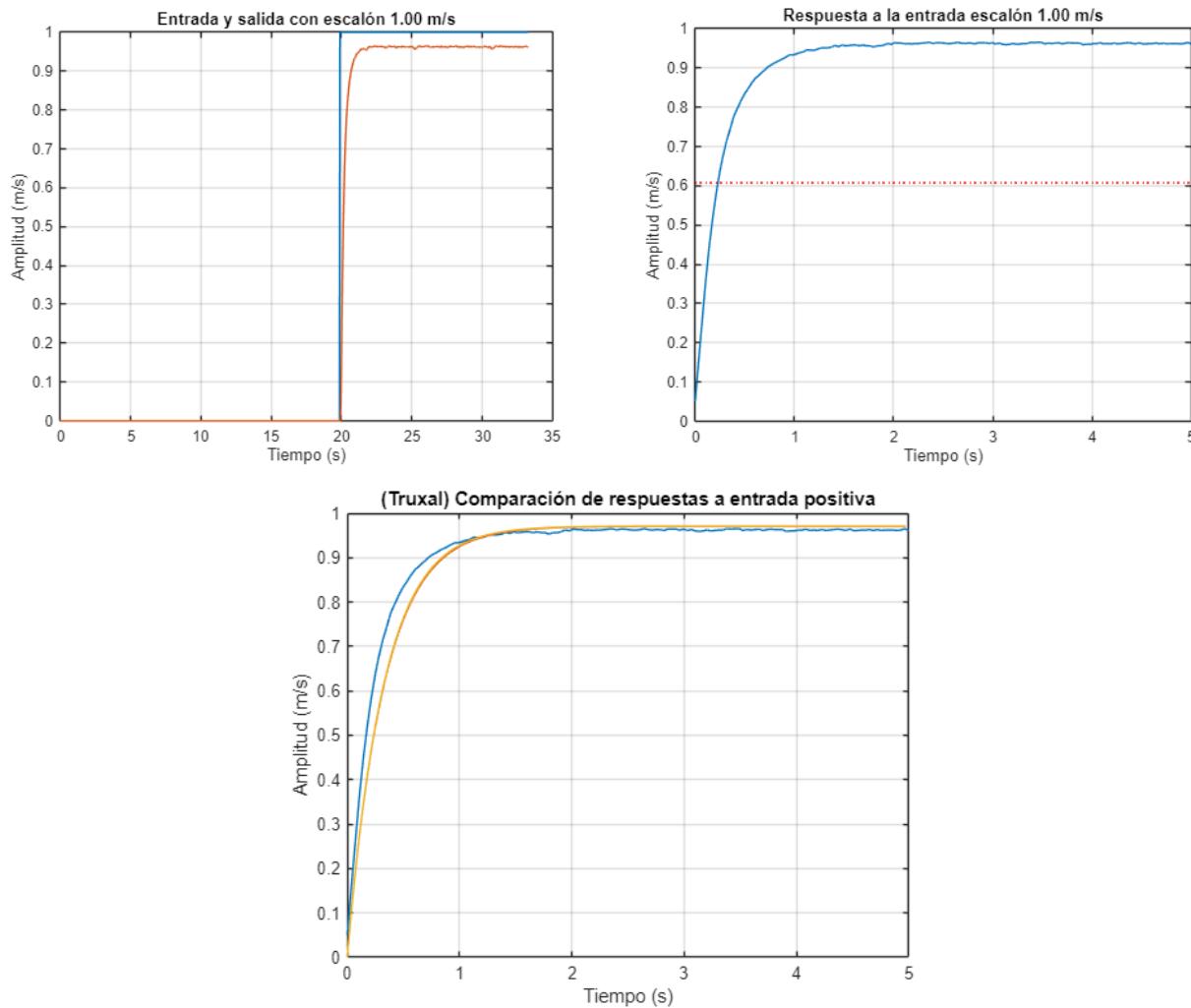


Figura 34. Comparación respuestas +1m/s Truxal

Las diferencias se deben a la aproximación de los decimales en los cálculos realizados. A la hora de completar la plantilla del controlador utilizamos el regulador desarrollado a papel con ayuda de la calculadora, es decir, que los decimales no eran exactos debido a aproximaciones.

Por esto mismo, al utilizar funciones de Matlab para los cálculos finales teóricos y las simulaciones, se han utilizado todos los decimales y el regulador difiere en pocas milésimas del calculado a mano. Aun así, esto es suficiente para que las características de las respuestas no coincidan.

Entrada +1m/s	Tiempo de establecimiento [s]	Salida (∞) [m/s]	Acción de control [V]
Teórica	0.980	0.9705	4.8268
Simulada sin alinealidades	0.979	0.9705	4.826
Simulada con alinealidades	0.979	0.9705	4.826
Real	0.847	0.965	4.62

Tabla 4. Comparación Truxal +1 m/s

Ejercicio 12

Igual que el ejercicio 11 pero con consigna de -1 m/s (bajando).

Mediante la interfaz GUI, se ha obtenido la siguiente respuesta a la entrada de 1 m/s con fichero que hemos introducido (fichero disponible en el anexo del documento):

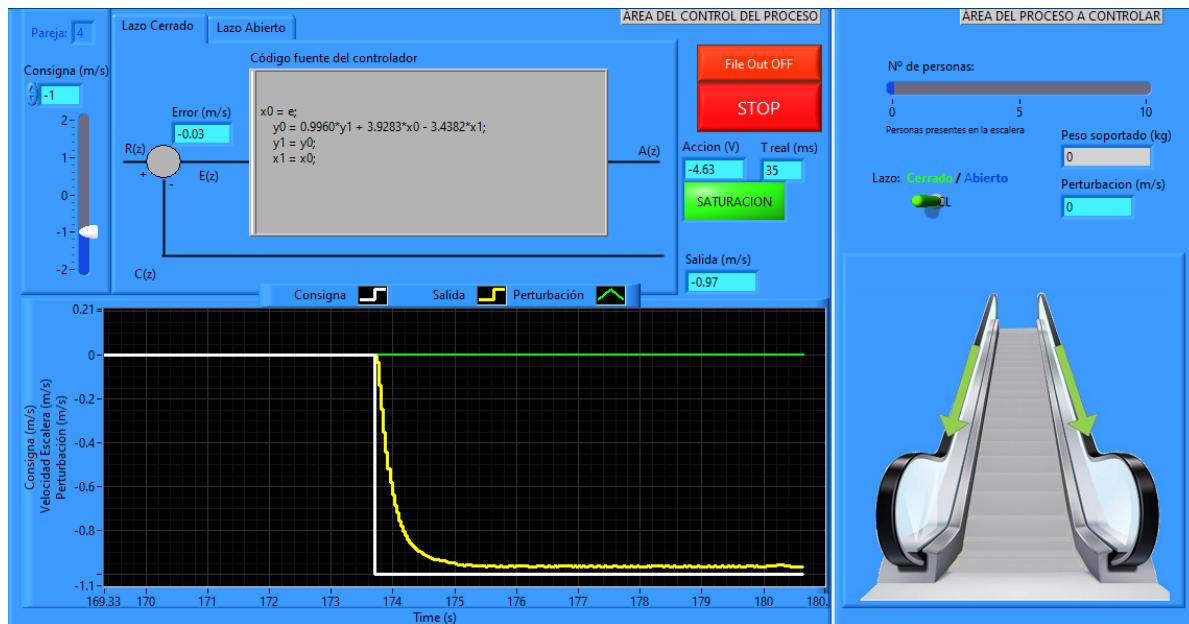


Figura 35. GUI Truxal -1 m/s

Guardando los ficheros con el contenido de esta captura, lo vamos a continuar mediante Matlab para mayor detalle:

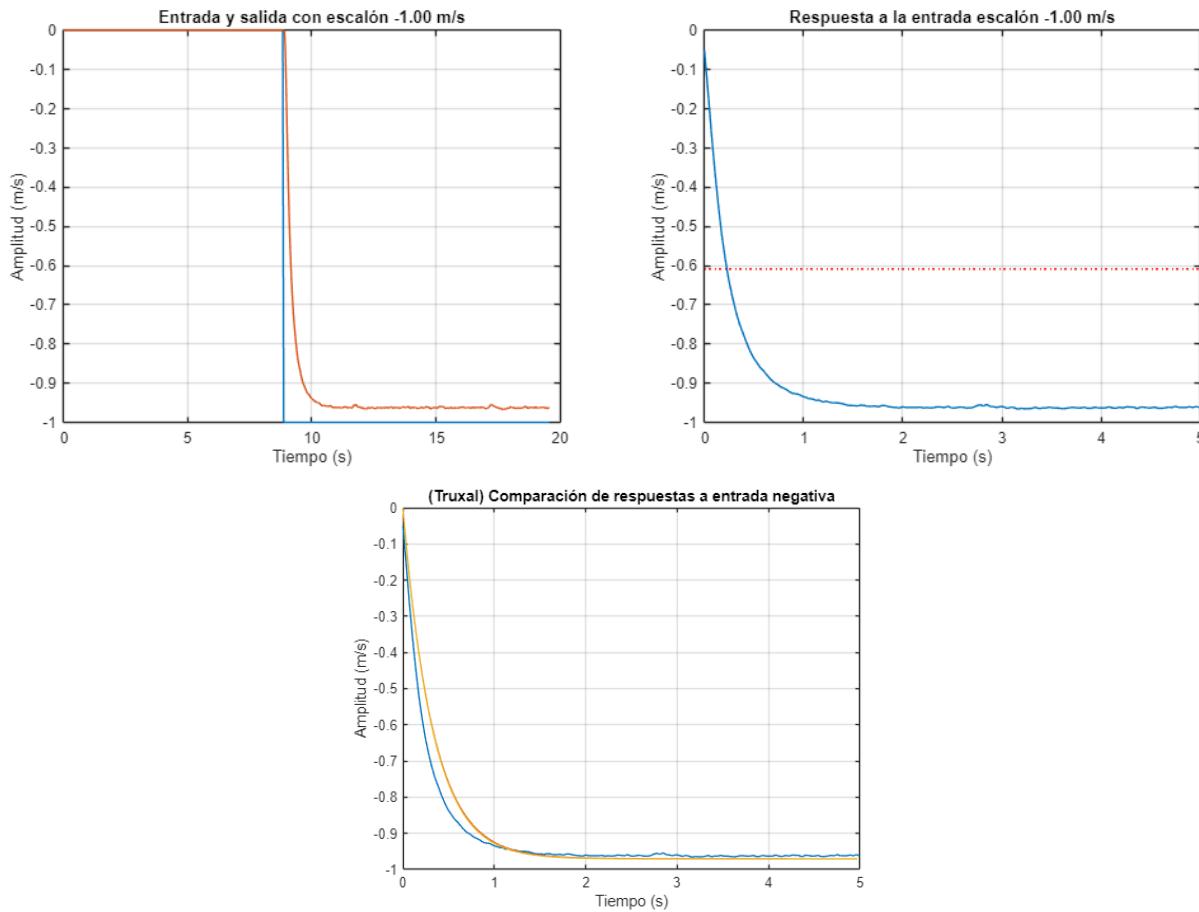


Figura 36. Comparación respuestas -1m/s Truxal

Las diferencias se deben a la aproximación de los decimales en los cálculos realizados. A la hora de completar la plantilla del controlador utilizamos el regulador desarrollado a papel con ayuda de la calculadora, es decir, que los decimales no eran exactos debido a aproximaciones.

Por esto mismo, al utilizar funciones de Matlab para los cálculos finales teóricos y las simulaciones, se han utilizado todos los decimales y el regulador difiere en pocas milésimas del calculado a mano. Aun así, esto es suficiente para que las características de las respuestas no coincidan.

Entrada -1m/s	Tiempo de establecimiento [s]	Salida (∞) [m/s]	Acción de control [V]
Teórica	0.980	-0.9705	-4.8268
Simulada sin alinealidades	0.979	-0.9705	-4.826
Simulada con alinealidades	0.979	-0.9705	-4.826
Real	0.808	-0.9610	-4.63

Tabla 5. Comparación Truxal -1 m/s

Ejercicio 13

Mida en régimen permanente tanto la acción de control como la velocidad de la escalera, subiendo y bajando en diferentes condiciones de carga de la escalera, para consignas de $\pm 1 \text{ m/s}$. Relacione los incrementos o decrementos de acción de control con el peso de las personas presentes en la escalera.

Mediante la interfaz GUI, se ha obtenido la siguiente respuesta a la entrada de 1 m/s y -1 m/s con fichero que hemos introducido (fichero disponible en el anexo del documento) y se han aplicado distintas perturbaciones:

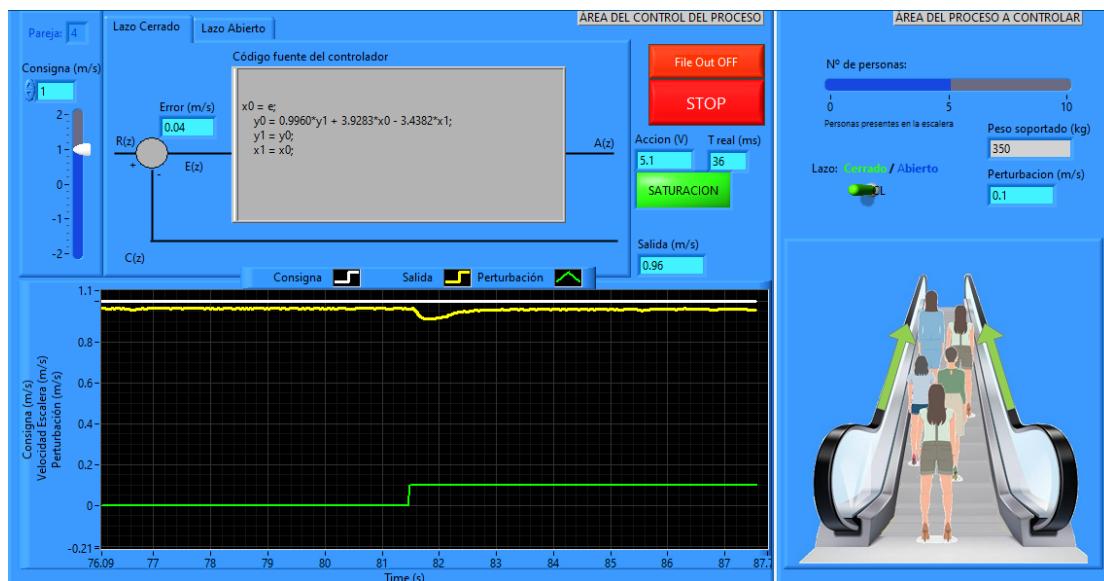


Figura 37 GUI LC Entrada 1 m/s con 5 Personas

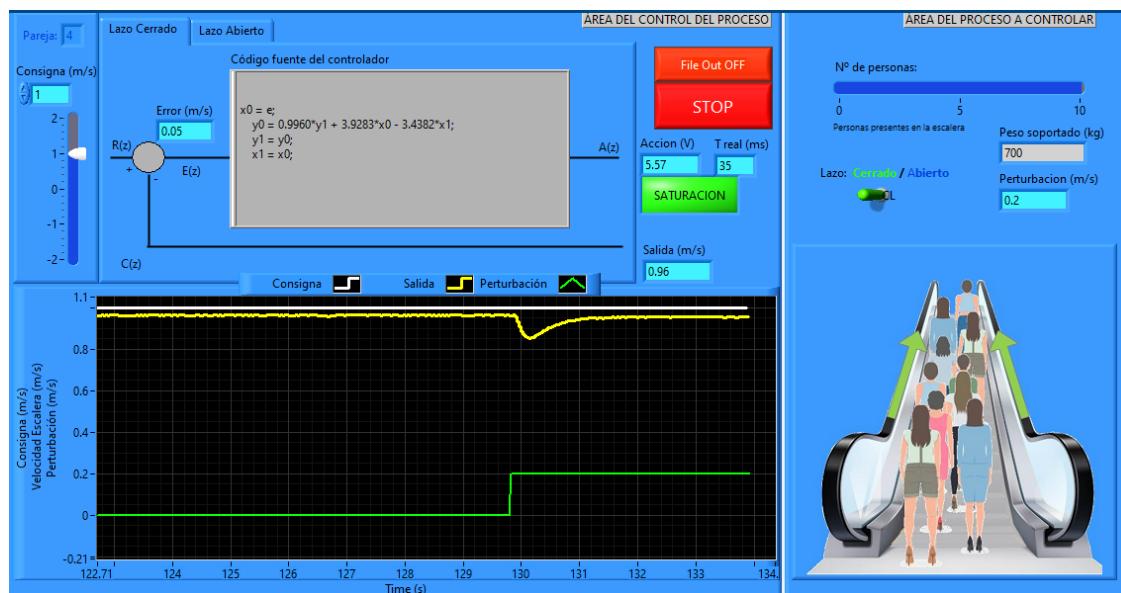


Figura 38 GUI LC Entrada 1 m/s con 10 Personas

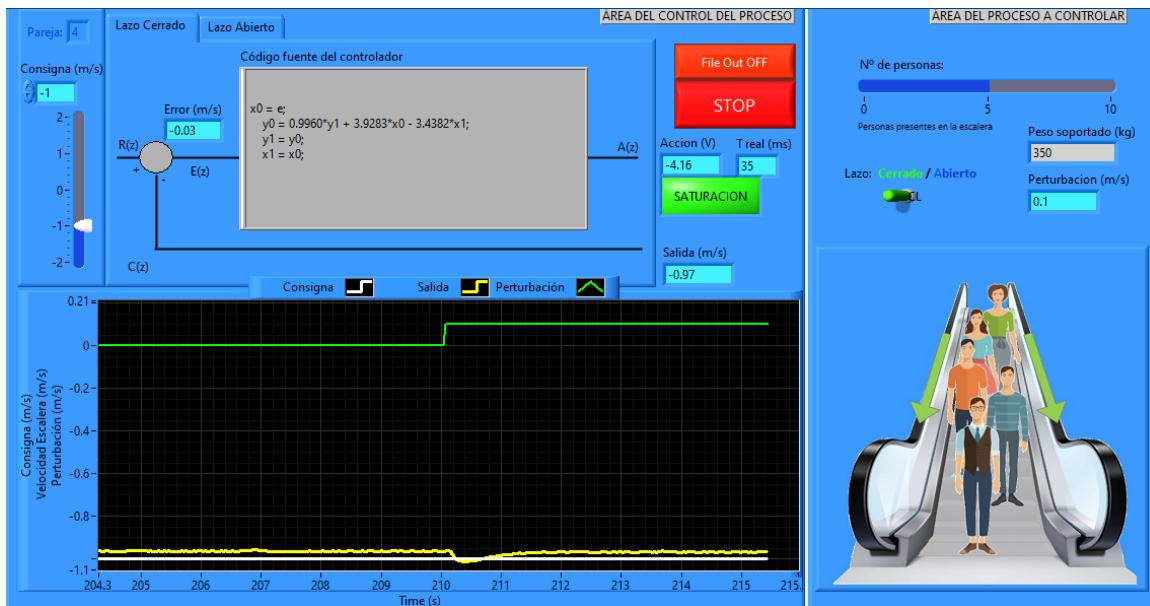


Figura 39 GUI LC Entrada -1 m/s con 5 Personas

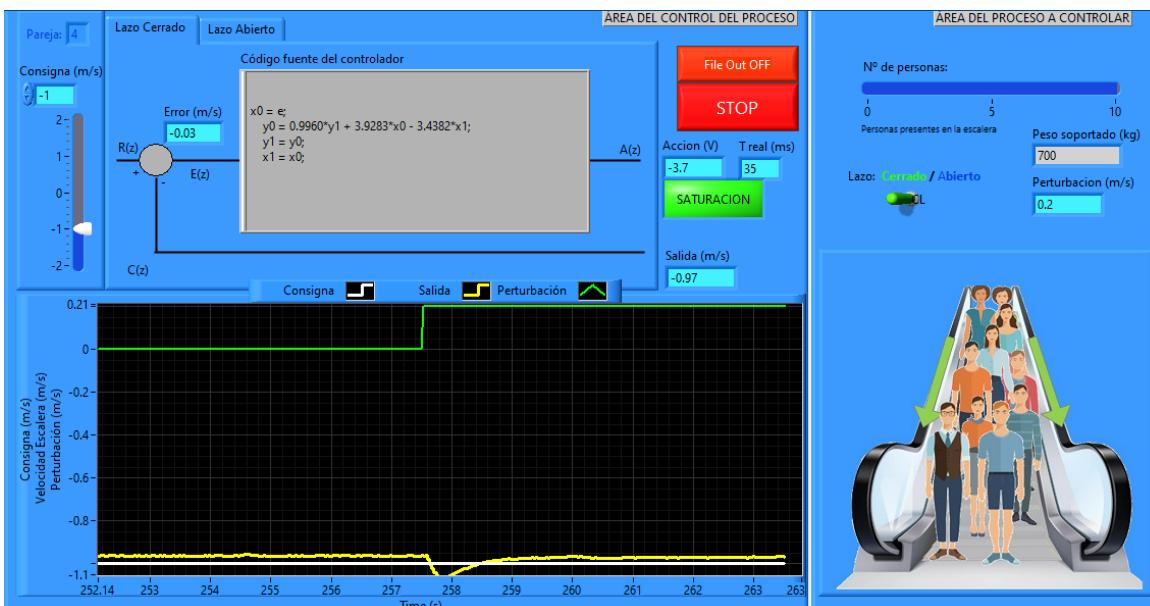


Figura 40 GUI LC Entrada -1 m/s con 10 Personas

Guardando los ficheros con el contenido de estas capturas, las vamos a ver a continuación mediante Matlab para mayor detalle, donde se encuentran las respuestas sin perturbación (en azul), con 5 personas (naranja) y con 10 personas (naranja):

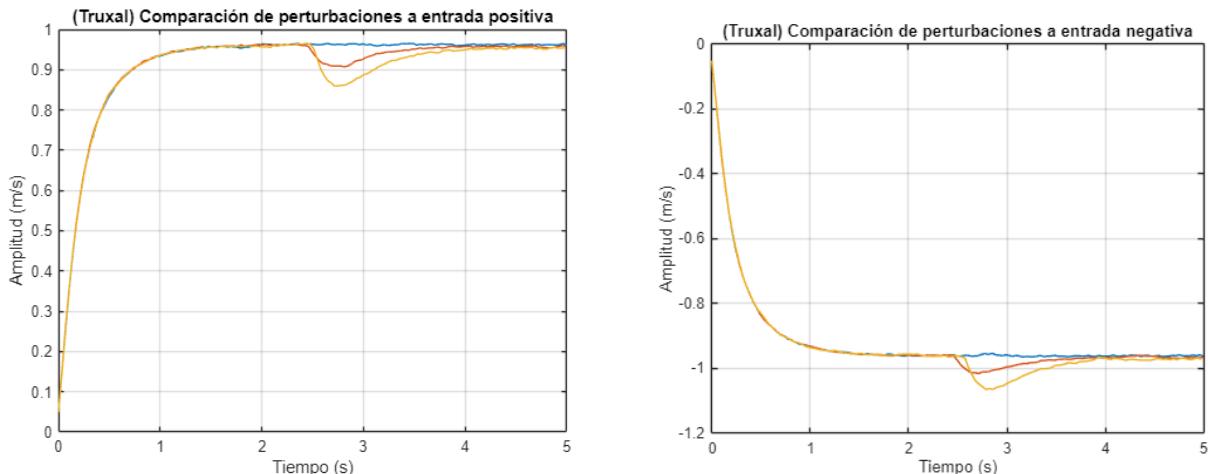


Figura 41. Comparación perturbaciones Truxal

De estas capturas obtenemos la siguiente tabla de resultados:

	0 personas		5 personas		10 personas	
Entrada	1 m/s	-1 m/s	1 m/s	-1 m/s	1 m/s	-1 m/s
Velocidad de la escalera [m/s]	0.96	-0.97	0.96	-0.97	0.96	-0.97
Acción de control [V]	4.62	-4.63	5.1	-4.16	5.57	-3.7

Tabla 6. Características perturbaciones Truxal

Como se puede observar, cuantas más personas se suben a la escalera, mayor es el peso que ejercen:

- **En el caso de la subida:** la señal de acción aumenta dado que se necesita más energía para mover la escalera y que se mantenga constante la velocidad.
- **En el caso de la bajada:** cuando mayor es el peso, más facilidad tiene la escalera para bajar, por lo que la señal de control disminuye (en módulo) para que pueda mantenerse constante la velocidad.

Además, en este caso se especificó un error del 3% en régimen permanente, error que se puede observar en la velocidad de la escalera, ya que esta no llega a ser igual que la entrada escalón, sino que tiene un error de aproximadamente 3% en todos los casos.

Al ser un sistema realimentado, cuando la perturbación se produce, el sistema rápidamente compensa la perturbación y recupera la velocidad, tal y como se puede observar en las figuras anteriores. Esto se logra gracias a la acción de control, modificando su valor según la perturbación y velocidad medidas.

Ejercicio 14

Extraiga cuantas reflexiones estime oportunas sobre el cumplimiento de los objetivos y sobre la comparativa entre especificaciones de diseño y resultados teóricos, simulados y reales.

Como se ha desarrollado en los ejercicios anteriores, los valores reales obtenidos del modelo realizado por el método directo de Truxal no nos dan unos valores que concuerden al 100% con lo esperado.

Esto se ha debido a diversos factores que, aunque de manera independiente no aportan una gran cantidad de error, juntos han provocado que haya una gran disparidad con lo que podría esperar de lo calculado de manera teórica.

Entre estos errores están:

- **Aproximación de $\ln(0.05) \approx -3$:** Si bien es cierto que esta aproximación es igual de válida que la aproximación de este valor a $-\pi$, sigue sin ser realmente el valor real. Por lo que esto introduce un ligero error.
- **Decimales en el cálculo teórico:** A la hora de desarrollar los cálculos teóricos del modelo se consideraron una menor cantidad de cifras significativas que las que se toman en Matlab. Este error es en sí la mayor fuente de error con la que contamos.

Estos errores afectan principalmente al régimen transitorio en el orden de apenas unos ~ 100 ms. Que aún con estos errores en juego, cabe destacar que el modelo obtenido cumple correctamente con los resultados esperados, hasta el punto de dar un error máximo en régimen permanente entorno al 3%.

En conclusión, se puede afirmar que el modelo planteado es más que correcto y sería necesario corregir los errores indicados anteriormente para aportar más precisión al mismo.

Fase 4: Diseño del regulador PID

A continuación, se plantea resolver el sistema de control en lazo cerrado mediante un regulador **PID discreto en la variante más sencilla** posible que permita **cancelar la dinámica dominante de la planta** para asegurar en el lazo cerrado un **tiempo de establecimiento de 1.12 s, sin oscilaciones** en régimen transitorio y con **error nulo** ante escalón en régimen permanente.

Ejercicio 15

Calcule la FdeT del regulador así como sus unidades, aplicando la teoría que corresponda y justificando adecuadamente la variante de PID elegida, el resultado debe presentarlo de la forma factorizada, polinomial y con potencias negativas en z. Presente la ecuación en diferencias del regulador. Obtenga las ganancias proporcional, integral y derivativa, así como los tiempos de acción integral y derivativa del PID (parámetros de sintonía).

Recordamos que tenemos los siguientes parámetros:

- **Del enunciado de la fase:**

$$t_s = 1.12 \text{ [s]}$$

$$e_{rpp} = 0$$

- **Del apartado anterior:**

$$T_s = 0.035 \text{ [s]}$$

$$B_o G(z) = \frac{0.0251}{z - 0.8752} \left[\frac{m/s}{V} \right]$$

1. **Obtenemos el módulo del polo buscado:**

$$k_s = \left\lceil \frac{t_s}{T_s} \right\rceil = \left\lceil \frac{1.12}{0.035} \right\rceil = 32$$

$$|p| = e^{-\frac{3}{k_s}} = e^{-\frac{3}{32}} = 0.9105$$

Al tener oscilación nula, el polo solo puede estar en el eje real positivo, por lo tanto:

$$p = 0.9105$$

2. **Obtenemos el regulador:**

- **Regulador P:**

Descartado, no puede ser un regulador P ya que no hay ninguna K_p para la que el polo sea el que estamos buscado. Esto se puede observar en el LdeR de la cadena abierta de este regulador:

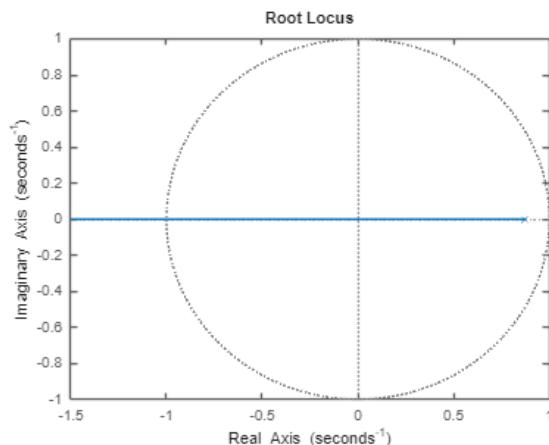


Figura 42. Lugar de raíces de la cadena abierta resultante con un regulador P

- **Regulador I:**

No puede ser un regulador I ya que este añadiría un nuevo polo y el polo anterior no sería el dominante. Además, si realizamos el LdeR de la cadena abierta de este regulador, obtenemos el siguiente resultado:

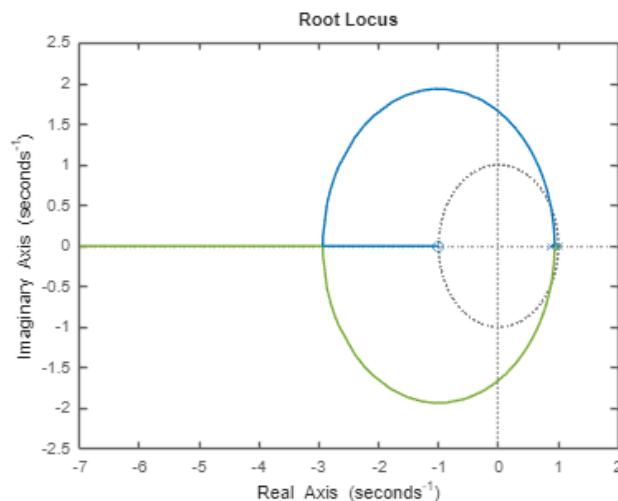


Figura 43. Lugar de raíces de la cadena abierta resultante con un regulador I

- **Regulador PI:**

Es un candidato ya que el cero que introduce eliminaría el polo no dominante, para asegurarnos así de que el polo dominante es el buscado. Eso se puede conseguir con el siguiente regulador:

$$F(z) = K_f \frac{z - 0.8752}{z - 1}$$

Con esto, solo falta encontrar el valor de K_f para el que el polo valga 0.9105. Para ello aplicamos la condición modular:

$$|K_f| = \frac{1}{|X|} = \left| \frac{z - 1}{0.0251} \right| = \left| \frac{0.9105 - 1}{0.0251} \right| = 3.5657$$

Finalmente, la función del regulador sería la siguiente:

$$F(z) = 3.5657 \frac{z - 0.8752}{z - 1} = \frac{3.5657 - 3.1207z^{-1}}{1 - z^{-1}} [\frac{V}{m/s}]$$

$$TF^{-1}\{A(z) * F_D(z)\} = TF^{-1}\{E(z) * F_N(z)\}$$

$$a[k] = 3.5657e[k] - 3.1207e[k-1] + a[k-1]$$

3. Parámetros de sintonía:

$$T_i = \frac{T_s}{2} * \frac{1+c}{1-c} = \frac{0.035}{2} * \frac{1+0.8752}{1-0.8752} = 0.2629 [s]$$

$$K_p = K_f * \frac{2T_i}{2T_i + T} = 3.5657 * \frac{2 * 0.2629}{2 * 0.2629 + 0.035} = 3.3432$$

$$K_i = \frac{K_p}{T_i} = \frac{3.3432}{0.2629} = 12.7166$$

$$q_0 = K_p + \frac{K_i * T}{2} = 3.5657$$

$$q_1 = -K_p + \frac{K_i * T}{2} = -3.1207$$

Finalmente:

$$F(z) = \frac{3.5657z^2 - 3.1207z}{z(z-1)} = \frac{3.566z - 3.121}{z-1} [\frac{V}{m/s}]$$

Ejercicio 16

Calcule la FdeT del sistema y obtenga el/los polo/s del sistema:

Con los datos anteriores:

$$B_o G(z) = \frac{0.0251}{z - 0.8752} [\frac{m/s}{V}]$$

$$F(z) = \frac{3.5657 - 3.1207z^{-1}}{1 - z^{-1}} [\frac{V}{m/s}]$$

Y mediante la función de Matlab:

```
Mz = minreal(feedback(series(Fz, BoG), 1));
```

Podemos obtener la FdeT del sistema:

$$M(z) = \frac{0.0895}{z - 0.9105} \left[\frac{m/s}{m/s} \right]$$

Donde los polos del sistema son:

$$p_1 = 0.9106$$

$$p_2 = 0.8751$$

Además, si realizamos el LdeR del sistema, obtenemos el siguiente resultado:

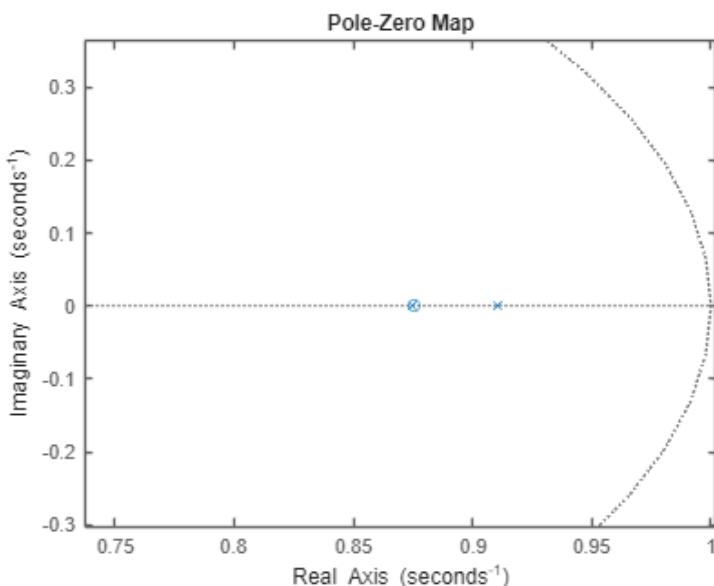


Figura 44. Polos y ceros del sistema

Ejercicio 17

Caracterice teóricamente, tanto en régimen transitorio como permanente, la salida del sistema ante entradas escalón de $\pm 1 \text{ m/s}$.

