



**Escuela Superior
de Ingeniería y Tecnología**
Universidad de La Laguna

Trabajo fin de grado

Sobre el efecto de los ruidos en sistemas con controladores de orden no entero

Autores:

Luis Alejandro Hernández Somaza

Miriam Rizo García

Tutor:

Leopoldo Acosta Sánchez

SEPTIEMBRE, 2021



AGRADECIMIENTOS

A nuestros padres, por habernos dado las primeras oportunidades de la vida y luego permitirnos tomar nuestro propio camino, siempre llevando de la mano su amor, sus consejos y sus enseñanzas. Gracias principalmente a ellos por nunca dejarnos caer y ante las dificultades siempre ser esa persona que necesitábamos. Gracias por ser nuestro pilar de vida.

A nuestros abuelos, por siempre darnos esos empujoncitos pequeños de ánimo que ante un mal día nos hacían recuperar la sonrisa. Gracias a ellos por todas sus historias y experiencias únicas pero sobre todo por muchas veces ser nuestro padre y madre.

A nuestros tíos/tías, por ser esa voz de conciencia que nos daba el último consejo en muchos momentos.

A mi hermana y primos, por ser parte fundamental de mi inspiración. Que este trabajo les sirva siempre de ejemplo de que todo en la vida se puede lograr si te lo propones y que aunque siempre te puedes encontrar piedras en el camino, con constancia y esfuerzo puedes siempre llegar a tus sueños y metas.

A nuestro tutor y profesor Leopoldo Acosta, por no sólo ser pieza fundamental en el desarrollo de nuestros conocimientos, sino también por acompañarnos siempre en todo el trayecto hacia la obtención del título. Gracias por su paciencia durante el desarrollo del TFG y por su disposición, buena voluntad y ayuda. Gracias por más que ser un profesor, ha sido un gran amigo.

Esto es para tí Papá, gracias por siempre acompañarme.

Para ustedes abuelas, por ser mi luz que nunca se apagará.





ÍNDICE:

Introducción.	1
1.1. Objetivo.	1
1.2. Abstract.	1
2. Metodología.	1
2.1. Fundamentos de los controladores PID.	1
2.2. PID Clásico.	2
2.3. PID Fraccionario.	3
3. Algoritmo y simulaciones	6
3.1. Estudio control con PID Clásico.	8
3.2. Estudio control con PID fraccionario.	9
3.3. Algoritmo.	10
4. Resultados (Alpha = -0.9 y Beta = 0.9)	13
4.1. 100 simulaciones sin Ruido de perturbación.	14
4.1.1. Salidas.	14
4.1.2. Densidad de probabilidad.	15
4.1.3. Media.	16
4.1.4. Desviación.	17
4.2. 100 simulaciones con Ruido de perturbación.	18
4.2.1. Salidas.	18
4.2.2. Densidad de probabilidad.	19
4.2.3. Media.	20
4.2.4. Desviación.	21
4.3. 500 simulaciones sin Ruido de perturbación.	22
4.3.1. Salidas.	22
4.3.2. Densidad de probabilidad.	23
4.3.3. Media.	24
4.3.4. Desviación.	25
4.4. 500 simulaciones con Ruido de perturbación.	26
4.4.1. Salidas.	26
4.4.2. Densidad de probabilidad.	28



4.4.3. Media.	29
4.4.4. Desviación.	30
5. Resultados (Alpha = -0.8 y Beta = 0.8)	31
5.1. 100 simulaciones sin Ruido de perturbación.	31
5.1.1. Salidas.	31
5.1.2. Densidad de probabilidad.	32
5.1.3. Media.	33
5.1.4. Desviación.	34
5.2. 100 simulaciones con Ruido de perturbación.	35
5.2.1. Salidas.	35
5.2.2. Densidad de probabilidad.	36
5.2.3. Media.	37
5.2.4. Desviación.	38
5.3. 500 simulaciones sin Ruido de perturbación.	39
5.3.1. Salidas.	39
5.3.2. Densidad de probabilidad.	40
5.3.3. Media.	41
5.3.4. Desviación.	42
5.4. 500 simulaciones con Ruido de perturbación.	43
5.4.1. Salidas.	43
5.4.2. Densidad de probabilidad.	44
5.4.3. Media.	45
5.4.4. Desviación.	46
6. Resultados (Alpha = -0.6 y Beta = 0.6)	47
6.1. 100 simulaciones sin Ruido de perturbación.	47
6.1.1. Salidas.	47
6.1.2. Densidad de probabilidad.	48
6.1.3. Media.	49
6.1.4. Desviación.	50
6.2. 100 simulaciones con Ruido de perturbación.	51
6.2.1. Salidas.	51



6.2.2. Densidad de probabilidad.	52
6.2.3. Media.	53
6.2.4. Desviación.	54
6.3. 500 simulaciones sin Ruido de perturbación.	55
6.3.1. Salidas.	55
6.3.2. Densidad de probabilidad.	56
6.3.3. Media.	56
6.3.4. Desviación.	57
6.4. 500 simulaciones con Ruido de perturbación.	58
6.4.1. Salidas.	58
6.4.2. Densidad de probabilidad.	59
6.4.3. Media.	60
6.4.4. Desviación.	61
7. Resultados (Alpha = -0.4 y Beta = 0.4)	62
7.1. 100 simulaciones sin Ruido de perturbación.	62
7.1.1. Salidas.	62
7.1.2. Densidad de probabilidad.	63
7.1.3. Media.	64
7.1.4. Desviación.	65
7.2. 100 simulaciones con Ruido de perturbación.	66
7.2.1. Salidas.	66
7.2.2. Densidad de probabilidad.	67
7.2.3. Media.	68
7.2.4. Desviación.	69
7.3. 500 simulaciones sin Ruido de perturbación.	70
7.3.1. Salidas.	70
7.3.2. Densidad de probabilidad.	71
7.3.3. Media.	72
7.3.4. Desviación.	73
7.4. 500 simulaciones con Ruido de perturbación.	74
7.4.1. Salidas.	74



7.4.2. Densidad de probabilidad.	75
7.4.3. Media.	76
7.4.4. Desviación.	77
8. Resultados (Alpha = -0.2 y Beta = 0.2)	78
8.1.1. Salidas.	78
8.1.2. Densidad de probabilidad.	79
8.1.3. Media.	80
8.1.4. Desviación.	81
8.2. 100 simulaciones con Ruido de perturbación.	82
8.2.1. Salidas.	82
8.2.2. Densidad de probabilidad.	83
8.2.3. Media.	84
8.2.4. Desviación.	85
8.3. 500 simulaciones sin Ruido de perturbación.	86
8.3.1. Salidas.	86
8.3.2. Densidad de probabilidad.	87
8.3.3. Media.	88
8.3.4. Desviación.	89
8.4. 500 simulaciones con Ruido de perturbación.	90
8.4.1. Salidas.	90
8.4.2. Densidad de probabilidad.	91
8.4.3. Media.	92
8.4.4. Desviación.	93
9. Análisis Salidas según parámetros Alpha y Beta	93
9.1 Salidas sin ruido de perturbación (500 simulaciones).	94
9.2 Medias de Salidas con ruido de perturbación (500 simulaciones).	95
10. Análisis Desviación Típica según parámetros Alpha y Beta	95
10.1 Desviaciones Típicas con ruido de perturbación (100 simulaciones)	95
10.2 Desviaciones Típicas con ruido de perturbación (500 simulaciones).	97



11. Análisis Media Desviación Típica según parámetros Alpha y Beta.	98
11.1 Media Desviaciones típicas con ruido de perturbación manteniendo Alpha fija y variando Beta (500 simulaciones).	98
11.2 Media Desviaciones típicas con ruido de perturbación manteniendo Beta fija y variando Alpha (500 simulaciones).	99
12. Conclusiones de los datos obtenidos.	99
13. Final Degree Project's General Conclusion.	101
14. Referencias.	101
15. Anexos.	102
15.1. Códigos.	102





ÍNDICE DE FIGURAS:

Figura 1. Efectos del orden fraccionario en la acción integral.	4
Figura 2. Efectos del orden fraccionario en la acción derivativa.	5
Figura 3. Ilustración gráfica de las aportaciones de un PID clásico y fraccionario.	6
Figura 4. Esquema de procedimiento de simulación y ficheros .m	8
Figura 5. Esquema de PID clásico realizado en Simulink.	9
Figura 6. Esquema de PID fraccionario realizado en Simulink.	9
Figura 7. Salida sin Ruido de Perturbación (100 simulaciones).	14
Figura 8. Densidad de Probabilidad sin Ruido de Perturbación (100 simulaciones).	15
Figura 9. Salida sin Ruido de Perturbación (100 simulaciones).	16
Figura 10. Desviación Sin Ruido de Perturbación (100 simulaciones).	17
Figura 11. Salida con Ruido de Perturbación (100 simulaciones) 1.	18
Figura 12. Salida con Ruido de Perturbación (100 simulaciones) 2.	18
Figura 13. Densidad de Probabilidad con Ruido de Perturbación (100 simulaciones) 1.	19
Figura 14. Densidad de Probabilidad con Ruido de Perturbación (100 simulaciones) 2.	19
Figura 15. Media con Ruido de Perturbación (100 simulaciones).	20
Figura 16. Desviación con Ruido de Perturbación (100 simulaciones).	21
Figura 17. Salida sin Ruido de Perturbación (500 simulaciones).	22
Figura 18. Densidad de Probabilidad sin Ruido de Perturbación (500 simulaciones).	23
Figura 19. Media sin Ruido de Perturbación (500 simulaciones).	24
Figura 20. Desviación sin Ruido de Perturbación (500 simulaciones).	25



Figura 21. Salida con Ruido de Perturbación (500 simulaciones) 1.	26
Figura 22. Salida con Ruido de Perturbación (500 simulaciones) 2.	27
Figura 23. Densidad de Probabilidad con Ruido de Perturbación (500 simulaciones) 1.	28
Figura 24. Densidad de Probabilidad con Ruido de Perturbación (500 simulaciones) 2.	28
Figura 25. Media con Ruido de Perturbación (500 simulaciones).	29
Figura 26. Media con Ruido de Perturbación (500 simulaciones).	30
Figura 27. Salida sin Ruido de Perturbación (100 simulaciones).	31
Figura 28. Densidad de Probabilidad sin Ruido de Perturbación (100 simulaciones).	32
Figura 29. Media sin Ruido de Perturbación (100 simulaciones).	33
Figura 30. Desviación sin Ruido de Perturbación (100 simulaciones).	34
Figura 31. Salida con Ruido de Perturbación (100 simulaciones) 1.	35
Figura 32. Salida con Ruido de Perturbación (100 simulaciones) 2.	35
Figura 33. Densidad de Probabilidad con Ruido de Perturbación (100 simulaciones) 1.	36
Figura 34. Densidad de Probabilidad con Ruido de Perturbación (100 simulaciones) 2.	36
Figura 35. Media con Ruido de Perturbación (100 simulaciones).	37
Figura 36. Media con Ruido de Perturbación (100 simulaciones).	38
Figura 37. Salida sin Ruido de Perturbación (500 simulaciones).	39
Figura 38. Densidad de Probabilidad sin Ruido de Perturbación (500 simulaciones).	40
Figura 39. Media sin Ruido de Perturbación (500 simulaciones).	41
Figura 40. Desviación sin Ruido de Perturbación (500 simulaciones).	42
Figura 41. Salida con Ruido de Perturbación (500 simulaciones) 1.	43



Figura 42. Salida con Ruido de Perturbación (500 simulaciones) 2.	43
Figura 43. Densidad de Probabilidad con Ruido de Perturbación (500 simulaciones) 1.	44
Figura 44. Densidad de Probabilidad con Ruido de Perturbación (500 simulaciones) 2.	44
Figura 45. Media con Ruido de Perturbación (500 simulaciones).	45
Figura 46. Desviación con Ruido de Perturbación (500 simulaciones).	46
Figura 47. Salida sin Ruido de Perturbación (100 simulaciones).	47
Figura 48. Densidad de Probabilidad sin Ruido de Perturbación (100 simulaciones).	48
Figura 49. Media sin Ruido de Perturbación (100 simulaciones).	49
Figura 50. Desviación sin Ruido de Perturbación (100 simulaciones).	50
Figura 51. Salida con Ruido de Perturbación (100 simulaciones) 1.	51
Figura 52. Salida con Ruido de Perturbación (100 simulaciones) 2.	51
Figura 53. Densidad de Probabilidad con Ruido de Perturbación (100 simulaciones) 1.	52
Figura 54. Densidad de Probabilidad con Ruido de Perturbación (100 simulaciones) 2.	52
Figura 55. Media con Ruido de Perturbación (100 simulaciones).	53
Figura 56. Desviación con Ruido de Perturbación (100 simulaciones).	54
Figura 57. Salida sin Ruido de Perturbación (500 simulaciones).	55
Figura 58. Densidad de Probabilidad sin Ruido de Perturbación (500 simulaciones).	56
Figura 59. Media sin Ruido de Perturbación (500 simulaciones).	56
Figura 60. Desviación sin Ruido de Perturbación (500 simulaciones).	57
Figura 61. Salida con Ruido de Perturbación (500 simulaciones) 1.	58
Figura 62. Salida con Ruido de Perturbación (500 simulaciones) 2.	58



Figura 63. Densidad de Probabilidad con Ruido de Perturbación (500 simulaciones) 1.	59
Figura 64. Densidad de Probabilidad con Ruido de Perturbación (500 simulaciones) 2.	59
Figura 65. Media con Ruido de Perturbación (500 simulaciones).	60
Figura 66. Media con Ruido de Perturbación (500 simulaciones).	61
Figura 67. Salida sin Ruido de Perturbación (100 simulaciones).	62
Figura 68. Densidad de Probabilidad sin Ruido de Perturbación (100 simulaciones).	63
Figura 69. Media sin Ruido de Perturbación (100 simulaciones).	64
Figura 70. Desviación sin Ruido de Perturbación (100 simulaciones).	65
Figura 71. Salida con Ruido de Perturbación (100 simulaciones) 1.	66
Figura 72. Salida con Ruido de Perturbación (100 simulaciones) 2.	66
Figura 73. Densidad de Probabilidad con Ruido de Perturbación (100 simulaciones) 1.	67
Figura 74. Densidad de Probabilidad con Ruido de Perturbación (100 simulaciones) 2.	67
Figura 75. Media con Ruido de Perturbación (100 simulaciones).	68
Figura 76. Desviación con Ruido de Perturbación (100 simulaciones).	69
Figura 77. Salida sin Ruido de Perturbación (500 simulaciones).	70
Figura 78. Densidad de Probabilidad sin Ruido de Perturbación (500 simulaciones).	71
Figura 79. Media sin Ruido de Perturbación (500 simulaciones).	72
Figura 80. Desviación sin Ruido de Perturbación (500 simulaciones).	73
Figura 81. Salida con Ruido de Perturbación (500 simulaciones) 1.	74
Figura 82. Salida con Ruido de Perturbación (500 simulaciones) 2.	74
Figura 83. Densidad de Probabilidad con Ruido de Perturbación (500 simulaciones) 1.	75



Figura 84. Densidad de Probabilidad con Ruido de Perturbación (500 simulaciones) 2.	75
Figura 85. Media con Ruido de Perturbación (500 simulaciones).	76
Figura 86. Desviación con Ruido de Perturbación (500 simulaciones).	77
Figura 87. Salida sin Ruido de Perturbación (100 simulaciones).	78
Figura 88. Densidad de Probabilidad sin Ruido de Perturbación (100 simulaciones).	79
Figura 89. Media sin Ruido de Perturbación (100 simulaciones).	80
Figura 90. Desviación sin Ruido de Perturbación (100 simulaciones).	81
Figura 91. Salida con Ruido de Perturbación (100 simulaciones) 1.	82
Figura 92. Salida con Ruido de Perturbación (100 simulaciones) 2.	82
Figura 93. Densidad de Probabilidad con Ruido de Perturbación (100 simulaciones) 1.	83
Figura 94. Densidad de Probabilidad con Ruido de Perturbación (100 simulaciones) 2.	83
Figura 95. Media con Ruido de Perturbación (100 simulaciones).	84
Figura 96. Desviación con Ruido de Perturbación (100 simulaciones).	85
Figura 97. Salida sin Ruido de Perturbación (500 simulaciones).	86
Figura 98. Densidad de Probabilidad sin Ruido de Perturbación (500 simulaciones).	87
Figura 99. Media sin Ruido de Perturbación (500 simulaciones).	88
Figura 100. Desviación sin Ruido de Perturbación (500 simulaciones).	89
Figura 101. Salida con Ruido de Perturbación (500 simulaciones) 1.	90
Figura 102. Salida con Ruido de Perturbación (500 simulaciones) 2.	90
Figura 103. Densidad de Probabilidad con Ruido de Perturbación (500 simulaciones) 1.	91
Figura 104. Densidad de Probabilidad con Ruido de Perturbación (500 simulaciones) 2.	91



Figura 105. Media con Ruido de Perturbación (500 simulaciones).	92
Figura 106. Desviación con Ruido de Perturbación (500 simulaciones).	93
Figura 107. Salidas según parámetros Alpha y Beta sin Ruido de Perturbación (500 simulaciones).	94
Figura 108. Salidas según parámetros Alpha y Beta con Ruido de Perturbación (500 simulaciones).	95
Figura 109. Desviación según parámetros Alpha y Beta (100 simulaciones).	96
Figura 110. Desviación según parámetros Alpha y Beta (500 simulaciones).	97
Figura 111. Media Desviaciones Típicas con Alfa Fija.	98
Figura 112. Media Desviaciones Típicas con Beta Fija.	99



1. Introducción.

1.1. Objetivo.

El objetivo del presente Trabajo Fin de Grado es la realización de un estudio mediante simulaciones del comportamiento del ruido en un sistema modificando los parámetros de controladores de orden no entero, para estudiar si se producen efectos que benefician la evolución de la densidad de probabilidad del estado.

Se diseñará un programa de simulación con el software MATLAB/Simulink, con el cual comenzaremos planteando un sistema sencillo de PID clásico, este sistema lo iremos modificando hasta obtener un sistema más complejo de PID fraccionario. Una vez hayamos obtenido resultados de ambos sistemas procederemos a compararlos.

1.2. Abstract.

The objective of this Final Degree Project is to develop a study through simulations of the behaviour of noise in a system by modifying the parameters of non-integer order controllers, in order to study if there are effects that benefit the evolution of the probability density of state.

We will design a simulation programme with the MATLAB/Simulink software, with which we will start with a simple classical PID system, which we will modify until we obtain a more complex fractional PID system. Once we have obtained results from both systems, we will proceed to compare them.

2. Metodología.

2.1. Fundamentos de los controladores PID.

Un controlador PID (Proporcional-Integral-Derivativo) es un controlador que debido a su simplicidad y al hecho de englobar 3 acciones diferentes, se ha convertido en el controlador más habitual en entornos industriales hoy en día. Se puede encontrar en casi todos los tipos de equipos de control, ya sea como



controlador independiente (una sola estación), como bloque funcional en controladores lógicos programables (PLC) y en sistemas de control distribuidos (DCS). El éxito de dichos controladores se debe a su desempeño en esquemas de control más sofisticados en donde el control básico no es suficiente para obtener el rendimiento requerido.

Un sistema de control tiene como objetivo obtener una respuesta deseada para un sistema por lo que dicho control se puede hacer en lazo abierto o lazo cerrado. En el primer caso la señal de entrada del proceso que da el controlador se basa sólo en la señal de referencia mientras que en el lazo cerrado se utiliza también la medición de la salida, es decir, la señal de realimentación, siendo la señal de entrada al controlador el error entre la señal de referencia y la salida del proceso.

En definitiva, el control PID se basa en la realimentación que resulta de unir tres tipos de controladores, aportando cada uno de ellos su acción para que se alcance el estado de salida deseado.

2.2. PID Clásico.

El PID clásico es un controlador que combina tres acciones sobre la señal de error: la proporcional, la integral y la derivativa, mediante la ecuación (1) en el dominio de Laplace. Y cuyo fin es que el error en el estacionario entre la consigna y la salida se haga cero.

$$F(s) = K_p \cdot \left(1 + \frac{1}{T_i \cdot s} + T_d \cdot s\right) \quad (1)$$

Donde K_p es la ganancia proporcional; T_i el tiempo de acción integral y T_d el tiempo de acción derivativa.

- **Acción proporcional:**

Es proporcional a la señal de error. La acción proporcional multiplica la señal de error por la constante K_p . No se alcanzará el valor de consigna. Al aumentar K_p ocurre lo siguiente: se aumenta la velocidad de respuesta del sistema, disminuye el error en régimen estacionario y aumenta la inestabilidad. Los dos



primeros efectos son deseables, pero el último efecto es el que hay que minimizar, por lo que hay que buscar el punto en el que haya rapidez y reducción del error sin que el sistema se haga inestable.

- **Acción integral:**

Calcula la integral de la señal de error. La integral la podemos interpretar como la acumulación de la señal de error, es decir, a medida que pasa el tiempo los errores se van sumando, lo que provoca que la acción integral sea mayor. Elimina el error en el estacionario para alcanzar el valor de consigna.

Aumentando la acción integral K_i ($K_i = \frac{1}{T_i}$) ocurre lo siguiente: disminuye el error del sistema en estado estacionario, aumenta la inestabilidad del sistema y aumenta un poco la velocidad del sistema.

Debe aplicarse al menos con la acción proporcional y hace la respuesta más oscilatoria ya que, se añade inercia al sistema y por lo tanto se hace más inestable.

- **Acción derivativa:**

Es proporcional a la derivada del error, también llamada velocidad del error. Aumentando la constante de control derivativa K_d ($K_d = T_d$) ocurre lo siguiente: aumenta la estabilidad del sistema, disminuye un poco la velocidad y el error en régimen estacionario no cambiará.

Aporta una acción correctora con errores pequeños y servirá para estabilizar una respuesta que presente mucha oscilación. Esta acción también hay que aplicarla al menos con la acción proporcional.

2.3. PID Fraccionario.

Se considera cálculo fraccionario a aquella extensión del cálculo que considera la integración y la derivación en cualquier orden no entero. El control fraccionario es debido a los trabajos que aplican el cálculo fraccionario en control, que surgió de la necesidad de buscar dispositivos cuyo comportamiento dinámico venga caracterizado por los operadores del cálculo fraccionario y que



puedan reproducir con suficiente fidelidad el comportamiento ideal de los operadores, tanto en el dominio del tiempo como en el de la frecuencia.

Por lo tanto es una metodología para describir sistemas mediante modelos de orden fraccionario, y generalizar el PID clásico, manteniendo la simplicidad, pero haciendo posible tratar con una clase más general los problemas de control ya que, aporta un mayor número de parámetros sintonizables que aportan más robustez.

Como se ha dicho anteriormente, la acción integral hace la respuesta más lenta para conseguir estabilidad y elimina el error en el estacionario. En el dominio del tiempo, al disminuir el valor de la acción integral se produce un decrecimiento del tiempo de subida y un aumento del tiempo de establecimiento. La acción integral también provoca una sobreoscilación en la respuesta transitoria. En el dominio de la frecuencia se produce un aumento en la pendiente de la magnitud de 20dB y en la fase de 90°.

En la siguiente figura se muestran los efectos del orden fraccionario en la acción integral en una señal de error cuadrada:

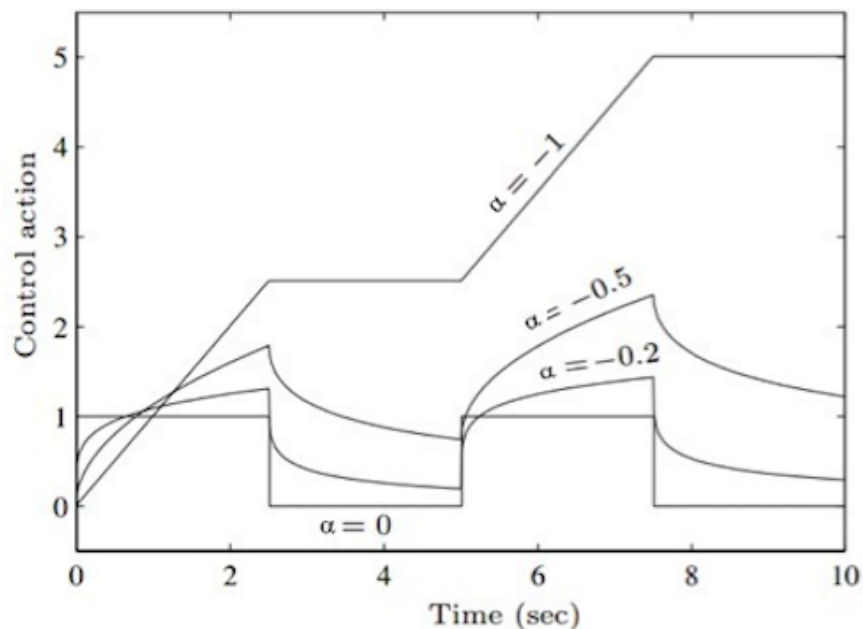


Figura 1. Efectos del orden fraccionario en la acción integral.

De esta figura podemos deducir lo siguiente:

$\alpha = 0$. La señal de error seguirá comportándose como una señal cuadrada.



$\alpha = -1$ Corresponde a la integral clásica entera, y se observa un incremento constante de la acción de control.

Mientras que en los valores intermedios de $-0,5$ y $-0,2$ la señal tiene una tendencia de crecimiento cuando el error es constante, pero los valores son más pequeños que en el caso entero, lo que implica mayor estabilidad.

La acción derivativa aumenta la estabilidad del sistema y acentúa los efectos de los ruidos y perturbaciones de alta frecuencia. En el dominio del tiempo se manifiesta por una disminución de la sobreoscilación y del tiempo de establecimiento. En el dominio de la frecuencia se añaden 20dB a la pendiente de la magnitud y 90° a la fase.

En la siguiente figura podemos observar los efectos del orden fraccionario en la acción derivativa en la señal de error:

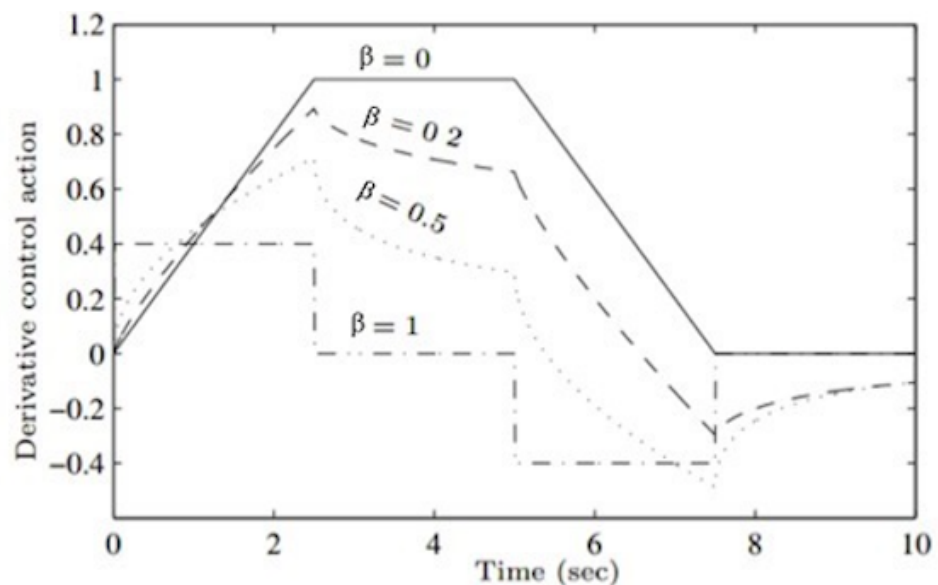


Figura 2. Efectos del orden fraccionario en la acción derivativa.

Cuando aplicamos un orden entero a la acción derivativa podemos observar que la acción de control sufre cambios más bruscos. En cambio, al aplicar un orden fraccionario, la acción de control varía más suavemente, lo que como se dijo anteriormente mejora la atenuación de ruidos de alta frecuencia.

El PID fraccionario combina las acciones anteriores y se representa de la siguiente manera en el dominio de Laplace:



$$F_f(s) = K_p \cdot \left(1 + \frac{1}{T_i \cdot s^\alpha} + T_d \cdot s^\beta\right) \quad (2)$$

A continuación, podemos ver de forma gráfica una descripción de los controladores PID enteros y fraccionarios:

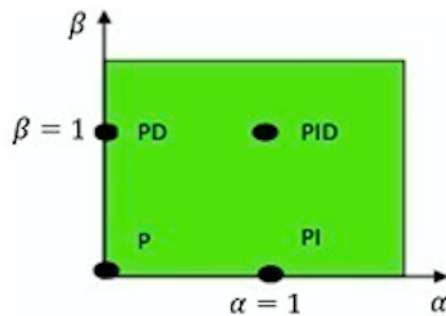


Figura 3. Ilustración gráfica de las aportaciones de un PID clásico y fraccionario.

Los puntos negros se refieren a los distintos tipos de controladores clásicos existentes, en cambio el cuadro verde representa todas las posibilidades que existen para los controladores fraccionarios, por lo que como se puede observar, estos controladores ofrecen un abanico mayor de posibilidades en los parámetros de ajuste. En lugar de disponer de tres parámetros, como ocurre en el PID clásico, se dispone en el caso fraccionario de cinco parámetros.

3. Algoritmo y simulaciones

Para realizar el estudio y las comparaciones de los sistemas controlados mediante un PID clásico y un PID fraccionario se utilizó la herramienta Matlab. En ella se crearon los lazos de control, se diseñó el algoritmo para hacer los estudios deseados y se simuló para ver sus respuestas. Se diseñaron varios experimentos mediante una serie de simulaciones y gráficos. En concreto cuatro para cada tipo de control. En estos cuatro estudios se variaron tanto el número de simulaciones como la existencia o ausencia de ruido para poder realizar comparaciones.

Dentro de cada lazo de control se busca estudiar el efecto que tiene el ruido en la salida del sistema cuando se le aplica una entrada específica. El controlador, en un instante de tiempo, llevará la salida a un valor fijo en el estacionario. Dependiendo de la entrada y del ruido, el transitorio se verá modificado por lo que el objeto del presente estudio es ver cómo se comportan las salidas para los 2 tipos de control.



Las entradas y salidas del sistema son dos factores a tener en cuenta a la hora de realizar el estudio y de analizar los valores, pero también tendremos en cuenta la densidad de probabilidad de las salidas obtenidas al introducir una entrada de números aleatorios.

Para entender exactamente lo que se ha descrito en el párrafo anterior habrá que tener claros dos conceptos: qué es la densidad de probabilidad y qué es una variable aleatoria. Una variable aleatoria viene definida por una serie de valores numéricos que corresponden al resultado de un experimento aleatorio. Por otro lado, la distribución de probabilidad describe la forma en la que varían los resultados del experimento aleatorio, es decir, nos da las probabilidades de todos los posibles resultados, en nuestro caso se trata de una función de densidad continua. La curva que define la función de densidad de probabilidad nos sirve para identificar las regiones de mayor y menor probabilidad de valores de la variable aleatoria.

En el caso que tratamos, obtendremos densidades de probabilidad que siguen el orden de una distribución normal o gaussiana. La distribución gaussiana es la distribución continua más importante y es una función de probabilidad para variables aleatorias continuas. Algunas de las características más importantes de la densidad de probabilidad de esta función son las siguientes: tiene forma de campana, es simétrica y puede tomar valores entre menos infinito y más infinito. Por lo tanto, de esto podemos decir lo siguiente, los valores centrales de la variables se presentan con más frecuencia que los extremos, es simétrica por lo que sus medidas de posición central (media, mediana y moda) coinciden y por último presenta como asíntota el eje x por lo que se aproxima a este sin cortarlo.

La función de densidad solo depende de dos parámetros la media μ y la varianza σ^2 . La distribución normal que tiene de media $\mu = 0$ y $\sigma^2 = 1$, se denomina distribución normal estándar $N(0,1)$. La gráfica tendrá simetría respecto a la media y el ancho de esta dependerá del valor de la varianza, cuanto mayor sea más achatada será.

Una vez explicados estos conceptos se explicará cómo se ha implementado el procedimiento. Hemos creado una matriz de números aleatorios en la que cada columna cumple con una distribución normal, que será nuestro ruido del sistema. A la hora de generar los números se procedió a identificar los valores máximos y mínimos y dividir en intervalos iguales, de esa forma podemos obtener la cantidad de números que hay en cada intervalo y obtener así la densidad de probabilidad que tiene forma de campana de Gauss.



Para un mejor entendimiento del código, éste se dividió en varios archivos **.m** los cuáles vienen especificados para cada caso (con o sin ruido de proceso).

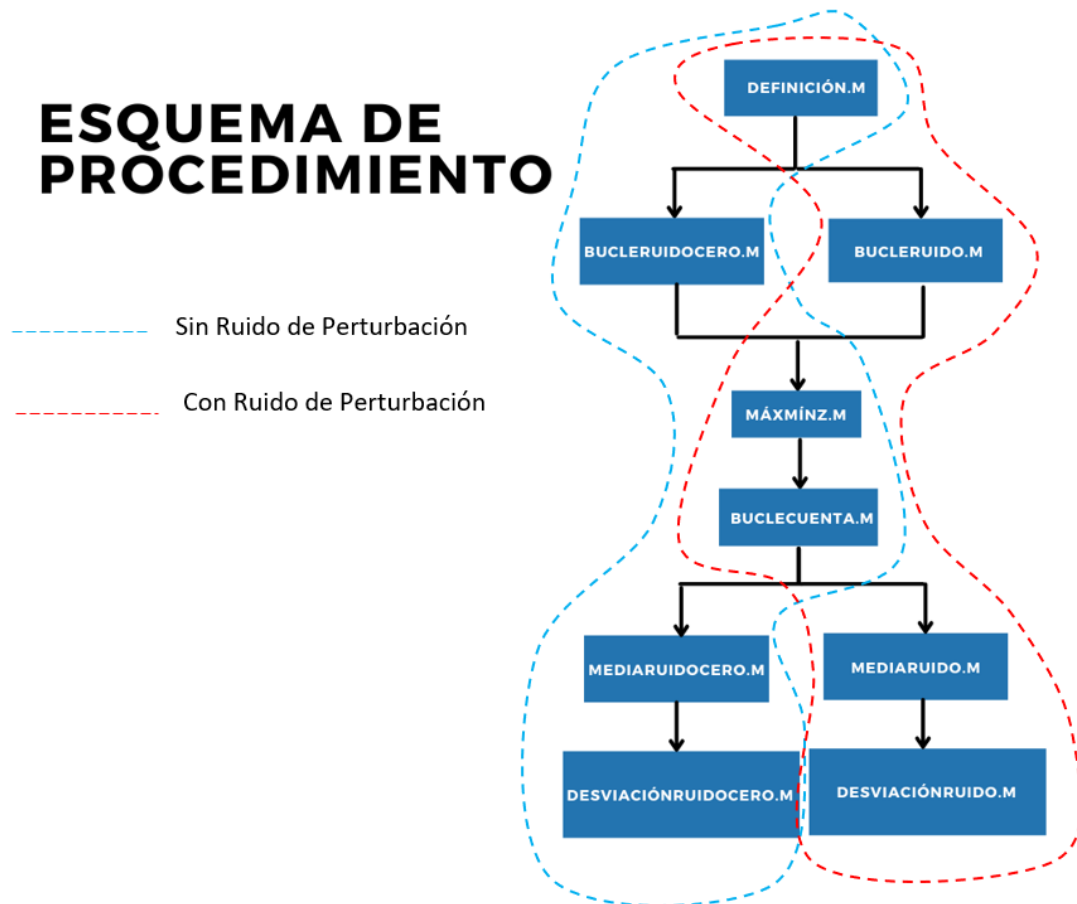


Figura 4. Esquema de procedimiento de simulación y ficheros **.m**

3.1. Estudio control con PID Clásico.

Mediante la herramienta Matlab, se creó en Simulink un sistema sencillo para simular un lazo de control PID clásico:

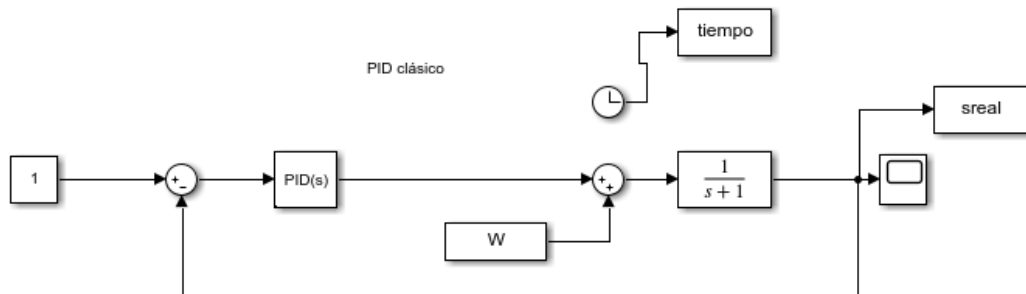


Figura 5. Esquema de PID clásico realizado en Simulink.

En la figura se pueden observar: la entrada (la cual se fijó en 1), el controlador PID (con sus acciones proporcional, integral y derivativa), la matriz de ruido o perturbación externa W, la función de transferencia de primer orden, la variable de salida “sreal”, el scope que nos permitió observar los resultados y la variable tiempo para definir el intervalo de estudio.

3.2. Estudio control con PID fraccionario.

De igual manera que el caso anterior, mediante la herramienta Matlab, se creó en Simulink un sistema sencillo para simular un lazo de control PID fraccionario:

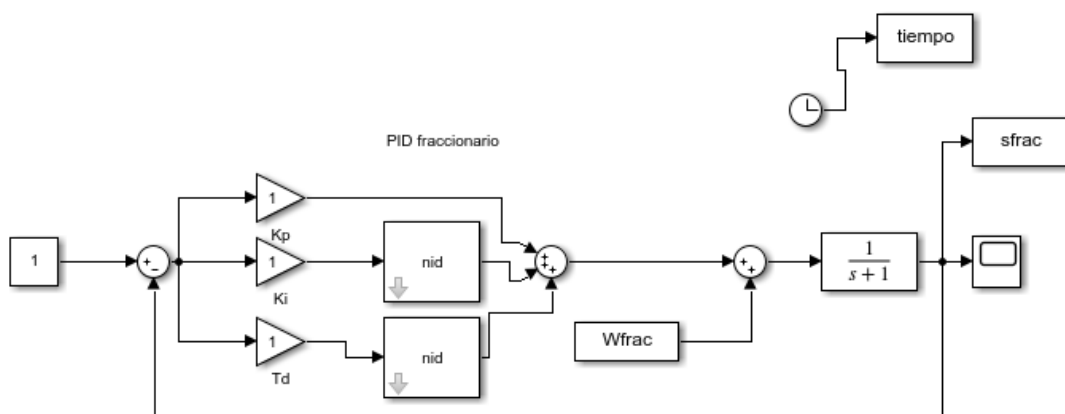


Figura 6. Esquema de PID fraccionario realizado en Simulink.

Para este lazo de control se utilizaron los bloques “nid” (fractional derivative) el cual simula un controlador fraccionario ya sea integral o derivativo según el valor que se le de:



- ”Alpha”: variable del controlador integral fraccionario (valor negativo).
- ”Beta”: variable del controlador derivativo fraccionario (valor positivo).

y permite de esta manera tener un orden no entero en dichas acciones.

Por otro lado, para separar el controlador PID, se crearon 3 bloques de ganancia:

- Kp: Ganancia Proporcional
- Ki: Ganancia Integral
- Td: Ganancia Derivativa

Cabe destacar que los valores del bloque “nid” están comprendidos entre 0 y 1 por lo que cuanto más se acerque el valor a 1, más cercano es el comportamiento a un PID clásico.

Del mismo modo que en el caso del PID clásico se tiene una entrada (la cual se fijó en 1), la matriz de ruido o perturbación externa Wfrac, la función de transferencia de primer orden, la variable de salida “sfrac”, el scope que nos permite observar los resultados y la variable tiempo para definir el intervalo de estudio.

3.3. Algoritmo.

Como se comentó anteriormente, el código se dividió en varios archivos para su mejor visualización y orden. Para cada controlador se utilizó el mismo código sólo diferenciando entre los nombres de las variables, como por ejemplo la matriz de perturbación externa ($W \rightarrow Wfrac$) y en los nombres de los archivos **.m** para especificar la ausencia o presencia de ruido (BluceRuido \rightarrow BucleRuidoCero), pero todo esto sin hacer cambio en la función que tienen dentro de la codificación. Es por ello que se explicará una sola versión y luego en los anexos se verá su símil.

El estudio conlleva la especificación del controlador, que entrada se usará, qué ruido de perturbación y el tiempo de la simulación, así como la manera de introducirlo en el sistema (un número, una matriz, conjunto de datos, etc.).

Los archivos utilizados fueron los siguientes:



-Definición: Aquí se encuentran las bases de la simulación, es decir, las primeras variables y los primeros parámetros.

Como se comentó antes, para cada controlador se hicieron 4 pruebas (de 100 y 500 simulaciones con ruido y sin ruido), por lo tanto, se define en primera instancia una matriz de ceros que va a almacenar las salidas obtenidas en las simulaciones a lo largo del tiempo establecido. Dichas salidas llenarán las columnas de la matriz siendo el número de filas el tiempo, es decir, en cada fila se tendrán los 100 valores del primer instante de tiempo, en la segunda fila los 100 valores correspondientes al segundo instante y así sucesivamente. Es por ello que luego se define la variable tiempo con 105 valores espaciados entre ellos por 0.0961539 aproximadamente. Luego se añadió otra variable de tiempo igual para tomarla de referencia en una futura interpolación ya que, al simular, las dimensiones de la variable original se perdían debido a ajustes que hace el Simulink. Seguidamente se definió el ruido como una matriz de ceros de dos columnas y 105 filas en la que la primera es el tiempo y la segunda los datos. La siguiente matriz llamada “entrada” recoge 100 simulaciones, con ruidos gaussianos de media 0 y desviación típica 1, en sus columnas, mientras que nuevamente tiene 105 filas (tiempo). Para llenar la matriz “entrada” se creó una variable aparte la cuál, mediante un bucle, va creando columnas de los números aleatorios antes descritos; de esa manera se asegura que el ruido que entra al sistema siempre sea gaussiano. En caso de no haber ruido, ésta matriz se define de las mismas dimensiones pero de ceros. Por último, se crea una matriz de ceros que define el tamaño de la salida inicial (105 filas) y otra igual de referencia para una futura interpolación.

-BucleRuido / BucleRuidoCero: En este archivo se encuentra el primer bucle donde se ejecuta el lazo de control y se almacenan las variables.

El bucle consiste en recorrer de 1 a 100 (ó 500 según el número de simulaciones establecidas) para almacenar una a una cada columna de la matriz “entrada” (ruidos gaussianos o ceros) en la segunda columna de la matriz de ruido. Seguidamente se ejecuta el Simulink, dónde ya se encuentra el lazo de control, y se realiza la interpolación de las salidas para llevarlas al tiempo que queremos (variables de referencia que definimos en el archivo anterior). Dentro del bucle, una vez tenemos la salida con los datos que queremos y dentro del



tiempo establecido, se hace la traspuesta y se van llenando las columnas de la matriz que almacena las salidas.

-MáxMinZ: La densidad de probabilidad nos va a mostrar el comportamiento real del sistema, para ello se especifican en el código una serie de parámetros.

Para poder implementar el cálculo de la densidad de probabilidad, primero se especificaron los valores máximo y mínimo de la variable Z (matriz que almacena las salidas). Dentro de este rango se hicieron 25 divisiones (ó 125 si son 500 simulaciones). Como lo que se quería lograr era ver cuantos valores de cada fila de la matriz Z se acumulaban dentro de cada división y así hacer la densidad de probabilidad, se creó un vector de ceros de 107 filas y 26 columnas dónde se acumularan los valores y se colocaron en la última fila los valores divisionales para que la fila 106 sirviera de contador.

-BucleCuenta: En este archivo se hace la cuenta de los valores mediante un bucle dentro de un bucle condicional.

Para conseguir los valores (por ejemplo en la prueba de 100 simulaciones) primero se establece la condición de que se repita el bucle siempre que un valor sea menor o igual a 105, es decir, que se repita hasta que recorramos los 105 instantes de tiempo, luego añadimos un bucle para recorrer las 100 columnas o mejor dicho las 100 simulaciones (500 en el otro caso) y otro para las 25 columnas del vector acumulativo que definimos en el archivo anterior. Dentro del último bucle se va acotando cada uno de los valores de la matriz Z y según entre qué valores esté (valores de las divisiones especificados en la fila 107 del vector acumulativo) se suma 1 a la posición correspondiente de esa división en el mismo vector (contador, fila 106). Una vez hayamos recorrido las 100 columnas de la primera fila y tengamos los valores acumulados, los vamos copiando en todo el vector y así tener en cada fila cada salida acumulada. Una vez terminada la fila, se pasa a la siguiente.

-DesviaciónRuido / DesviaciónRuidoCero: Para un posterior estudio se saca la desviación típica de cada columna de la matriz Z y se almacena en un vector de una fila y 105 columnas.



-MediaRuido / MediaRuidoCero: De la misma manera, para un estudio posterior, se calcula la media de cada columna de la matriz Z y se almacena en un vector de una fila y columnas.

-Gráficos: Se agrupan en este archivo las gráficas individuales a realizar para su posterior estudio las cuales son: salida del sistema, desviación típica y media de la salida y la densidad de probabilidad.

-Gráficos combinados: En este archivo se encuentran los gráficos ya agrupados para su comparación. Los gráficos son los mismos que el archivo **.m** anterior.

-Alfa Fija / Beta Fija: Para poder estudiar cómo afectan los valores Alfa y Beta por separado, se creó este archivo en el que primeramente se definen unas variables que van a almacenar la media de las desviaciones de cada uno de los casos en que un valor se queda fijo y el otro varía. Luego, para poder hacer este estudio sólo en el estacionario, se acotó la desviación y luego se almacenó en otras variables. Por último, se halló la media de dicha variable y se graficaron todas en un sólo plot.

-Gráficos Salidas / Gráficos Medias Salidas / Gráficos Desviación: En estos archivos se encuentran los códigos para los últimos análisis.

4. Resultados (Alpha = -0.9 y Beta = 0.9)

Una vez hecho el algoritmo de las 4 pruebas, se procede a comparar cada una de las pruebas del PID Clásico con el PID Fraccionario, representando gráficamente las salidas, las desviaciones típicas y las medias de los resultados de cada simulación. Cabe destacar que inicialmente se usaron las variables del controlador fraccionario cercanas a 1, por lo que se espera que los comportamientos de ambos PID sean parecidos. Por otro lado, para los casos con ruido de perturbación, en las gráficas de salida y densidad de probabilidad se tomaron datos de varias simulaciones, en cambio para la gráfica de salida de los casos sin ruido de perturbación se hizo una única simulación.



4.1. 100 simulaciones sin Ruido de perturbación.

4.1.1. Salidas.

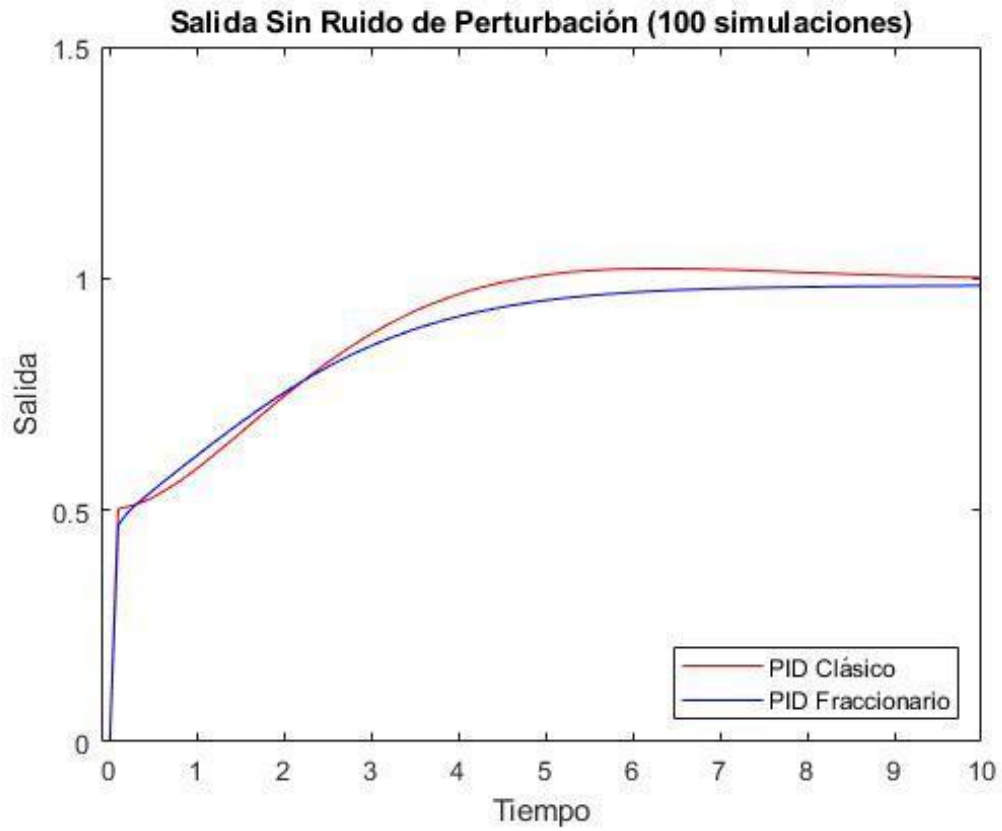


Figura 7. Salida Sin Ruido de Perturbación (100 simulaciones).



4.1.2. Densidad de probabilidad.

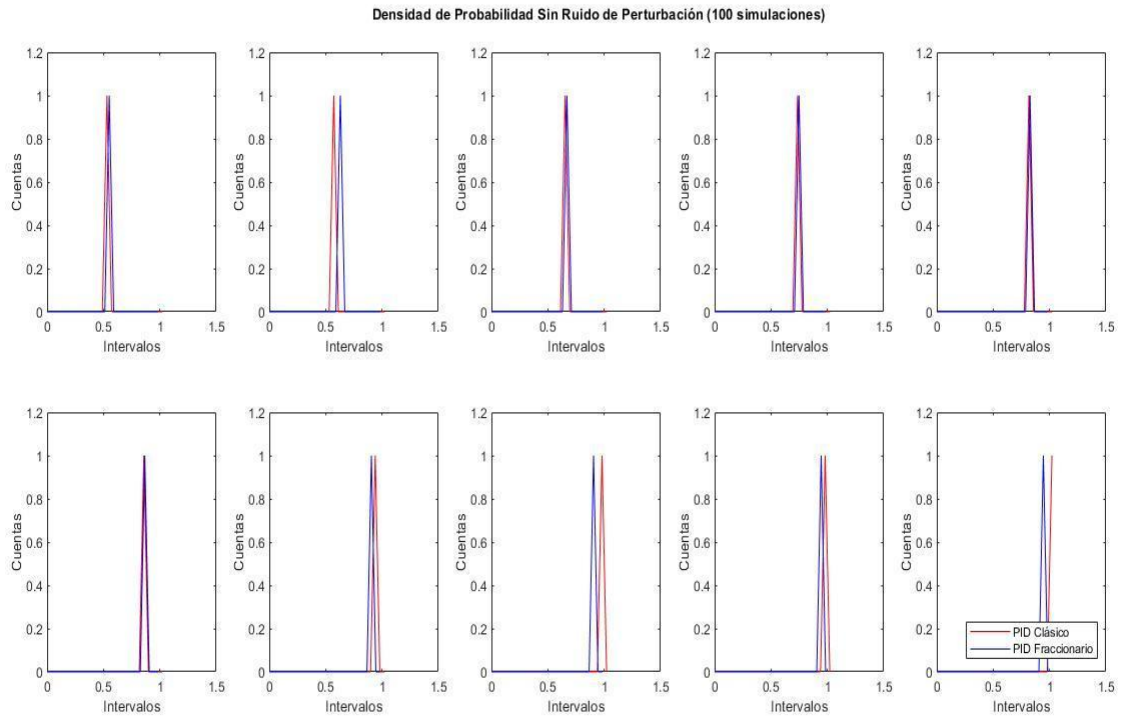


Figura 8. Densidad de Probabilidad Sin Ruido de Perturbación (100 simulaciones).



4.1.3. Media.

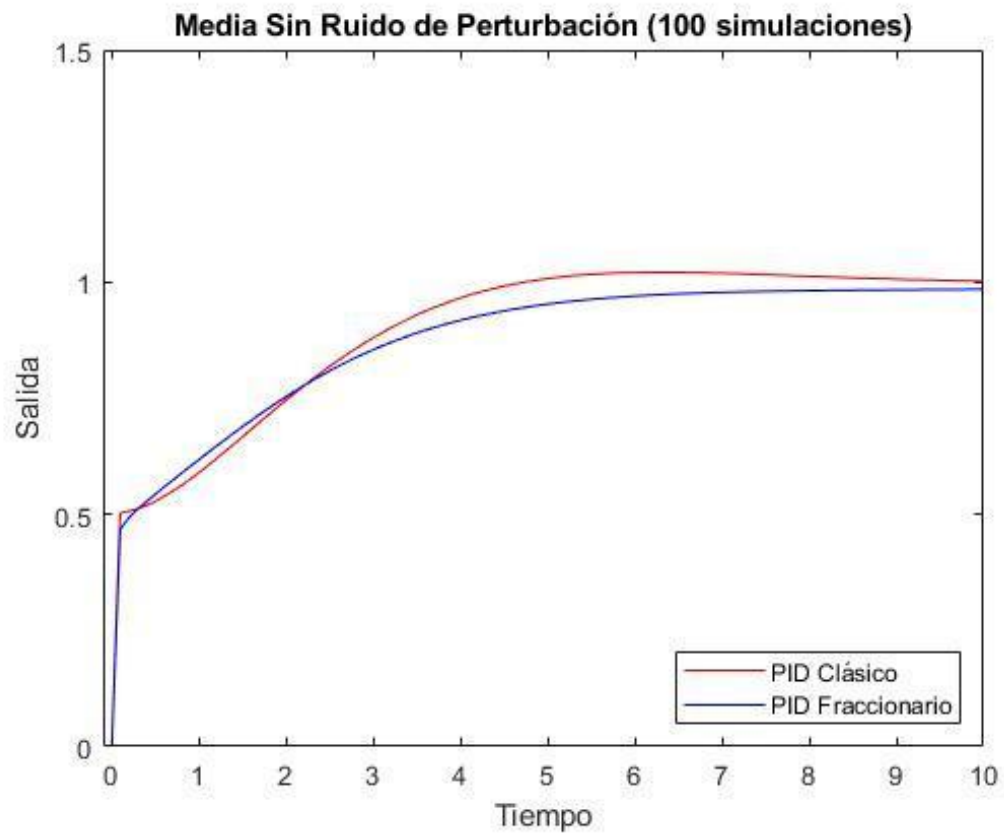


Figura 9. Media Sin Ruido de Perturbación (100 simulaciones).



4.1.4. Desviación.

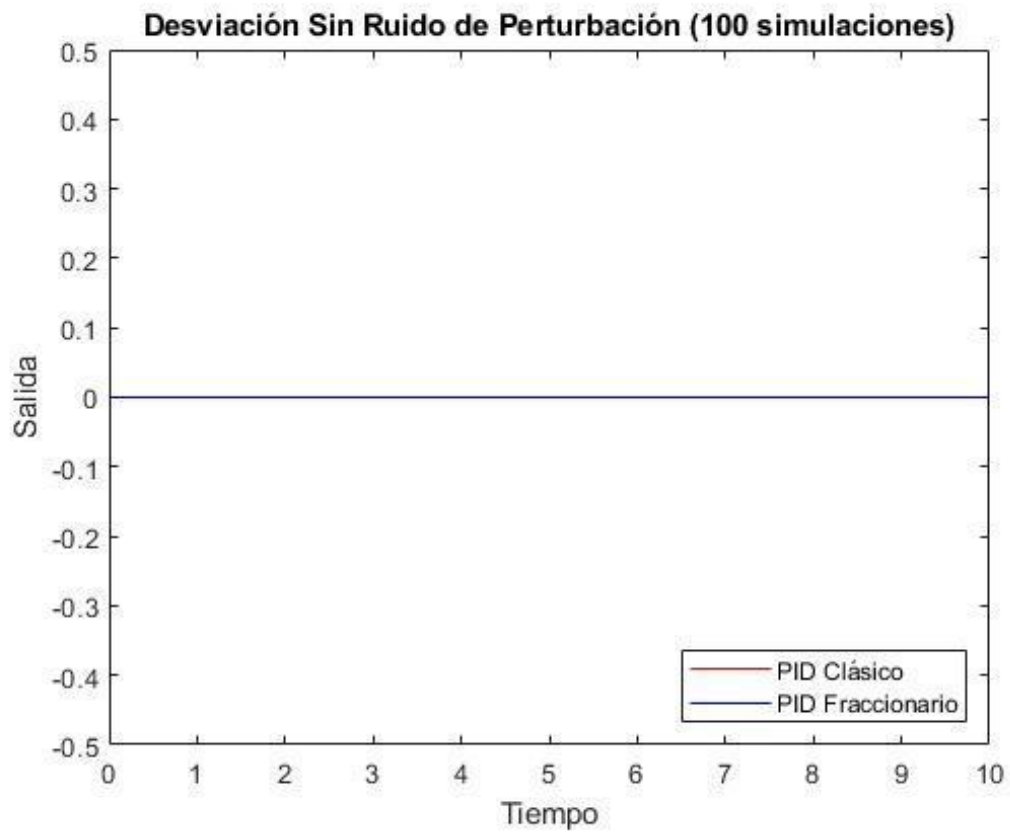


Figura 10. Desviación Sin Ruido de Perturbación (100 simulaciones).



4.2. 100 simulaciones con Ruido de perturbación.

4.2.1. Salidas.

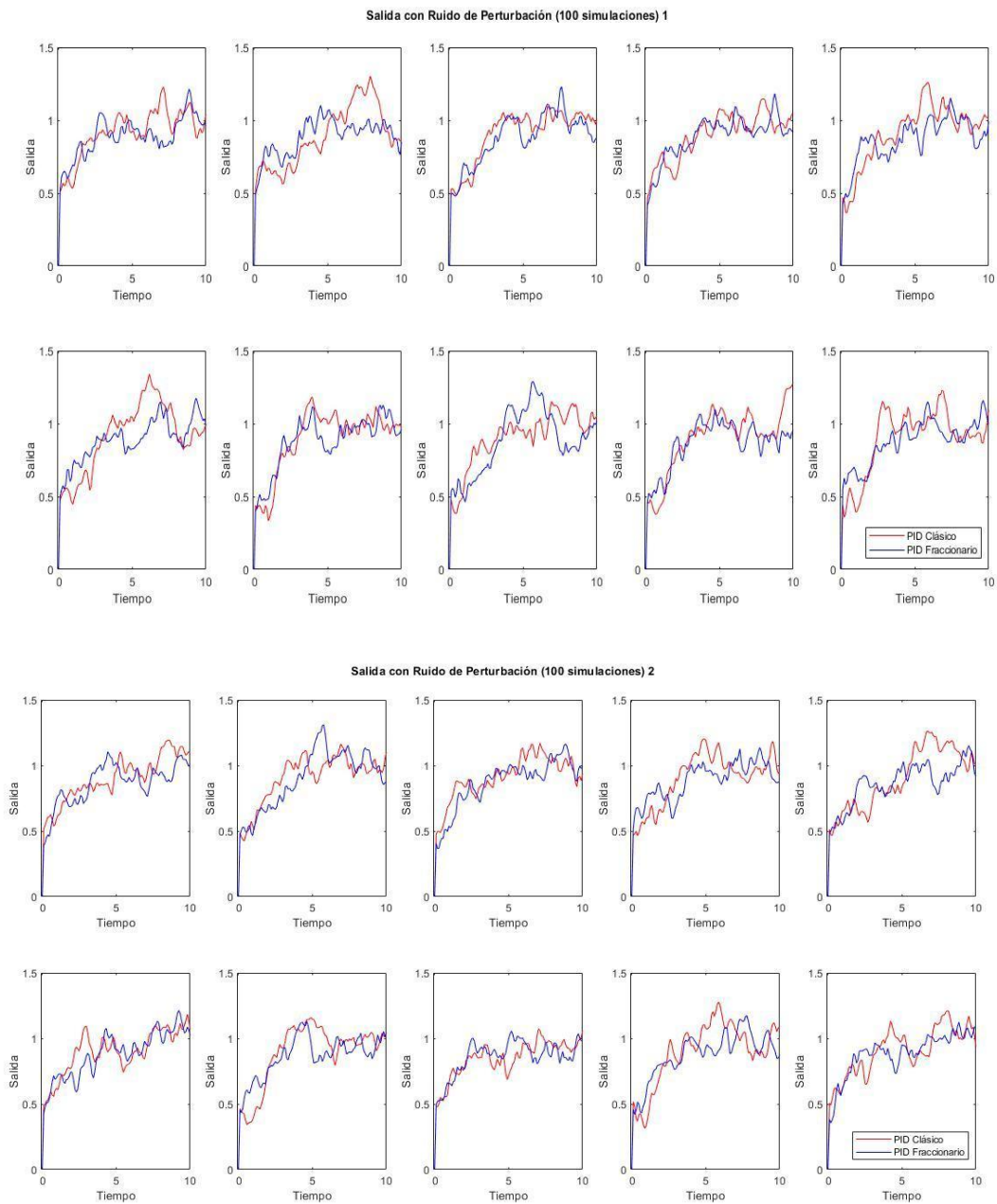


Figura 11. Salida Con Ruido de Perturbación (100 simulaciones) 1.

Figura 12. Salida Con Ruido de Perturbación (100 simulaciones) 2.



4.2.2. Densidad de probabilidad.

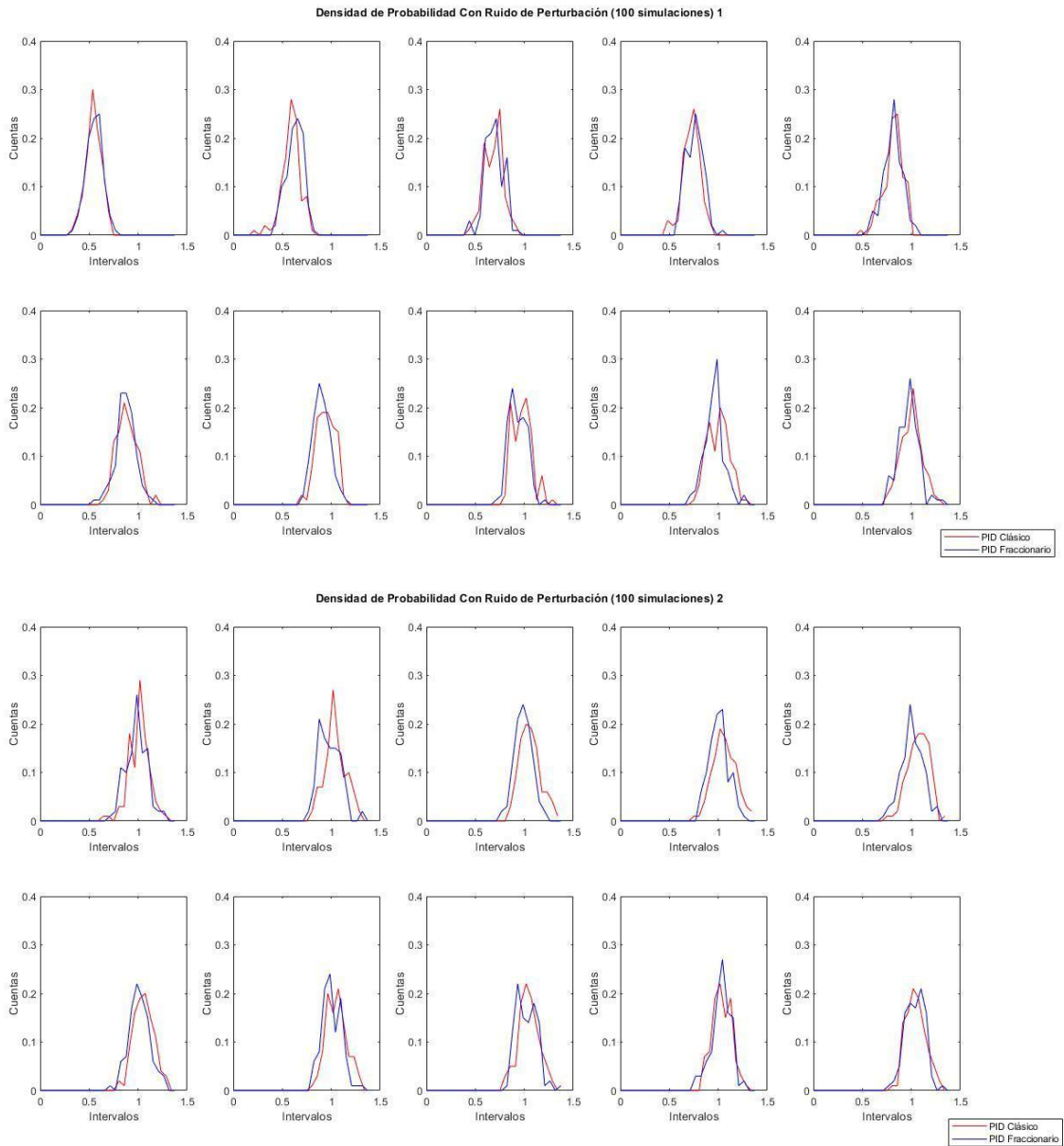


Figura 13. Densidad de Probabilidad Con Ruido de Perturbación (100 simulaciones) 1.
Figura 14. Densidad de Probabilidad Con Ruido de Perturbación (100 simulaciones) 2.



4.2.3. Media.

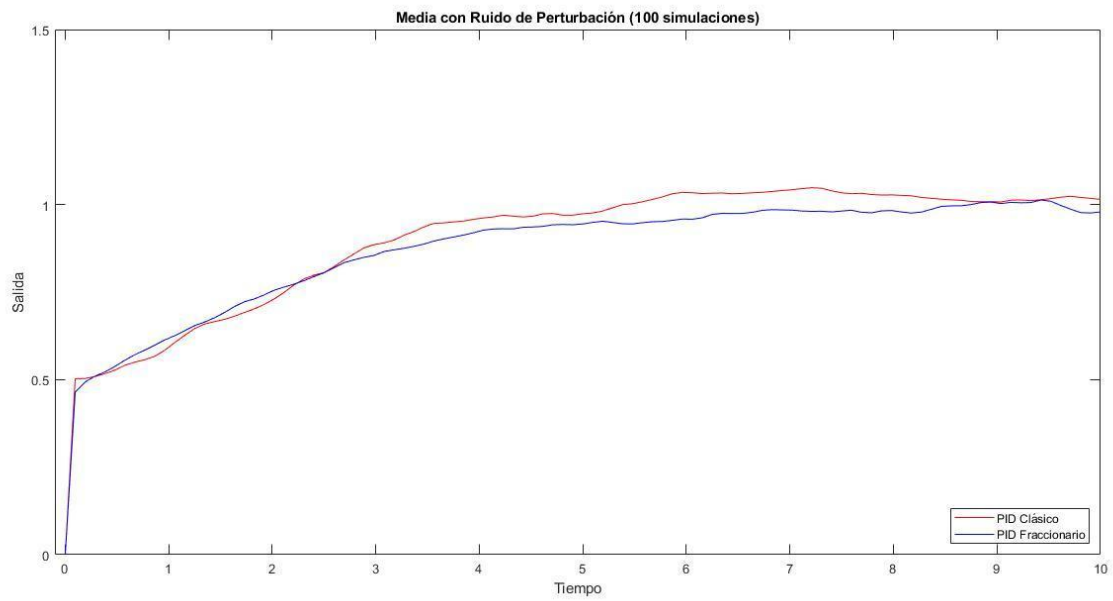


Figura 15. Media Con Ruido de Perturbación (100 simulaciones).



4.2.4. Desviación.

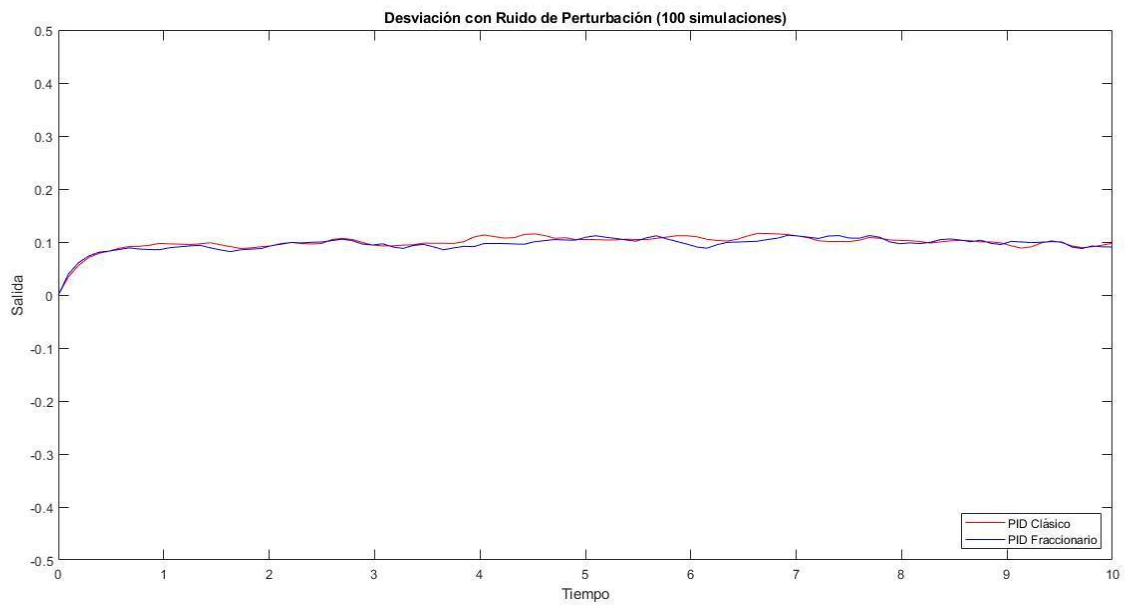


Figura 16. Desviación con Ruido de Perturbación (100 simulaciones).



4.3. 500 simulaciones sin Ruido de perturbación.

4.3.1. Salidas.

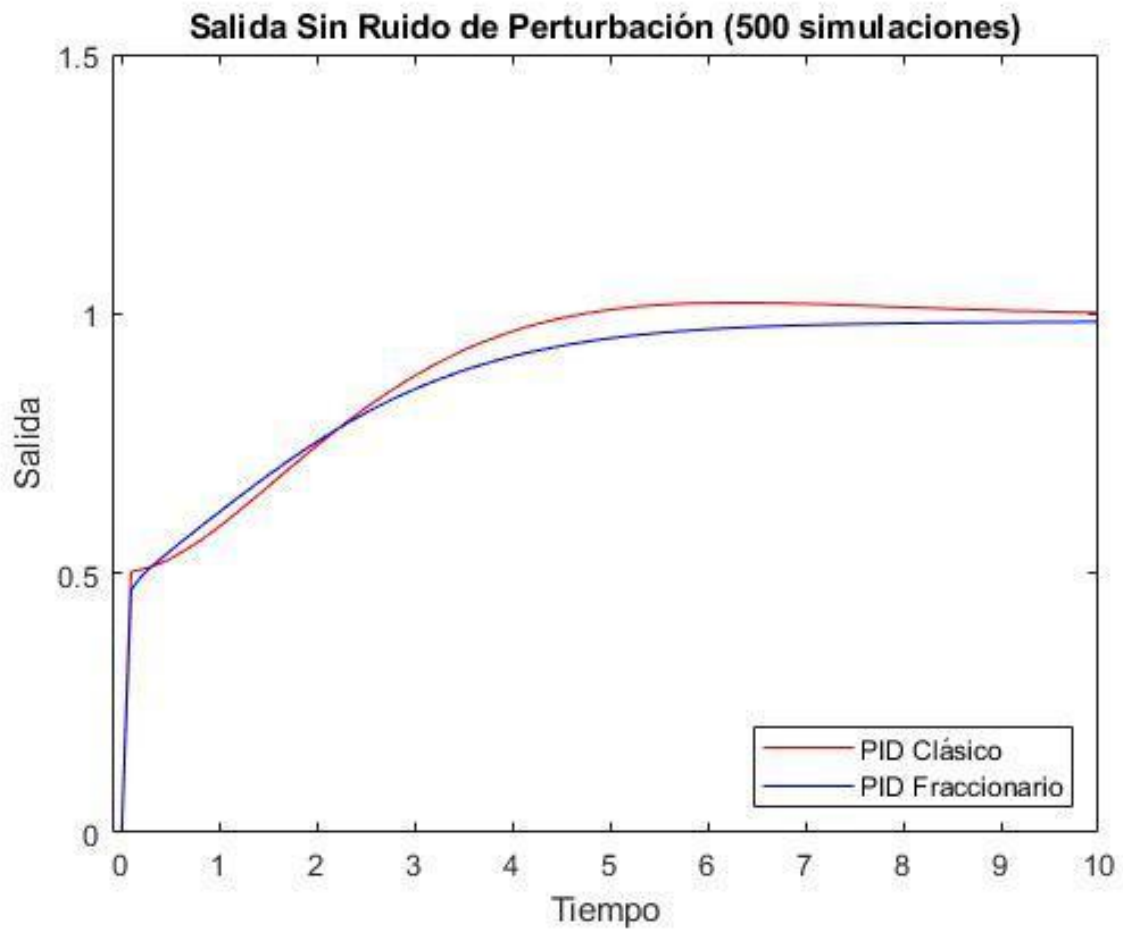


Figura 17. Salida sin Ruido de Perturbación (500 simulaciones).