Régimen permanente:

$$c(\infty) = R * \lim_{z \rightarrow 1} M(z) \left\{ \begin{array}{l} R = 1 \text{ m/s} \rightarrow c(\infty) = 1 * 1 = 1 \text{ m/s} \\ R = -1 \text{ m/s} \rightarrow c(\infty) = -1 * 1 = -1 \text{ m/s} \end{array} \right.$$

Régimen transitorio:

Contamos con un polo real simple, por lo tanto, el sistema es sobreamortiguado:

$$p = 0.9105$$

$$k_s = \frac{\ln(0.05)}{\ln(|p|)} = \frac{-3}{\ln(0.9105)} = 32$$

$$t_s = k_s T_s = 1.12 [s]$$

Ejercicio 18

Calcule en régimen permanente tanto la acción de control como la velocidad de la escalera subiendo y bajando en diferentes condiciones de carga de la escalera, para consignas de $\pm 1 \text{ m/s}$. Relacione los incrementos o decrementos de acción de control con el peso de las personas presentes en la escalera.

Para la realización de este apartado, se ha creado una función escalera en Matlab que realiza las siguientes operaciones:

$$k_0[M(z)_{\text{entrada}}] = R * \lim_{z \rightarrow 1} M(z)$$

$$k_0[M(z)_{\text{pert}}] = P * \lim_{z \rightarrow 1} \frac{1}{1 + F(z) * B_o G(z)}$$

$$k_0[B_o G(z)] = \lim_{z \rightarrow 1} B_o G(z)$$

$$c(\infty) = k_0[M(z)_{\text{entrada}}] + k_0[M(z)_{\text{pert}}]$$

$$a(\infty) = \frac{c(\infty) + P}{k_0[B_o G(z)]}$$

Entrada escalón [m/s]	Perturbación [m/s]	Acción de control [V]	Velocidad de la escalera [m/s]
1	0	4.9737	1
-1	0	-4.9737	-1
1	0.1	5.4710	1
-1	0.1	-4.4763	-1
1	0.2	5.9684	1
-1	0.2	-3.9789	-1

Tabla 7. Velocidad de la escalera y acción de control a entrada escalón $\pm 1 \text{ m/s}$ con diferentes perturbaciones (PID)

Como se puede observar, cuantas más personas se suben a la escalera, mayor es el peso que ejercen:

- **En el caso de la subida:** la señal de acción aumenta dado que se necesita más energía para mover la escalera y que se mantenga constante la velocidad.
- **En el caso de la bajada:** cuando mayor es el peso, más facilidad tiene la escalera para bajar, por lo que la señal de control disminuye (en módulo) para que pueda mantenerse constante la velocidad.

Además, en este caso no hay error, por lo que la velocidad de la escalera corresponde con la entrada escalón.

Fase 5: Validación del regulador PID

En lazo cerrado, justifique que el regulador diseñado cumple con lo esperado, realizando las siguientes tareas:

Ejercicio 19

Simule en Simulink el sistema de control con y sin alinealidades. Mida las características de las respuestas (tanto el régimen permanente como el transitorio) ante entradas escalón de $\pm 1 \text{ m/s}$. Analice las diferencias de las medidas respecto de los cálculos teóricos.

En la herramienta de Simulink, se han realizado los siguientes modelos para validar el regulador desarrollado en la fase anterior:

- **Modelo sin alinealidades:**

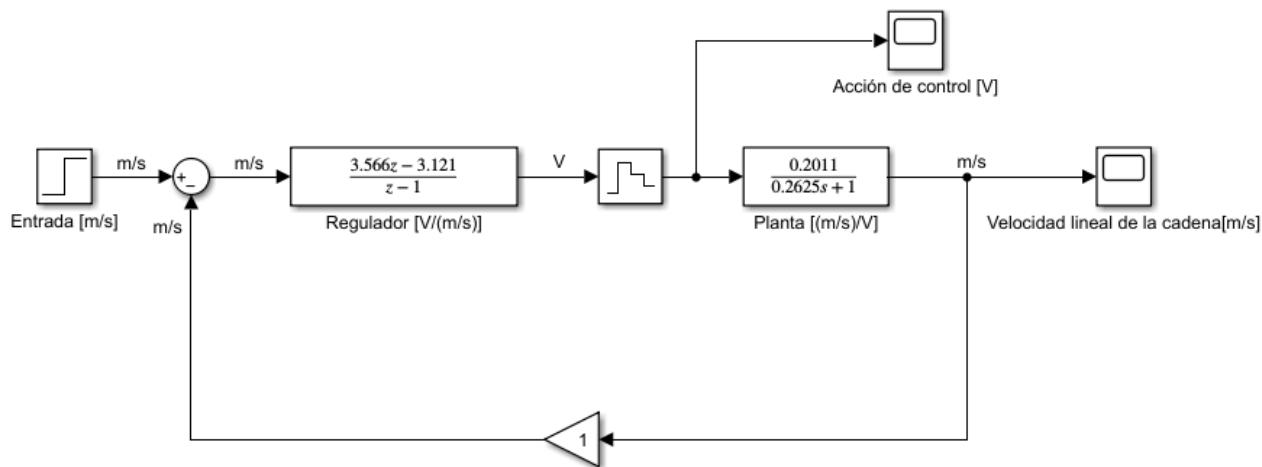


Figura 45. Modelo sistema con regulador PID sin alinealidades

- **Modelo con alinealidades:**

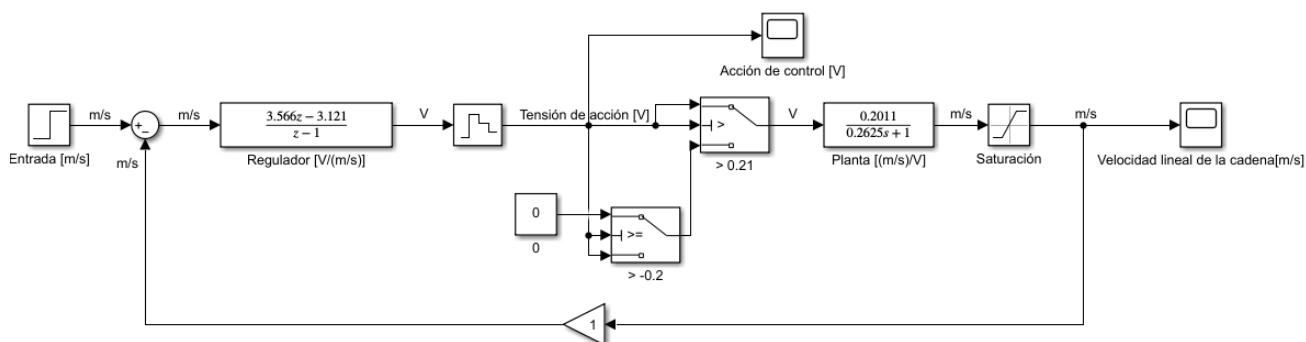


Figura 46. Modelo sistema con regulador PID con alinealidades

Simulando las mismas señales que en el primer ejercicio obtenemos las siguientes respuestas, de la entrada (azul) y la salida (naranja):

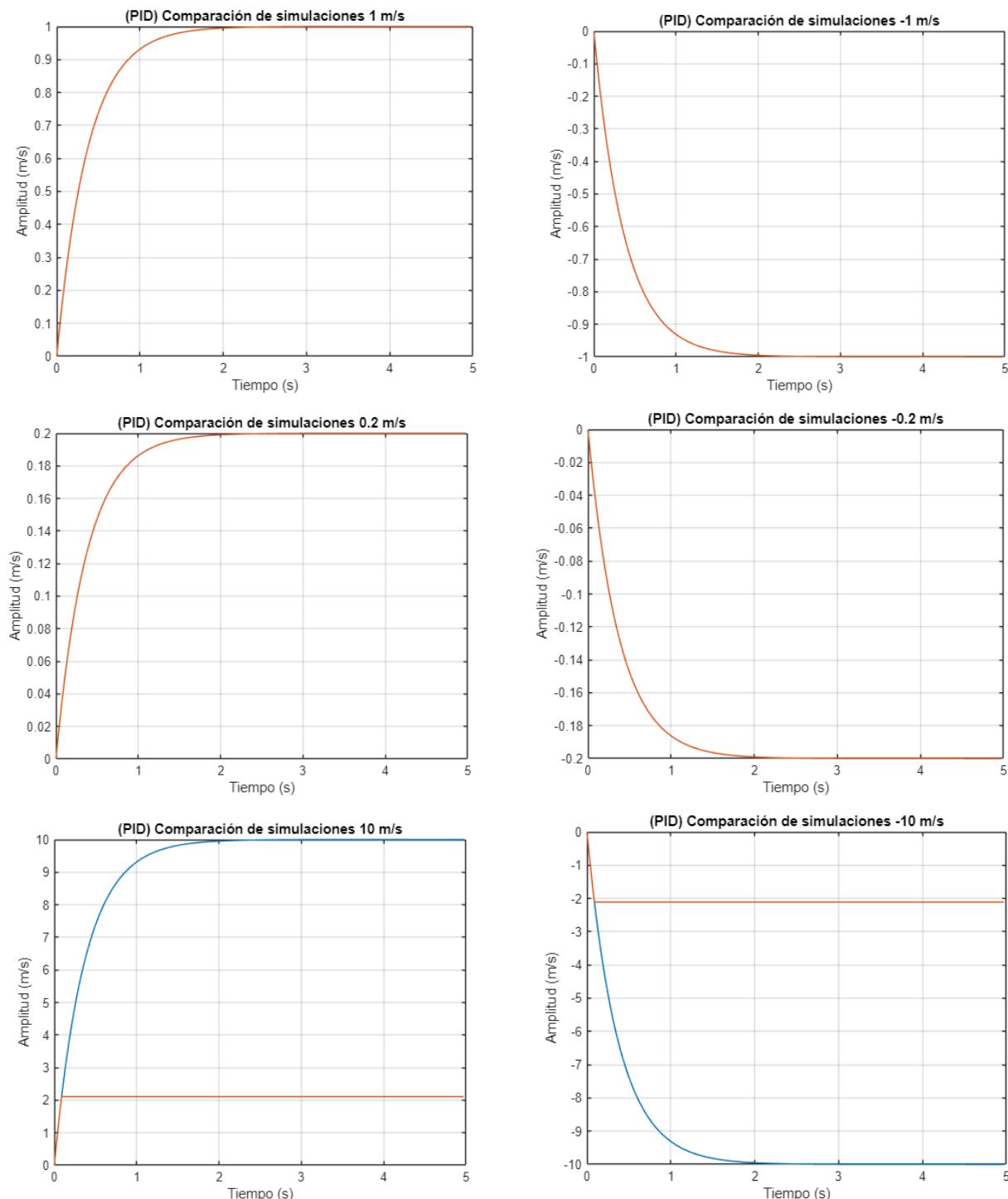


Figura 47. Comparación de simulaciones PID

Podemos observar que:

- **Para $\pm 1V$:** las dos señales coinciden.
- **Para $\pm 10V$:** las dos señales no coinciden. Esto se debe a que la saturación es respecto a la salida, por lo que sigue ocurriendo cuando la salida es mayor que 2.1098 m/s o menor que -2.1123 m/s .

- Para $\pm 0.2V$: se observa que las dos salidas coinciden. Esto se debe a que la zona muerta ha cambiado dado que la acción de control ya no coincide con la señal de entrada.

En la siguiente tabla se puede observar la comparación de las características de las respuestas (tanto en régimen permanente como en transitorio) de los cálculos teóricos y las simulaciones sin y con alinealidades para una entrada escalón de $\pm 1 \text{ m/s}$.

Entrada	Tiempo de establecimiento [s]	Salida (∞) [m/s]	Acción de control [V]
Teórica $+1 \text{ m/s}$	1.120	1	4.9737
$+1 \text{ m/s}$ sin alinealidades	1.118	1	4.9730
$+1 \text{ m/s}$ con alinealidades	1.118	1	4.9730
Teórica -1 m/s	1.120	-1	-4.9737
-1 m/s sin alinealidades	1.118	-1	-4.9730
-1 m/s con alinealidades	1.118	-1	-4.9730

Tabla 8. Características PID

Se observa que, para las simulaciones, los valores no cambian dado que para entrada $\pm 1 \text{ m/s}$ no afectan las alinealidades a la respuesta.

Ejercicio 20

Mediante el GUI, obtenga la captura de la respuesta del sistema para una consigna de 1 m/s (subiendo) y sin personas transportadas (sin carga). Adjunte una captura de pantalla de la ventana de gestión del sistema de control en la que se aprecie con la suficiente nitidez el código fuente del regulador (Figura 2.b). Mida las características de la respuesta y coteje los resultados con los de la simulación del ejercicio anterior, así como con los cálculos teóricos realizados en la fase de diseño del regulador.