4.3.2. Densidad de probabilidad.

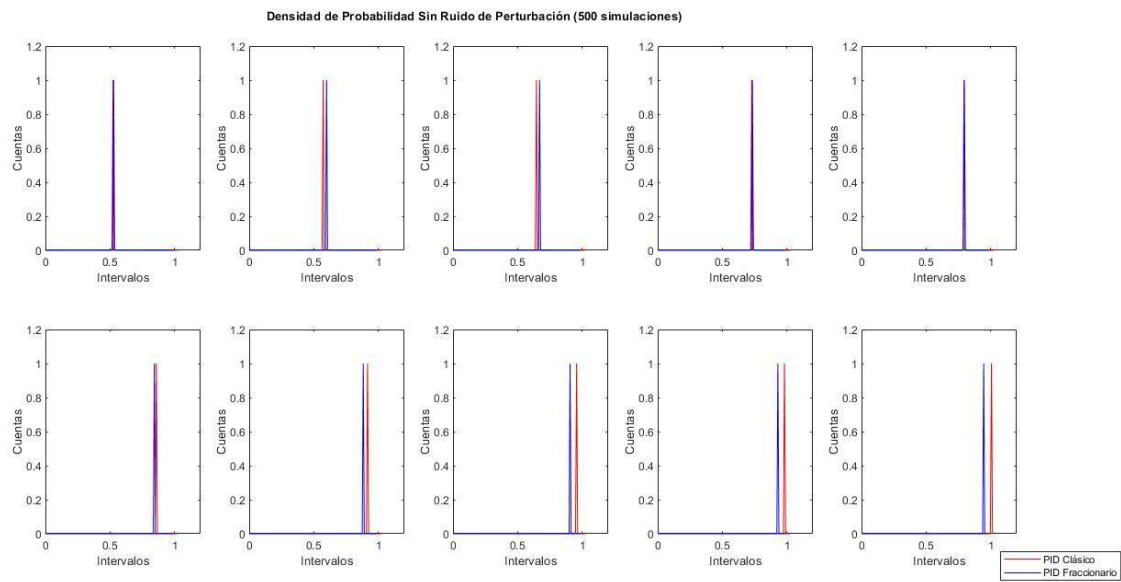


Figura 18. Densidad de Probabilidad sin Ruido de Perturbación (500 simulaciones).



4.3.3. Media.

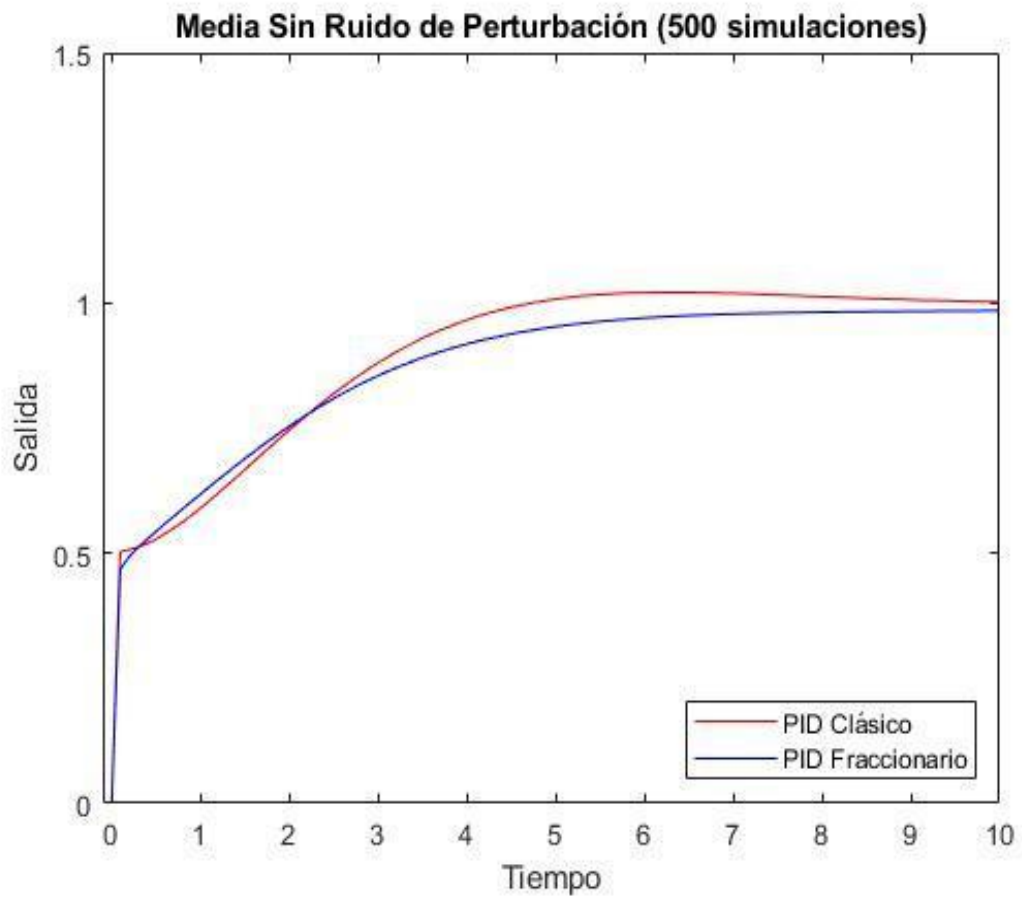


Figura 19. Media sin Ruido de Perturbación (500 simulaciones).



4.3.4. Desviación.

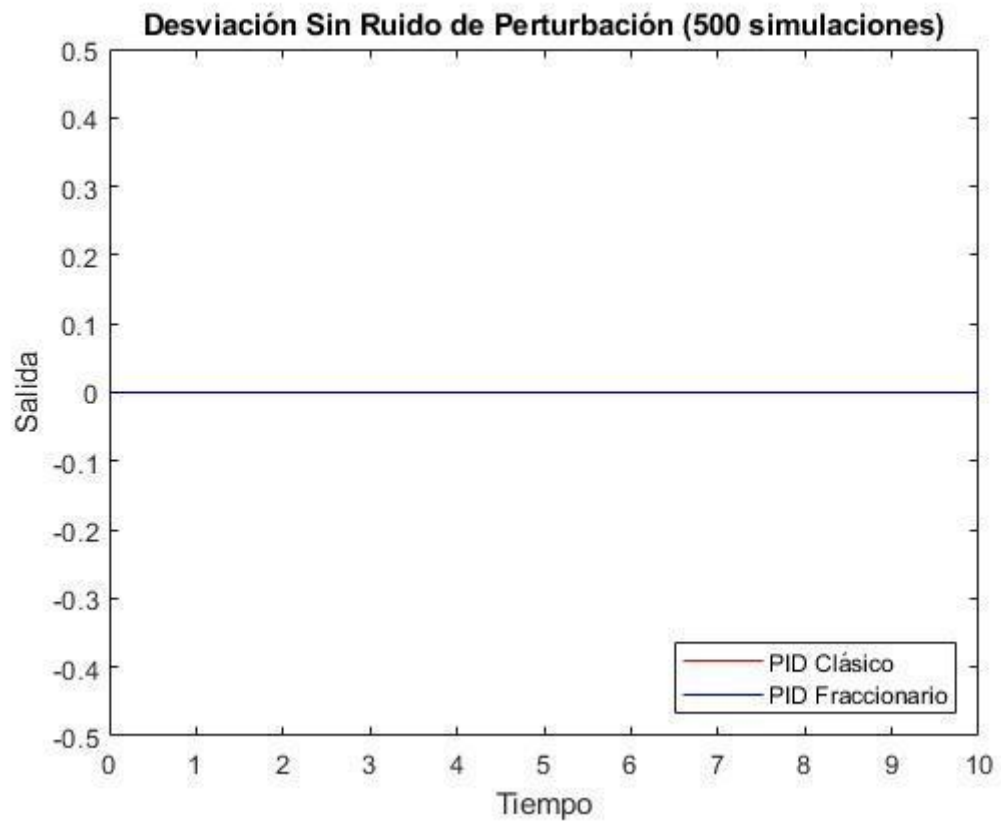


Figura 20. Desviación sin Ruido de Perturbación (500 simulaciones).



4.4. 500 simulaciones con Ruido de perturbación.

4.4.1. Salidas.

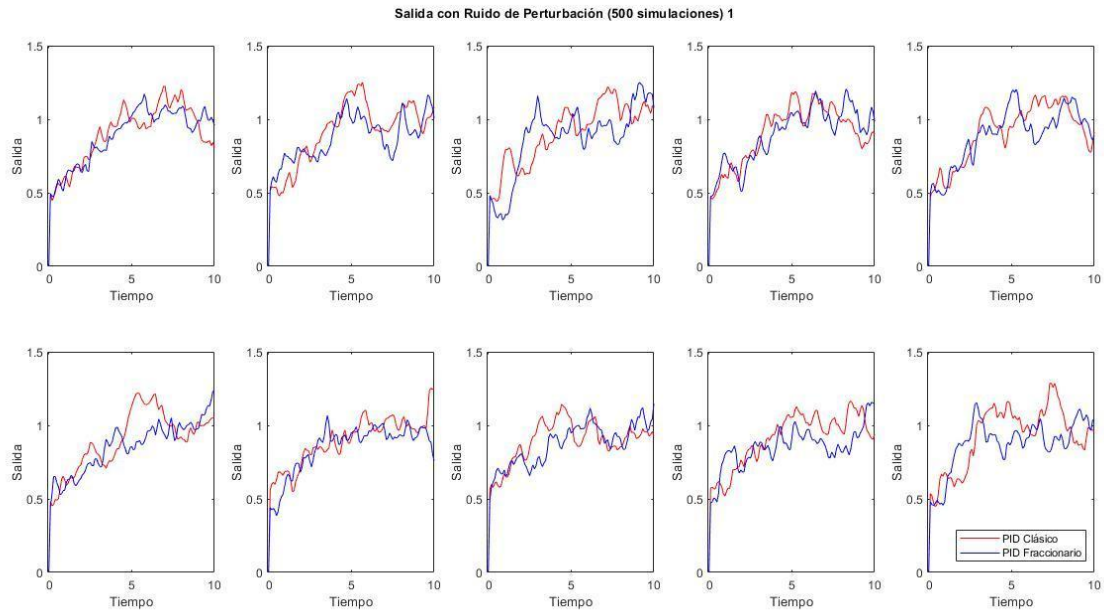


Figura 21. Salida con Ruido de Perturbación (500 simulaciones) 1.

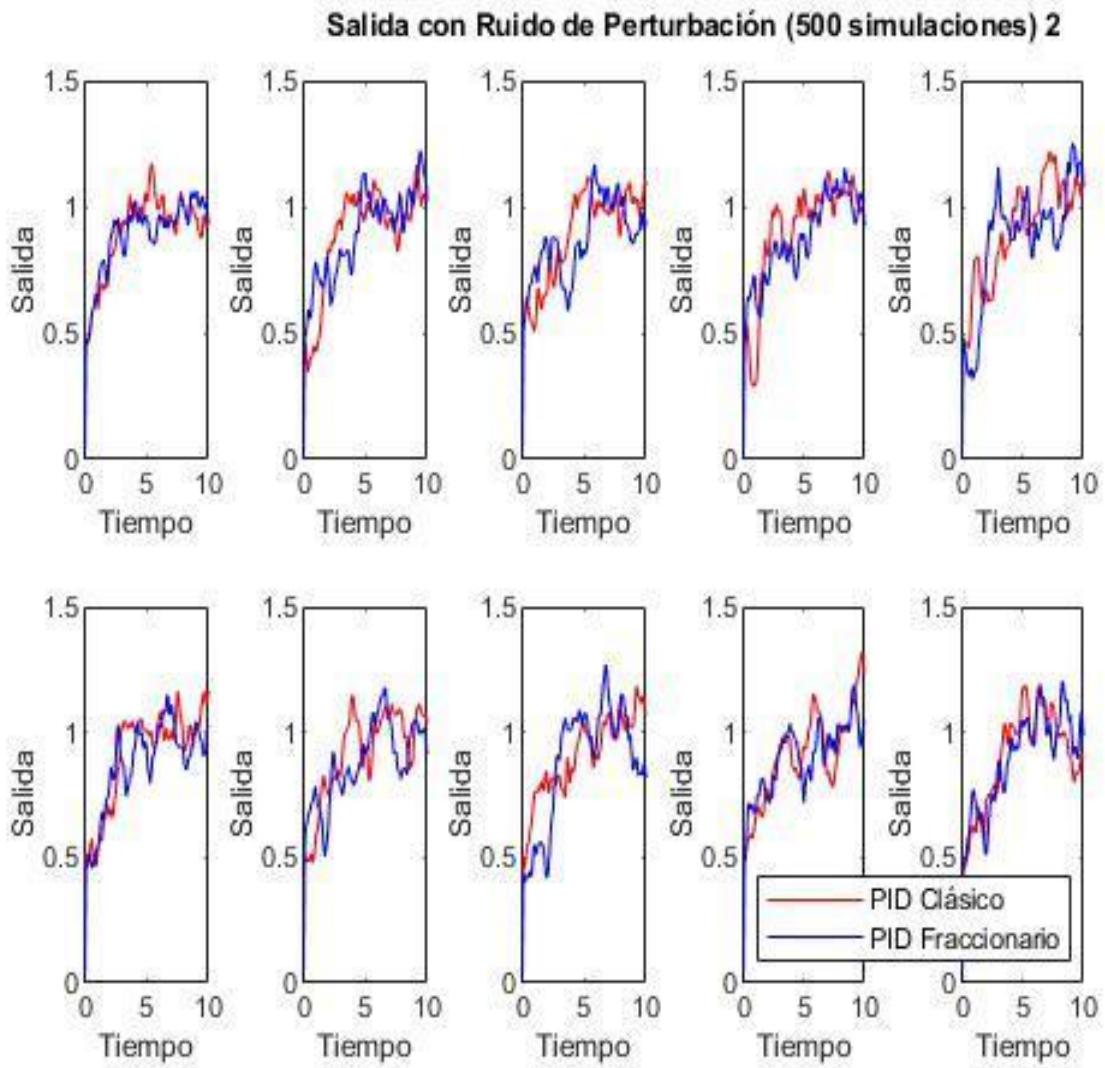


Figura 22. Salida con Ruido de Perturbación (500 simulaciones) 2.



4.4.2. Densidad de probabilidad.

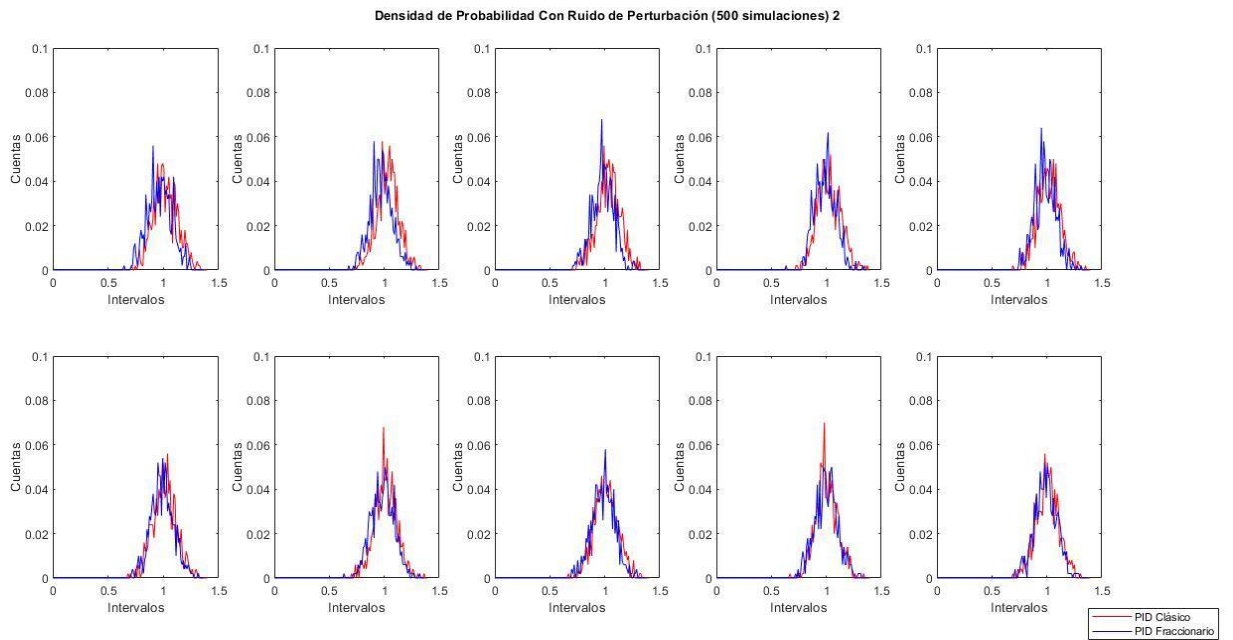
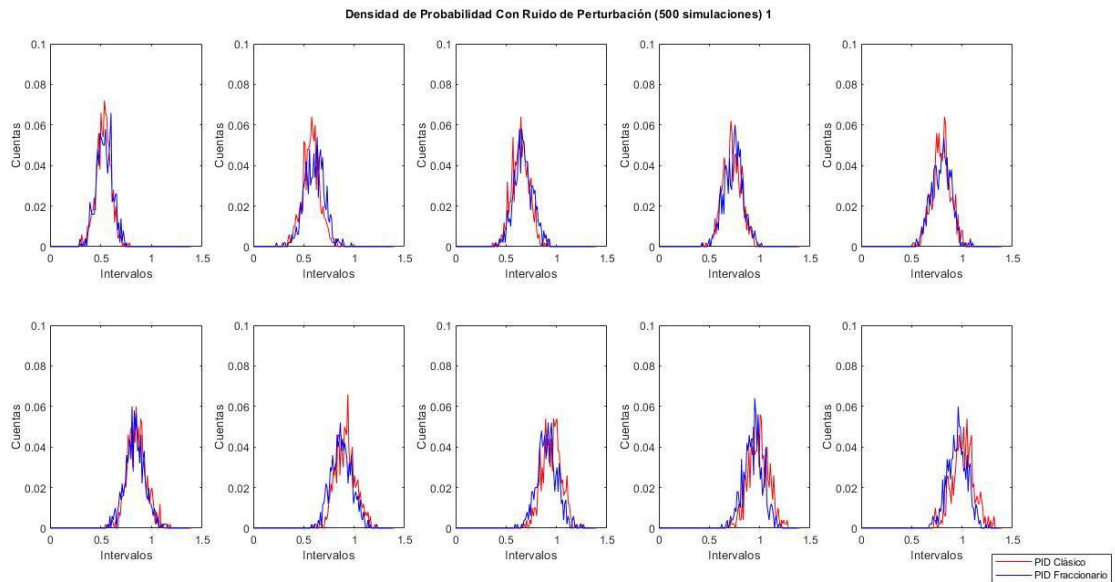


Figura 23. Densidad de Probabilidad con Ruido de Perturbación (500 simulaciones) 1.

Figura 24. Densidad de Probabilidad con Ruido de Perturbación (500 simulaciones) 2.



4.4.3. Media.

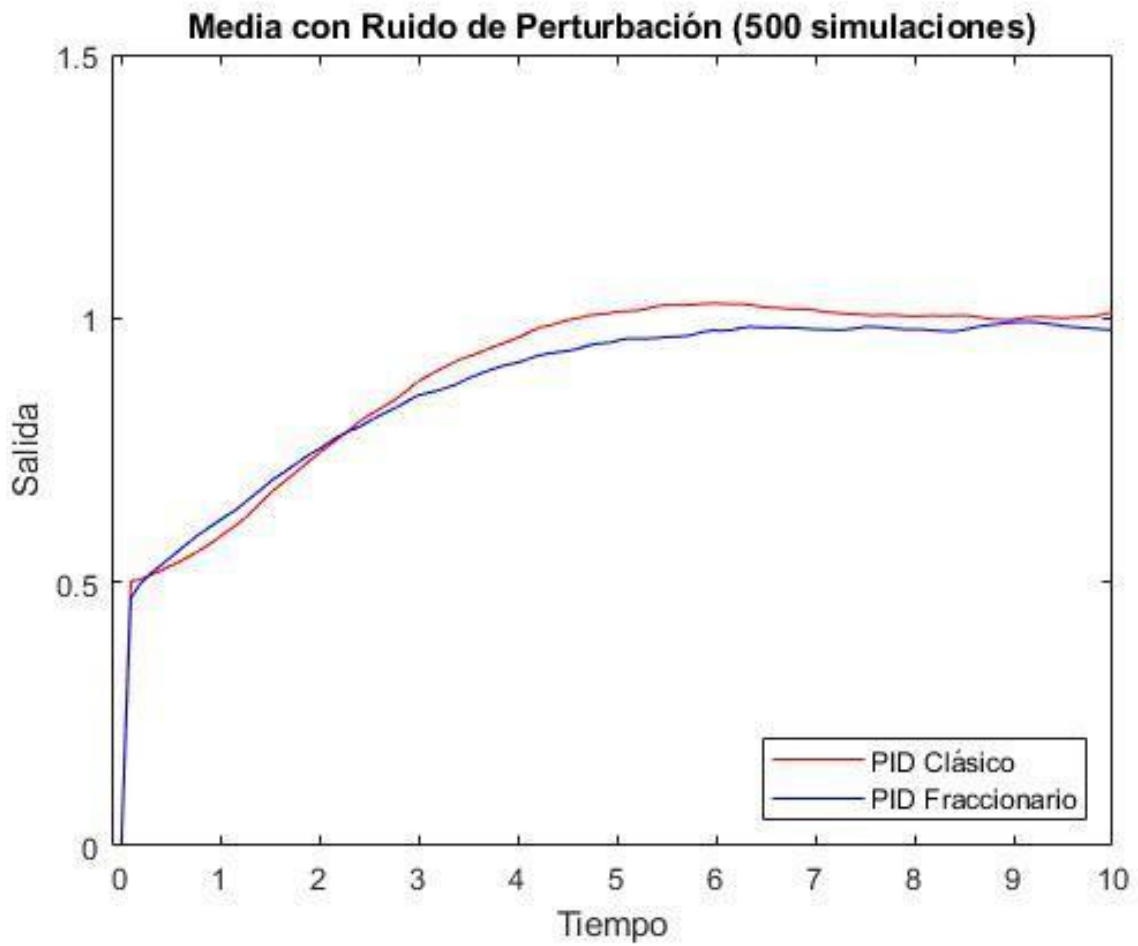


Figura 25. Media con Ruido de Perturbación (500 simulaciones).



4.4.4. Desviación.

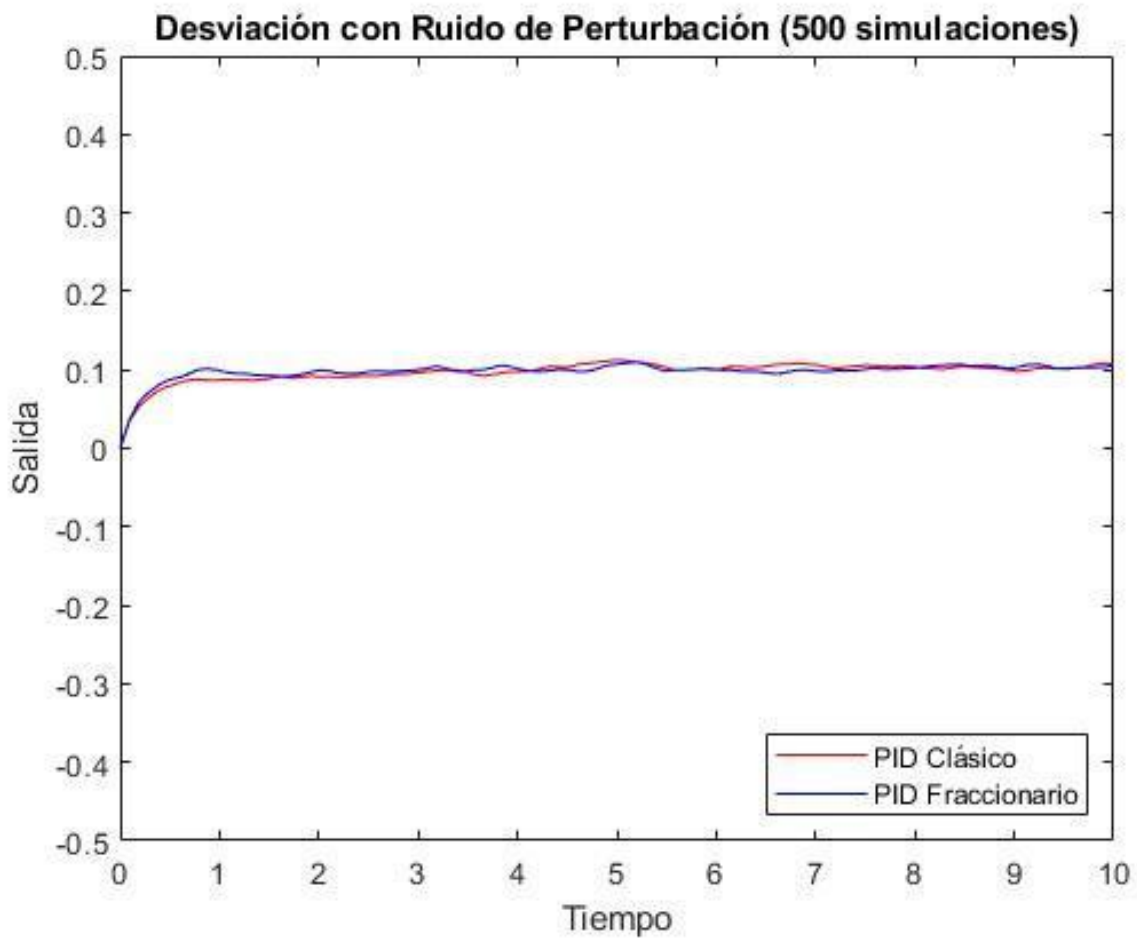


Figura 26. Media con Ruido de Perturbación (500 simulaciones).



5. Resultados (Alpha = -0.8 y Beta = 0.8)

5.1. 100 simulaciones sin Ruido de perturbación.

5.1.1. Salidas.

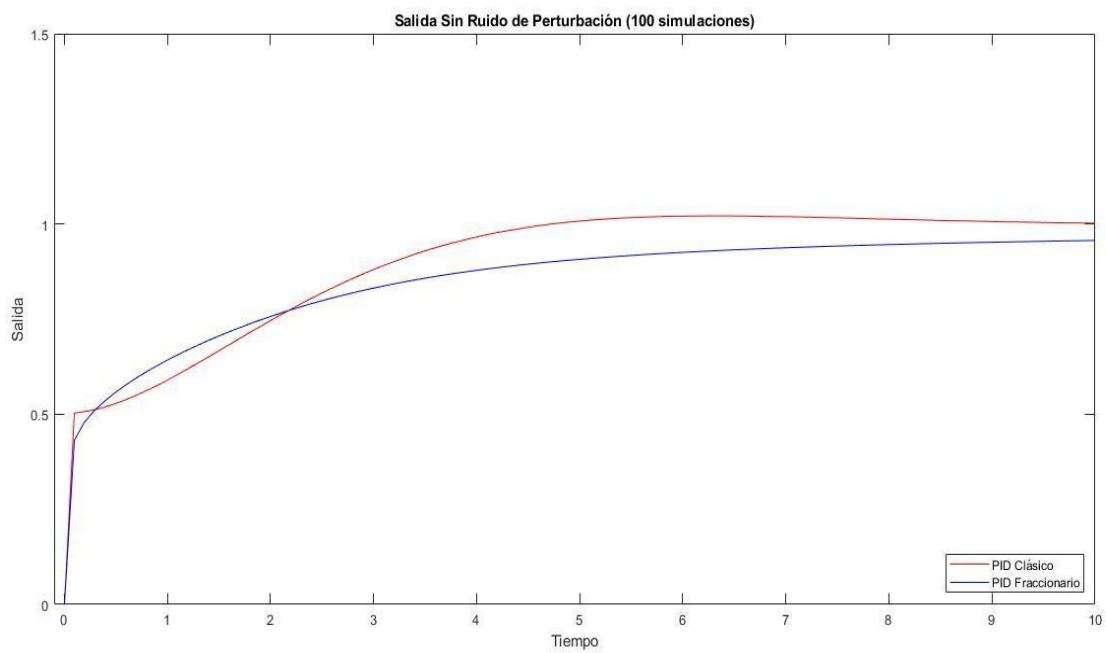


Figura 27. Salida sin Ruido de Perturbación (100 simulaciones).



5.1.2. Densidad de probabilidad.

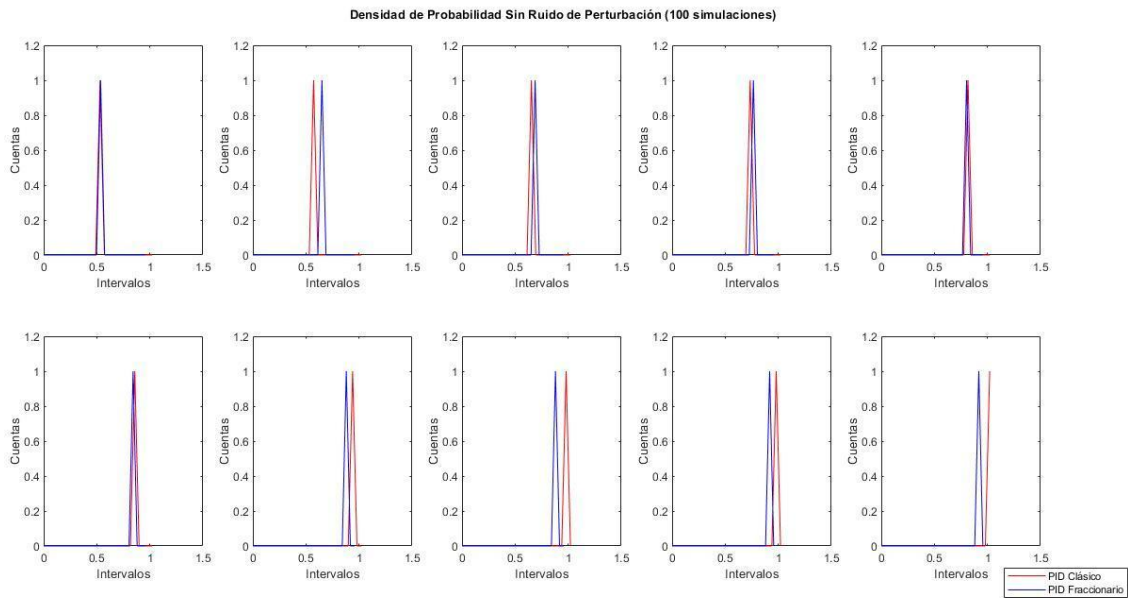


Figura 28. Densidad de Probabilidad sin Ruido de Perturbación (100 simulaciones).



5.1.3. Media.

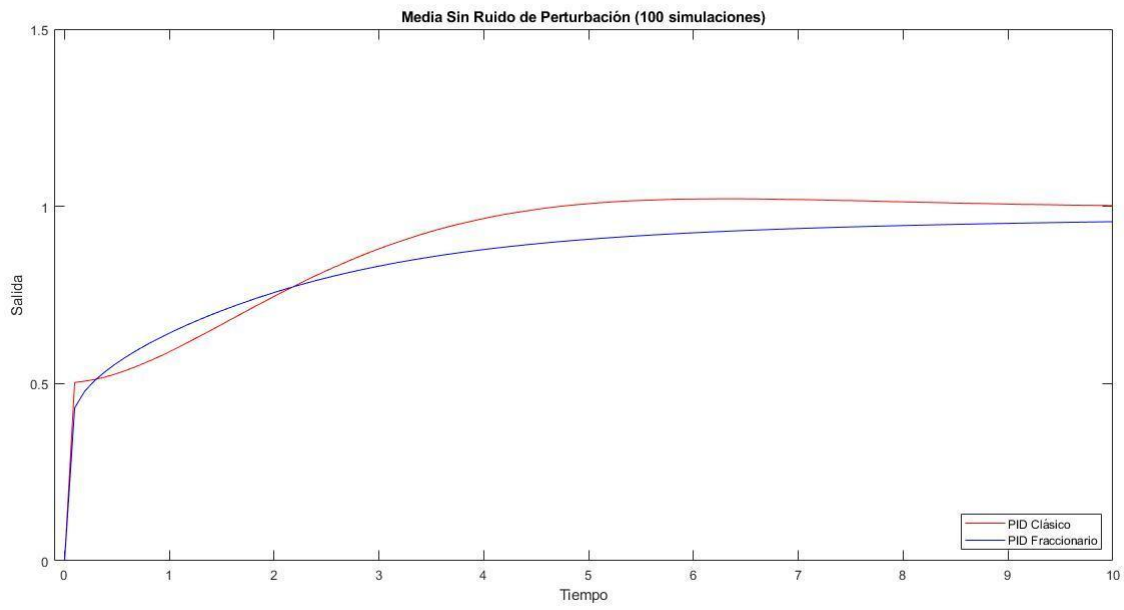


Figura 29. Media sin Ruido de Perturbación (100 simulaciones).



5.1.4. Desviación.

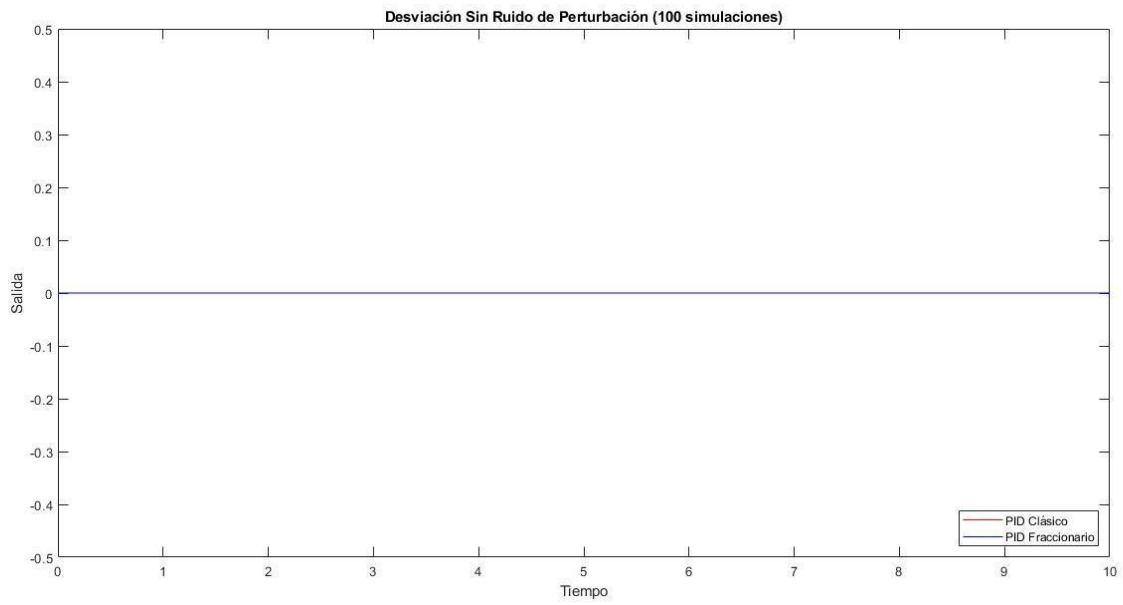


Figura 30.Desviación sin Ruido de Perturbación (100 simulaciones).



5.2. 100 simulaciones con Ruido de perturbación.

5.2.1. Salidas.

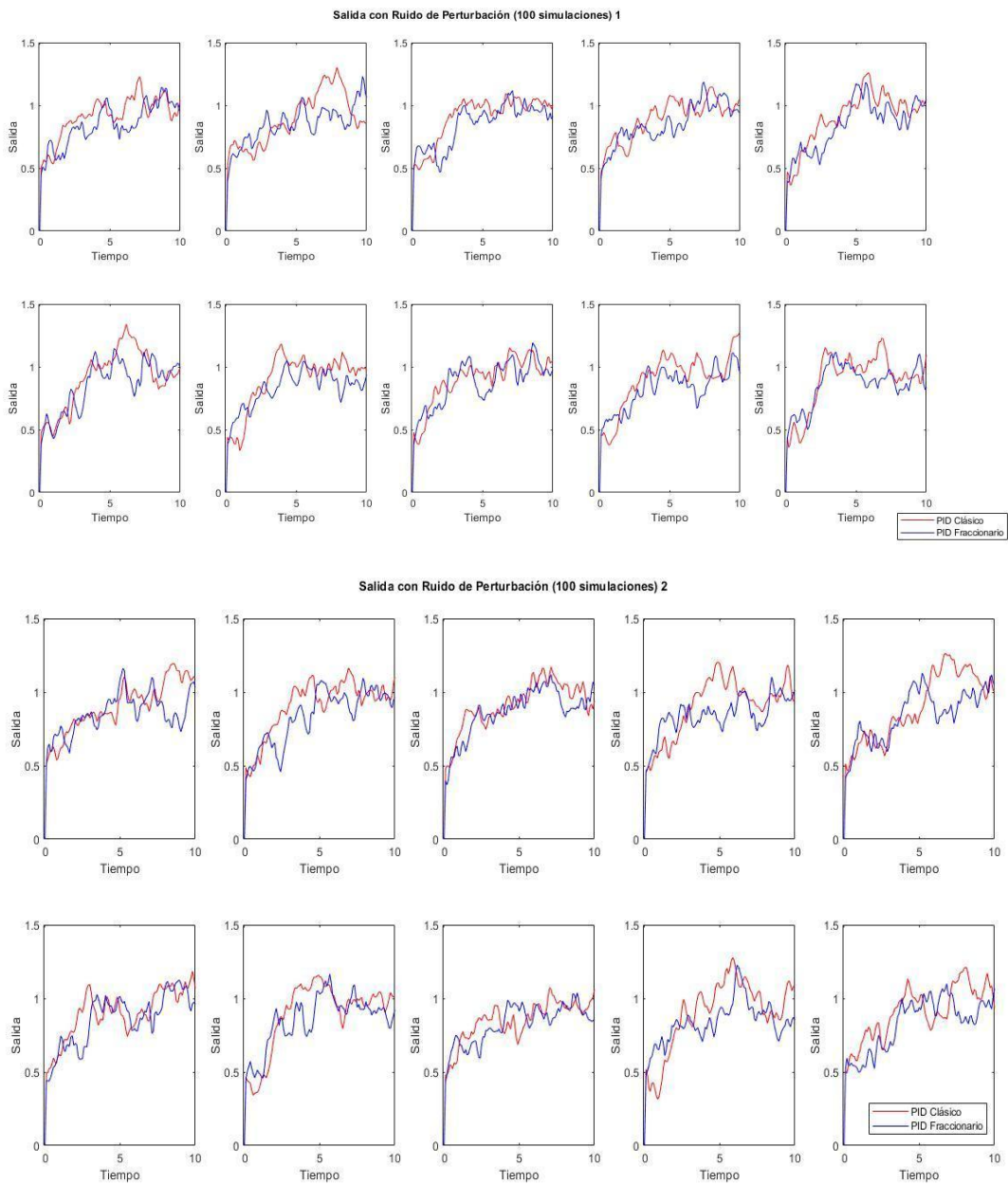


Figura 31.Salida con Ruido de Perturbación (100 simulaciones) 1.

Figura 32.Salida con Ruido de Perturbación (100 simulaciones) 2.



5.2.2. Densidad de probabilidad.

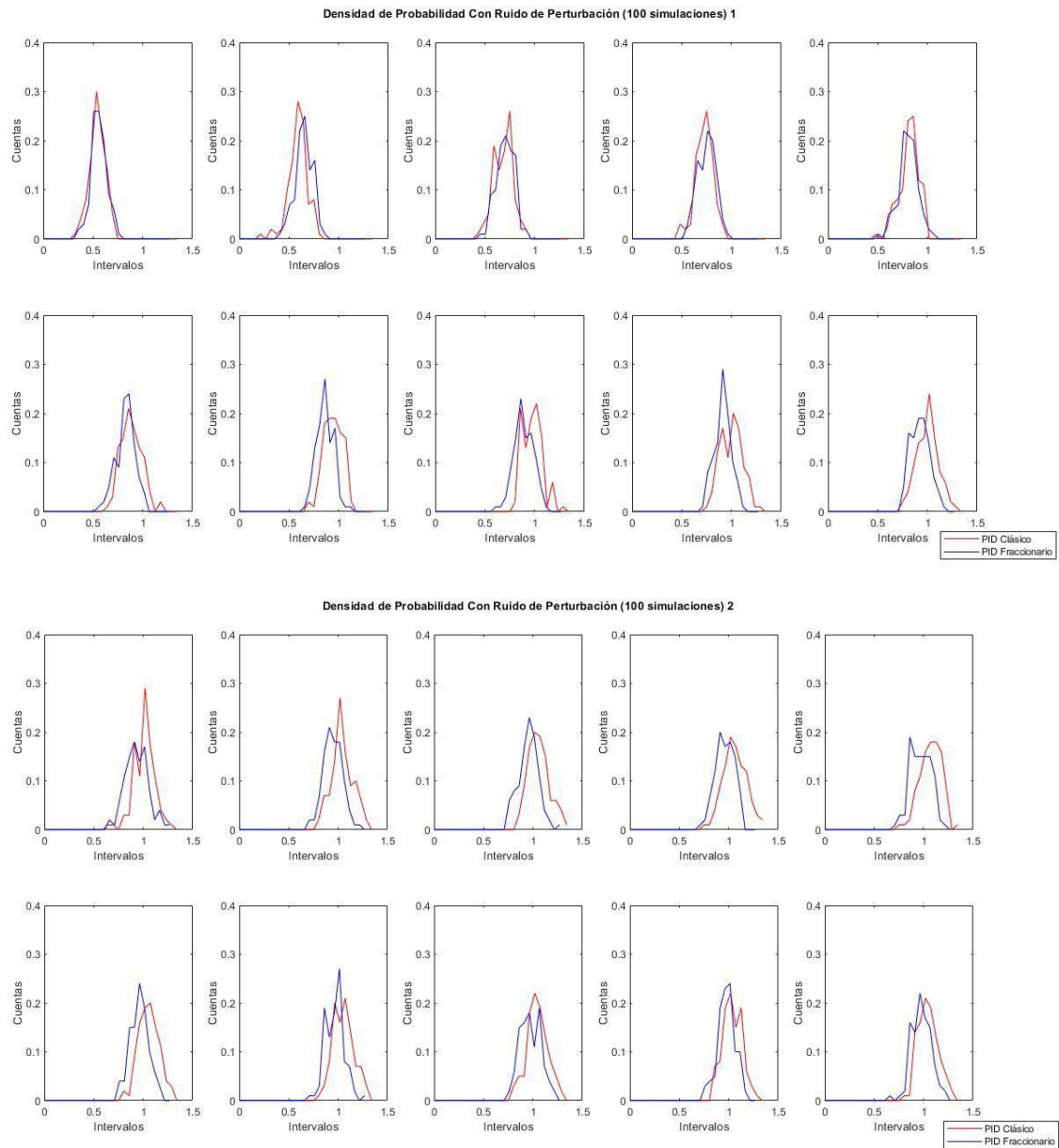


Figura 33. Densidad de Probabilidad con Ruido de Perturbación (100 simulaciones) 1.

Figura 34. Densidad de Probabilidad con Ruido de Perturbación (100 simulaciones) 2.



5.2.3. Media.

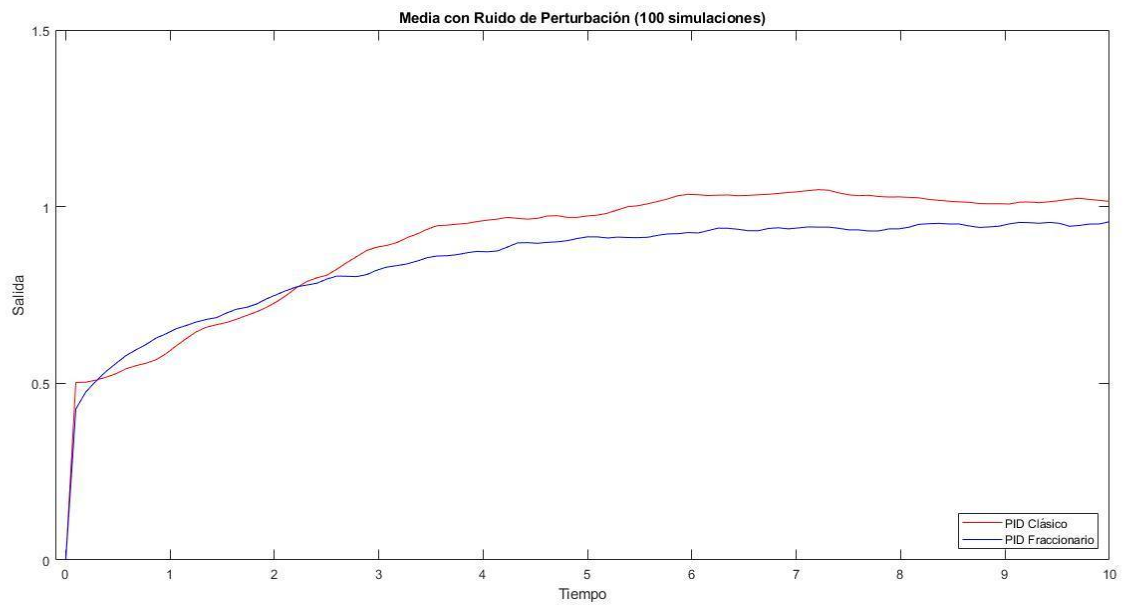


Figura 35. Media con Ruido de Perturbación (100 simulaciones).



5.2.4. Desviación.

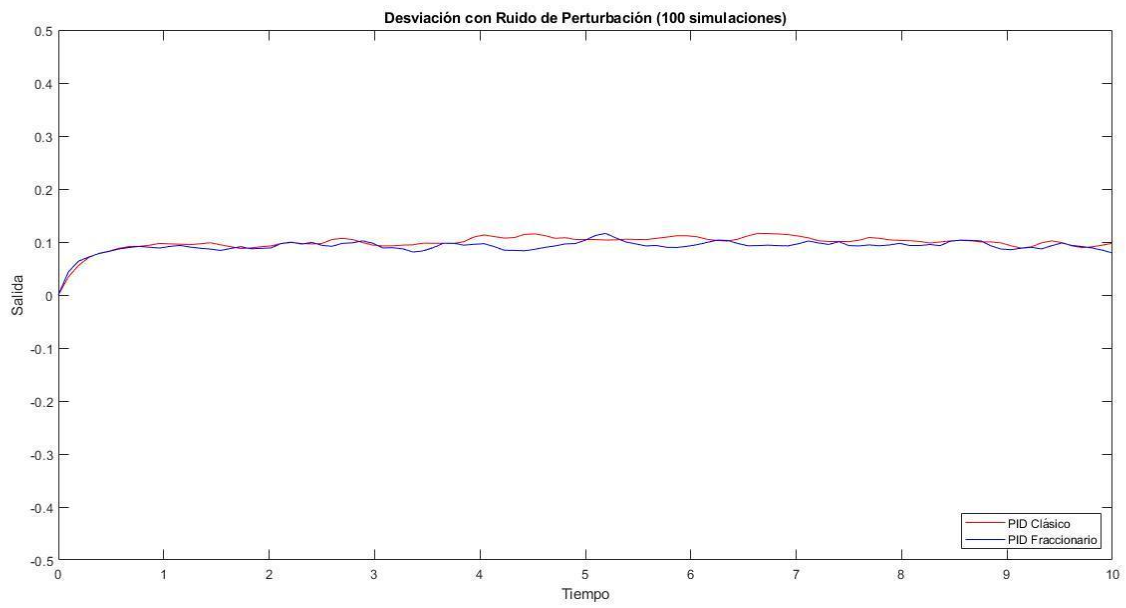


Figura 36. Media con Ruido de Perturbación (100 simulaciones).



5.3. 500 simulaciones sin Ruido de perturbación.

5.3.1. Salidas.

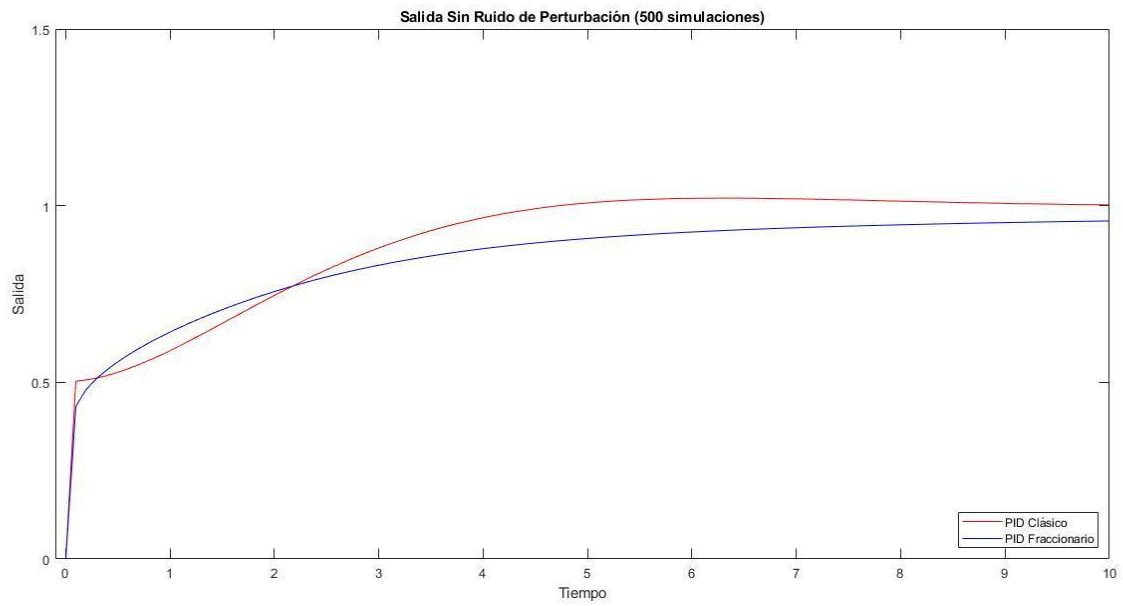


Figura 37. Salida sin Ruido de Perturbación (500 simulaciones).



5.3.2. Densidad de probabilidad.

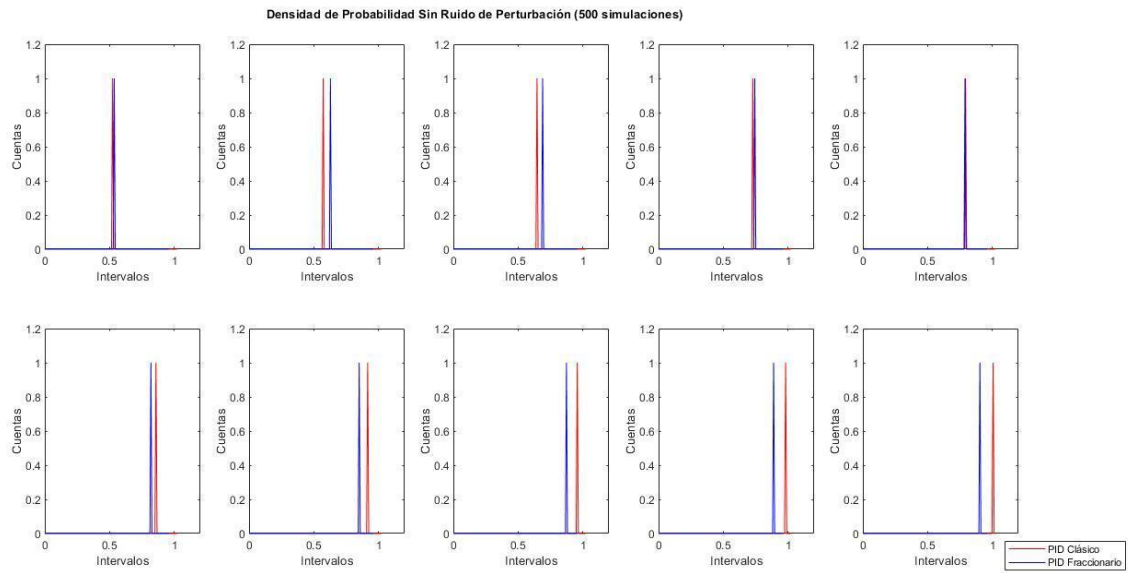


Figura 38. Densidad de Probabilidad sin Ruido de Perturbación (500 simulaciones).



5.3.3. Media.

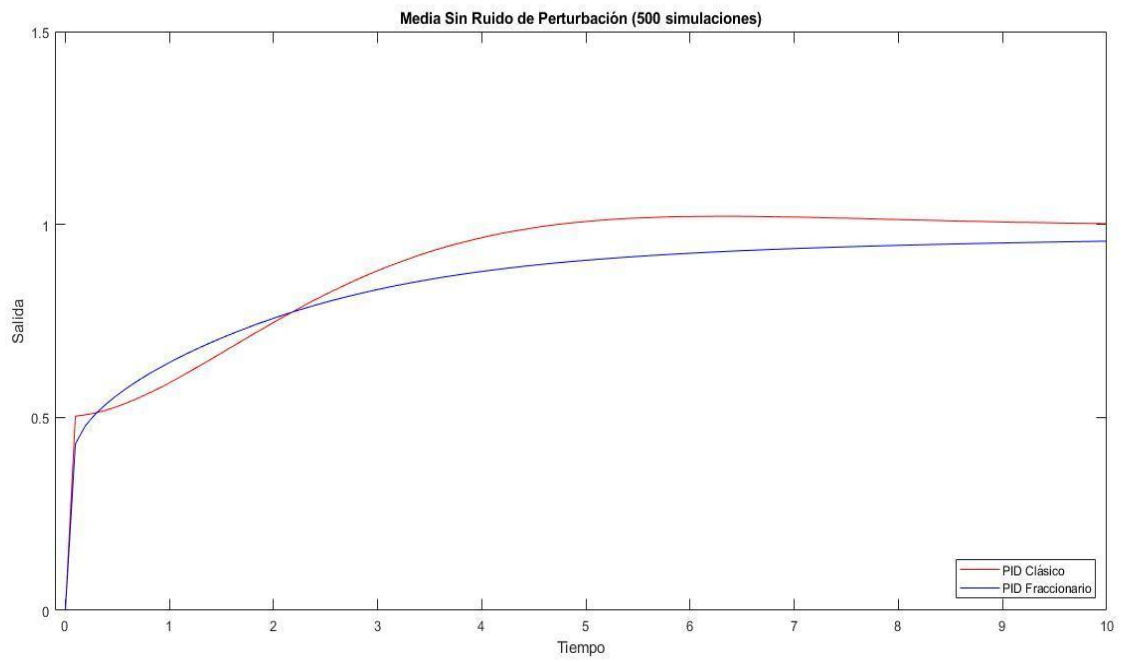


Figura 39. Media sin Ruido de Perturbación (500 simulaciones).



5.3.4. Desviación.

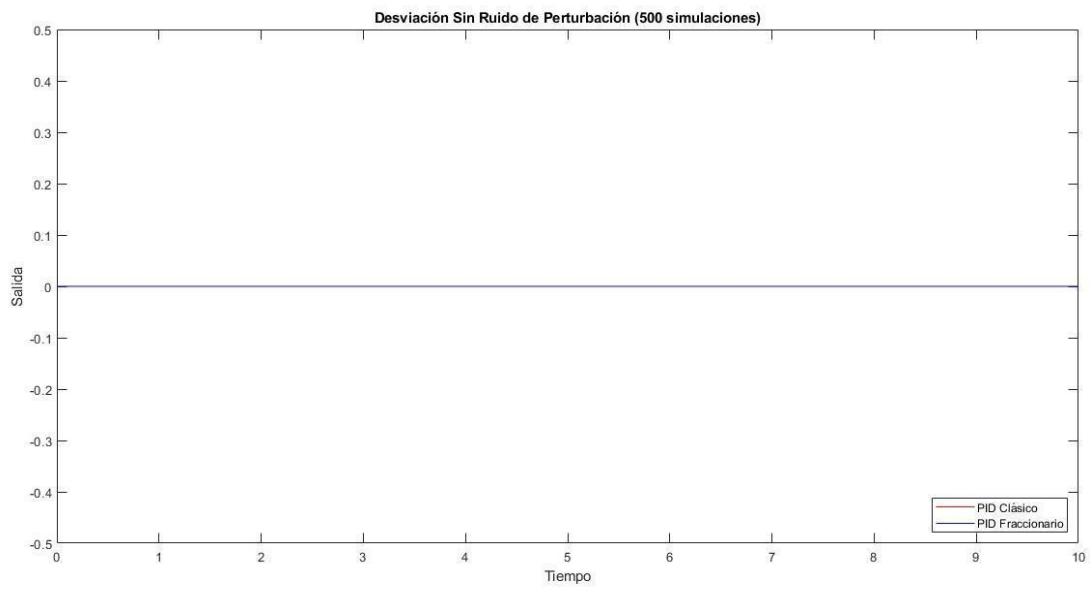


Figura 40.Desviación sin Ruido de Perturbación (500 simulaciones).



5.4. 500 simulaciones con Ruido de perturbación.

5.4.1. Salidas.

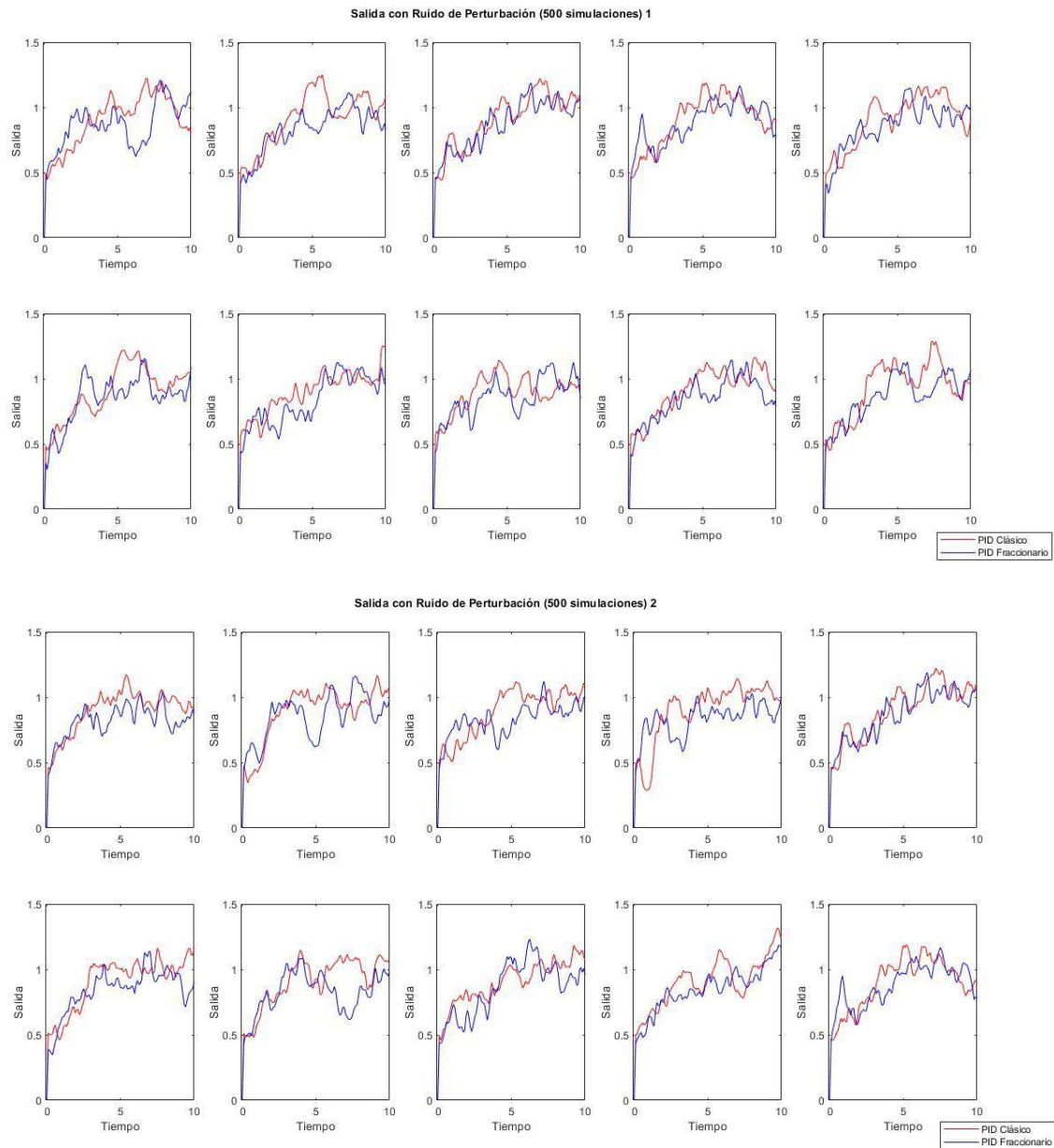


Figura 41. Salida con Ruido de Perturbación (500 simulaciones) 1.

Figura 42. Salida con Ruido de Perturbación (500 simulaciones) 2.



5.4.2. Densidad de probabilidad.

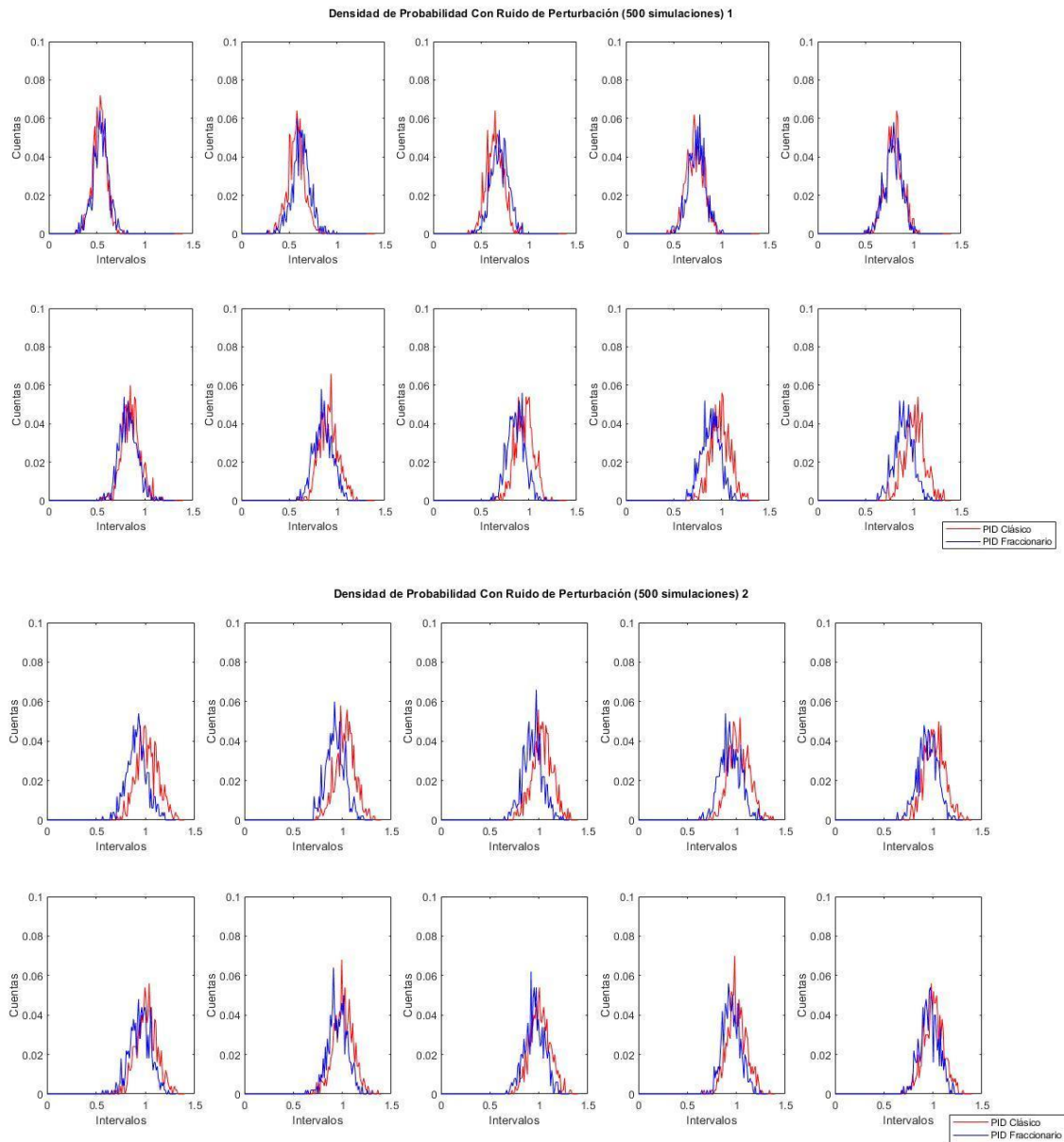


Figura 43. Densidad de Probabilidad con Ruido de Perturbación (500 simulaciones) 1.
Figura 44. Densidad de Probabilidad con Ruido de Perturbación (500 simulaciones) 2.



5.4.3. Media.

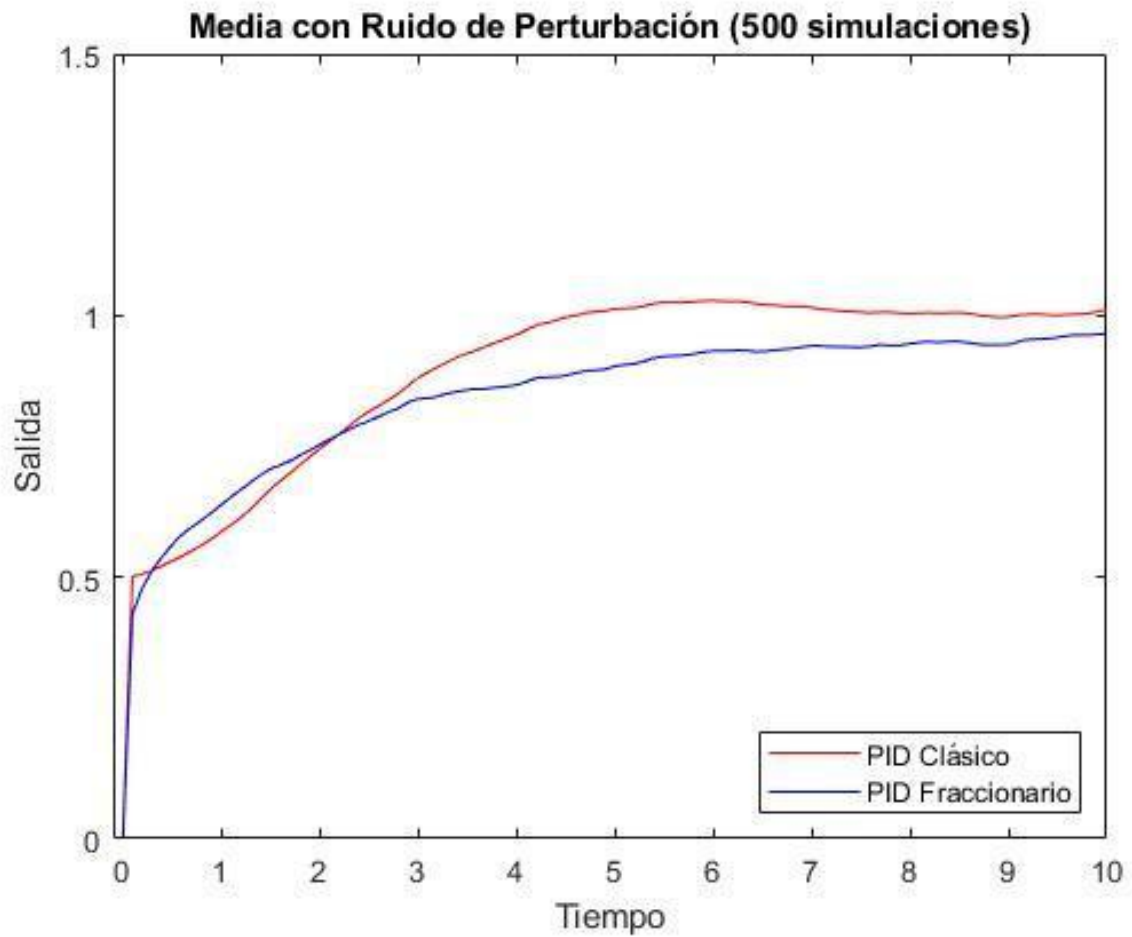


Figura 45. Media con Ruido de Perturbación (500 simulaciones).



5.4.4. Desviación.

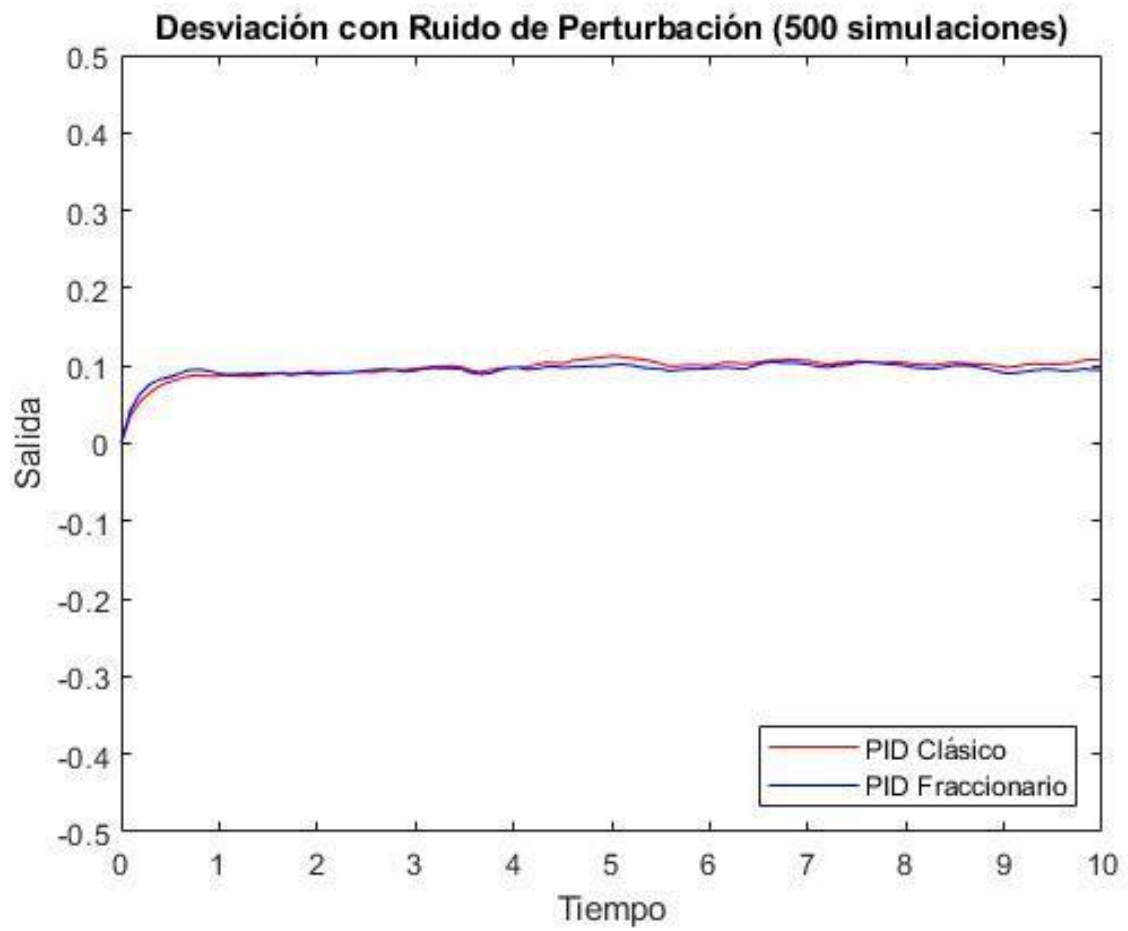


Figura 46.Desviación con Ruido de Perturbación (500 simulaciones).