Mediante la interfaz GUI, se ha obtenido la siguiente respuesta a la entrada de 1 m/s con fichero que hemos introducido (fichero disponible en el anexo del documento):

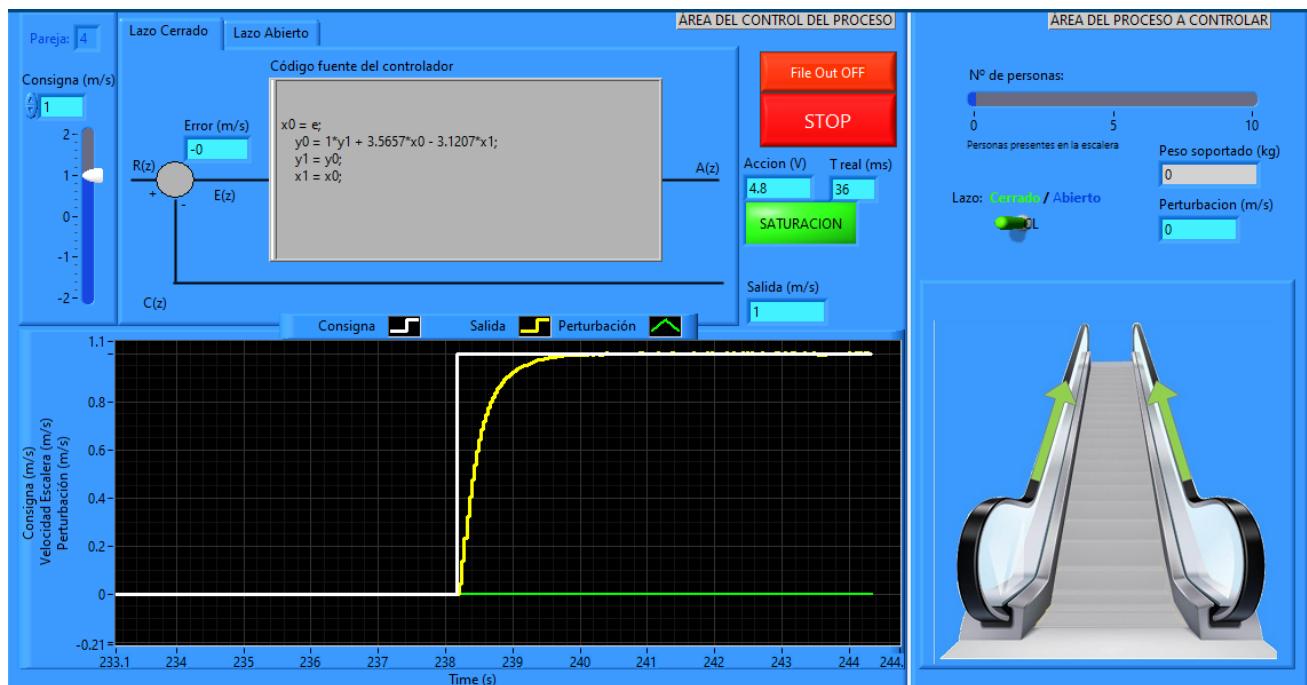


Figura 48. GUI PID 1 m/s

Guardando los ficheros con el contenido de esta captura, lo vamos a continuación mediante Matlab para mayor detalle:

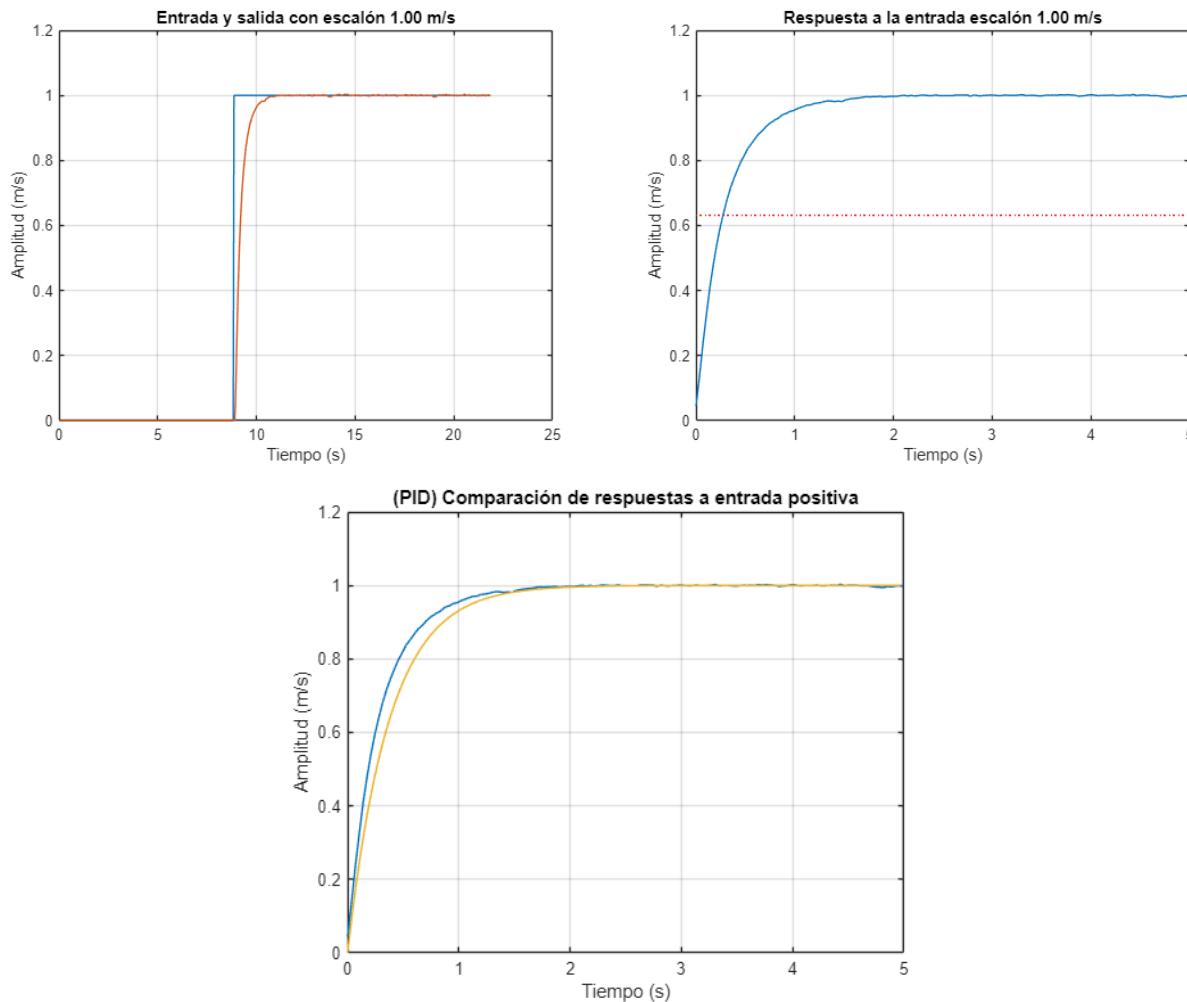


Figura 49. Comparación respuestas +1m/s PID

En este caso, debido a que se ha utilizado el mismo regulador para todos los casos, las diferencias entre las características teóricas/simuladas y las reales se deben a que es un sistema físico y presenta características que no se pueden caracterizar.

Entrada +1m/s	Tiempo de establecimiento [s]	Salida (∞) [m/s]	Acción de control [V]
Teórica	1.12	1	4.9737
Simulada sin alinealidades	1.118	1	4.973
Simulada con alinealidades	1.118	1	4.973
Real	0.948	0.999	4.8

Tabla 9. Comparación PID +1 m/s

Ejercicio 21

Igual que el ejercicio 11 pero con consigna de -1 m/s (bajando).

Mediante la interfaz GUI, se ha obtenido la siguiente respuesta a la entrada de 1 m/s con fichero que hemos introducido (fichero disponible en el anexo del documento):

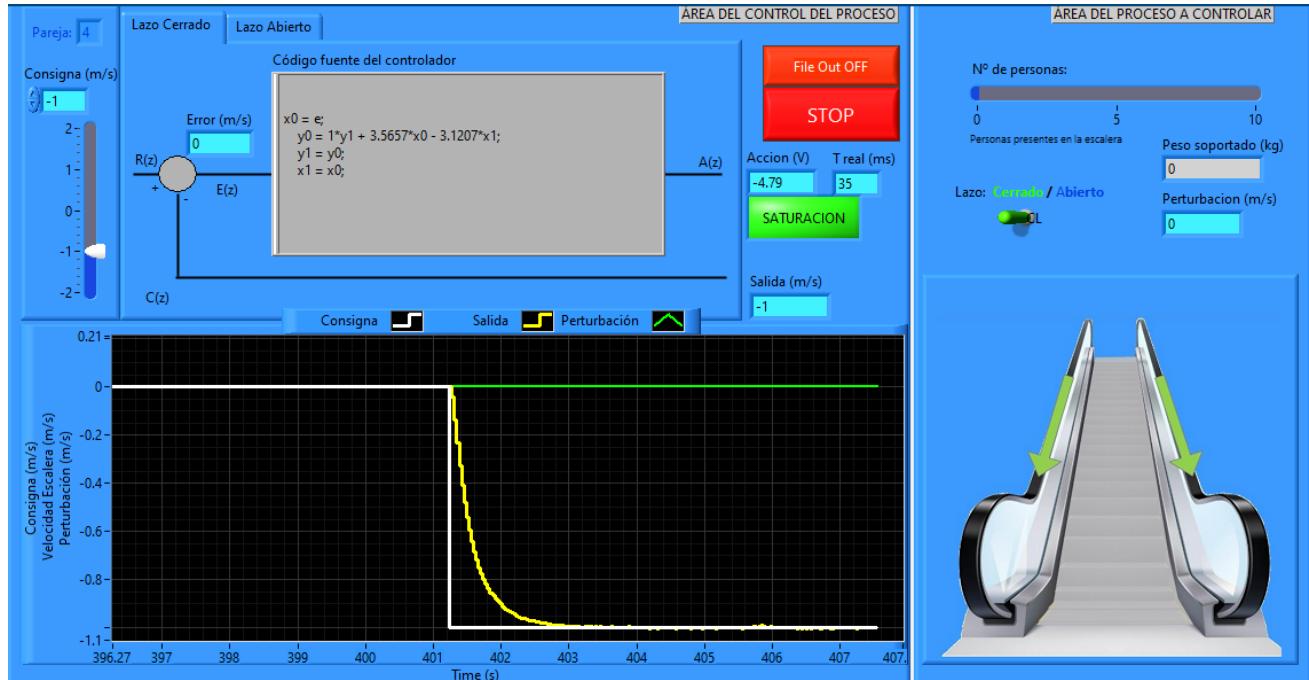


Figura 50. GUI PID -1 m/s

Guardando los ficheros con el contenido de esta captura, lo vamos a continuar mediante Matlab para mayor detalle:

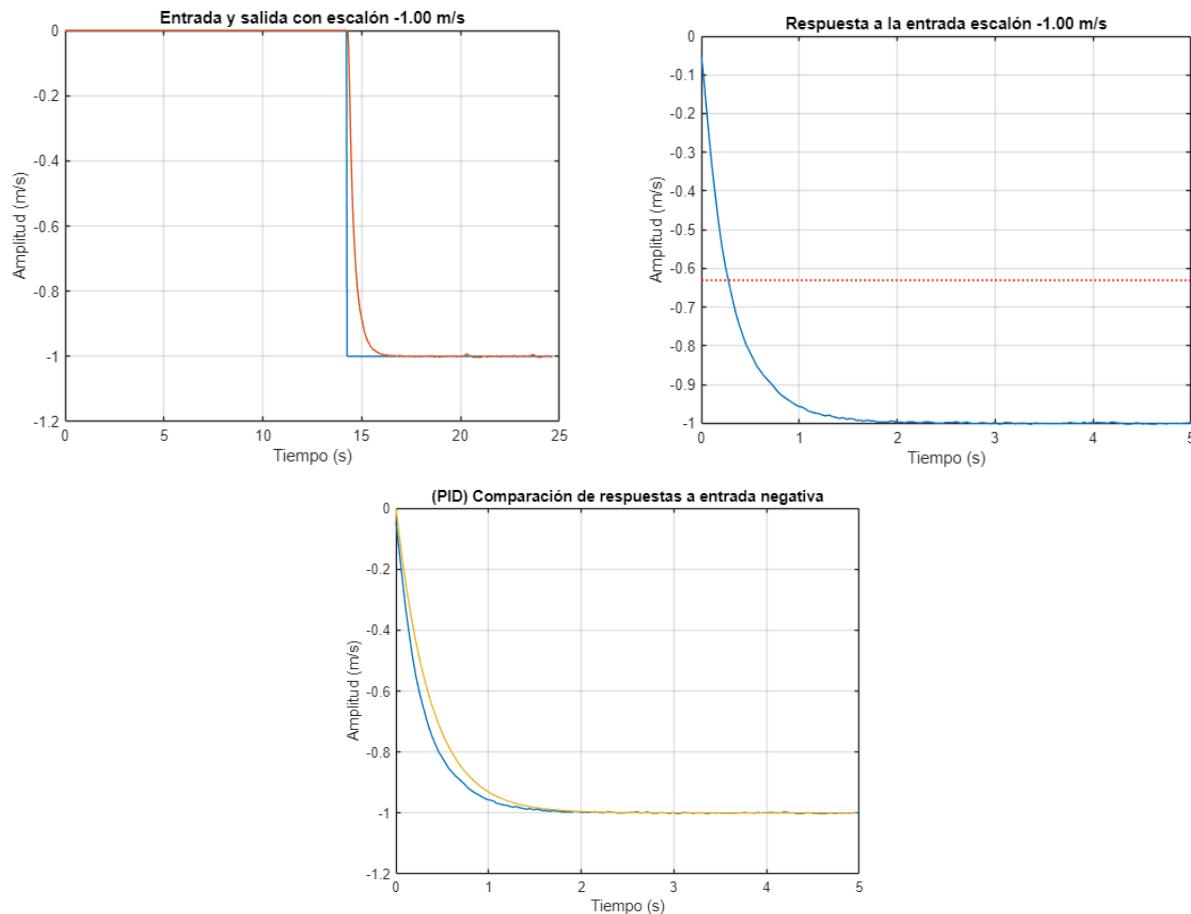


Figura 51. Comparación respuestas -1m/s PID

En este caso, debido a que se ha utilizado el mismo regulador para todos los casos, las diferencias entre las características teóricas/simuladas y las reales se deben a que es un sistema físico y presenta características que no se pueden caracterizar.

Entrada -1m/s	Tiempo de establecimiento [s]	Salida (∞) [m/s]	Acción de control [V]
Teórica	1.12	-1	-4.9737
Simulada sin alinealidades	1.118	-1	-4.973
Simulada con alinealidades	1.118	-1	-4.973
Real	0.952	-0.999	-4.79

Tabla 10. Comparación PID -1 m/s

Ejercicio 22

Mida en régimen permanente tanto la acción de control como la velocidad de la escalera, subiendo y bajando en diferentes condiciones de carga de la escalera, para consignas de $\pm 1 \text{ m/s}$. Relacione los incrementos o decrementos de acción de control con el peso de las personas presentes en la escalera.