6. Resultados (Alpha = -0.6 y Beta = 0.6)

6.1. 100 simulaciones sin Ruido de perturbación.

6.1.1. Salidas.

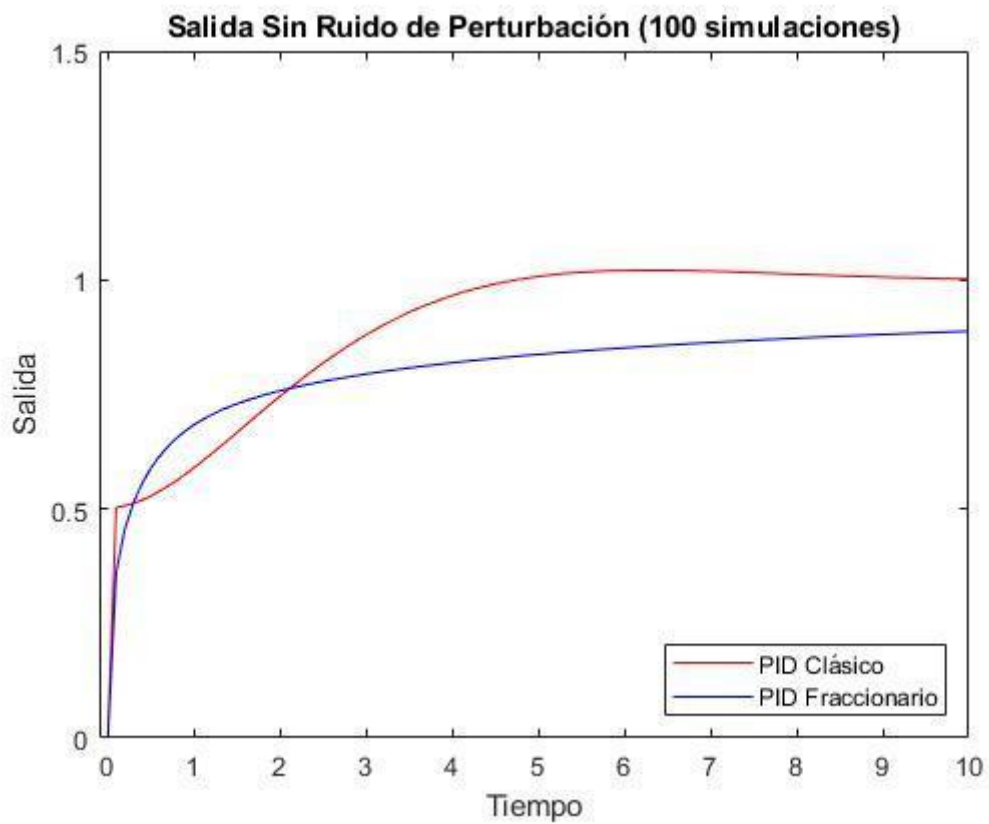


Figura 47. Salida sin Ruido de Perturbación (100 simulaciones).



6.1.2. Densidad de probabilidad.

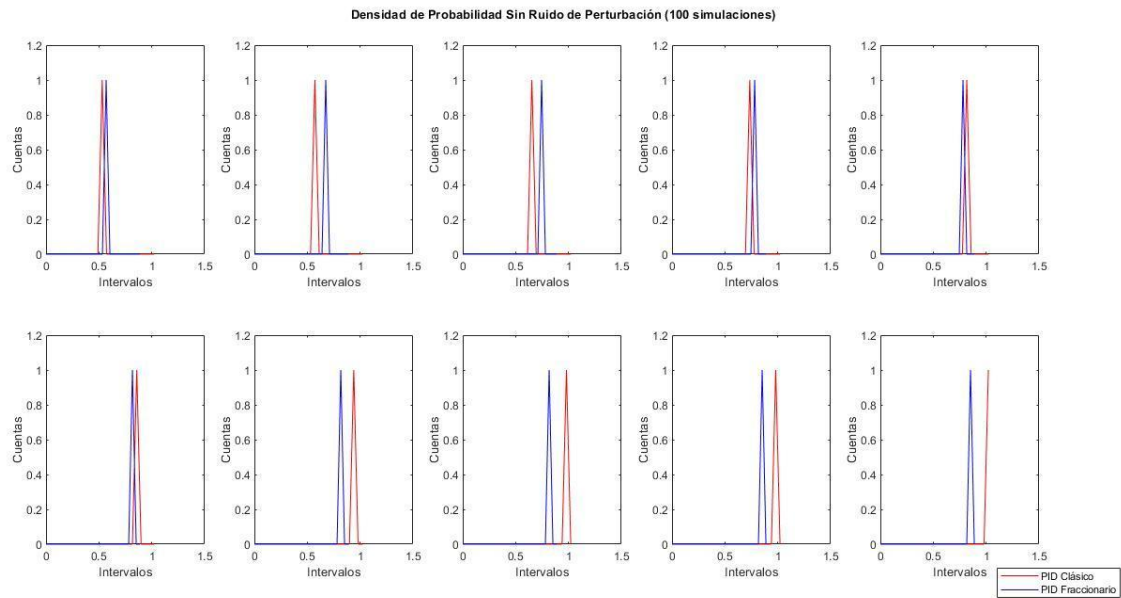


Figura 48. Densidad de Probabilidad sin Ruido de Perturbación (100 simulaciones).



6.1.3. Media.

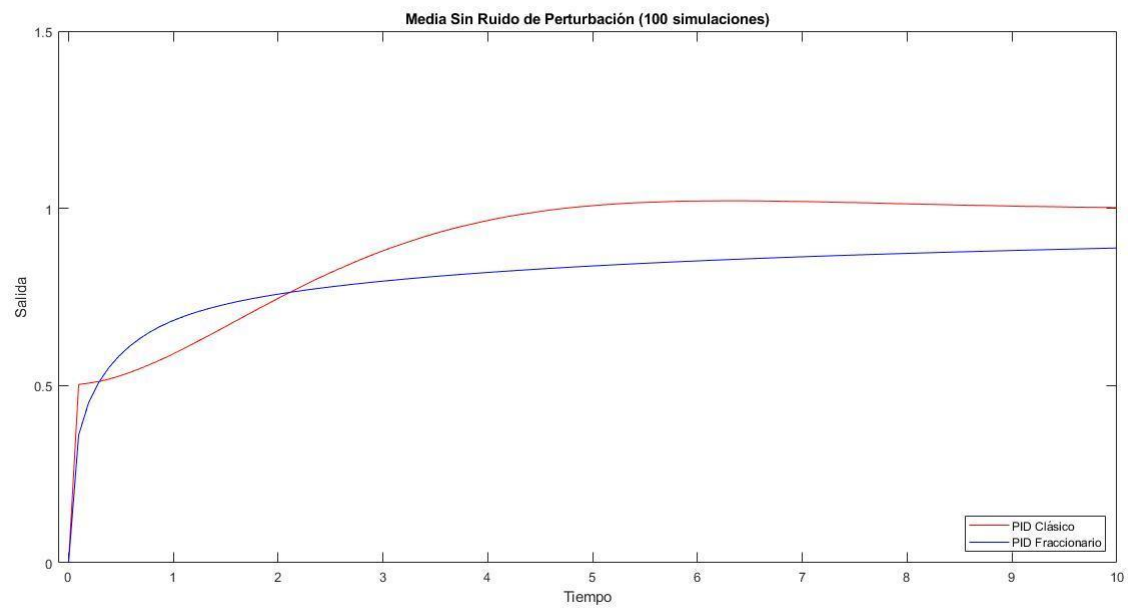


Figura 49. Media sin Ruido de Perturbación (100 simulaciones).



6.1.4. Desviación.

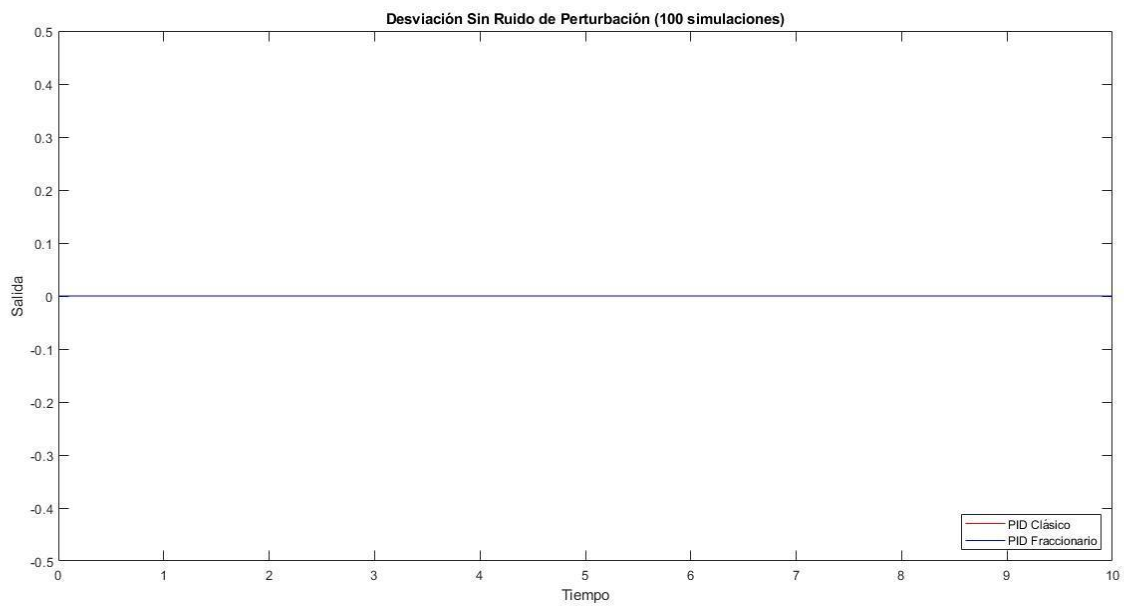


Figura 50.Desviación sin Ruido de Perturbación (100 simulaciones).



6.2. 100 simulaciones con Ruido de perturbación.

6.2.1. Salidas.

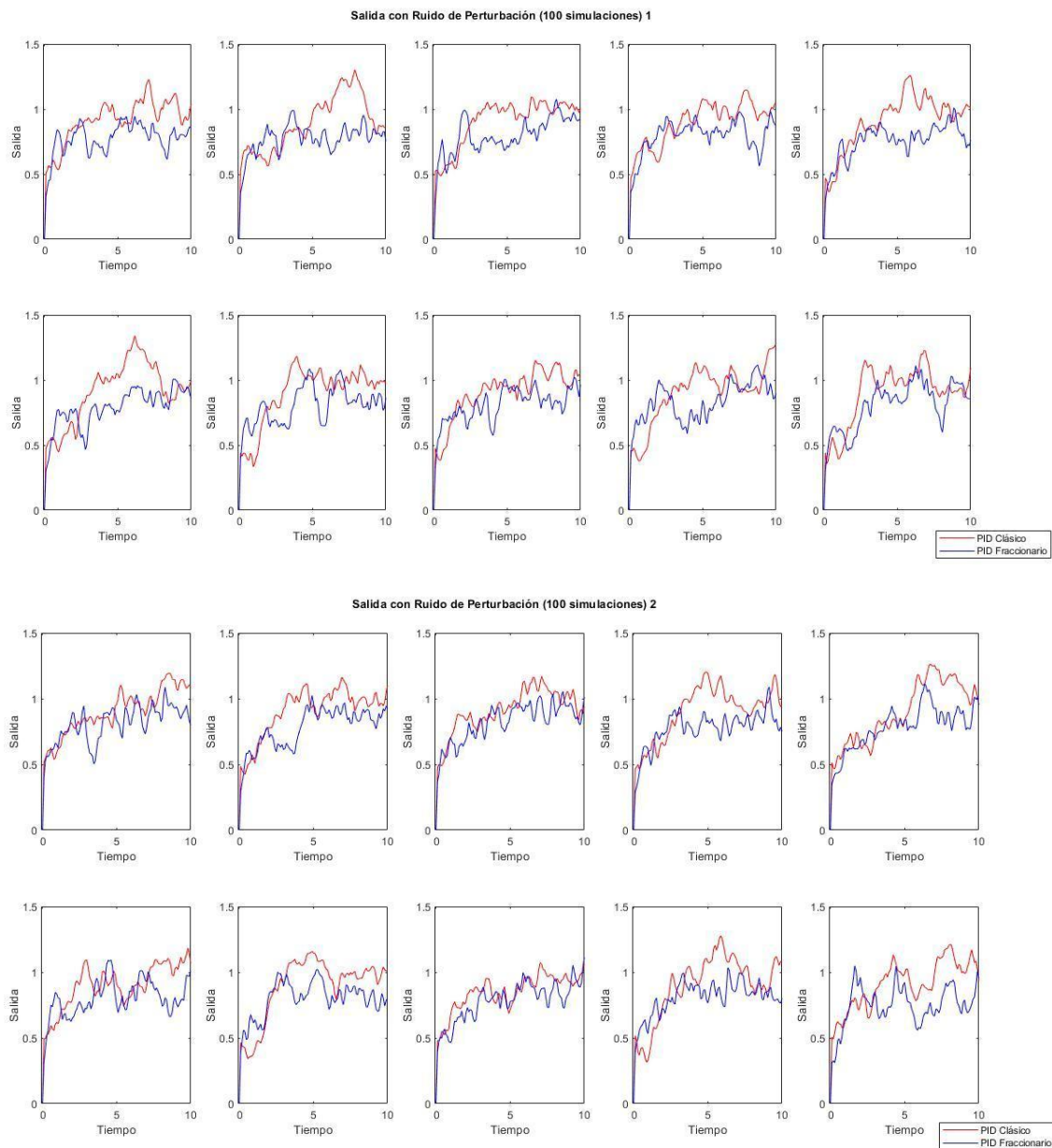


Figura 51. Salida con Ruido de Perturbación (100 simulaciones) 1.

Figura 52. Salida con Ruido de Perturbación (100 simulaciones) 2.



6.2.2. Densidad de probabilidad.

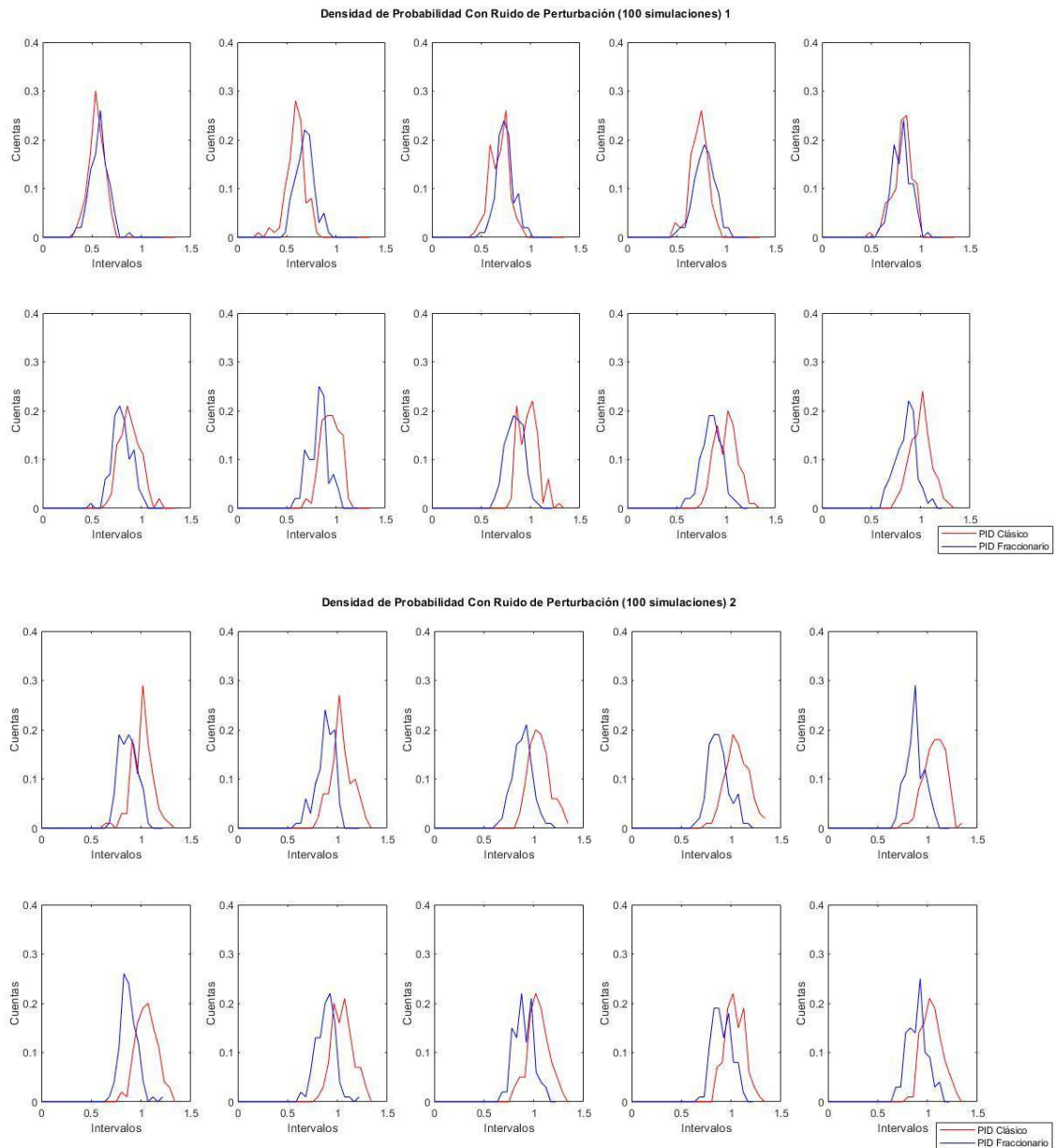


Figura 53. Densidad de Probabilidad con Ruido de Perturbación (100 simulaciones) 1.

Figura 54. Densidad de Probabilidad con Ruido de Perturbación (100 simulaciones) 2.



6.2.3. Media.

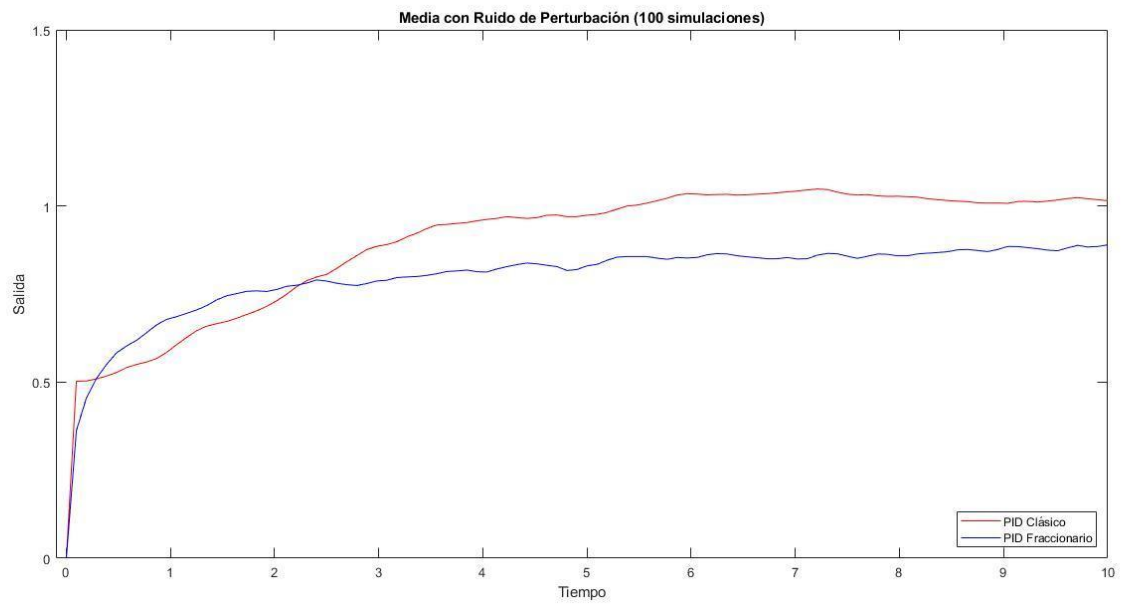


Figura 55. Media con Ruido de Perturbación (100 simulaciones).



6.2.4. Desviación.

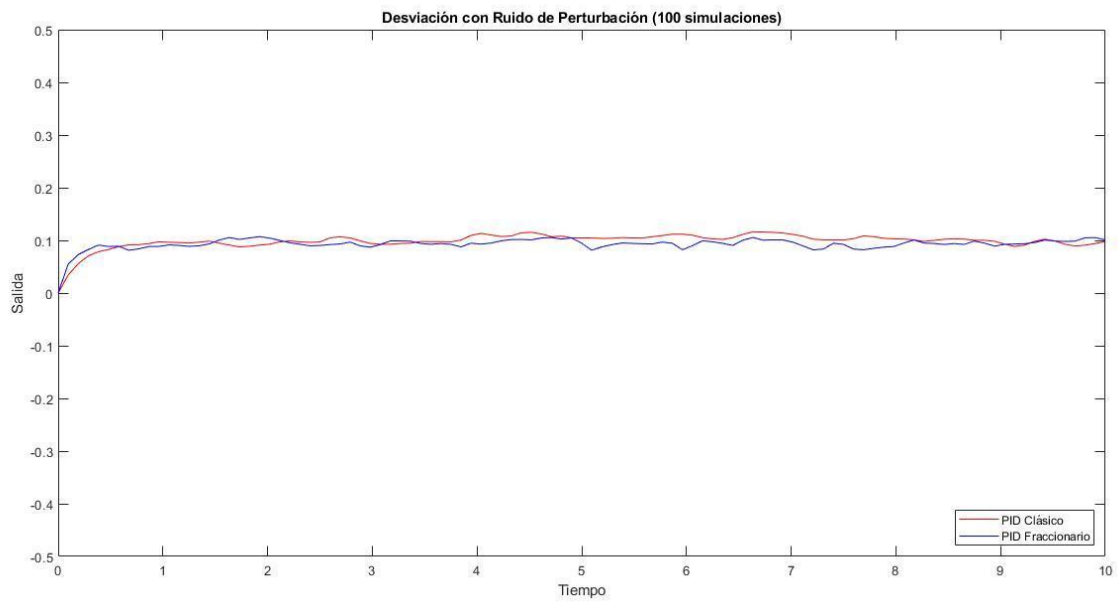


Figura 56.Desviación con Ruido de Perturbación (100 simulaciones).



6.3. 500 simulaciones sin Ruido de perturbación.

6.3.1. Salidas.

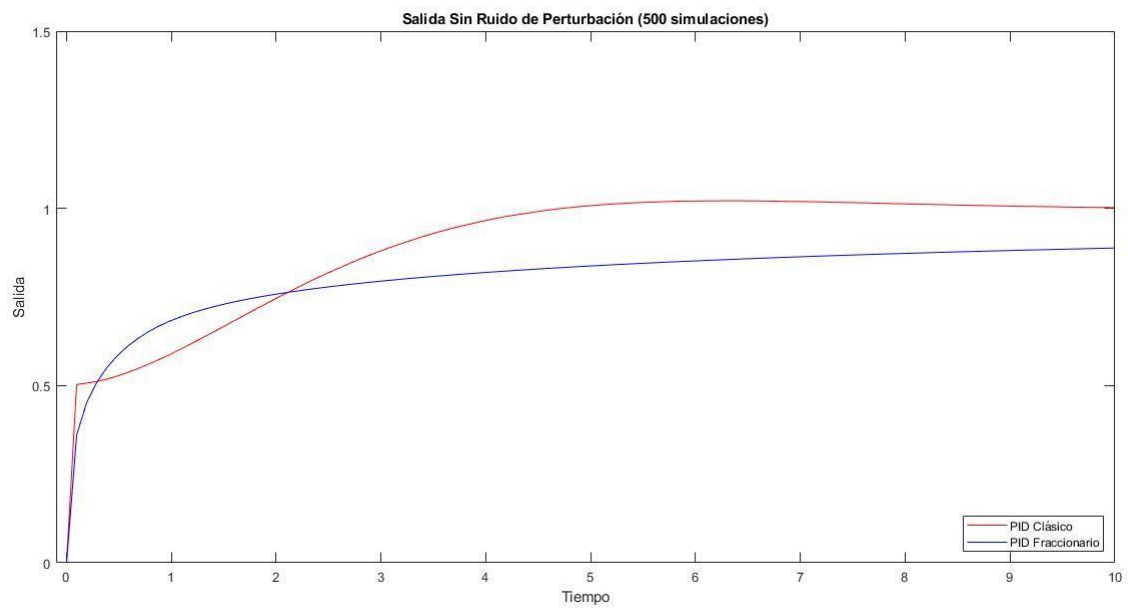


Figura 57. Salida sin Ruido de Perturbación (500 simulaciones).



6.3.2. Densidad de probabilidad.

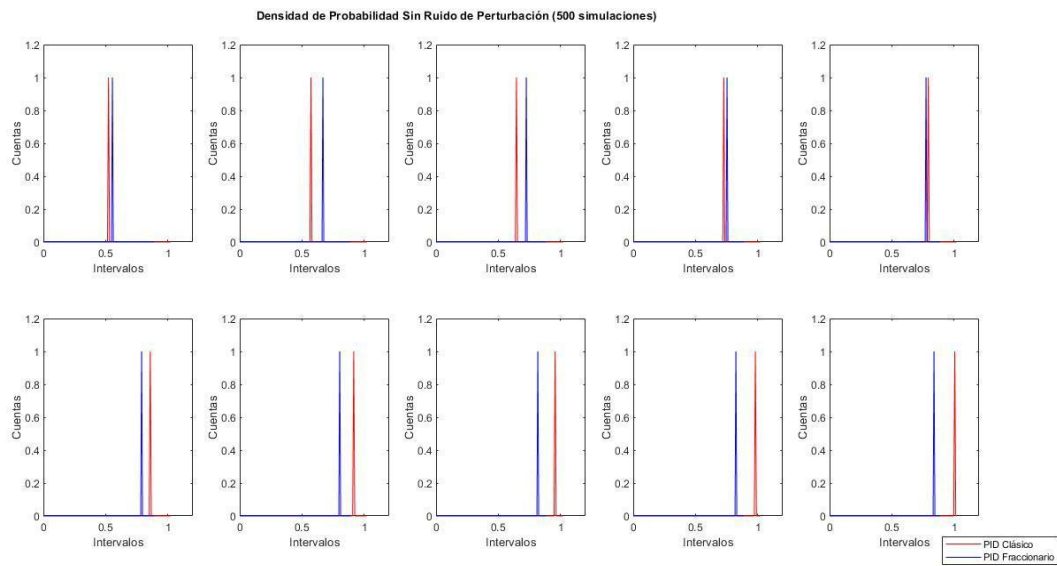


Figura 58. Densidad de Probabilidad sin Ruido de Perturbación (500 simulaciones).

6.3.3. Media.

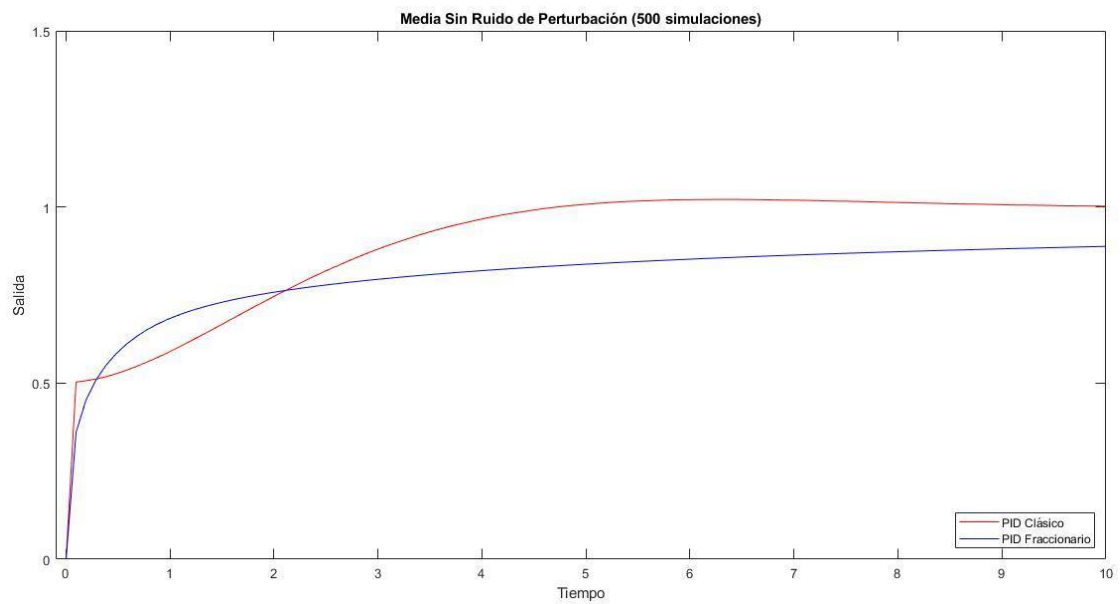


Figura 59. Media sin Ruido de Perturbación (500 simulaciones).



6.3.4. Desviación.

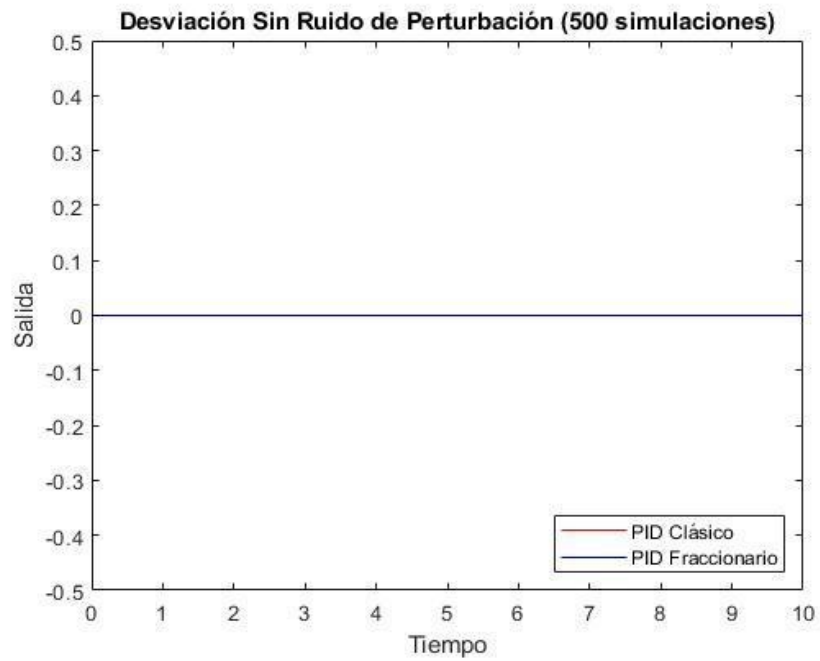


Figura 60.Desviación sin Ruido de Perturbación (500 simulaciones).



6.4. 500 simulaciones con Ruido de perturbación.

6.4.1. Salidas.

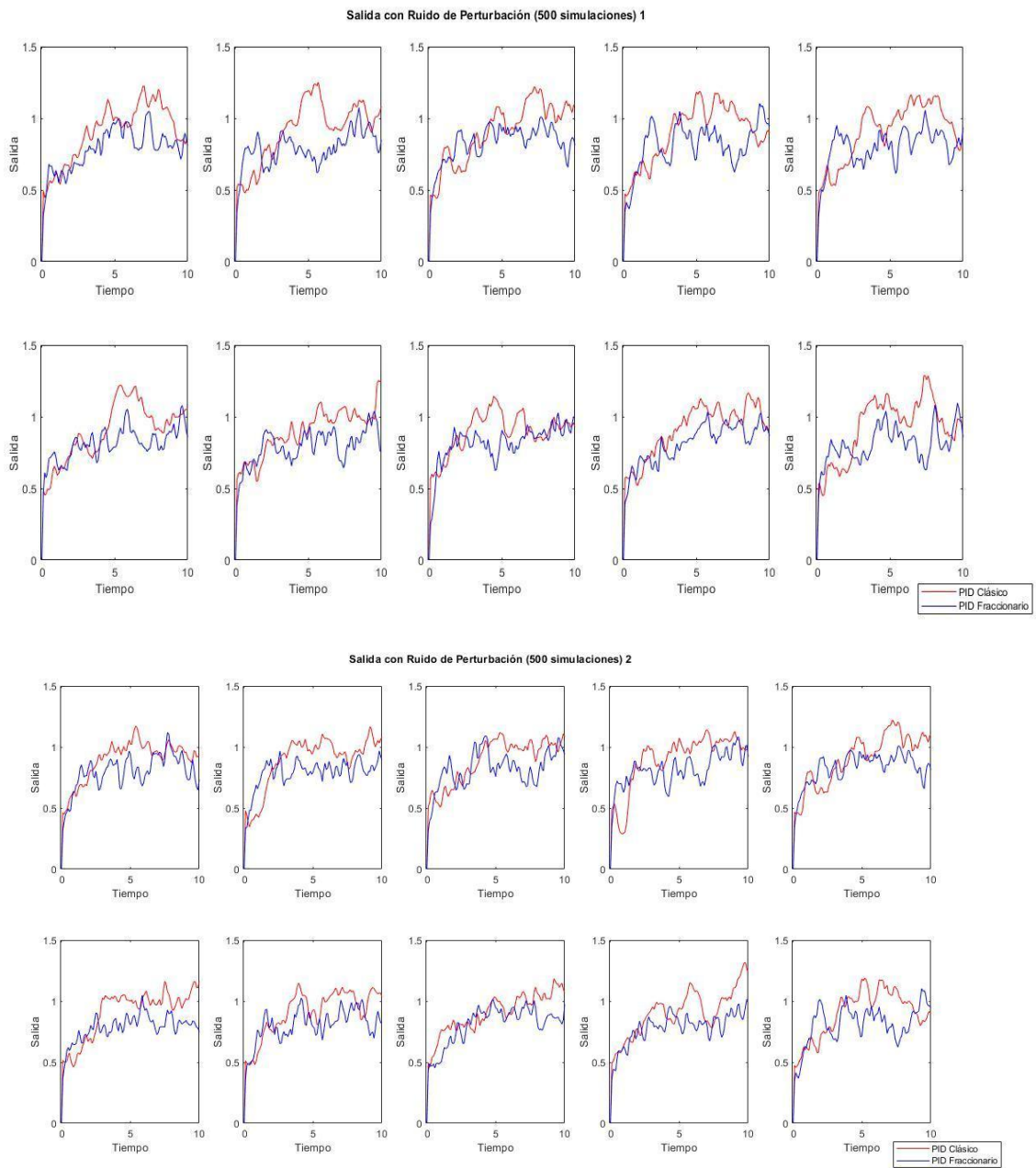


Figura 61. Salida con Ruido de Perturbación (500 simulaciones) 1.

Figura 62. Salida con Ruido de Perturbación (500 simulaciones) 2.



6.4.2. Densidad de probabilidad.

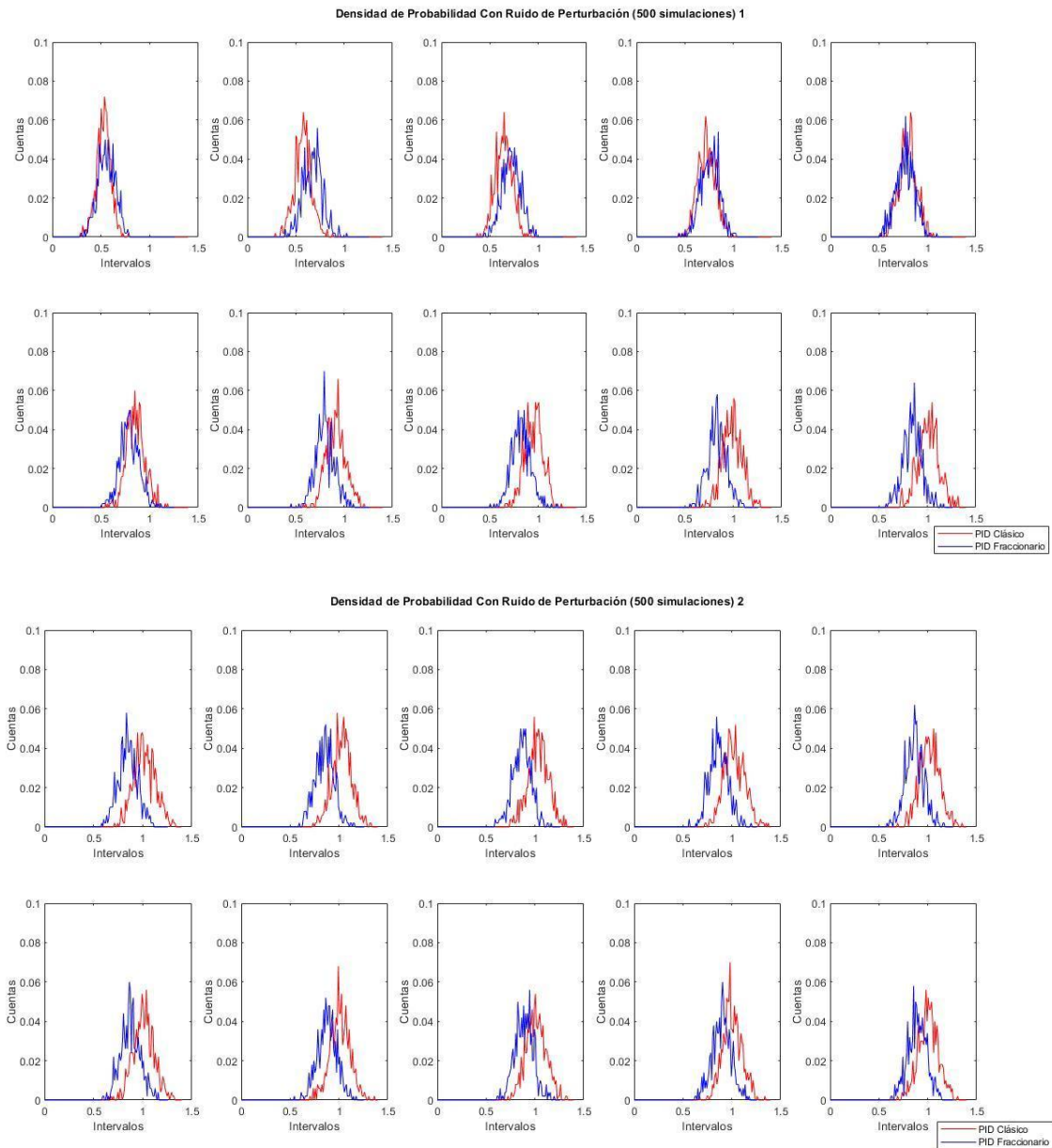


Figura 63. Densidad de Probabilidad con Ruido de Perturbación (500 simulaciones) 1.

Figura 64. Densidad de Probabilidad con Ruido de Perturbación (500 simulaciones) 2.



6.4.3. Media.

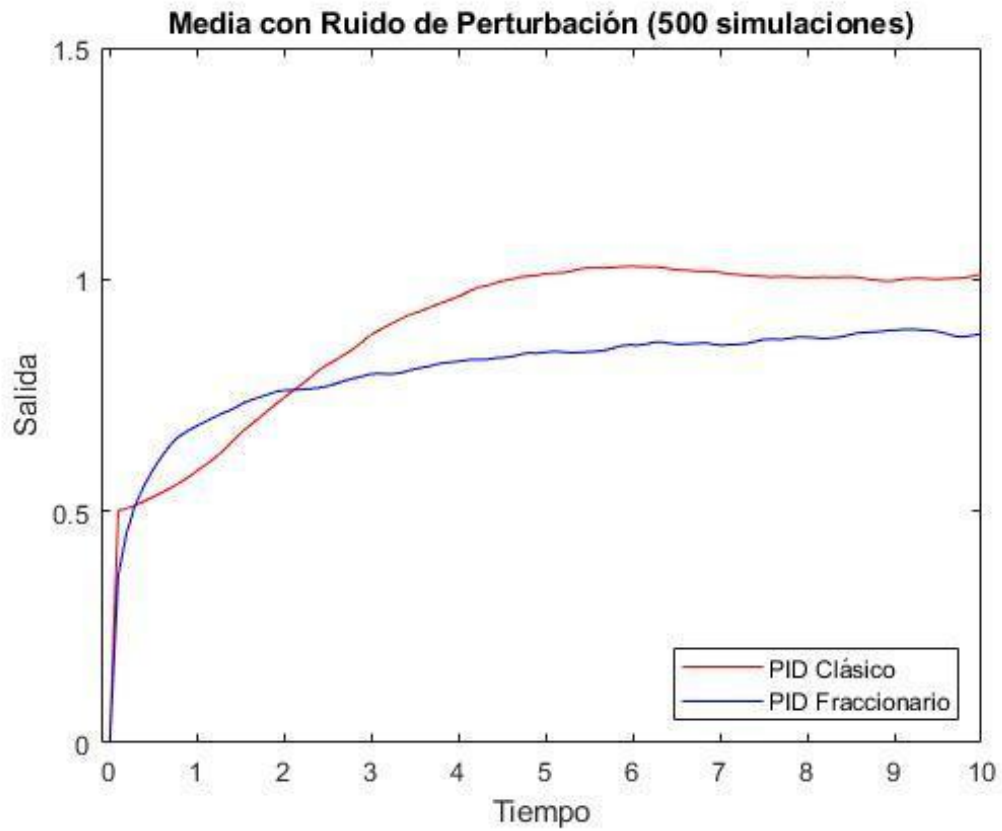


Figura 65. Media con Ruido de Perturbación (500 simulaciones).



6.4.4. Desviación.

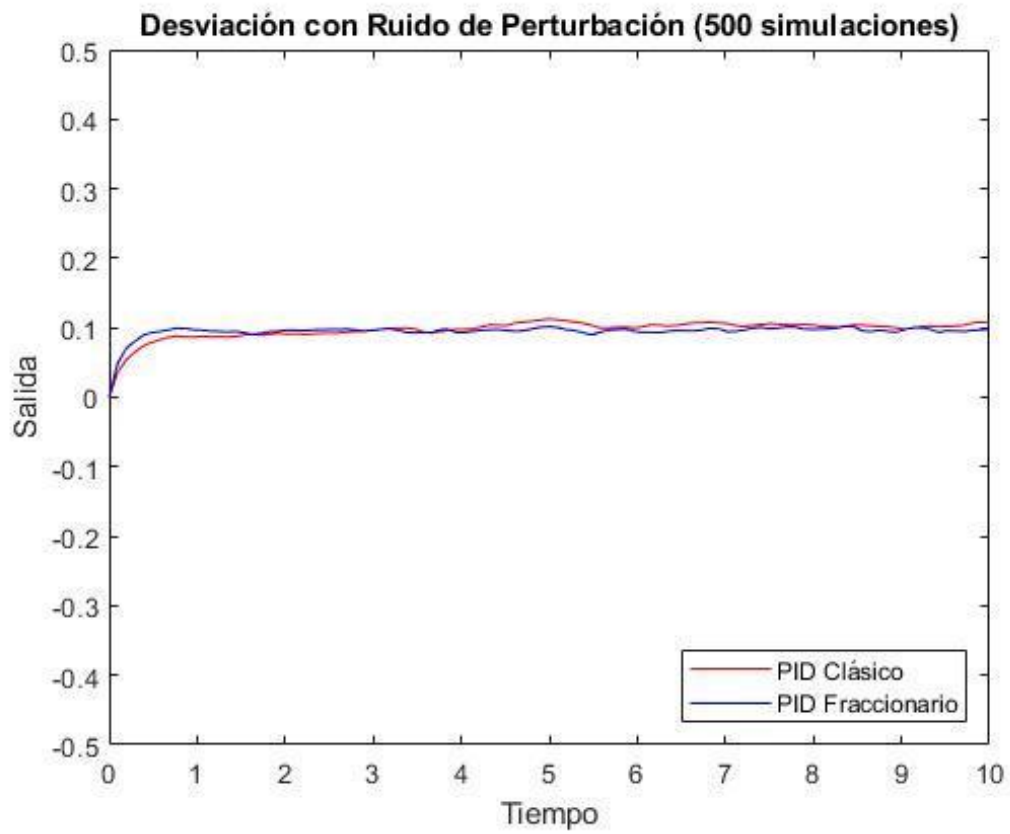


Figura 66. Media con Ruido de Perturbación (500 simulaciones).



7. Resultados (Alpha = -0.4 y Beta = 0.4)

7.1. 100 simulaciones sin Ruido de perturbación.

7.1.1. Salidas.

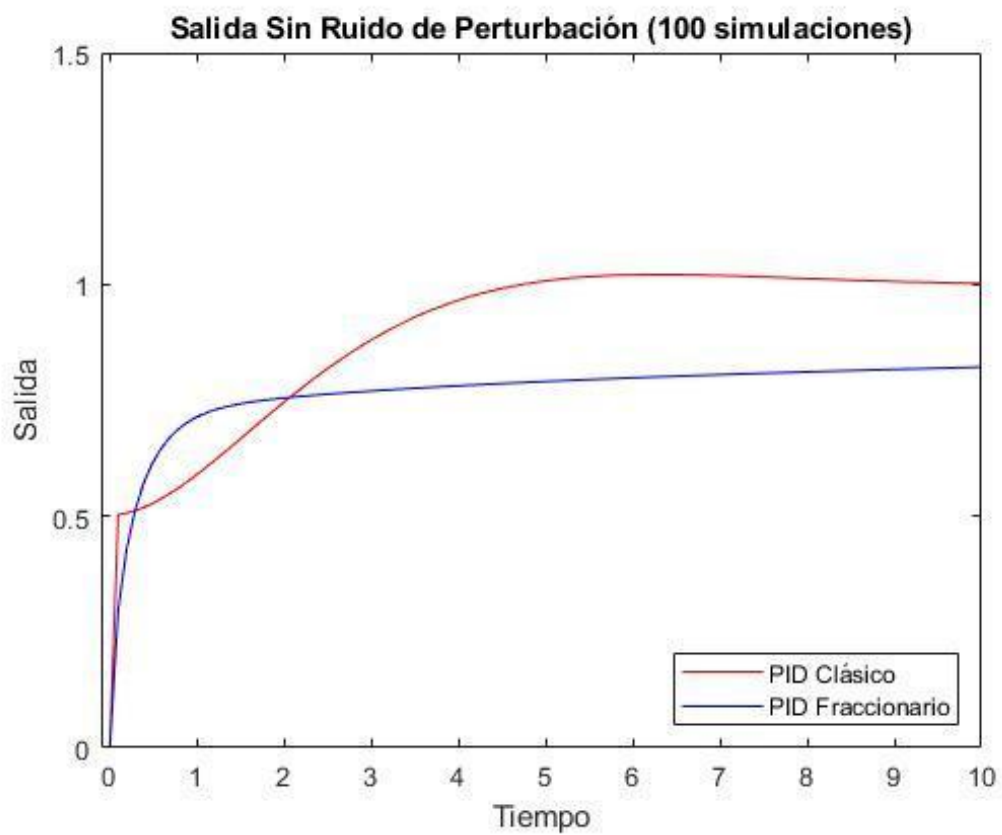


Figura 67. Salida sin Ruido de Perturbación (100 simulaciones).



7.1.2. Densidad de probabilidad.

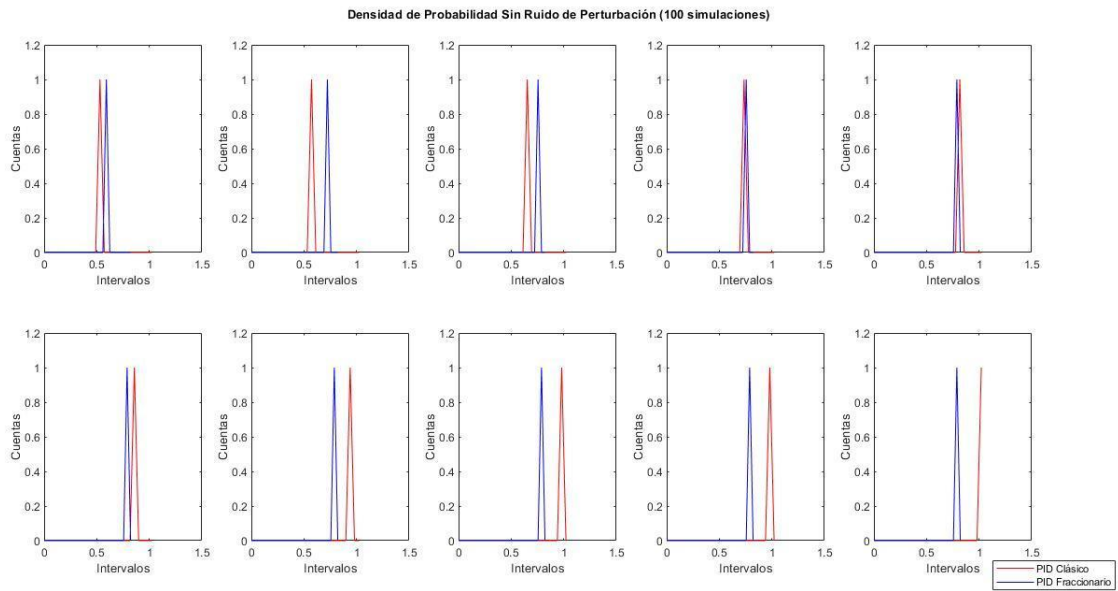


Figura 68. Densidad de Probabilidad sin Ruido de Perturbación (100 simulaciones).



7.1.3. Media.

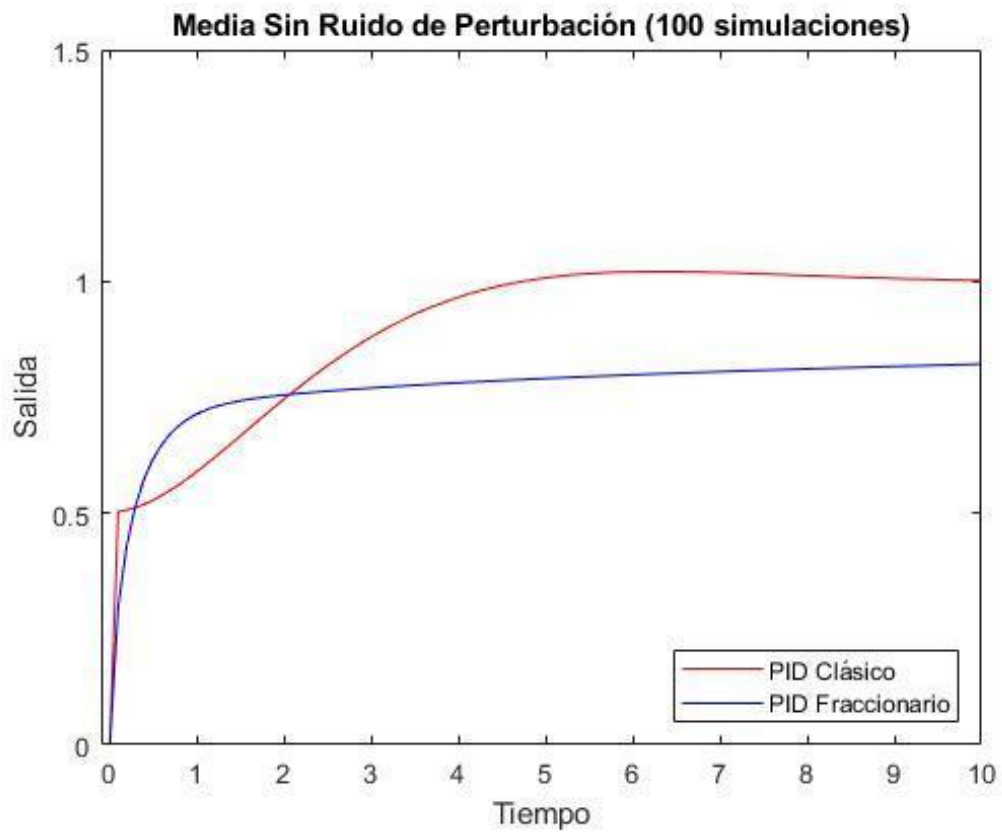


Figura 69. Media sin Ruido de Perturbación (100 simulaciones).



7.1.4. Desviación.

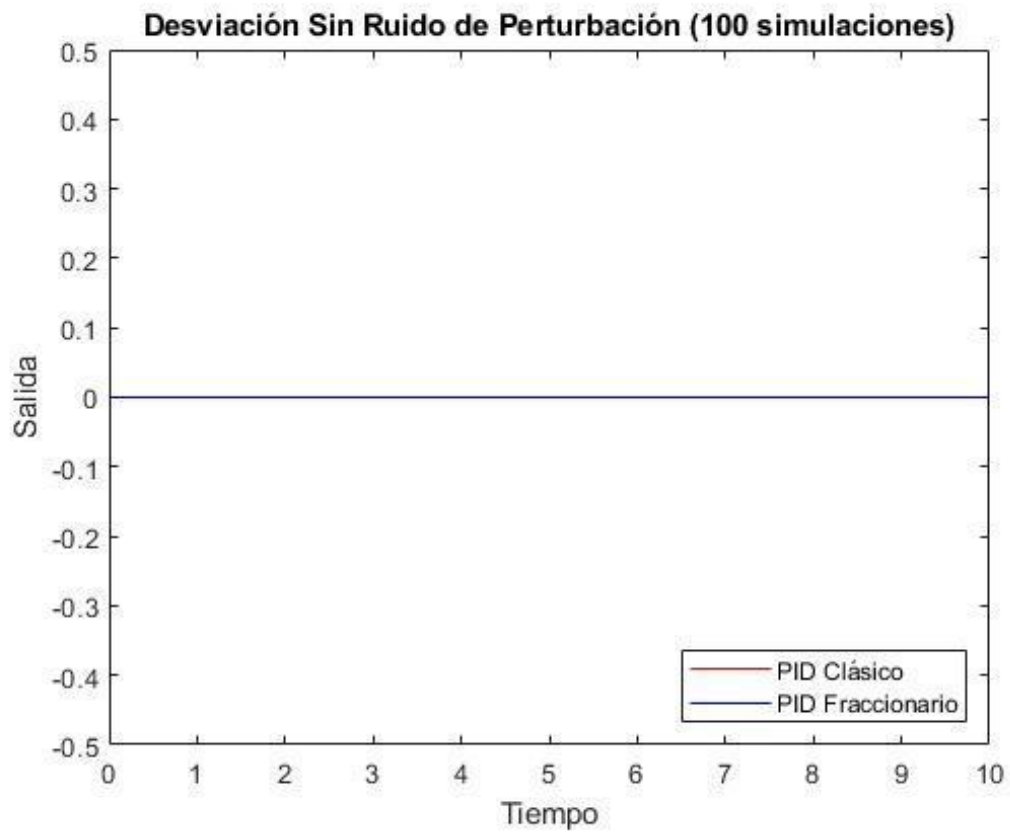


Figura 70.Desviación sin Ruido de Perturbación (100 simulaciones).



7.2. 100 simulaciones con Ruido de perturbación.

7.2.1. Salidas.

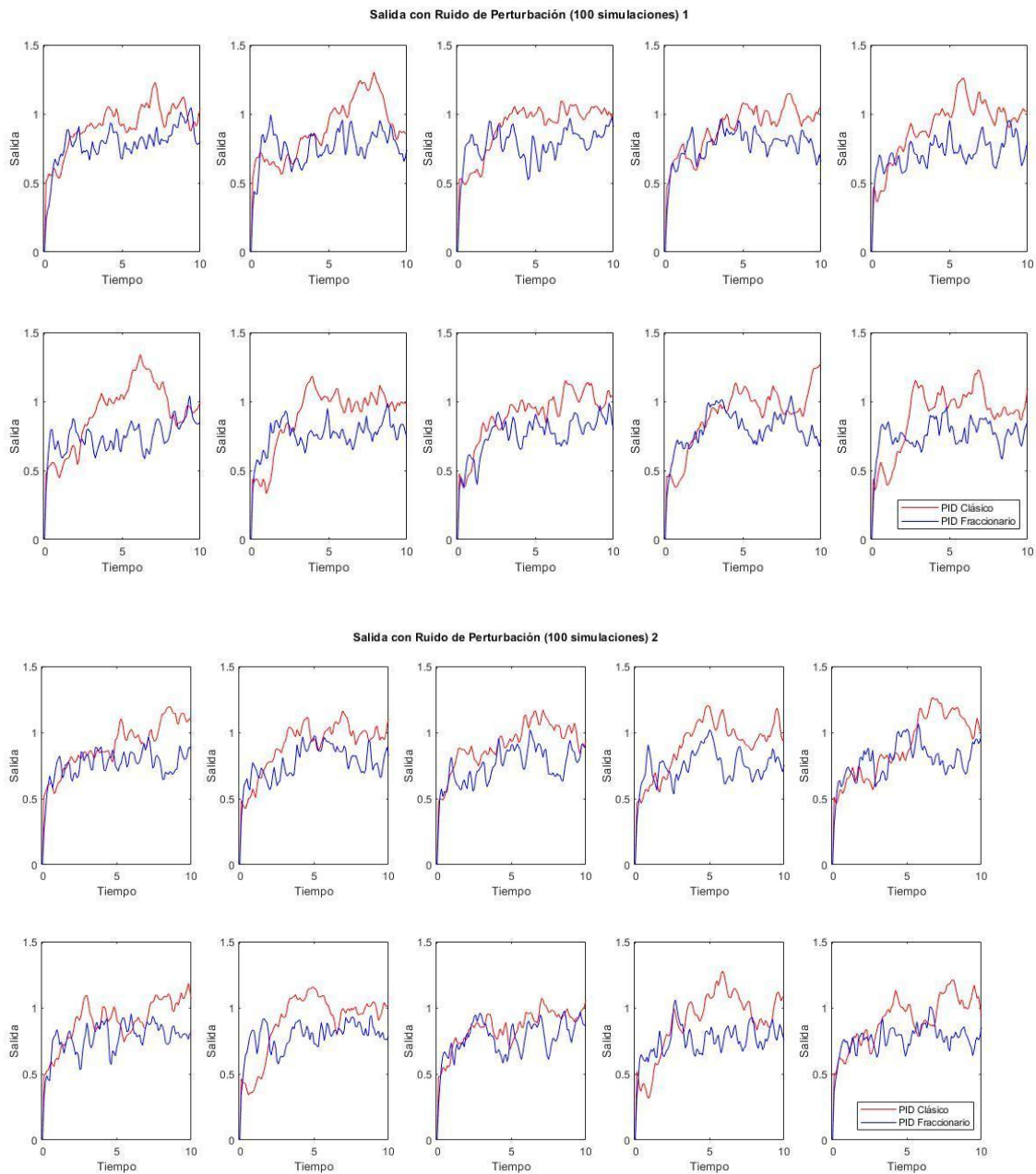


Figura 71.Salida con Ruido de Perturbación (100 simulaciones) 1.

Figura 72.Salida con Ruido de Perturbación (100 simulaciones) 2.



7.2.2. Densidad de probabilidad.

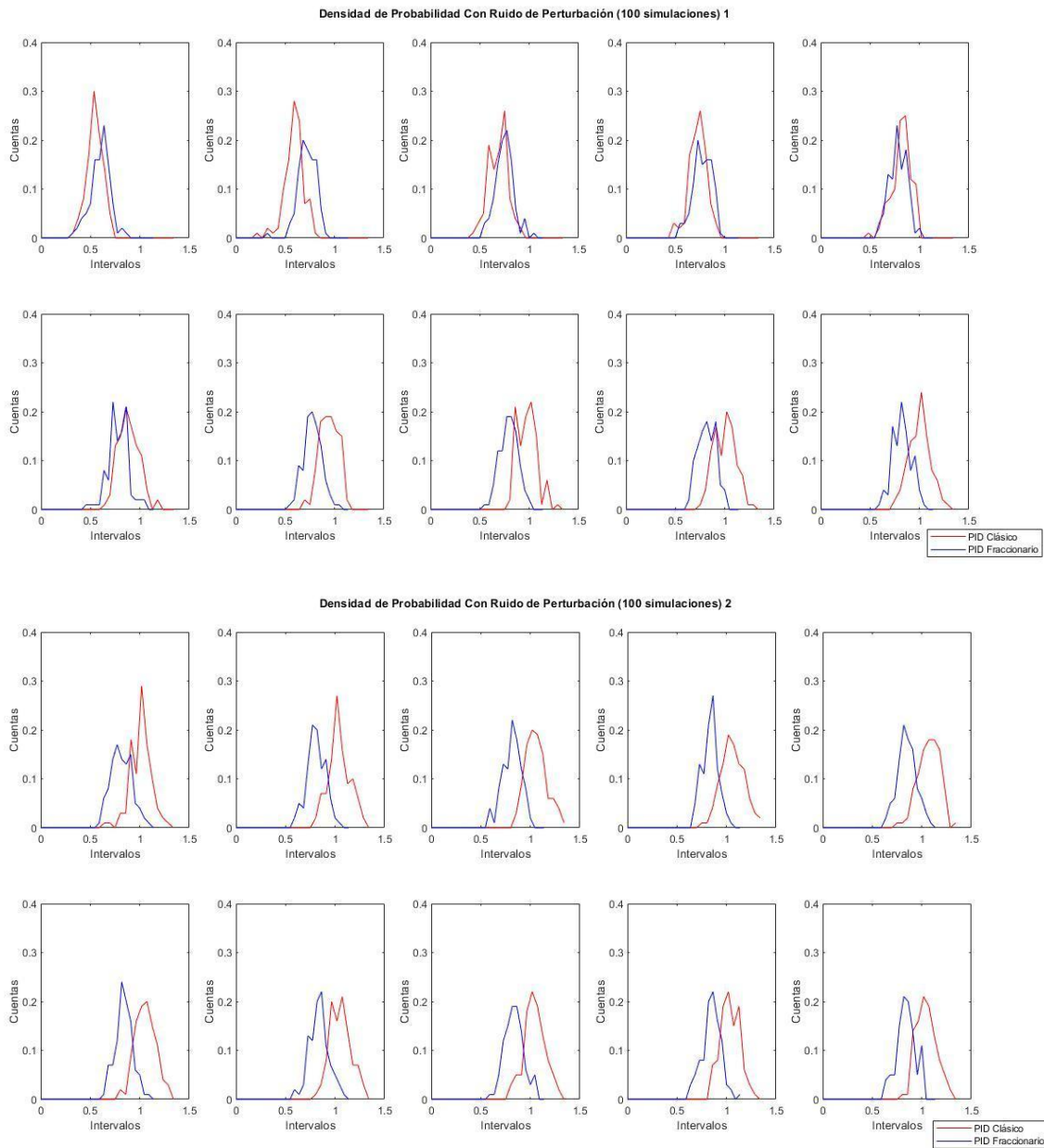


Figura 73. Densidad de Probabilidad con Ruido de Perturbación (100 simulaciones) 1.

Figura 74. Densidad de Probabilidad con Ruido de Perturbación (100 simulaciones) 2.



7.2.3. Media.

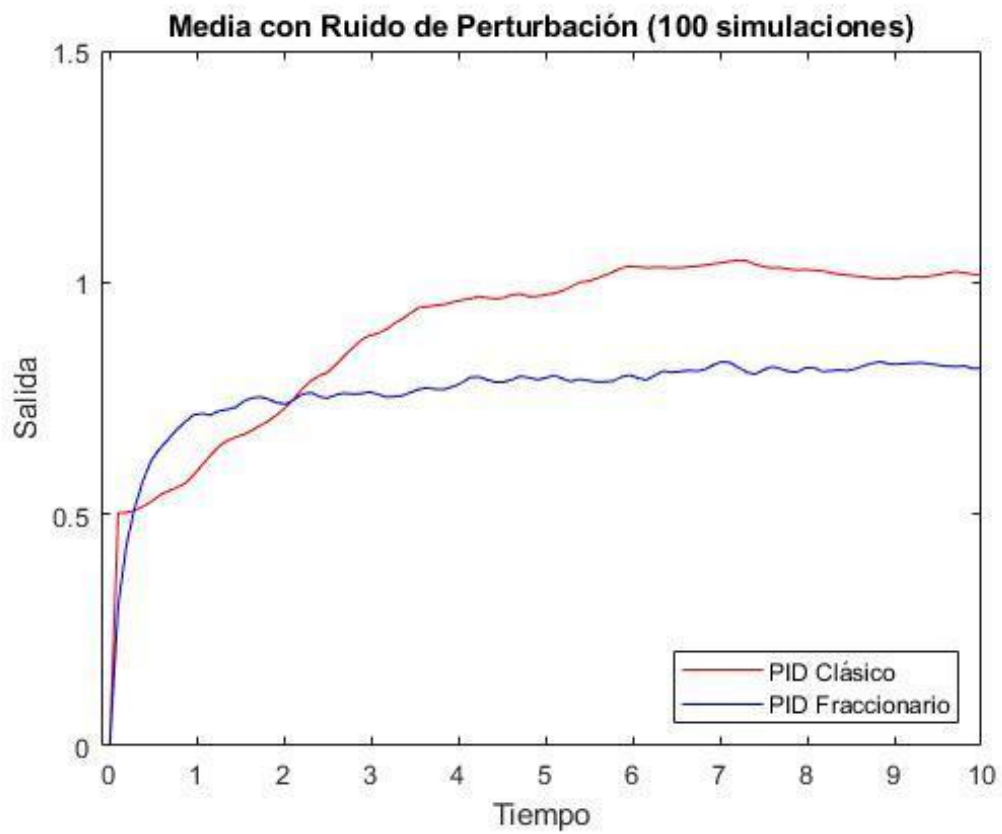


Figura 75. Media con Ruido de Perturbación (100 simulaciones).



7.2.4. Desviación.

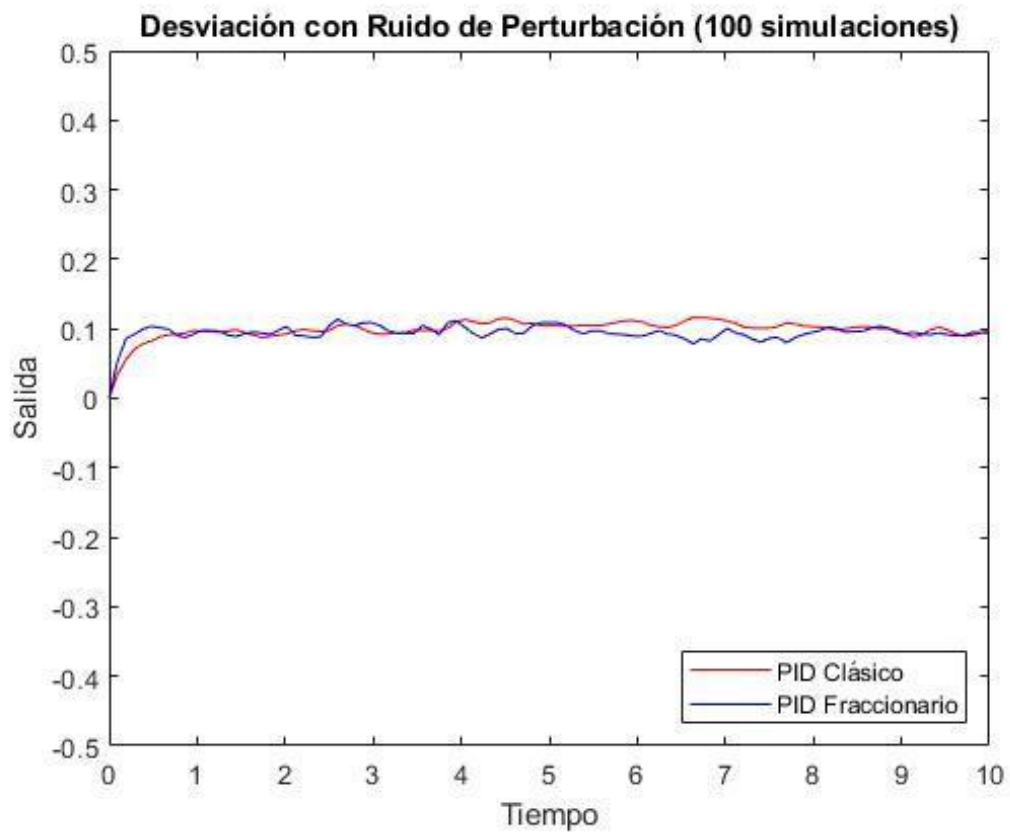


Figura 76.Desviación con Ruido de Perturbación (100 simulaciones).



7.3. 500 simulaciones sin Ruido de perturbación.

7.3.1. Salidas.

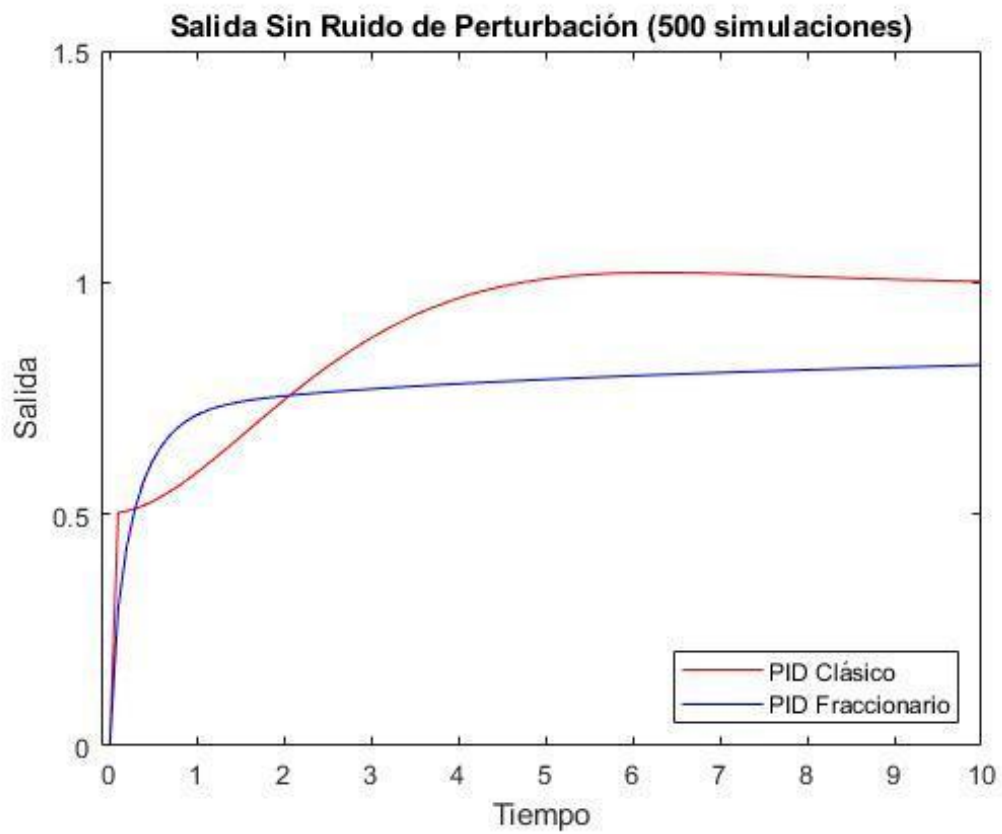


Figura 77. Salida sin Ruido de Perturbación (500 simulaciones).