Mediante la interfaz GUI, se ha obtenido la siguiente respuesta a la entrada de 1 m/s y -1 m/s con fichero que hemos introducido (fichero disponible en el anexo del documento) y se han aplicado distintas perturbaciones:

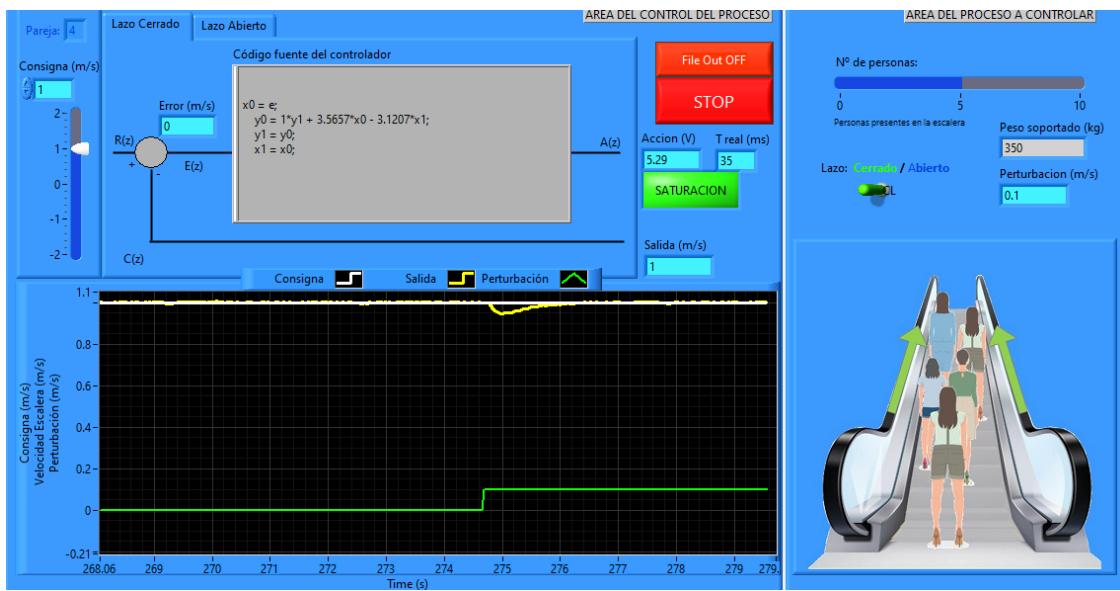


Figura 52 GUI LC F5 Entrada 1 m/s y 5 personas

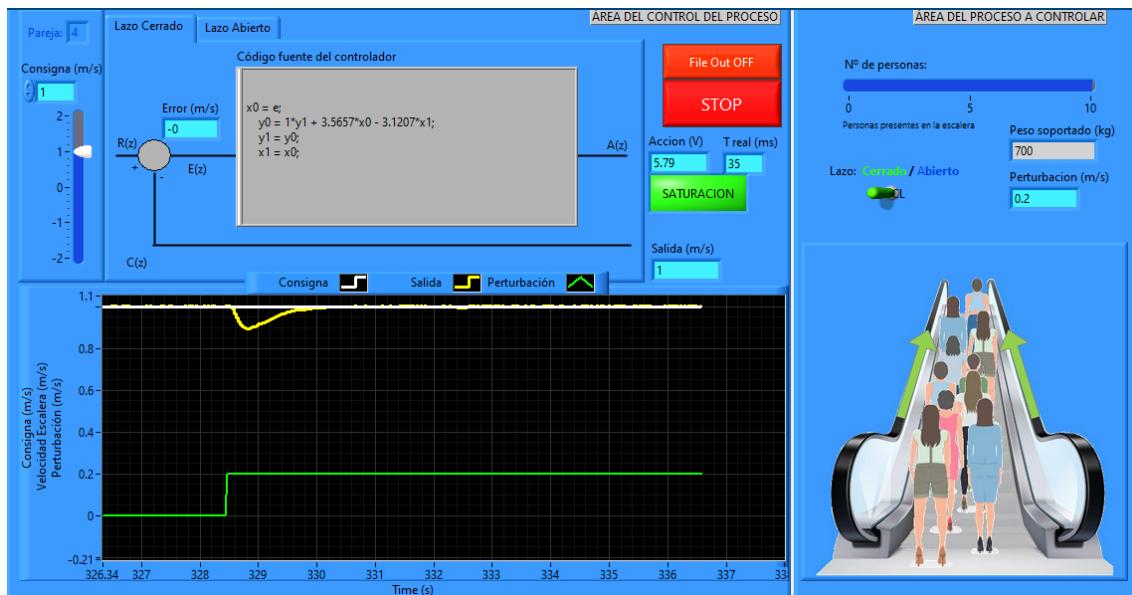


Figura 53 GUI LC F5 Entrada 1 m/s y 10 Personas

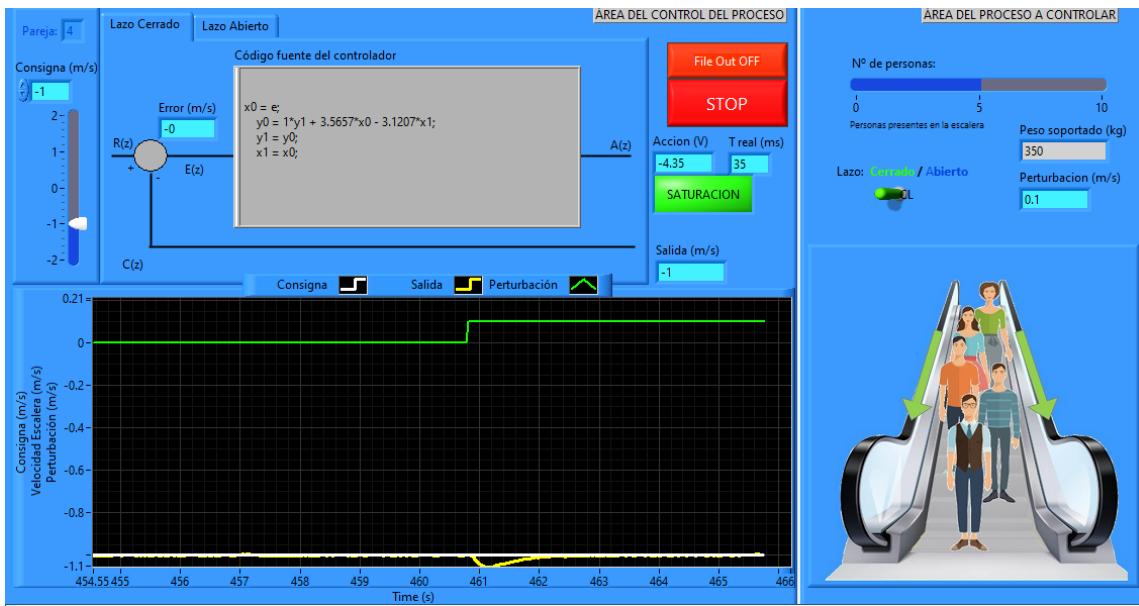


Figura 54 GUI LC F5 Entrada -1 m/s y 5 personas

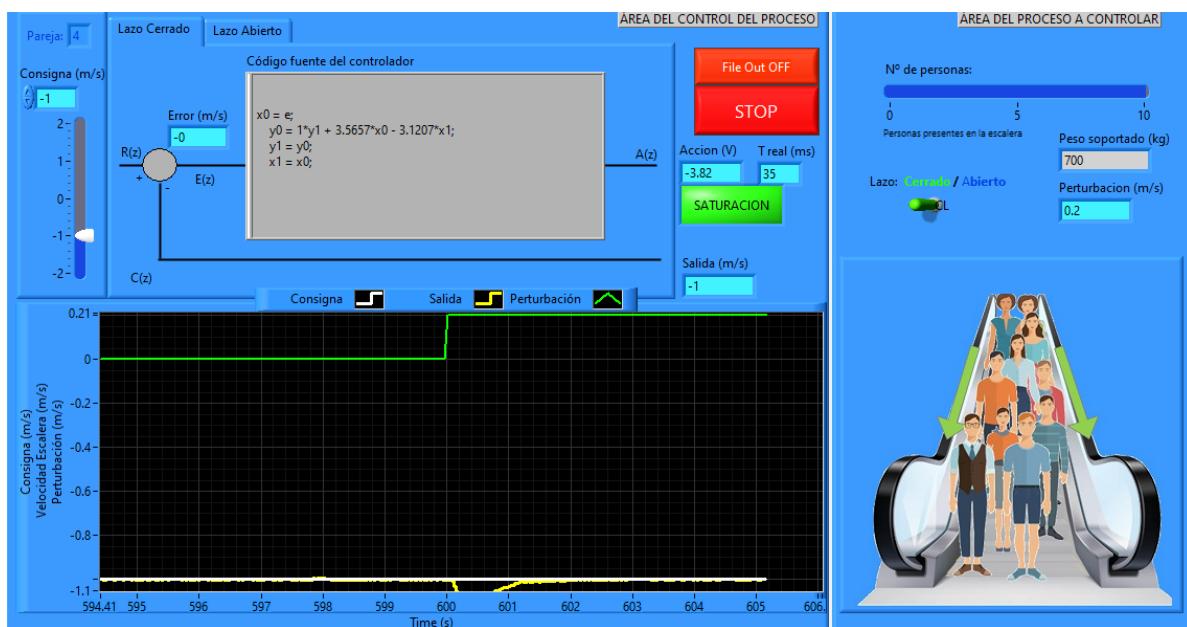


Figura 55 GUI LC F5 Entrada -1 m/s y 10 personas

Guardando los ficheros con el contenido de estas capturas, las vamos a ver a continuación mediante Matlab para mayor detalle, donde se encuentran las respuestas sin perturbación (en azul), con 5 personas (naranja) y con 10 personas (naranja):

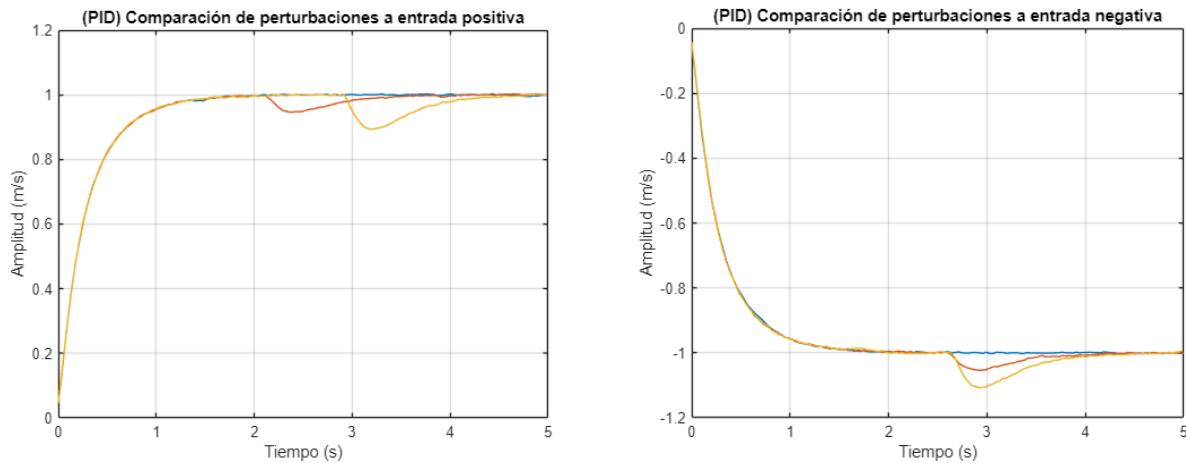


Figura 56. Comparación perturbaciones PID

De estas capturas obtenemos la siguiente tabla de resultados:

	0 personas		5 personas		10 personas	
Entrada	1 m/s	-1 m/s	1 m/s	-1 m/s	1 m/s	-1 m/s
Velocidad de la escalera [m/s]	1	-1	1	-1	1	-1
Acción de control [V]	4.8	-4.79	5.29	-4.35	5.79	-3.82

Tabla 11. Características perturbaciones PID

Como se puede observar, cuantas más personas se suben a la escalera, mayor es el peso que ejercen:

- **En el caso de la subida:** la señal de acción aumenta dado que se necesita más energía para mover la escalera y que se mantenga constante la velocidad.
- **En el caso de la bajada:** cuando mayor es el peso, más facilidad tiene la escalera para bajar, por lo que la señal de control disminuye (en módulo) para que pueda mantenerse constante la velocidad.

Al ser un sistema realimentado, cuando la perturbación se produce, el sistema rápidamente compensa la perturbación y recupera la velocidad, tal y como se puede observar en las figuras anteriores.

Esto se logra gracias a la acción de control, modificando su valor según la perturbación y velocidad medidas.

Ejercicio 23

Extraiga cuantas reflexiones estime oportunas sobre el cumplimiento de los objetivos y sobre la comparativa entre especificaciones de diseño y resultados teóricos, simulados y reales.

Como se ha visto y comentado en los ejercicios 20 y 21, los resultados tanto teóricos como simulados son exactamente iguales, mientras que los resultados reales se diferencian con los teóricos debido a que estamos trabajando con un sistema físico.

Este presenta ciertas alinealidades características que no se pueden caracterizar mas allá de su medición, ya que estas no dependen de nada relacionado con el modelo desarrollado.

Omitiendo este pequeño detalle, se puede asegurar que el modelo de regulador PID desarrollado cumple con los objetivos planteados y asegurando una gran precisión, coincidiendo los valores teóricos, simulados y reales.

Cabe destacar también que este regulador es significativamente mas estable que el regulador planteado mediante el método directo de Truxal de las fases 2 y 3.

Anexo 1: Fase 1

Fase 1: Modelado y caracterización de la planta en lazo abierto

```
clear
addpath('Funciones\');
addpath('Simulink\')

format long
Ts = 35e-3;
fprintf('El periodo de muestreo es: %.4f s.\n', Ts)
```

Ejercicio 1: Obtener la FdeT de la planta en lazo abierto.

Entrada escalón 1V:

```
[salida_1V, tiempo_1V] = representar('./Capturas/Fase1/PGA_G4_1V.txt', 1, 'V', 0);
[km_1V, Tau_1V, salida_1V, tiempo_1V] = salida(salida_1V, tiempo_1V, Ts, 1, 'V', 0);
fprintf('km con entrada escalón de 1V: %.4f.\n', km_1V)
fprintf('Tau con entrada escalón de 1V: %.4f s.\n', Tau_1V)
```

Entrada escalón 2V:

```
[salida_2V, tiempo_2V] = representar('./Capturas/Fase1/PGA_G4_2V.txt', 2, 'V', 0);
[km_2V, Tau_2V, salida_2V, tiempo_2V] = salida(salida_2V, tiempo_2V, Ts, 2, 'V', 0);
fprintf('km con entrada escalón de 2V: %.4f.\n', km_2V)
fprintf('Tau con entrada escalón de 2V: %.4f s.\n', Tau_2V)
```

Entrada escalón 5V:

```
[salida_5V, tiempo_5V] = representar('./Capturas/Fase1/PGA_G4_5V.txt', 5, 'V', 0);
[km_5V, Tau_5V, salida_5V, tiempo_5V] = salida(salida_5V, tiempo_5V, Ts, 5, 'V', 0);
fprintf('km con entrada escalón de 5V: %.4f.\n', km_5V)
fprintf('Tau con entrada escalón de 5V: %.4f s.\n', Tau_5V)
```

Entrada escalón -1V:

```
[salida_n1V, tiempo_n1V] = representar('./Capturas/Fase1/PGA_G4_n1V.txt', -1, 'V', 0);
[km_n1V, Tau_n1V, salida_n1V, tiempo_n1V] = salida(salida_n1V, tiempo_n1V, Ts, -1, 'V', 0);
fprintf('km con entrada escalón de -1V: %.4f.\n', km_n1V)
fprintf('Tau con entrada escalón de -1V: %.4f s.\n', Tau_n1V)
```

Entrada escalón -2V:

```
[salida_n2V, tiempo_n2V] = representar('./Capturas/Fase1/PGA_G4_n2V.txt', -2, 'V', 0);
[km_n2V, Tau_n2V, salida_n2V, tiempo_n2V] = salida(salida_n2V, tiempo_n2V, Ts, -2, 'V', 0);
```

```
fprintf('km con entrada escalón de -2V: %.4f.\n', km_n2V)
fprintf('Tau con entrada escalón de -2V: %.4f s.\n', Tau_n2V)
```

Entrada escalón -5V:

```
[salida_n5V, tiempo_n5V] = representar('./Capturas/Fase1/PGA_G4_n5V.txt', -5, 'V', 0);
[km_n5V, Tau_n5V, salida_n5V, tiempo_n5V] = salida(salida_n5V, tiempo_n5V, Ts, -5, 'V', 0);
fprintf('km con entrada escalón de -5V: %.4f.\n', km_n5V)
fprintf('Tau con entrada escalón de -5V: %.4f s.\n', Tau_n5V)
```

Media de las kms y Taus:

```
km = mean([km_1V (km_2V) (km_5V) (km_n1V) (km_n2V) (km_n5V)]);
Tau = mean([Tau_1V Tau_2V Tau_5V Tau_n1V Tau_n2V Tau_n5V]);
fprintf('La km de la planta es: %.4f.\n', km)
fprintf('La Tau de la planta es: %.4f s.\n', Tau)
```

FdeT de la planta:

```
Gs = tf(km, [Tau 1]);
```

Valores de interés:

```
p = pole(Gs);
fprintf('El polo de la planta es: %.4f.\n', p)
ts = -3/p;
fprintf('El tiempo de establecimiento de la planta es: %.4f.\n', ts)
```

Ejercicio 2: Validar el modelo mediante Simulink.**Entrada escalón 1V:**

```
step_value = 1;
[salida_simulado, tiempo_simulado] = simulink('modeloFdeT', '10');
comparacion(salida_simulado, tiempo_simulado, salida_1V, tiempo_1V, 'null', 'null', 'Comparación de
respuestas con escalón 1V')
```

Entrada escalón -1V:

```
step_value = -1;
[salida_simulado, tiempo_simulado] = simulink('modeloFdeT', '10');
comparacion(salida_simulado, tiempo_simulado, salida_n1V, tiempo_n1V, 'null', 'null', 'Comparación de
respuestas con escalón -1V')
```

Entrada escalón 2V:

```
step_value = 2;
[salida_simulado, tiempo_simulado] = simulink('modeloFdeT', '10');
comparacion(salida_simulado, tiempo_simulado, salida_2V, tiempo_2V, 'null', 'null', 'Comparación de
respuestas con escalón 2V')
```

Entrada escalón -2V:

```
step_value = -2;
[salida_simulado, tiempo_simulado] = simulink('modeloFdeT', '10');
comparacion(salida_simulado, tiempo_simulado, salida_n2V, tiempo_n2V, 'null', 'null', 'Comparación de
respuestas con escalón -2V')
```

Entrada escalón 5V:

```
step_value = 5;
[salida_simulado, tiempo_simulado] = simulink('modeloFdeT', '10');
comparacion(salida_simulado, tiempo_simulado, salida_5V, tiempo_5V, 'null', 'null', 'Comparación de
respuestas con escalón 5V')
```

Entrada escalón -5V:

```
step_value = -5;
[salida_simulado, tiempo_simulado] = simulink('modeloFdeT', '10');
comparacion(salida_simulado, tiempo_simulado, salida_n5V, tiempo_n5V, 'null', 'null', 'Comparación de
respuestas con escalón -5V')
```

Ejercicio 3: Modelación de la saturación y la zona muerta.**Saturación positiva:**

```
[salida_10V, tiempo_10V] = representar('./Capturas/Fase1/PGA_G4_10V.txt', 10, 'V', 0);
[km_10V, Tau_10V, salida_10V, tiempo_10V] = salida(salida_10V, tiempo_10V, Ts, 10, 'V', 0);
[salida_20V, tiempo_20V] = representar('./Capturas/Fase1/PGA_G4_20V.txt', 20, 'V', 0);
[km_20V, Tau_20V, salida_20V, tiempo_20V] = salida(salida_20V, tiempo_20V, Ts, 20, 'V', 0);
comparacion(salida_10V, tiempo_10V, salida_20V, tiempo_20V, 'null', 'null', 'Comparación saturación
positiva')
fprintf('Salida saturada positiva: %.4f.\n', (km_10V*10 + km_20V*20)/2)
```

Saturación negativa:

```
[salida_n10V, tiempo_n10V] = representar('./Capturas/Fase1/PGA_G4_n10V.txt', -10, 'V', 0);
[km_n10V, Tau_n10V, salida_n10V, tiempo_n10V] = salida(salida_n10V, tiempo_n10V, Ts, -10, 'V',
0);
```

```
[salida_n20V, tiempo_n20V] = representar('./Capturas/Fase1/PGA_G4_n20V.txt', -20, 'V', 0);
[km_n20V, Tau_n20V, salida_n20V, tiempo_n20V] = salida(salida_n20V, tiempo_n20V, Ts, -20, 'V', 0);
comparacion(salida_n10V, tiempo_n10V, salida_n20V, tiempo_n20V, 'null', 'null', 'Comparación
saturación negativa')
fprintf('Salida saturada negativa: %.4f.\n', (-1)*(km_n10V*10 + km_n20V*20)/2)
```

Zona muerta positiva:

```
[salida_0_19V, tiempo_0_19V] = representar('./Capturas/Fase1/PGA_G4_0.19V.txt', 0.19, 'V', 0);
[salida_0_2V, tiempo_0_2V] = representar('./Capturas/Fase1/PGA_G4_0.2V.txt', 0.2, 'V', 0);
[salida_0_21V, tiempo_0_21V] = representar('./Capturas/Fase1/PGA_G4_0.21V.txt', 0.21, 'V', 0);
[salida_0_22V, tiempo_0_22V] = representar('./Capturas/Fase1/PGA_G4_0.22V.txt', 0.22, 'V', 0);
salida(salida_0_22V, tiempo_0_22V, Ts, 0.22, 'V', 0);
```

Zona muerta negativa:

```
[salida_n0_19V, tiempo_n0_19V] = representar('./Capturas/Fase1/PGA_G4_n0.19V.txt', -0.19, 'V', 0);
[salida_n0_2V, tiempo_n0_2V] = representar('./Capturas/Fase1/PGA_G4_n0.2V.txt', -0.2, 'V', 0);
[salida_n0_21V, tiempo_n0_21V] = representar('./Capturas/Fase1/PGA_G4_n0.21V.txt', -0.21, 'V', 0);
salida(salida_n0_21V, tiempo_n0_21V, Ts, -0.21, 'V', 0);
```

Modelo Simulink:**Entradas positivas:**

```
step_value = 1;
[salida_sim_1V, tiempo_sim_1V] = simulink('modeloSatDead', '10');

step_value = 0.21;
[salida_sim_0_21V, tiempo_sim_0_21V] = simulink('modeloSatDead', '10');

step_value = 20;
[salida_sim_20V, tiempo_sim_20V] = simulink('modeloSatDead', '10');

comparacion(salida_sim_1V, tiempo_sim_1V, salida_sim_0_21V, tiempo_sim_0_21V, salida_sim_20V, tiempo_
sim_20V, 'Comparación entradas positivas')
```

Entradas negativas:

```
step_value = -1;
[salida_sim_n1V, tiempo_sim_n1V] = simulink('modeloSatDead', '10');

step_value = -0.2;
[salida_sim_n0_2V, tiempo_sim_n0_2V] = simulink('modeloSatDead', '10');
```

```
step_value = -20;
[salida_sim_n20V, tiempo_sim_n20V] = simulink('modeloSatDead', '10');

comparacion(salida_sim_n1V,tiempo_sim_n1V,salida_sim_n0_2V,tiempo_sim_n0_2V,salida_sim_n20V,tiempo_sim_n20V,'Comparación entradas negativas')
```

Ejercicio 4: Equivalente discreto en régimen permanente y transitorio.

Equivalente discreto:

```
BoG = c2d(Gs, Ts, 'zoh');

figure;

plot(tiempo_1V,salida_1V);
hold on;
plot(tiempo_sim_1V,salida_sim_1V);
[salida_eq, tiempo_eq] = step(BoG,5);
stairs(tiempo_eq,salida_eq);

hold off;
title('Comparación de respuestas');
xlabel('Tiempo (s)');
ylabel('Amplitud (V)');
xlim([0 5]);
grid on;
```

Ejercicio 5: Respuesta de la planta en régimen permanente para $\pm 1V$ con perturbación (10 personas = 700 kg).

```
[salida_1V_10P, tiempo_1V_10P] = representar('./Capturas/Fase1/PGA_G4_1V_10P.txt', 1, 'V', 10);
salida(salida_1V_10P, tiempo_1V_10P, Ts, 1, 'V', 10);
[salida_n1V_10P, tiempo_n1V_10P] = representar('./Capturas/Fase1/PGA_G4_n1V_10P.txt', -1, 'V',
10);
salida(salida_n1V_10P, tiempo_n1V_10P, Ts, -1, 'V', 10);
```

Modelo en Simulink:

```
step_value = 1;
pert = 0;
[salida_sim_1V_0P, tiempo_sim_1V_0P] = simulink('modeloPert', '10');

pert = 0.1;
[salida_sim_1V_5P, tiempo_sim_1V_5P] = simulink('modeloPert', '10');

pert = 0.2;
```

```
[salida_sim_1V_10P, tiempo_sim_1V_10P] = simulink('modeloPert', '10');

comparacion(salida_sim_1V_0P,tiempo_sim_1V_0P,salida_sim_1V_5P,tiempo_sim_1V_5P,salida_sim_1V_10
P,tiempo_sim_1V_10P,'Comparación de perturbaciones a entrada positiva')
```

```
step_value = -1;
pert = 0;
[salida_sim_n1V_0P, tiempo_sim_n1V_0P] = simulink('modeloPert', '10');

pert = 0.1;
[salida_sim_n1V_5P, tiempo_sim_n1V_5P] = simulink('modeloPert', '10');

pert = 0.2;
[salida_sim_n1V_10P, tiempo_sim_n1V_10P] = simulink('modeloPert', '10');

comparacion(salida_sim_n1V_0P,tiempo_sim_n1V_0P,salida_sim_n1V_5P,tiempo_sim_n1V_5P,salida_sim_n
1V_10P,tiempo_sim_n1V_10P,'Comparación de perturbaciones a entrada negativa')
```

Anexo 2: Fase 2

Fase 2: Diseño directo del regulador

Nuevos parámetros iniciales:

```
ts = 0.980;
errp = 0.03;
```

Ejercicio 6: Calcular la FdeT del regulador con el método directo Truxal.

Planteamiento del modelo, aplicación de la especificación dinámica:

```
ks = ceil(ts/Ts);
fprintf('La ks es: %.4f.\n', ks)
mod_p = exp(-3/ks);
fprintf('El módulo del polo es: %.4f.\n', mod_p)
```

Aplicación de la especificación estática: errp = 3%.

```
km = (1 - errp) * (1 - 0.8984);
fprintf('La km es: %.4f.\n', km)
```

Modelo final:

```
Mz = zpk([],0.8984, 0.0986, Ts);
```

Ejercicio 7: Dibujar el diagrama de bloques del sistema.

```
%open_system('diagramaBloquesTruxal.slx')
```

Ejercicio 8: Calcular la FdeT del sistema y caracterizar su salida a entradas escalón ± 1 m/s.

FdeT del sistema:

```
Fz = tf(minreal(Mz / (BoG * (1 - Mz))));
Mz = minreal(feedback(series(Fz, BoG), 1));
```

Ejercicio 9: Acción de control y velocidad de la escalera en régimen permanente.

Velocidad de la escalera y Acción de control con 0 personas:

```
escalera(BoG, Fz, 1, 0);
```

```
escalera(BoG, Fz, -1, 0);
```

Velocidad de la escalera y Acción de control con 5 personas:

```
escalera(BoG, Fz, 1, 0.1);
escalera(BoG, Fz, -1, 0.1);
```

Velocidad de la escalera y Acción de control con 10 personas:

```
escalera(BoG, Fz, 1, 0.2);
escalera(BoG, Fz, -1, 0.2);
```

Anexo 3: Fase 3

Fase 3: Validación del diseño directo

Ejercicio 10: Simulación del sistema de control en Simulink.

Entrada +1m/s:

```
step_value = 1;
[salida_sim_1ms_truxal_sin, tiempo_sim_1ms_truxal_sin] = simulink('sistemaTruxalSin', '10');
salida(salida_sim_1ms_truxal_sin, tiempo_sim_1ms_truxal_sin, Ts, 1, 'm/s', 0);
[salida_sim_1ms_truxal_con, tiempo_sim_1ms_truxal_con] = simulink('sistemaTruxalCon', '10');
salida(salida_sim_1ms_truxal_con, tiempo_sim_1ms_truxal_con, Ts, 1, 'm/s', 0);
comparacion(salida_sim_1ms_truxal_sin, tiempo_sim_1ms_truxal_sin, salida_sim_1ms_truxal_con, tiempo_sim_1ms_truxal_con, 'null', 'null', '(Truxal) Comparación de simulaciones 1 m/s')
```

Entrada -1m/s:

```
step_value = -1;
[salida_sim_n1ms_truxal_sin, tiempo_sim_n1ms_truxal_sin] = simulink('sistemaTruxalSin', '10');
salida(salida_sim_n1ms_truxal_sin, tiempo_sim_n1ms_truxal_sin, Ts, -1, 'm/s', 0);
[salida_sim_n1ms_truxal_con, tiempo_sim_n1ms_truxal_con] = simulink('sistemaTruxalCon', '10');
salida(salida_sim_n1ms_truxal_con, tiempo_sim_n1ms_truxal_con, Ts, -1, 'm/s', 0);
comparacion(salida_sim_n1ms_truxal_sin, tiempo_sim_n1ms_truxal_sin, salida_sim_n1ms_truxal_con, tiempo_sim_n1ms_truxal_con, 'null', 'null', '(Truxal) Comparación de simulaciones -1 m/s')
```

Entrada +0.2m/s:

```
step_value = 0.2;
[salida_sim_0_2ms_truxal_sin, tiempo_sim_0_2ms_truxal_sin] = simulink('sistemaTruxalSin', '10');
salida(salida_sim_0_2ms_truxal_sin, tiempo_sim_0_2ms_truxal_sin, Ts, 0.2, 'm/s', 0);
[salida_sim_0_2ms_truxal_con, tiempo_sim_0_2ms_truxal_con] = simulink('sistemaTruxalCon', '10');
salida(salida_sim_0_2ms_truxal_con, tiempo_sim_0_2ms_truxal_con, Ts, 0.2, 'm/s', 0);
comparacion(salida_sim_0_2ms_truxal_sin, tiempo_sim_0_2ms_truxal_sin, salida_sim_0_2ms_truxal_con, tiempo_sim_0_2ms_truxal_con, 'null', 'null', '(Truxal) Comparación de simulaciones 0.2 m/s')
```

Entrada -0.2m/s:

```
step_value = -0.2;
[salida_sim_n0_2ms_truxal_sin, tiempo_sim_n0_2ms_truxal_sin] = simulink('sistemaTruxalSin', '10');
salida(salida_sim_n0_2ms_truxal_sin, tiempo_sim_n0_2ms_truxal_sin, Ts, -0.2, 'm/s', 0);
[salida_sim_n0_2ms_truxal_con, tiempo_sim_n0_2ms_truxal_con] = simulink('sistemaTruxalCon', '10');
salida(salida_sim_n0_2ms_truxal_con, tiempo_sim_n0_2ms_truxal_con, Ts, -0.2, 'm/s', 0);
comparacion(salida_sim_n0_2ms_truxal_sin, tiempo_sim_n0_2ms_truxal_sin, salida_sim_n0_2ms_truxal_con, tiempo_sim_n0_2ms_truxal_con, 'null', 'null', '(Truxal) Comparación de simulaciones -0.2 m/s')
```

Entrada +10m/s:

```
step_value = 10;
[salida_sim_10ms_truxal_sin, tiempo_sim_10ms_truxal_sin] = simulink('sistemaTruxalSin', '10');
salida(salida_sim_10ms_truxal_sin, tiempo_sim_10ms_truxal_sin, Ts, 10, 'm/s', 0);
[salida_sim_10ms_truxal_con, tiempo_sim_10ms_truxal_con] = simulink('sistemaTruxalCon', '10');
salida(salida_sim_10ms_truxal_con, tiempo_sim_10ms_truxal_con, Ts, 10, 'm/s', 0);
comparacion(salida_sim_10ms_truxal_sin, tiempo_sim_10ms_truxal_sin, salida_sim_10ms_truxal_con, tiempo_sim_10ms_truxal_con, 'null', 'null', '(Truxal) Comparación de simulaciones 10 m/s')
```

Entrada -10m/s:

```
step_value = -10;
[salida_sim_n10ms_truxal_sin, tiempo_sim_n10ms_truxal_sin] = simulink('sistemaTruxalSin', '10');
salida(salida_sim_n10ms_truxal_sin, tiempo_sim_n10ms_truxal_sin, Ts, -10, 'm/s', 0);
[salida_sim_n10ms_truxal_con, tiempo_sim_n10ms_truxal_con] = simulink('sistemaTruxalCon', '10');
salida(salida_sim_n10ms_truxal_con, tiempo_sim_n10ms_truxal_con, Ts, -10, 'm/s', 0);
comparacion(salida_sim_n10ms_truxal_sin, tiempo_sim_n10ms_truxal_sin, salida_sim_n10ms_truxal_con, tiempo_sim_n10ms_truxal_con, 'null', 'null', '(Truxal) Comparación de simulaciones -10 m/s')
```

Ejercicio 11: Características de la respuesta para entrada +1 m/s.

```
[salida_1ms_truxal, tiempo_1ms_truxal] = representar('./Capturas/Fase3/PGA_G4_1ms_0P.txt', 1, 'm/s', 0);
[km_1ms_truxal, Tau_1ms_truxal, salida_1ms_truxal, tiempo_1ms_truxal] =
salida(salida_1ms_truxal, tiempo_1ms_truxal, Ts, 1, 'm/s', 0);
fprintf('km con entrada escalón de 1V: %.4f.\n', km_1ms_truxal)
comparacion(salida_1ms_truxal, tiempo_1ms_truxal, salida_sim_1ms_truxal_sin, tiempo_sim_1ms_truxal_sin, salida_sim_1ms_truxal_con, tiempo_sim_1ms_truxal_con, '(Truxal) Comparación de respuestas a entrada positiva')
```