7.3.2. Densidad de probabilidad.

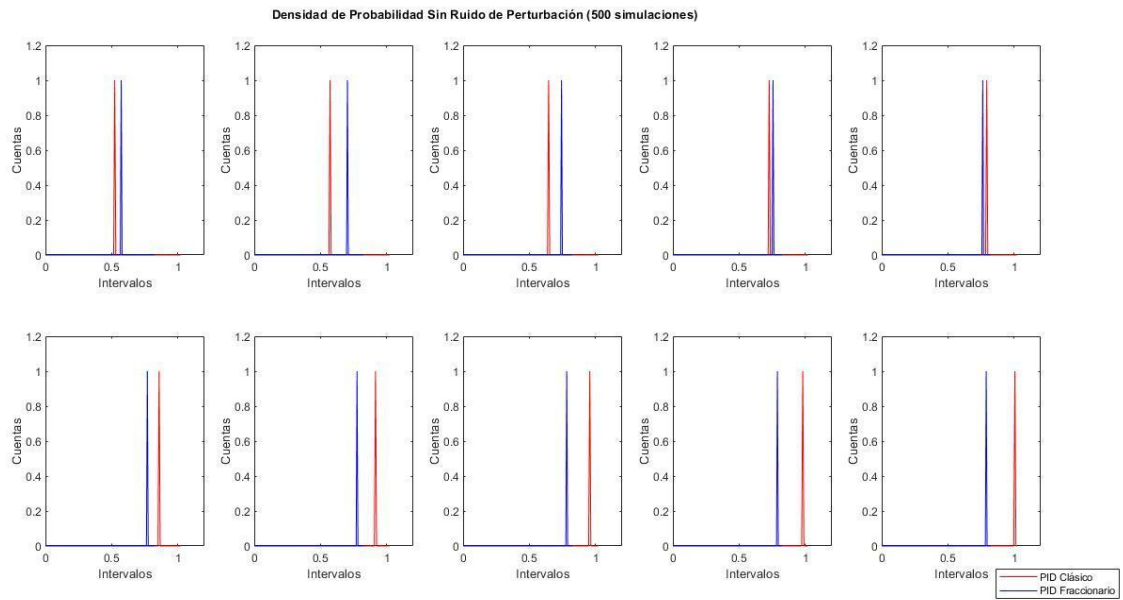


Figura 78. Densidad de Probabilidad sin Ruido de Perturbación (500 simulaciones).



7.3.3. Media.

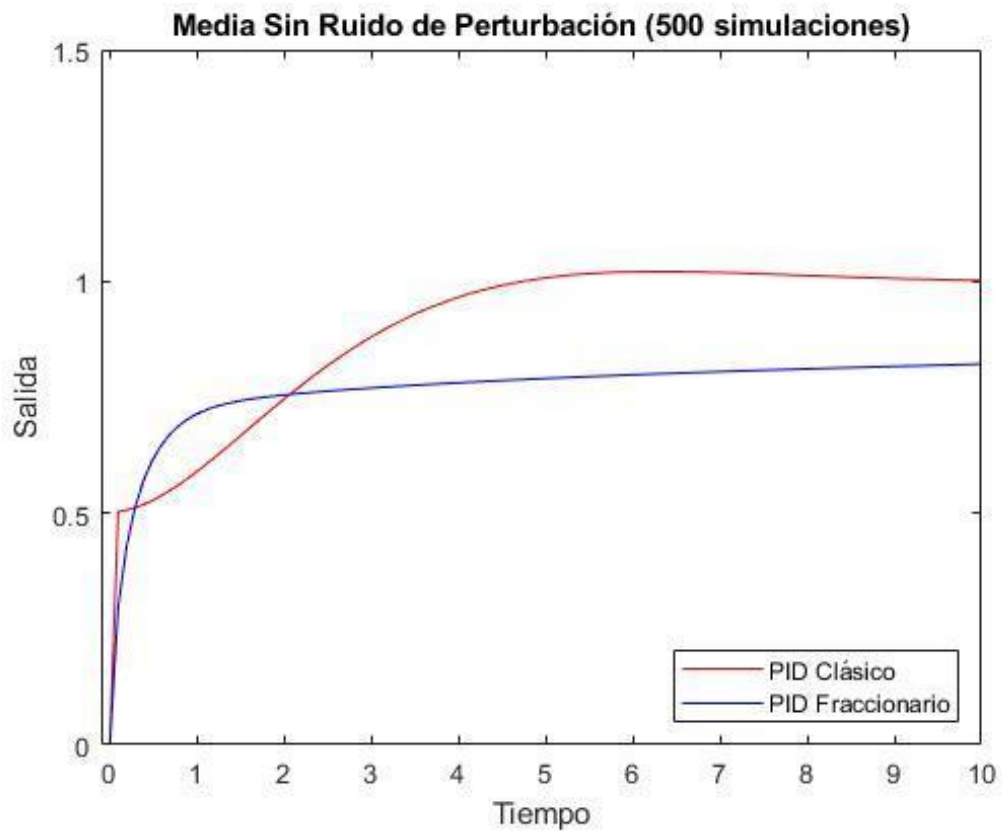


Figura 79. Media sin Ruido de Perturbación (500 simulaciones).



7.3.4. Desviación.

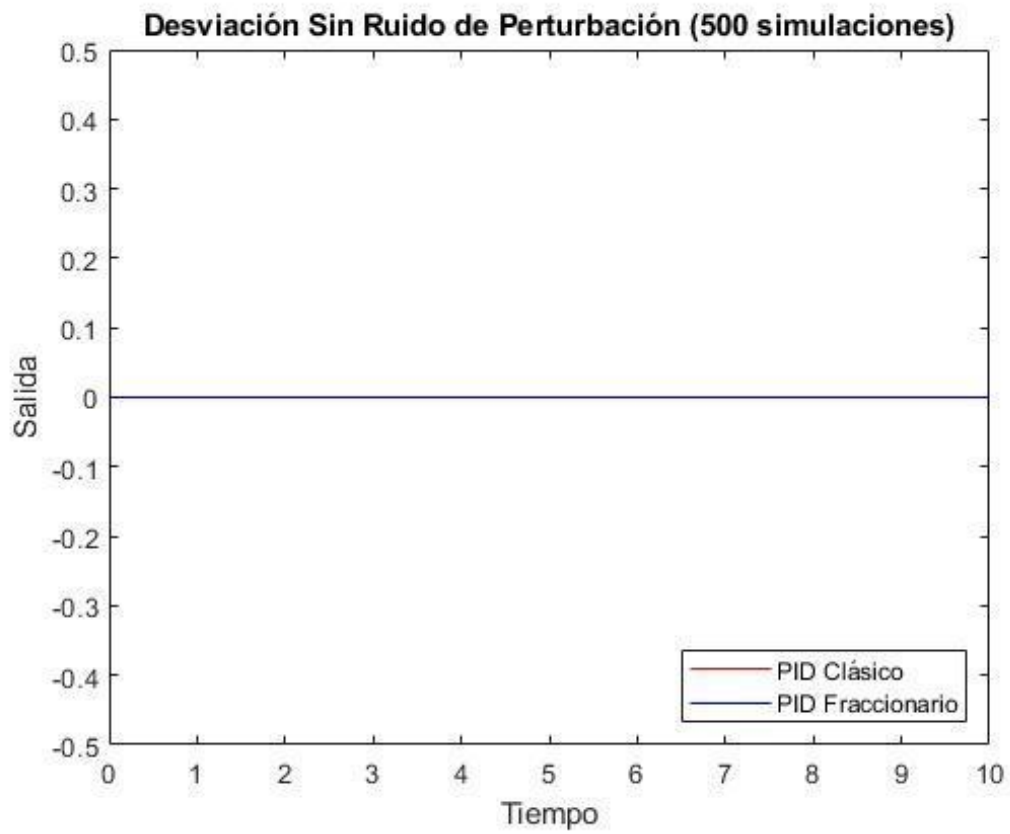


Figura 80.Desviación sin Ruido de Perturbación (500 simulaciones).



7.4. 500 simulaciones con Ruido de perturbación.

7.4.1. Salidas.

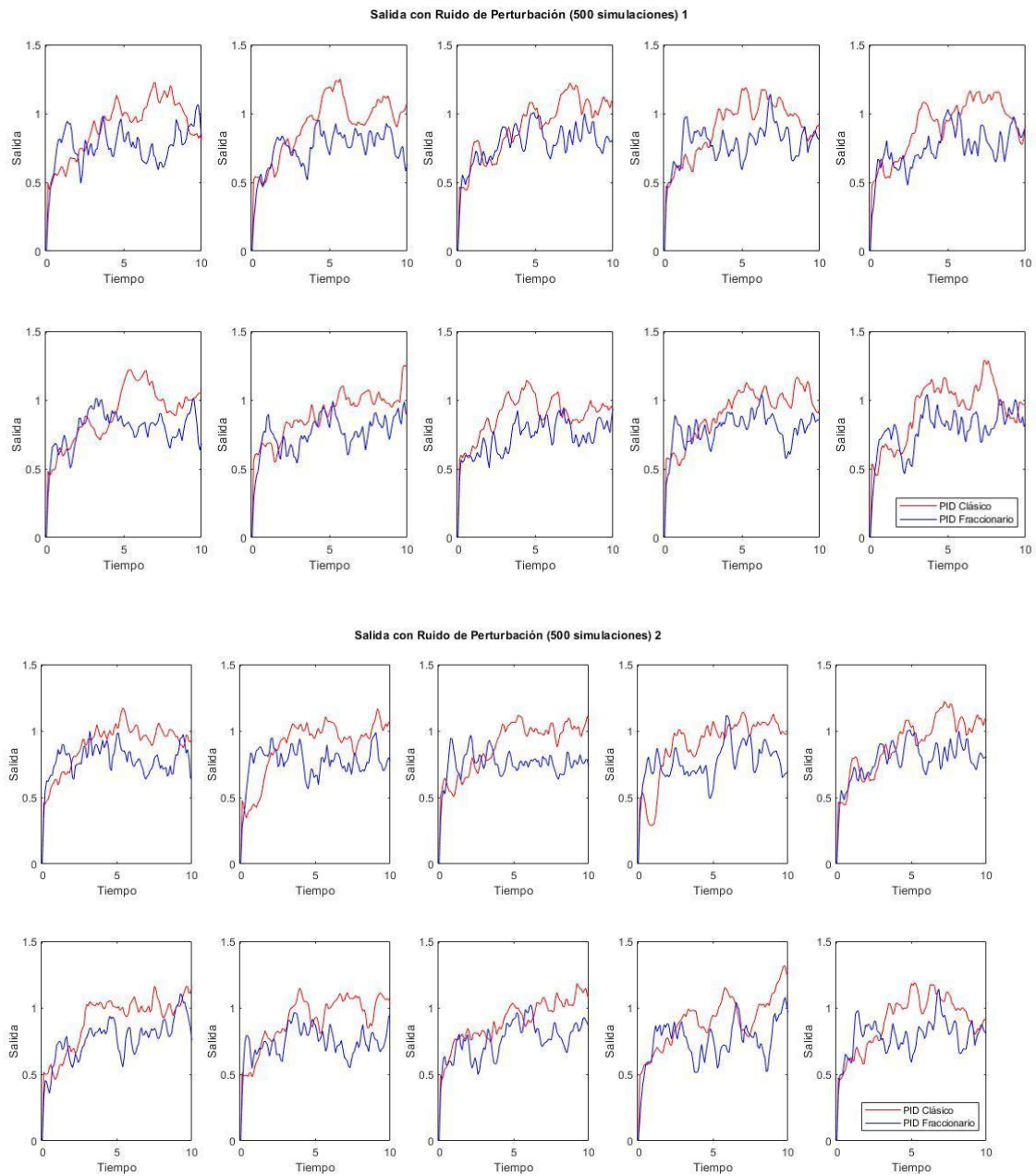


Figura 81. Salida con Ruido de Perturbación (500 simulaciones) 1.

Figura 82. Salida con Ruido de Perturbación (500 simulaciones) 2.



7.4.2. Densidad de probabilidad.

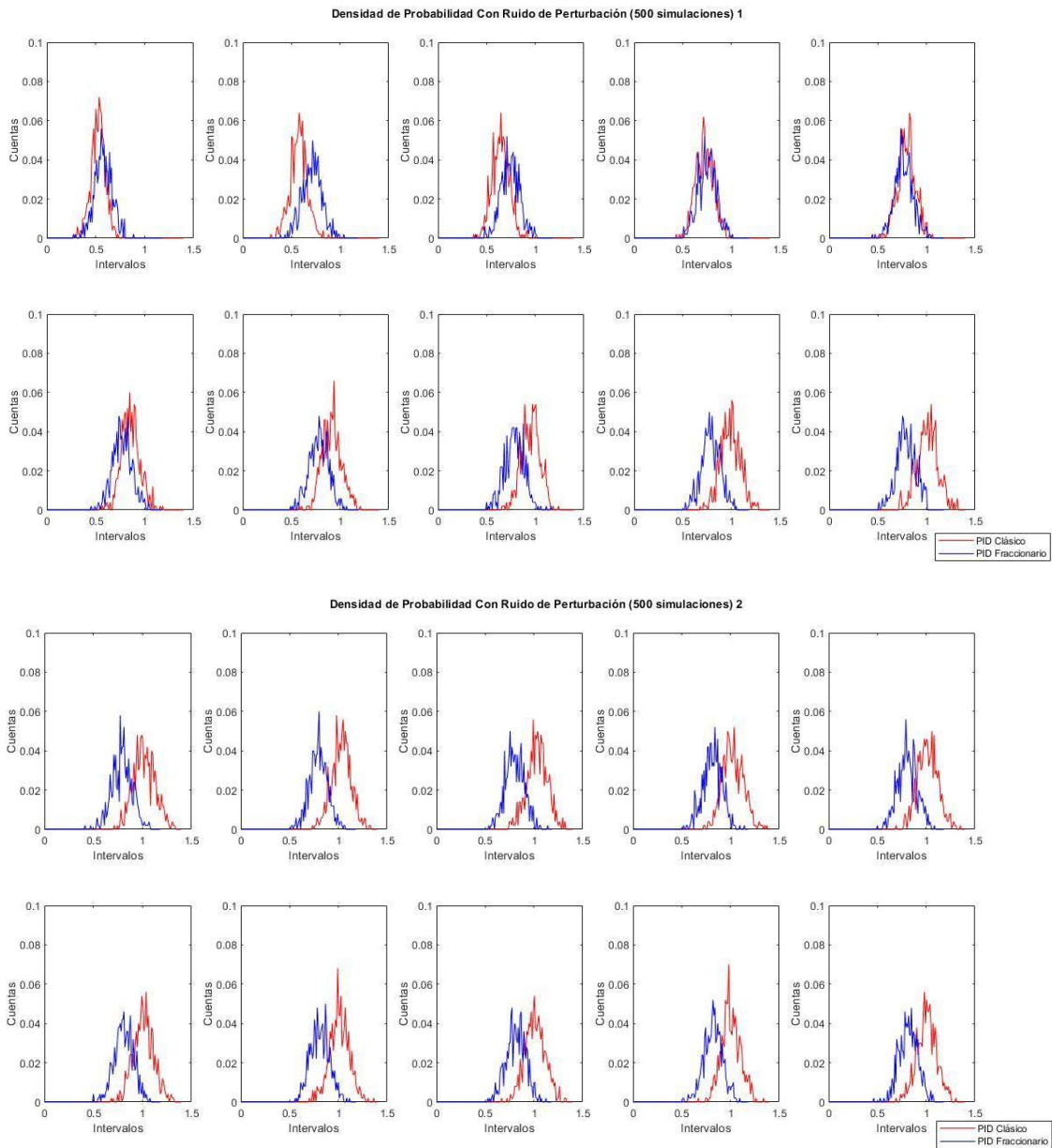


Figura 83. Densidad de Probabilidad con Ruido de Perturbación (500 simulaciones) 1.

Figura 84. Densidad de Probabilidad con Ruido de Perturbación (500 simulaciones) 2.



7.4.3. Media.

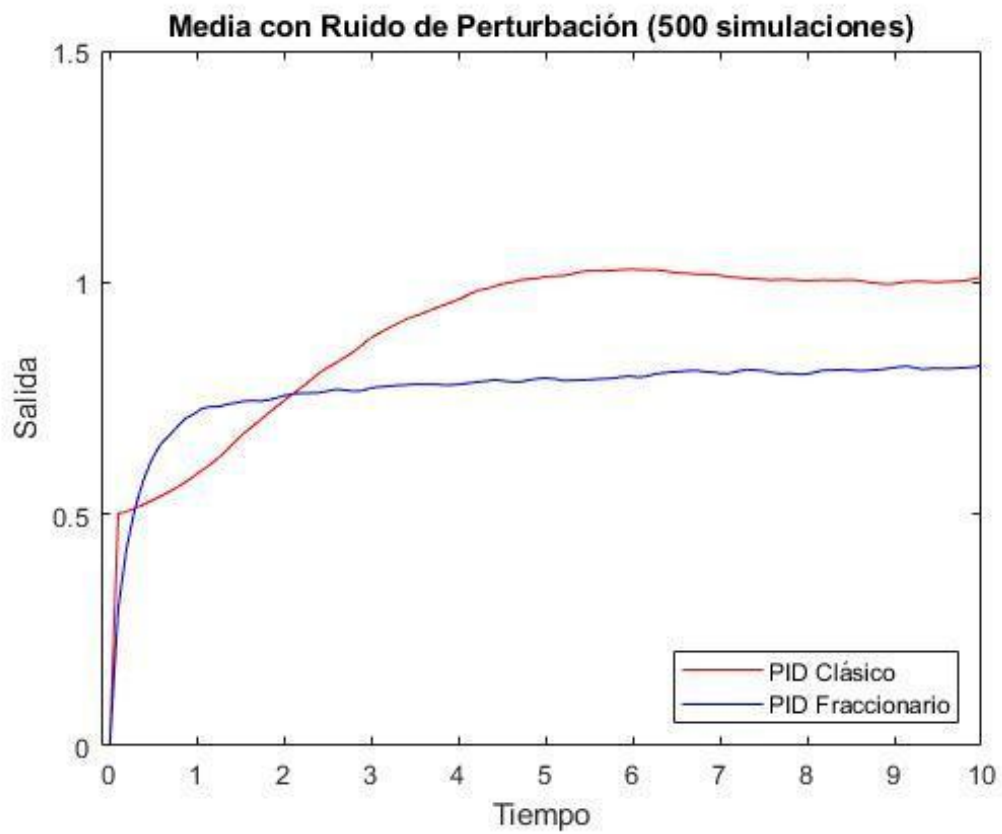


Figura 85. Media con Ruido de Perturbación (500 simulaciones).



7.4.4. Desviación.

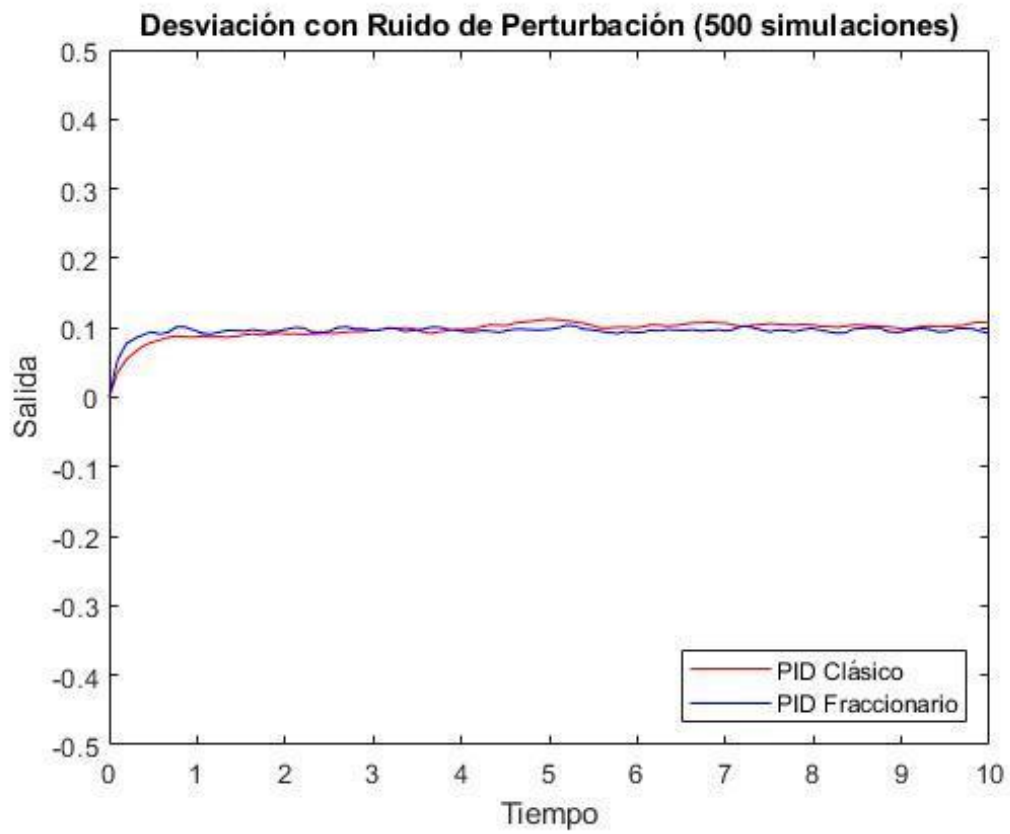


Figura 86.Desviación con Ruido de Perturbación (500 simulaciones).



8. Resultados (Alpha = -0.2 y Beta = 0.2)

8.1. 100 simulaciones sin Ruido de perturbación.

8.1.1. Salidas.

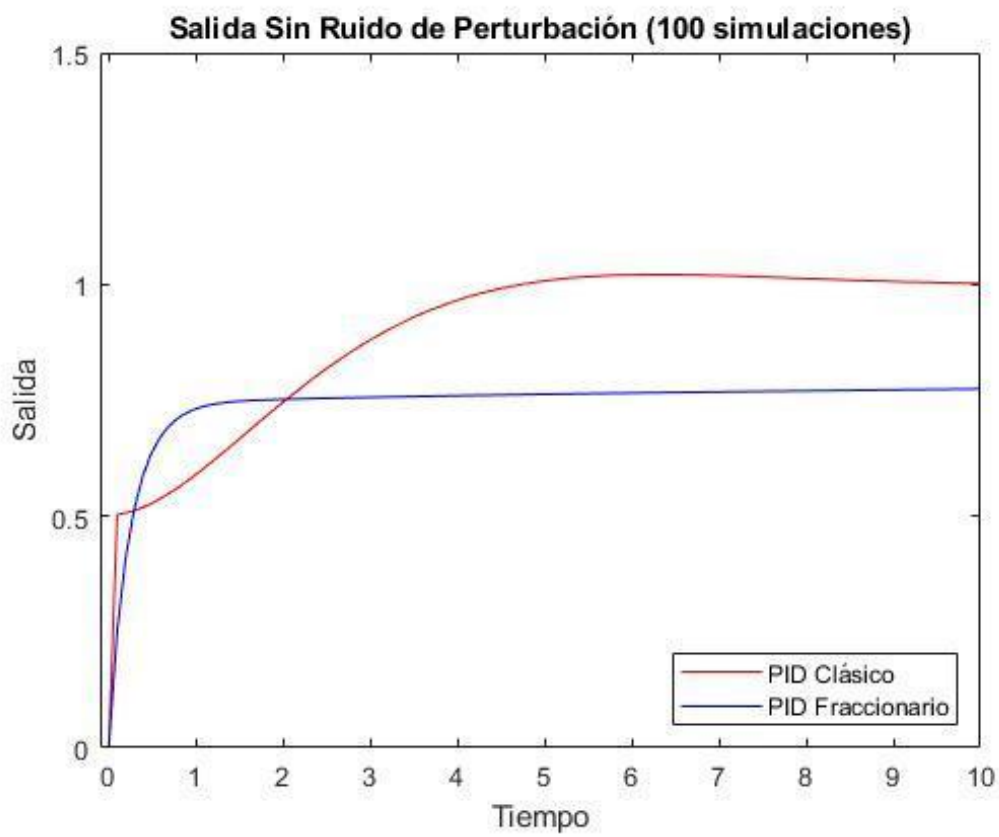


Figura 87. Salida sin Ruido de Perturbación (100 simulaciones).



8.1.2. Densidad de probabilidad.

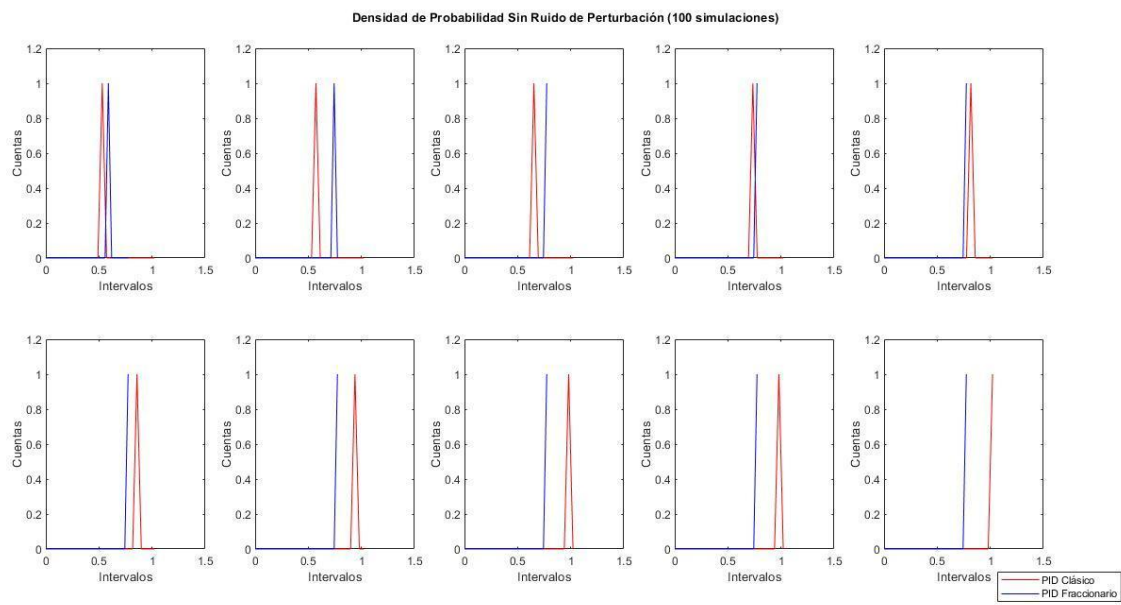


Figura 88. Densidad de Probabilidad sin Ruido de Perturbación (100 simulaciones).



8.1.3. Media.

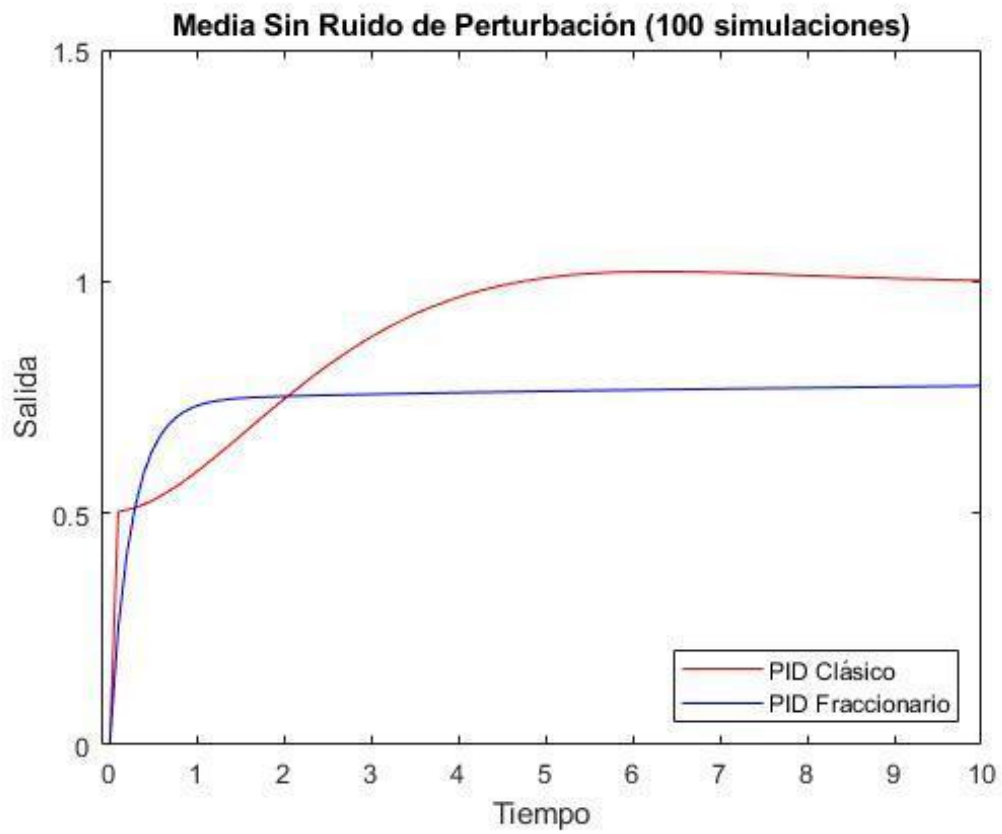


Figura 89. Media sin Ruido de Perturbación (100 simulaciones).



8.1.4. Desviación.

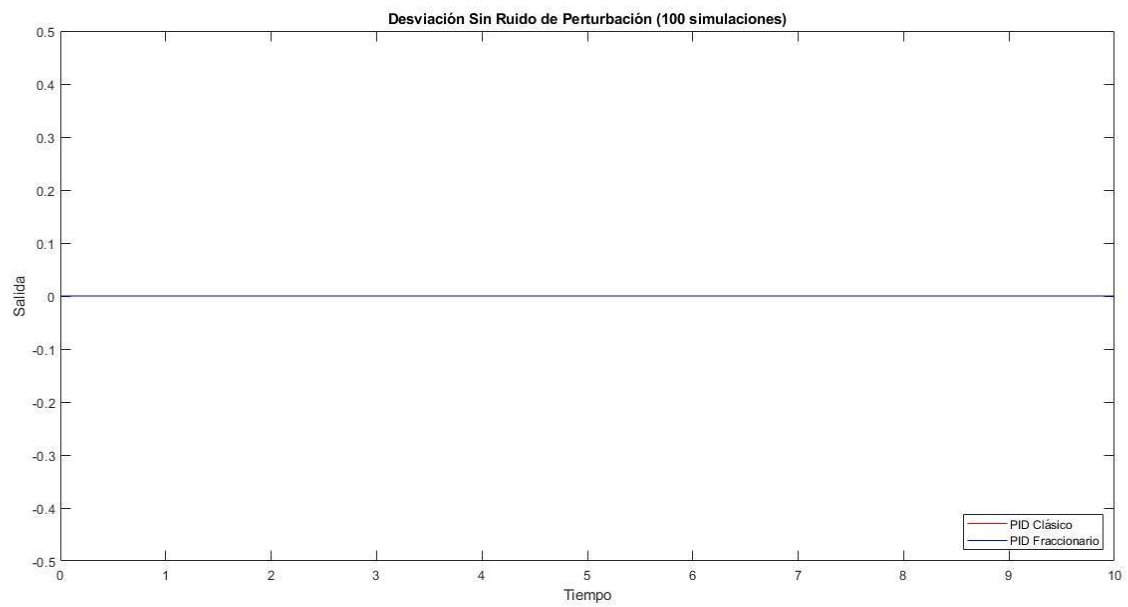


Figura 90.Desviación sin Ruido de Perturbación (100 simulaciones).



8.2. 100 simulaciones con Ruido de perturbación.

8.2.1. Salidas.

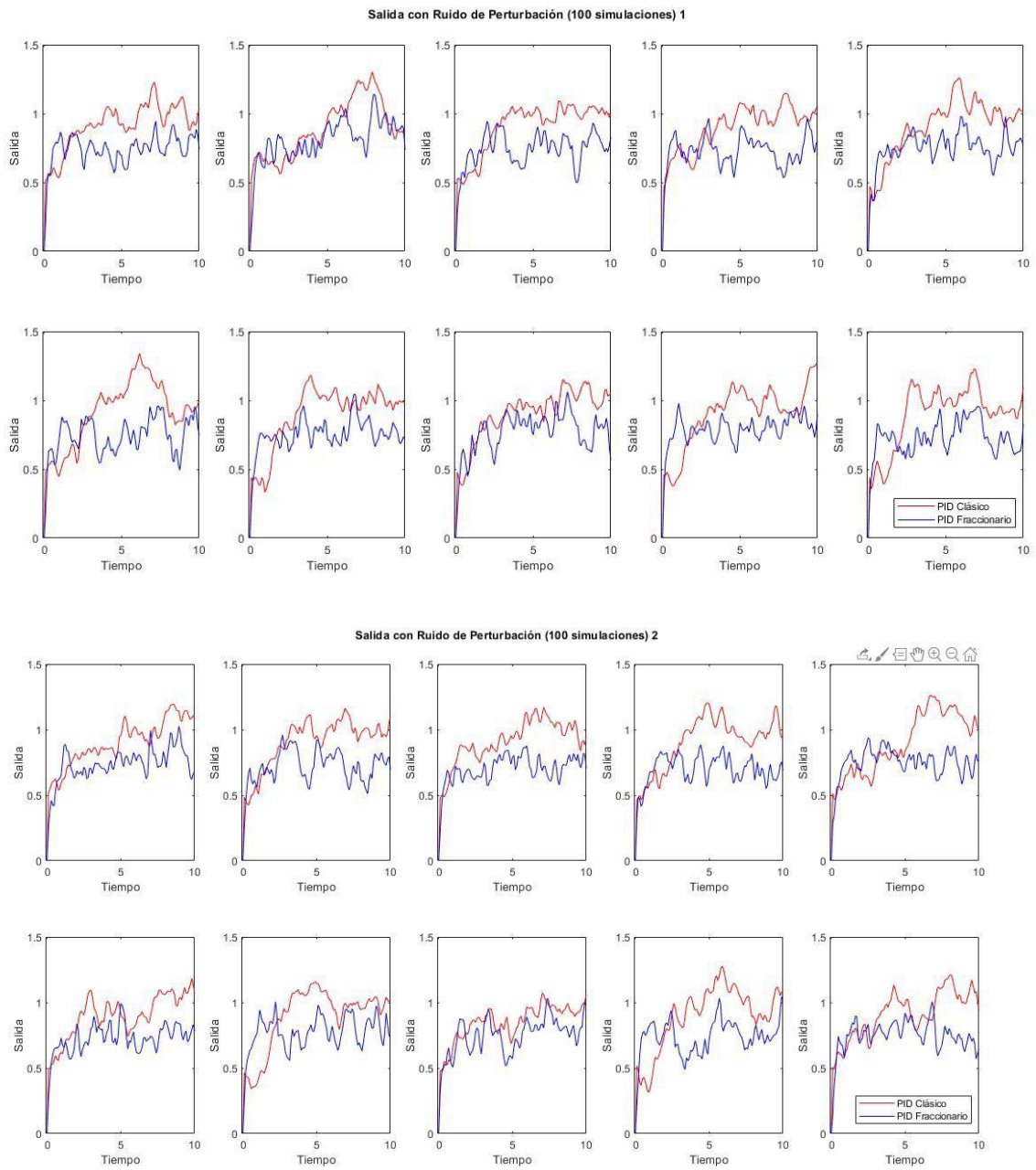


Figura 91. Salida con Ruido de Perturbación (100 simulaciones) 1.

Figura 92. Salida con Ruido de Perturbación (100 simulaciones) 2.