Ejercicio 12: Características de la respuesta para entrada -1 m/s.

```
[salida_n1ms_truxal, tiempo_n1ms_truxal] = representar('./Capturas/Fase3/PGA_G4_n1ms_0P.txt', -1, 'm/s', 0);
[km_n1ms_truxal, Tau_n1ms_truxal, salida_n1ms_truxal, tiempo_n1ms_truxal] =
salida(salida_n1ms_truxal, tiempo_n1ms_truxal, Ts, -1, 'm/s', 0);
fprintf('km con entrada escalón de 1V: %.4f.\n', km_n1ms_truxal)
comparacion(salida_n1ms_truxal, tiempo_n1ms_truxal, salida_sim_n1ms_truxal_sin, tiempo_sim_n1ms_truxal_sin, salida_sim_n1ms_truxal_con, tiempo_sim_n1ms_truxal_con, '(Truxal) Comparación de respuestas a entrada negativa')
```

Ejercicio 13: Régimen permanente de la acción de control y la velocidad de la escalera.**Sin carga:**

```
[salida_1ms_0P_truxal, tiempo_1ms_0P_truxal] = representar('./Capturas/Fase3/PGA_G4_1ms_0P.txt', 1, 'm/s', 0);
[km_1ms_0P_truxal, Tau_1ms_0P_truxal, salida_1ms_0P_truxal, tiempo_1ms_0P_truxal] =
salida(salida_1ms_0P_truxal, tiempo_1ms_0P_truxal, Ts, 1, 'm/s', 0);
[salida_n1ms_0P_truxal, tiempo_n1ms_0P_truxal] =
representar('./Capturas/Fase3/PGA_G4_n1ms_0P.txt', -1, 'm/s', 0);
[km_n1ms_0P_truxal, Tau_n1ms_0P_truxal, salida_n1ms_0P_truxal, tiempo_n1ms_0P_truxal] =
salida(salida_n1ms_0P_truxal, tiempo_n1ms_0P_truxal, Ts, -1, 'm/s', 0);
```

Con carga - 5 personas:

```
[salida_1ms_5P_truxal, tiempo_1ms_5P_truxal] = representar('./Capturas/Fase3/PGA_G4_1ms_5P.txt', 1, 'm/s', 5);
[km_1ms_5P_truxal, Tau_1ms_5P_truxal, salida_1ms_5P_truxal, tiempo_1ms_5P_truxal] =
salida(salida_1ms_5P_truxal, tiempo_1ms_5P_truxal, Ts, 1, 'm/s', 5);
[salida_n1ms_5P_truxal, tiempo_n1ms_5P_truxal] =
representar('./Capturas/Fase3/PGA_G4_n1ms_5P.txt', -1, 'm/s', 5);
[km_n1ms_5P_truxal, Tau_n1ms_5P_truxal, salida_n1ms_5P_truxal, tiempo_n1ms_5P_truxal] =
salida(salida_n1ms_5P_truxal, tiempo_n1ms_5P_truxal, Ts, -1, 'm/s', 5);
```

Con carga - 10 personas:

```
[salida_1ms_10P_truxal, tiempo_1ms_10P_truxal] =
representar('./Capturas/Fase3/PGA_G4_1ms_10P.txt', 1, 'm/s', 10);
[km_1ms_10P_truxal, Tau_1ms_10P_truxal, salida_1ms_10P_truxal, tiempo_1ms_10P_truxal] =
salida(salida_1ms_10P_truxal, tiempo_1ms_10P_truxal, Ts, 1, 'm/s', 10);
[salida_n1ms_10P_truxal, tiempo_n1ms_10P_truxal] =
representar('./Capturas/Fase3/PGA_G4_n1ms_10P.txt', -1, 'm/s', 10);
[km_n1ms_10P_truxal, Tau_n1ms_10P_truxal, salida_n1ms_10P_truxal, tiempo_n1ms_10P_truxal] =
salida(salida_n1ms_10P_truxal, tiempo_n1ms_10P_truxal, Ts, -1, 'm/s', 10);
```

Comparación:

```
comparacion(salida_1ms_0P_truxal, tiempo_1ms_0P_truxal, salida_1ms_5P_truxal, tiempo_1ms_5P_truxal,
salida_1ms_10P_truxal, tiempo_1ms_10P_truxal, '(Truxal) Comparación de perturbaciones a entrada
positiva')
comparacion(salida_n1ms_0P_truxal, tiempo_n1ms_0P_truxal, salida_n1ms_5P_truxal, tiempo_n1ms_5P_tru-
xal, salida_n1ms_10P_truxal, tiempo_n1ms_10P_truxal, '(Truxal) Comparación de perturbaciones a
entrada negativa')
```

Anexo 4: Fase 4

Fase 4: Diseño del regulador PID

Nuevos parámetros iniciales:

```
ts = 1.12;
errp = 0;
```

Ejercicio 15: Calcular la FdeT del regulador PID.

Módulo del polo buscado:

```
ks = ceil(ts/Ts);
mod_p = exp(-3/ks);
fprintf('El módulo del polo es: %.4f.\n', mod_p)
```

Regulador P:

```
rlocus(BoG)
```

Regulador I:

```
Fz = zpk(-1, [0.8752 1], 1, Ts);
rlocus(Fz)
```

Regulador PI:

```
K = 3.5657;
Ti = (Ts/2) * ((1+0.8752)/(1-0.8752));
Kp = K * (2*Ti)/(2*Ti+Ts);
Ki = Kp/Ti;
q0 = Kp + (Ki*Ts)/2;
q1 = -Kp + (Ki*Ts)/2;
Fz = minreal(tf([q0 q1 0], [1 -1 0], Ts));
```

Ejercicio 16: Calcular la FdeT del sistema y obtener los polos.

```
Mz = minreal(feedback(series(Fz, BoG), 1));
pzplot(Mz)
p = pole(Mz);
fprintf('El polo es: %.4f.\n', p)
```

Ejercicio 18: Acción de control y velocidad de la escalera en régimen permanente para entradas escalón $\pm 1 \text{ m/s}$.

Velocidad de la escalera y Acción de control con 0 personas:

```
escalera(BoG, Fz, 1, 0);  
escalera(BoG, Fz, -1, 0);
```

Velocidad de la escalera y Acción de control con 5 personas:

```
escalera(BoG, Fz, 1, 0.1);  
escalera(BoG, Fz, -1, 0.1);
```

Velocidad de la escalera y Acción de control con 10 personas:

```
escalera(BoG, Fz, 1, 0.2);  
escalera(BoG, Fz, -1, 0.2);
```

Anexo 5: Fase 5

Fase 5: Validación del regulador PID

Ejercicio 19: Simulación del sistema de control en Simulink.

Entrada +1m/s:

```
step_value = 1;
[salida_sim_1ms_PID_sin, tiempo_sim_1ms_PID_sin] = simulink('sistemaPIDSin', '10');
[km_sim_1ms_PID_sin, Tau_sim_1ms_PID_sin, salida_sim_1ms_PID_sin, tiempo_sim_1ms_PID_sin] =
salida(salida_sim_1ms_PID_sin, tiempo_sim_1ms_PID_sin, Ts, 1, 'm/s', 0);
[salida_sim_1ms_PID_con, tiempo_sim_1ms_PID_con] = simulink('sistemaPIDCon', '10');
[km_sim_1ms_PID_con, Tau_sim_1ms_PID_con, salida_sim_1ms_PID_con, tiempo_sim_1ms_PID_con] =
salida(salida_sim_1ms_PID_con, tiempo_sim_1ms_PID_con, Ts, 1, 'm/s', 0);
comparacion(salida_sim_1ms_PID_sin, tiempo_sim_1ms_PID_sin, salida_sim_1ms_PID_con, tiempo_sim_1ms_PID_con, 'null', 'null', '(PID) Comparación de simulaciones 1 m/s')
```

Entrada -1m/s:

```
step_value = -1;
[salida_sim_n1ms_PID_sin, tiempo_sim_n1ms_PID_sin] = simulink('sistemaPIDSin', '10');
[km_sim_n1ms_PID_sin, Tau_sim_n1ms_PID_sin, salida_sim_n1ms_PID_sin, tiempo_sim_n1ms_PID_sin] =
salida(salida_sim_n1ms_PID_sin, tiempo_sim_n1ms_PID_sin, Ts, -1, 'm/s', 0);
[salida_siz_n1ms_PID_con, tiempo_sim_n1ms_PID_con] = simulink('sistemaPIDCon', '10');
[km_sim_n1ms_PID_con, Tau_sim_n1ms_PID_con, salida_sim_n1ms_PID_con, tiempo_sim_n1ms_PID_con] =
salida(salida_sim_n1ms_PID_con, tiempo_sim_n1ms_PID_con, Ts, -1, 'm/s', 0);
comparacion(salida_sim_n1ms_PID_sin, tiempo_sim_n1ms_PID_sin, salida_sim_n1ms_PID_con, tiempo_sim_n1ms_PID_con, 'null', 'null', '(PID) Comparación de simulaciones -1 m/s')
```

Entrada +0.2m/s:

```
step_value = 0.2;
[salida_sim_0_2ms_PID_sin, tiempo_sim_0_2ms_PID_sin] = simulink('sistemaPIDSin', '10');
[km_sim_0_2ms_PID_sin, Tau_sim_0_2ms_PID_sin, salida_sim_0_2ms_PID_sin,
tiempo_sim_0_2ms_PID_sin] = salida(salida_sim_0_2ms_PID_sin, tiempo_sim_0_2ms_PID_sin, Ts, 0.2,
'm/s', 0);
[salida_sim_0_2ms_PID_con, tiempo_sim_0_2ms_PID_con] = simulink('sistemaPIDCon', '10');
[km_sim_0_2ms_PID_con, Tau_sim_0_2ms_PID_con, salida_sim_0_2ms_PID_con,
tiempo_sim_0_2ms_PID_con] = salida(salida_sim_0_2ms_PID_con, tiempo_sim_0_2ms_PID_con, Ts, 0.2,
'm/s', 0);
comparacion(salida_sim_0_2ms_PID_sin, tiempo_sim_0_2ms_PID_sin, salida_sim_0_2ms_PID_con, tiempo_sim_0_2ms_PID_con, 'null', 'null', '(PID) Comparación de simulaciones 0.2 m/s')
```

Entrada -0.2m/s:

```
step_value = -0.2;
[salida_sim_n0_2ms_PID_sin, tiempo_sim_n0_2ms_PID_sin] = simulink('sistemaPIDSin', '10');
```

```
[km_sim_n0_2ms_PID_sin, Tau_sim_n0_2ms_PID_sin, salida_sim_n0_2ms_PID_sin,
tiempo_sim_n0_2ms_PID_sin] = salida(salida_sim_n0_2ms_PID_sin, tiempo_sim_n0_2ms_PID_sin, Ts, -
0.2, 'm/s', 0);
[salida_sim_n0_2ms_PID_con, tiempo_sim_n0_2ms_PID_con] = simulink('sistemaPIDCon', '10');
[km_sim_n0_2ms_PID_con, Tau_sim_n0_2ms_PID_con, salida_sim_n0_2ms_PID_con,
tiempo_sim_n0_2ms_PID_con] = salida(salida_sim_n0_2ms_PID_con, tiempo_sim_n0_2ms_PID_con, Ts, -
0.2, 'm/s', 0);
comparacion(salida_sim_n0_2ms_PID_sin, tiempo_sim_n0_2ms_PID_sin, salida_sim_n0_2ms_PID_con, tiempo_
sim_n0_2ms_PID_con, 'null', 'null', '(PID) Comparación de simulaciones -0.2 m/s')
```

Entrada +10m/s:

```
step_value = 10;
[salida_sim_10ms_PID_sin, tiempo_sim_10ms_PID_sin] = simulink('sistemaPIDSin', '10');
[km_sim_10ms_PID_sin, Tau_sim_10ms_PID_sin, salida_sim_10ms_PID_sin, tiempo_sim_10ms_PID_sin] =
salida(salida_sim_10ms_PID_sin, tiempo_sim_10ms_PID_sin, Ts, 10, 'm/s', 0);
[salida_sim_10ms_PID_con, tiempo_sim_10ms_PID_con] = simulink('sistemaPIDCon', '10');
[km_sim_10ms_PID_con, Tau_sim_10ms_PID_con, salida_sim_10ms_PID_con, tiempo_sim_10ms_PID_con] =
salida(salida_sim_10ms_PID_con, tiempo_sim_10ms_PID_con, Ts, 10, 'm/s', 0);
comparacion(salida_sim_10ms_PID_sin, tiempo_sim_10ms_PID_sin, salida_sim_10ms_PID_con, tiempo_sim_1
0ms_PID_con, 'null', 'null', '(PID) Comparación de simulaciones 10 m/s')
```

Entrada -10m/s:

```
step_value = -10;
[salida_sim_n10ms_PID_sin, tiempo_sim_n10ms_PID_sin] = simulink('sistemaPIDSin', '10');
[km_sim_n10ms_PID_sin, Tau_sim_n10ms_PID_sin, salida_sim_n10ms_PID_sin,
tiempo_sim_n10ms_PID_sin] = salida(salida_sim_n10ms_PID_sin, tiempo_sim_n10ms_PID_sin, Ts, -10,
'm/s', 0);
[salida_sim_n10ms_PID_con, tiempo_sim_n10ms_PID_con] = simulink('sistemaPIDCon', '10');
[km_sim_n10ms_PID_con, Tau_sim_n10ms_PID_con, salida_sim_n10ms_PID_con,
tiempo_sim_n10ms_PID_con] = salida(salida_sim_n10ms_PID_con, tiempo_sim_n10ms_PID_con, Ts, -10,
'm/s', 0);
comparacion(salida_sim_n10ms_PID_sin, tiempo_sim_n10ms_PID_sin, salida_sim_n10ms_PID_con, tiempo_si
m_n10ms_PID_con, 'null', 'null', '(PID) Comparación de simulaciones -10 m/s')
```

Ejercicio 20: Características de la respuesta para entrada +1 m/s.

```
[salida_1ms_PID, tiempo_1ms_PID] = representar('./Capturas/Fase5/PGA_G4_1ms_0P.txt', 1, 'm/s',
0);
[km_1ms_PID, Tau_1ms_PID, salida_1ms_PID, tiempo_1ms_PID] = salida(salida_1ms_PID,
tiempo_1ms_PID, Ts, 1, 'm/s', 0);
fprintf('(PID) km con entrada escalón de 1m/s: %.4f.\n', km_1ms_PID)
comparacion(salida_1ms_PID, tiempo_1ms_PID, salida_sim_1ms_PID_sin, tiempo_sim_1ms_PID_sin, salida_s
im_1ms_PID_con, tiempo_sim_1ms_PID_con, '(PID) Comparación de respuestas a entrada positiva')
```

Ejercicio 21: Características de la respuesta para entrada -1 m/s.

```
[salida_n1ms_PID, tiempo_n1ms_PID] = representar('./Capturas/Fase5/PGA_G4_n1ms_0P.txt', -1,
'm/s', 0);
[km_n1ms_PID, Tau_n1ms_PID, salida_n1ms_PID, tiempo_n1ms_PID] = salida(salida_n1ms_PID,
tiempo_n1ms_PID, Ts, -1, 'm/s', 0);
fprintf('(PID) km con entrada escalón de -1m/s: %.4f.\n', km_n1ms_PID)
comparacion(salida_n1ms_PID, tiempo_n1ms_PID, salida_sim_n1ms_PID_sin, tiempo_sim_n1ms_PID_sin, salida_sim_n1ms_PID_con, tiempo_sim_n1ms_PID_con, '(PID) Comparación de respuestas a entrada negativa')
```

Ejercicio 22: Régimen permanente de la acción de control y la velocidad de la escalera.**Sin carga:**

```
[salida_1ms_0P_PID, tiempo_1ms_0P_PID] = representar('./Capturas/Fase5/PGA_G4_1ms_0P.txt', 1,
'm/s', 0);
[km_1ms_0P_PID, Tau_1ms_0P_PID, salida_1ms_0P_PID, tiempo_1ms_0P_PID] =
salida(salida_1ms_0P_PID, tiempo_1ms_0P_PID, Ts, 1, 'm/s', 0);
[salida_n1ms_0P_PID, tiempo_n1ms_0P_PID] = representar('./Capturas/Fase5/PGA_G4_n1ms_0P.txt', -1,
'm/s', 0);
[km_n1ms_0P_PID, Tau_n1ms_0P_PID, salida_n1ms_0P_PID, tiempo_n1ms_0P_PID] =
salida(salida_n1ms_0P_PID, tiempo_n1ms_0P_PID, Ts, -1, 'm/s', 0);
```

Con carga - 5 personas:

```
[salida_1ms_5P_PID, tiempo_1ms_5P_PID] = representar('./Capturas/Fase5/PGA_G4_1ms_5P.txt', 1,
'm/s', 5);
[km_1ms_5P_PID, Tau_1ms_5P_PID, salida_1ms_5P_PID, tiempo_1ms_5P_PID] =
salida(salida_1ms_5P_PID, tiempo_1ms_5P_PID, Ts, 1, 'm/s', 5);
[salida_n1ms_5P_PID, tiempo_n1ms_5P_PID] = representar('./Capturas/Fase5/PGA_G4_n1ms_5P.txt', -1,
'm/s', 5);
[km_n1ms_5P_PID, Tau_n1ms_5P_PID, salida_n1ms_5P_PID, tiempo_n1ms_5P_PID] =
salida(salida_n1ms_5P_PID, tiempo_n1ms_5P_PID, Ts, -1, 'm/s', 5);
```

Con carga - 10 personas:

```
[salida_1ms_10P_PID, tiempo_1ms_10P_PID] = representar('./Capturas/Fase5/PGA_G4_1ms_10P.txt', 1,
'm/s', 10);
[km_1ms_10P_PID, Tau_1ms_10P_PID, salida_1ms_10P_PID, tiempo_1ms_10P_PID] =
salida(salida_1ms_10P_PID, tiempo_1ms_10P_PID, Ts, 1, 'm/s', 10);
[salida_n1ms_10P_PID, tiempo_n1ms_10P_PID] = representar('./Capturas/Fase5/PGA_G4_n1ms_10P.txt', -1,
'm/s', 10);
[km_n1ms_10P_PID, Tau_n1ms_10P_PID, salida_n1ms_10P_PID, tiempo_n1ms_10P_PID] =
salida(salida_n1ms_10P_PID, tiempo_n1ms_10P_PID, Ts, -1, 'm/s', 10);
```

Comparación:

```
comparacion(salida_1ms_0P_PID,tiempo_1ms_0P_PID,salida_1ms_5P_PID,tiempo_1ms_5P_PID,salida_1ms_1
0P_PID,tiempo_1ms_10P_PID,'(PID) Comparación de perturbaciones a entrada positiva')
comparacion(salida_n1ms_0P_PID,tiempo_n1ms_0P_PID,salida_n1ms_5P_PID,tiempo_n1ms_5P_PID,salida_n
1ms_10P_PID,tiempo_n1ms_10P_PID,'(PID) Comparación de perturbaciones a entrada negativa')
```

Anexo 6: Funciones

representar.m:

```
%{
@file representar.m
@brief Funcion para representar datos de entrada y salida desde un archivo.

Esta funcion carga datos desde un archivo especificado y representa los datos de entrada y salida
en una gráfica. También añade títulos y etiquetas apropiadas a la gráfica basándose en los
parámetros proporcionados.

@param archivo La ruta al archivo que contiene los datos a graficar.
@param valor_entrada El valor del escalón de entrada.
@param unidades Las unidades de los datos de entrada y salida.
@param pert El número de personas (perturbación) que afectan el sistema. Si pert es 0,
    el título no incluirá el número de personas.

@return salida Los datos de salida extraídos del archivo.
@return tiempo Los datos de tiempo extraídos del archivo.
%}
function [salida,tiempo] = representar(archivo, valor_entrada, unidades, pert)
figure;

datos = load(archivo); % Cargar los datos del archivo
tiempo = datos(:,1); % Asignar la primera columna de datos a la variable tiempo
entrada = datos(:,2); % Asignar la segunda columna de datos a la variable entrada
salida = datos(:,3); % Asignar la tercera columna de datos a la variable salida

plot(tiempo,entrada)
hold on
plot(tiempo,salida)
hold off

if(pert == 0)
    title(sprintf('Entrada y salida con escalón %.2f %s', valor_entrada, unidades))
    xlabel('Tiempo (s)')
    ylabel(sprintf('Amplitud (%s)', unidades))
    grid on;
else
    title(sprintf('Entrada y salida con escalón %.2f %s con %d personas', valor_entrada,
unidades, pert))
    xlabel('Tiempo (s)')
    ylabel(sprintf('Amplitud (%s)', unidades))
    grid on;
end
end
```

salida.m:

```
%{
@file salida.m
@brief Esta función calcula la respuesta del sistema a una entrada escalón y representa la respuesta.

@param salida Vector que contiene la señal de salida.
@param tiempo Vector que contiene los instantes de tiempo correspondientes a la señal de salida.
@param Ts Tiempo de muestreo.
@param valor_entrada Valor de la entrada escalón.
@param unidades Unidades del valor de entrada.
@param pert Número de personas perturbando el sistema (0 si no hay perturbación).

@return km Ganancia estática del sistema.
@return Tau Constante de tiempo del sistema.
@return salida_interes Señal de salida dentro del tiempo de interés.
@return tiempo_interes Instantes de tiempo dentro del tiempo de interés.

La función realiza los siguientes pasos:
1. Encuentra el tiempo inicial y final donde la salida es diferente de cero.
2. Extrae el tiempo y la señal de salida dentro del tiempo de interés.
3. Limita el tiempo de interés a los primeros 5 segundos.
4. Grafica la señal de salida y una línea horizontal en el 63.2% del valor final de la salida.
5. Calcula la ganancia estática (km) y la constante de tiempo (Tau) del sistema.
%}
function [km,Tau,salida_interes,tiempo_interes] = salida(salida, tiempo, Ts, valor_entrada, unidades, pert)

figure;

t_ini = tiempo(find(salida ~= 0, 1)); % Encontrar el tiempo inicial donde la salida no es cero
t_fin = tiempo(end); % Asignar el tiempo final
x_ini = find(tiempo==t_ini); % Encontrar el índice del tiempo inicial
x_fin = find(tiempo==t_fin); % Encontrar el índice del tiempo final
tiempo_interes = tiempo(x_ini:x_fin)-tiempo(x_ini); % Extraer el intervalo de tiempo de interés
salida_interes = salida(x_ini:x_fin); % Extraer la salida correspondiente al intervalo de tiempo de interés

% Limitar el tiempo de interés a los primeros 5 segundos
limite = find(tiempo_interes <= 5);
tiempo_interes = tiempo_interes(limite);
salida_interes = salida_interes(limite);

plot(tiempo_interes,salida_interes)
hold on
plot([0 tiempo_interes(end)], [1 1]*salida_interes(end)*0.632, 'r:') % Dibujar una línea horizontal en el 63.2% del valor final de la salida
hold off
xlim([0 5]);

if(pert == 0)
    title(sprintf('Respuesta a la entrada escalón %.2f %s', valor_entrada, unidades))
    xlabel('Tiempo (s)')
    ylabel(sprintf('Amplitud (%s)', unidades))
    grid on;
else
    title(sprintf('Respuesta a la entrada escalón %.2f %s con %d personas', valor_entrada, unidades, pert))
    xlabel('Tiempo (s)')
    ylabel(sprintf('Amplitud (%s)', unidades))
    grid on;
end

muestras = round(size(find(salida_interes >= 0))/2); % Calcular el número de muestras para el promedio
c_inf = mean(salida_interes(end-muestras, end)); % Calcular el valor final de la salida
muestra = find(abs(salida_interes) >= abs(c_inf)*0.632, 1); % Encontrar la muestra donde la salida alcanza el 63.2% del valor final
Tau = (muestra+muestra-1)/2*Ts; % Calcular la constante de tiempo Tau
km = c_inf/valor_entrada; % Calcular la ganancia estática km

end
```

comparacion.m:

```
%{
  Funcion para comparar y representar tres conjuntos de datos.

@param salida1 Array de datos de salida de la primera señal.
@param tiempo1 Array de datos de tiempo para la primera señal.
@param salida2 Array de datos de salida para la segunda señal.
@param tiempo2 Array de datos de tiempo para la segunda señal.
@param salida3 Array de datos de salida para la tercera señal. Si es 'null', no se representará.
@param tiempo3 Array de datos de tiempo para la tercera señal. Si es 'null', no se representará.
@param texto Texto del título para la gráfica.

Esta función crea una figura y grafica los conjuntos de datos proporcionados en el mismo gráfico.
También establece el título, las etiquetas para los ejes x e y, los límites para el eje x y
habilita la cuadrícula.

}

function [] = comparacion(salida1,tiempo1,salida2,tiempo2,salida3,tiempo3,Texto)

figure;

plot(tiempo1,salida1);
hold on;

plot(tiempo2,salida2);

if (salida3 ~= 'null')
    plot(tiempo3,salida3);
end

hold off;
title(Texto);
xlabel('Tiempo (s)');
ylabel('Amplitud (m/s)');
xlim([0 5]);
grid on;

end
```

escalera.m:

```
%{  
    Funcion para calcular la salida y la señal de control para una entrada escalon y una perturbacion  
dadas.  
  
    @param BoG Funcion de transferencia del sistema.  
    @param Fz Funcion de transferencia del regulador.  
    @param entrada Valor de la entrada escalon en m/s.  
    @param pert Valor de la perturbacion en m/s.  
    @return [c_inf, a_inf] Una tupla donde c_inf es la salida con entrada escalon y perturbacion en  
m/s,  
            y a_inf es la señal de control con entrada escalon y perturbacion en V.
```

La funcion calcula lo siguiente:

- g_inf: La ganancia DC del sistema.
- Mz_esc: La funcion de transferencia en lazo cerrado para la entrada escalon.
- Mz_pert: La funcion de transferencia en lazo cerrado para la perturbacion.
- c_inf: La salida con entrada escalón y perturbacion.
- a_inf: La señal de control con entrada escalón y perturbacion.

La funcion tambien imprime la salida calculada y la señal de control.

Ejemplo de uso:

```
[c_inf, a_inf] = escalera(BoG, Fz, entrada, pert);  
%}  
function [c_inf, a_inf] = escalera(BoG, Fz, entrada, pert)  
  
    g_inf = dcgain(BoG); % Calcular la ganancia del sistema  
    Mz_esc = minreal(feedback(series(Fz, BoG), 1)); % Calcular la función de transferencia en lazo  
    % cerrado para la entrada escalón  
    Mz_pert = minreal(1/(1+series(Fz, BoG))); % Calcular la función de transferencia en lazo  
    % cerrado para la perturbación  
  
    c_inf = entrada * dcgain(Mz_esc) - pert * dcgain(Mz_pert); % Calcular la salida con entrada  
    % escalón y perturbación  
    a_inf = (c_inf + pert)/g_inf; % Calcular la señal de control con  
    % entrada escalón y perturbación  
  
    % Imprimir la salida calculada  
    fprintf('La salida con entrada escalón %d m/s y perturbación %.1f m/s es: %.4f m/s.\n',  
    entrada, pert, c_inf)  
    % Imprimir la señal de control calculada  
    fprintf('La señal de acción con entrada escalón %d m/s y perturbación %.1f m/s es: %.4f V.\n',  
    entrada, pert, a_inf)  
  
end
```

simulink.m:

```
%{
    Función para simular un modelo de Simulink y extraer la salida de interés.

@param archivo La ruta al archivo del modelo de Simulink.
@param stop El tiempo de parada para la simulación.
@return salida_interes Los datos de salida de interés de la simulación.
@return tiempo_interes Los datos de tiempo de interés de la simulación.

Esta función realiza los siguientes pasos:
1. Carga el modelo de Simulink especificado.
2. Establece el tiempo de parada de la simulación.
3. Ejecuta la simulación.
4. Extrae los datos de tiempo y salida de los resultados de la simulación.
5. Encuentra el tiempo inicial donde la salida no es cero.
6. Extrae los datos de tiempo y salida de interés basados en el tiempo inicial.
7. Limita el tiempo de interés a los primeros 5 segundos.
8. Cierra el modelo de Simulink sin guardar cambios.

%}
function [salida_interes,tiempo_interes] = simulink(archivo, stop)
figure;

load_system(archivo); % Cargar el sistema de Simulink especificado por el
archivo
set_param(archivo, 'StopTime', stop); % Establecer el tiempo de simulación

simulacion = sim(archivo); % Ejecutar la simulación

tiempo = simulacion.out(:,1); % Extraer los datos de tiempo de la simulación
salida = simulacion.out(:,2); % Extraer los datos de salida de la simulación
t_ini = tiempo(find(salida ~= 0, 1)); % Encontrar el tiempo inicial donde la salida no es
cero

% Si se encuentra un tiempo inicial no vacío
if(t_ini~=[])
    t_fin = tiempo(end); % Asignar el tiempo final
    x_ini = find(tiempo==t_ini); % Encontrar el índice del tiempo
inicial
    x_fin = find(tiempo==t_fin); % Encontrar el índice del tiempo final
    tiempo_interes = tiempo(x_ini:x_fin)-tiempo(x_ini); % Extraer el intervalo de tiempo de
interés
    salida_interes = salida(x_ini:x_fin); % Extraer la salida correspondiente al
intervalo de tiempo de interés
else
    % Si no se encuentra un tiempo inicial, usar todo el tiempo y salida
    tiempo_interes = tiempo;
    salida_interes = salida;
end

% Limitar el tiempo de interés a los primeros 5 segundos
limite = find(tiempo_interes <= 5);
tiempo_interes = tiempo_interes(limite);
salida_interes = salida_interes(limite);

% Cerrar el sistema de Simulink sin guardar cambios
close_system(archivo, 0);

end
```

Anexo 7: Código fuente del regulador Truxal

```
//-----
// ÚNICAS VARIABLES DISPONIBLES PARA UTILIZAR:
int y0; //Salida del controlador en el instante k
int y1=0; //Salida del controlador en el instante k-1
int y2=0; //Salida del controlador en el instante k-2
int y3=0; //Salida del controlador en el instante k-3
int y4=0; //Salida del controlador en el instante k-4

int x0; //Entrada al controlador en el instante k
int x1=0; //Entrada al controlador en el instante k-1
int x2=0; //Entrada al controlador en el instante k-2
int x3=0; //Entrada al controlador en el instante k-3

int e;
//-----

//-----
//CÓDIGO EJECUTABLE:
while (1==1) {
    //Error del sistema:
    e=entrada-salida; //entrada y salida son variables globales

    ///////////////////////////////
    //INICIO CÓDIGO DE ALUMNO

    x0 = e;
    y0 = 0.9960*y1 + 3.9283*x0 - 3.4382*x1;
    y1 = y0;
    x1 = x0;

    //FIN CÓDIGO DE ALUMNO
    //////////////////////////

    //Retardo de ejecución del bucle:
    delay(T); //T es una variable global
}
//-----
```

Anexo 8: Código fuente del regulador PID

```
//-----
// ÚNICAS VARIABLES DISPONIBLES PARA UTILIZAR:
int y0; //Salida del controlador en el instante k
int y1=0; //Salida del controlador en el instante k-1
int y2=0; //Salida del controlador en el instante k-2
int y3=0; //Salida del controlador en el instante k-3
int y4=0; //Salida del controlador en el instante k-4

int x0; //Entrada al controlador en el instante k
int x1=0; //Entrada al controlador en el instante k-1
int x2=0; //Entrada al controlador en el instante k-2
int x3=0; //Entrada al controlador en el instante k-3

int e;
//-----

//-----
//CÓDIGO EJECUTABLE:
while (1==1) {
    //Error del sistema:
    e=entrada-salida; //entrada y salida son variables globales

    /////////////////////////
    //INICIO CÓDIGO DE ALUMNO

    x0 = e;
    y0 = 1*y1 + 3.5657*x0 - 3.1207*x1;
    y1 = y0;
    x1 = x0;

    //FIN CÓDIGO DE ALUMNO
    /////////////////////////

    //Retardo de ejecución del bucle:
    delay(T); //T es una variable global
}
//-----
```