8.2.2. Densidad de probabilidad.

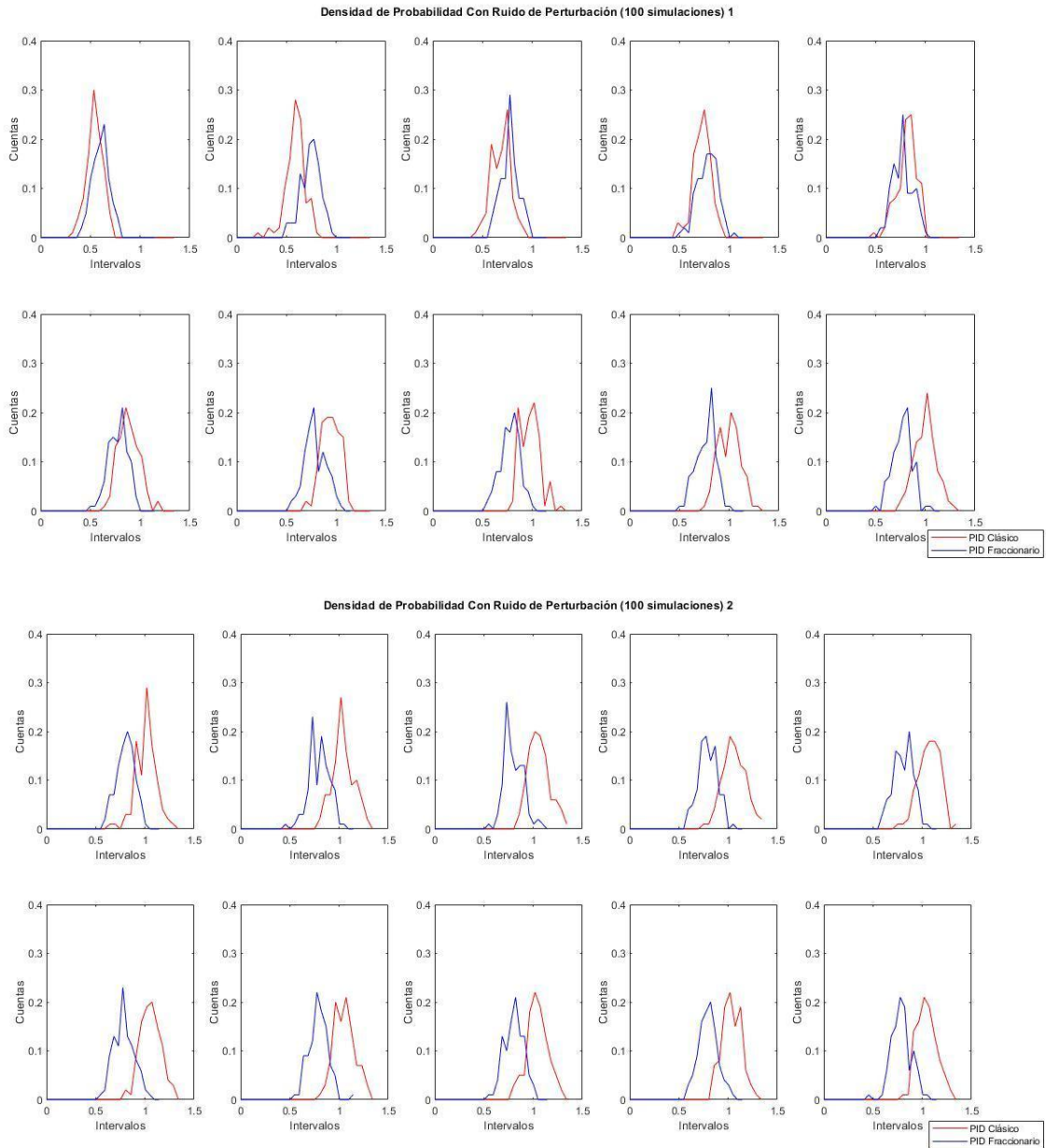


Figura 93. Densidad de Probabilidad con Ruido de Perturbación (100 simulaciones) 1.

Figura 94. Densidad de Probabilidad con Ruido de Perturbación (100 simulaciones) 2.



8.2.3. Media.

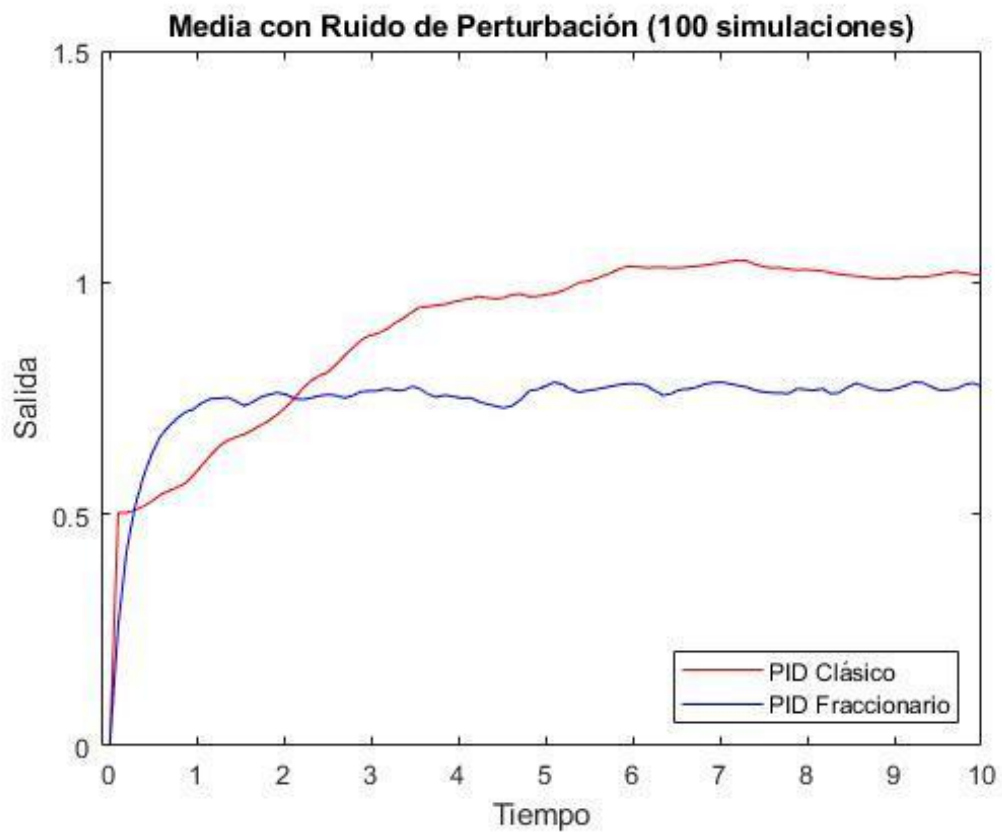


Figura 95. Media con Ruido de Perturbación (100 simulaciones).



8.2.4. Desviación.

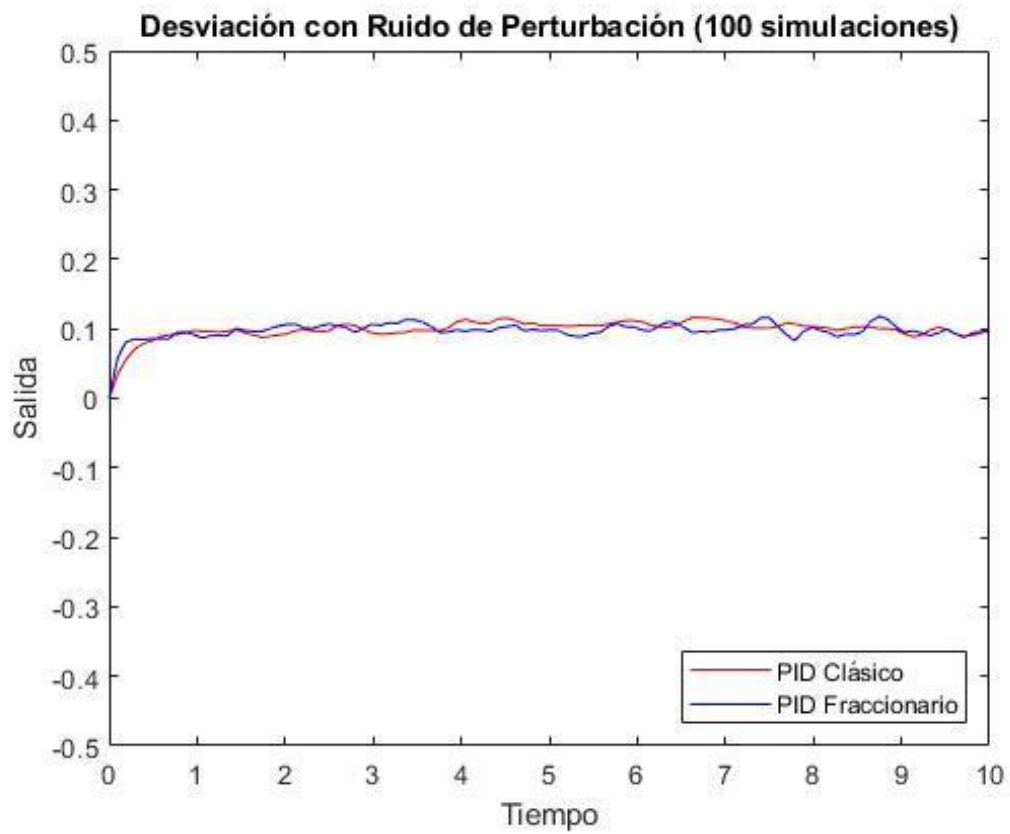


Figura 96.Desviación con Ruido de Perturbación (100 simulaciones).



8.3. 500 simulaciones sin Ruido de perturbación.

8.3.1. Salidas.

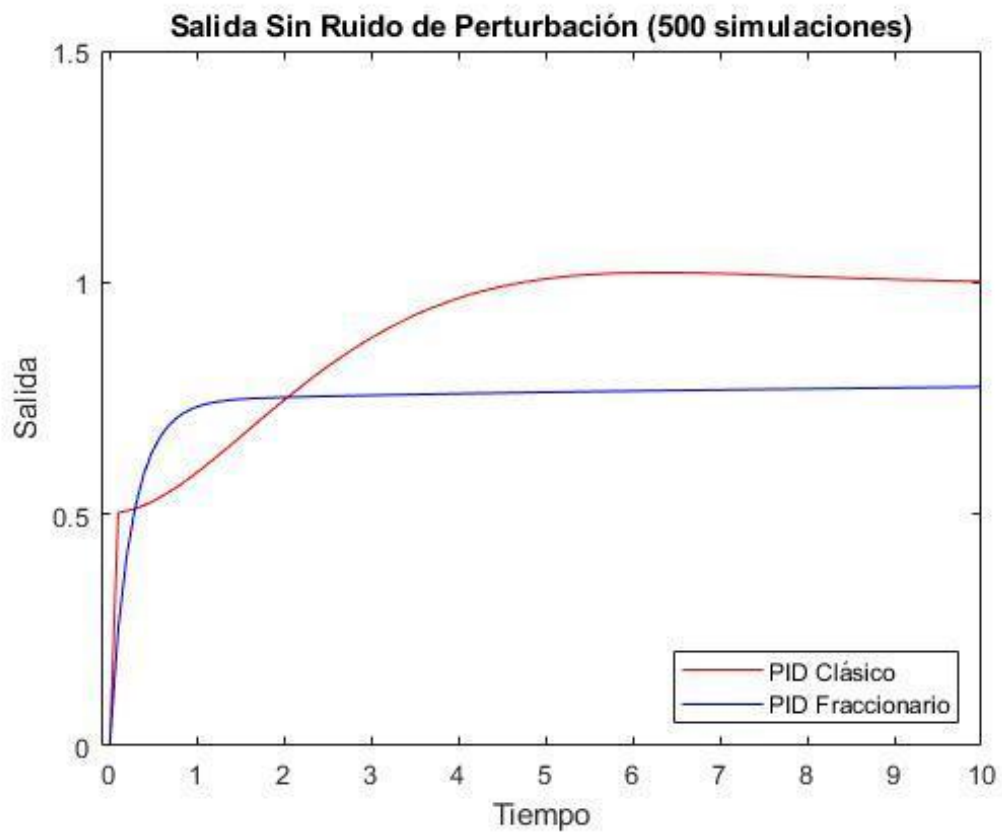


Figura 97. Salida sin Ruido de Perturbación (500 simulaciones).



8.3.2. Densidad de probabilidad.

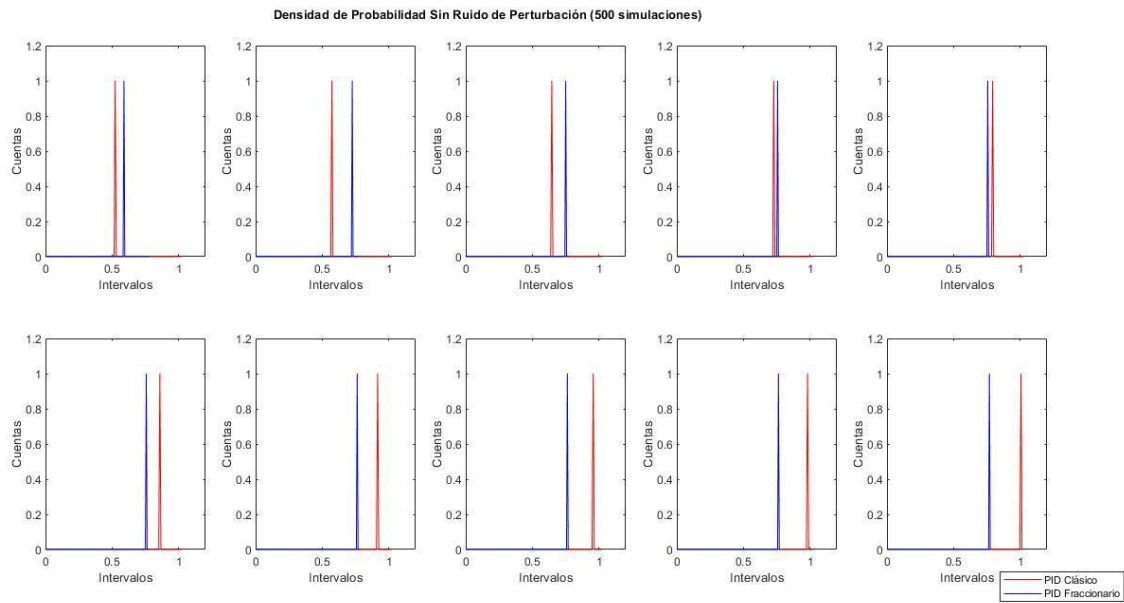


Figura 98. Densidad de Probabilidad sin Ruido de Perturbación (500 simulaciones).



8.3.3. Media.

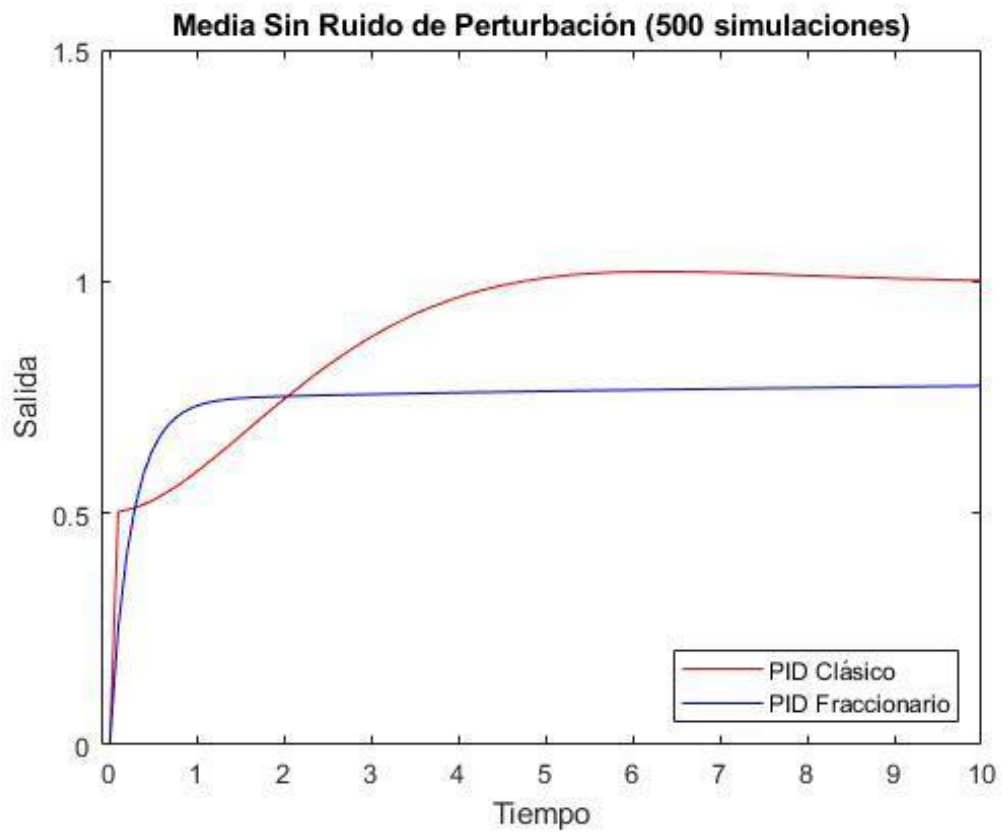


Figura 99. Media sin Ruido de Perturbación (500 simulaciones).



8.3.4. Desviación.

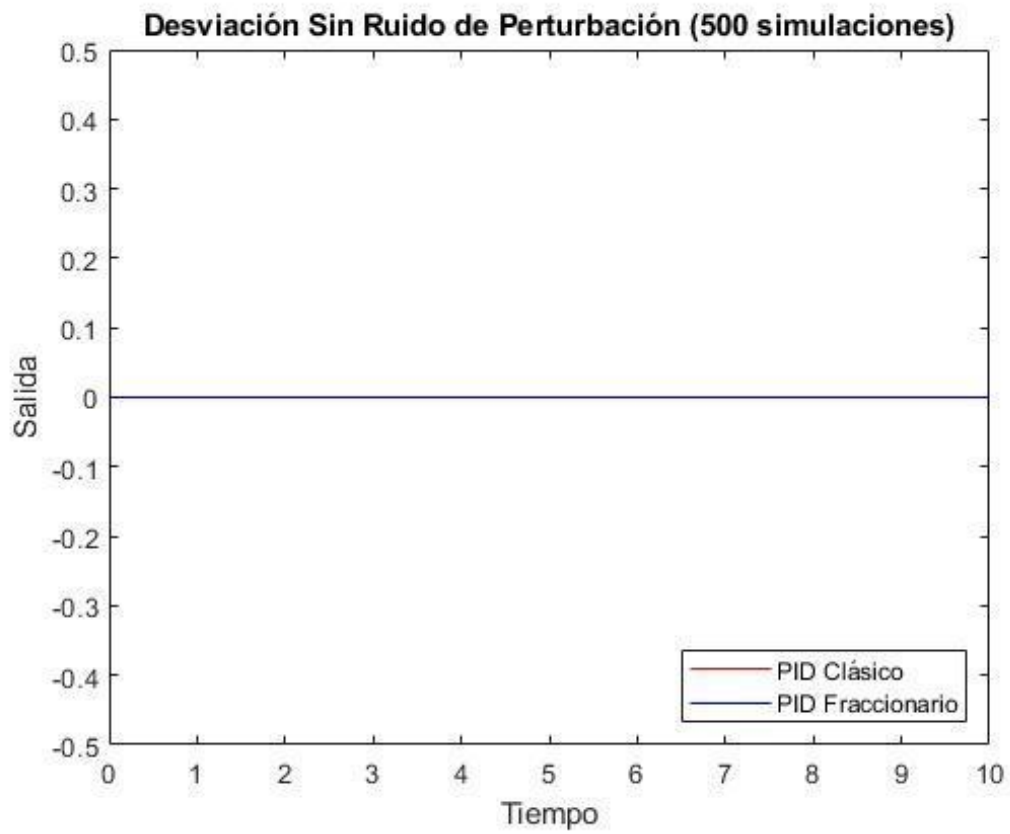


Figura 100.Desviación sin Ruido de Perturbación (500 simulaciones).



8.4. 500 simulaciones con Ruido de perturbación.

8.4.1. Salidas.

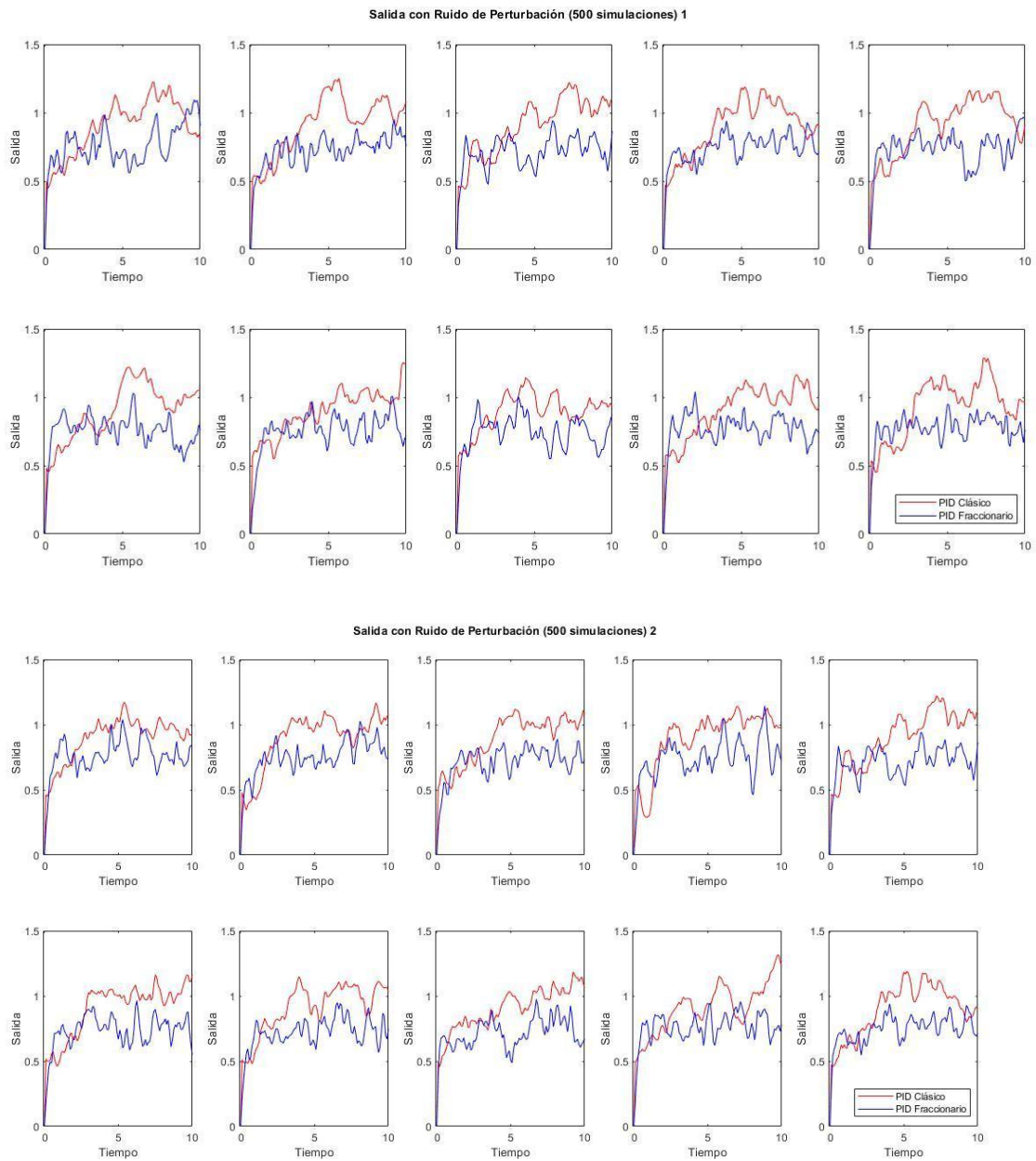


Figura 101.Salida con Ruido de Perturbación (500 simulaciones) 1.

Figura 102.Salida con Ruido de Perturbación (500 simulaciones) 2.



8.4.2. Densidad de probabilidad.

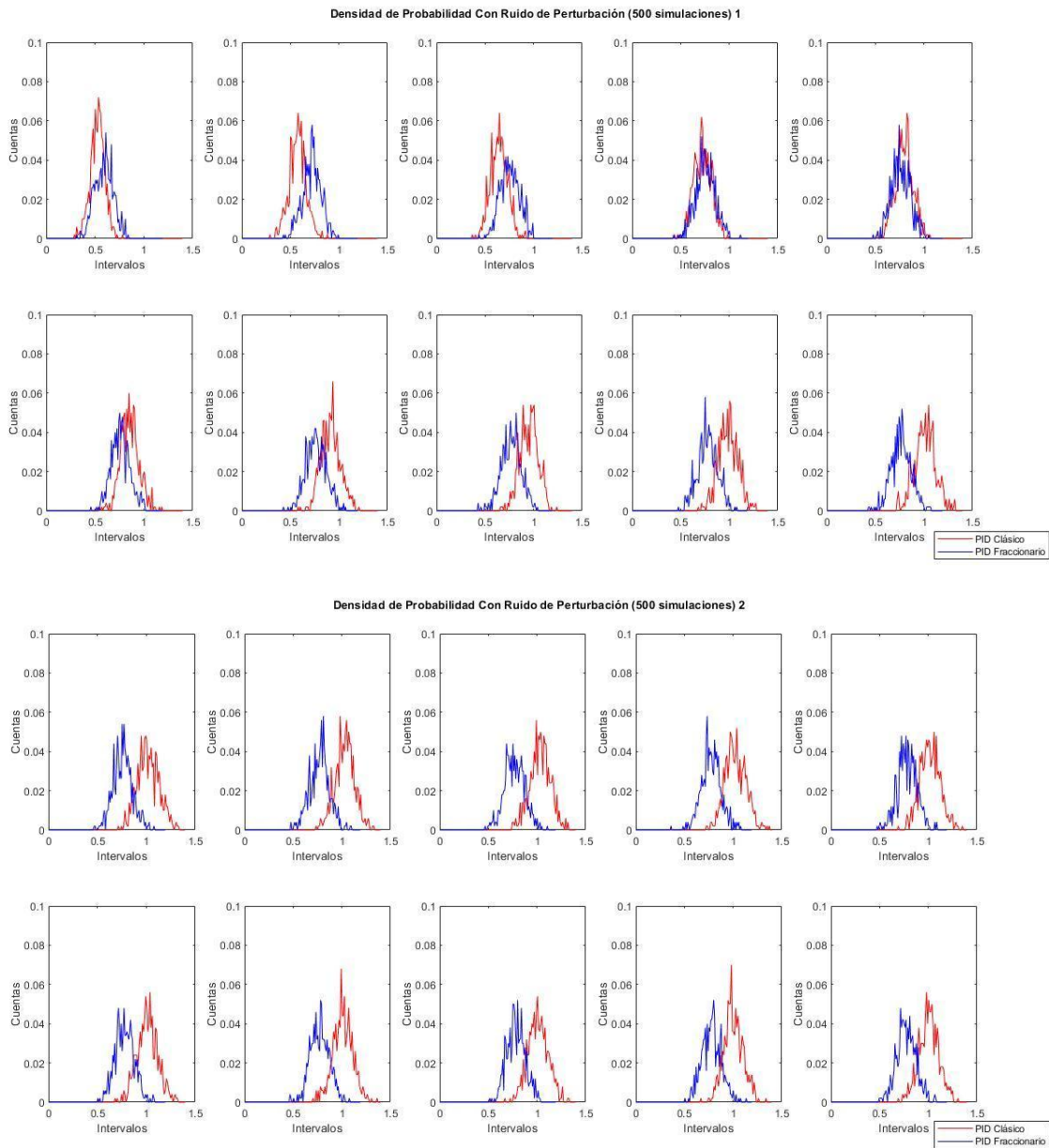


Figura 103. Densidad de Probabilidad con Ruido de Perturbación (500 simulaciones) 1.

Figura 104. Densidad de Probabilidad con Ruido de Perturbación (500 simulaciones) 2.



8.4.3. Media.

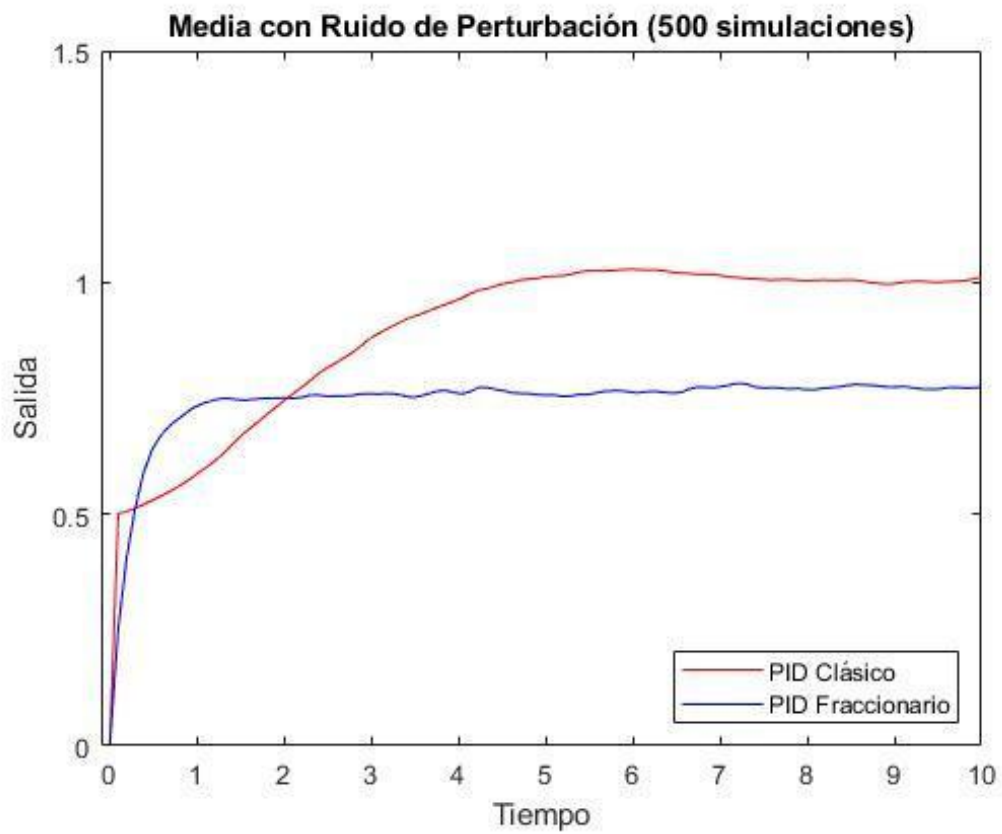


Figura 105. Media con Ruido de Perturbación (500 simulaciones).



8.4.4. Desviación.

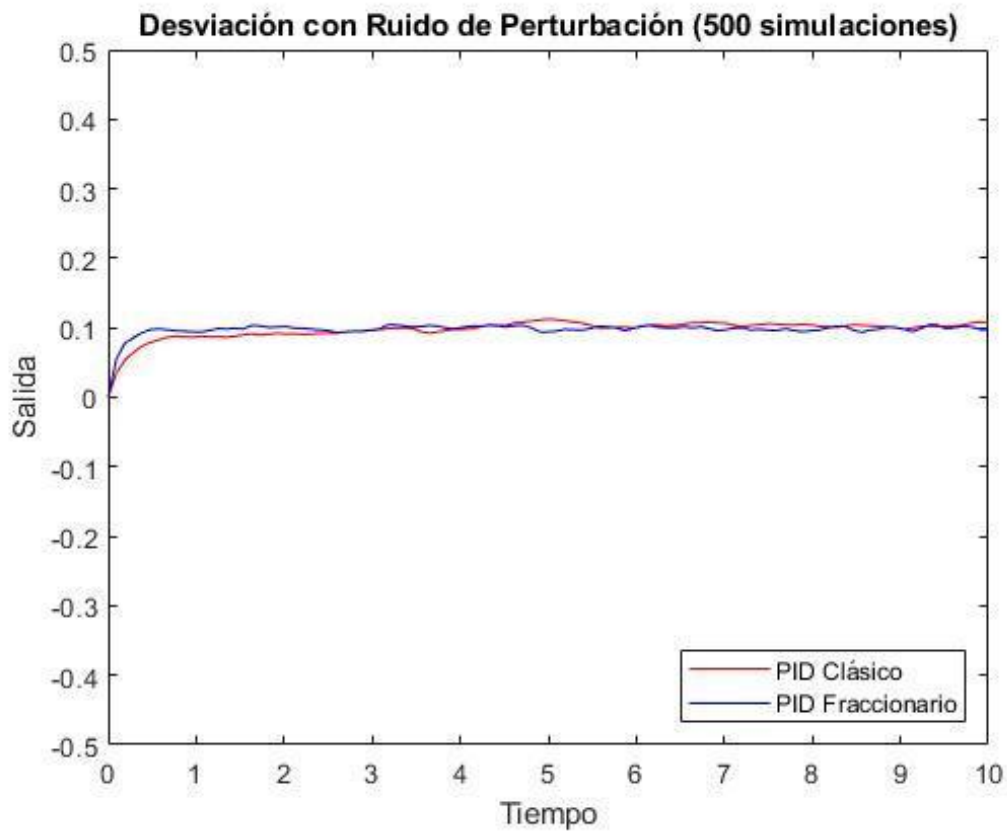


Figura 106.Desviación con Ruido de Perturbación (500 simulaciones).

9. Análisis Salidas según parámetros Alpha y Beta

Como primera instancia, se comparó el comportamiento de las salidas de cada sistema según los dos tipos de PID y sus parámetros Alfa y Beta.



9.1 Salidas sin ruido de perturbación (500 simulaciones).

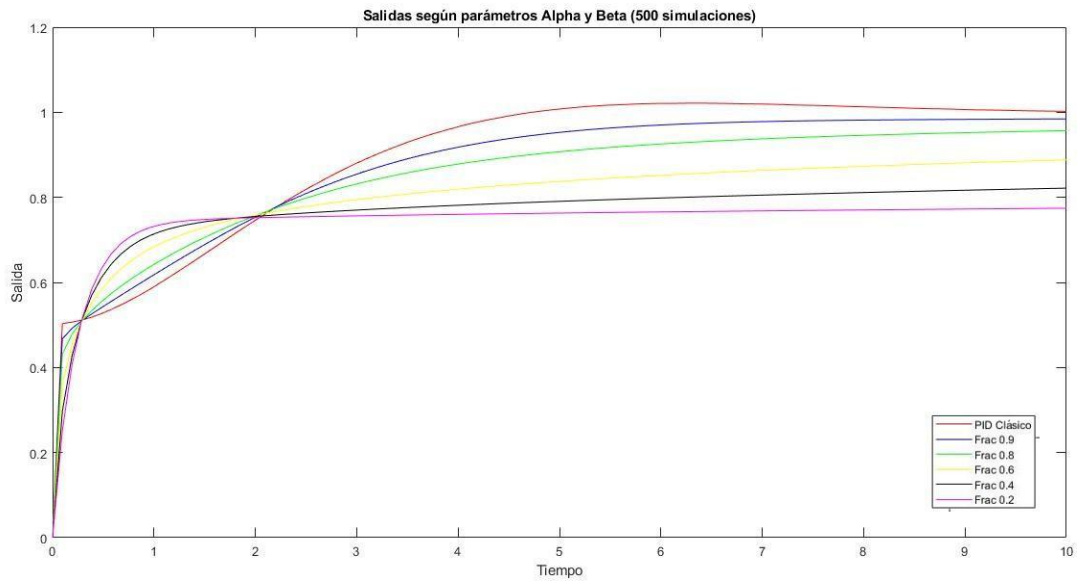


Figura 107. Salidas según parámetros Alpha y Beta sin Ruido de Perturbación (500 simulaciones).



9.2 Medias de Salidas con ruido de perturbación (500 simulaciones).

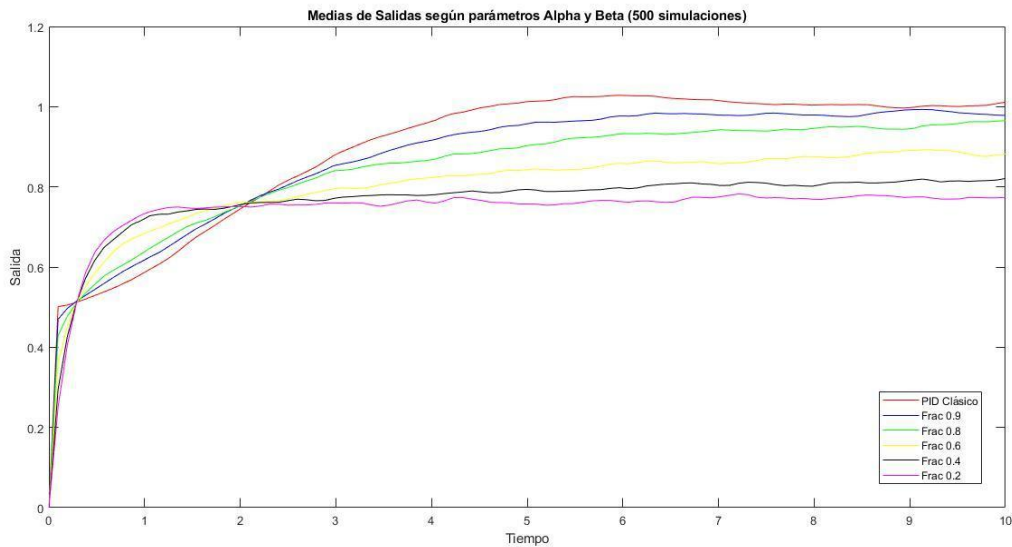


Figura 108. Salidas según parámetros Alpha y Beta con Ruido de Perturbación (500 simulaciones).

10. Análisis Desviación Típica según parámetros Alpha y Beta

Se ha procedido a realizar una comparación entre las gráficas de las desviaciones típicas variando los valores de Alpha y Beta de un PID fraccionario y un PID clásico, para observar cuál era el comportamiento del sistema modificando estos parámetros.

10.1 Desviaciones Típicas con ruido de perturbación (100 simulaciones)

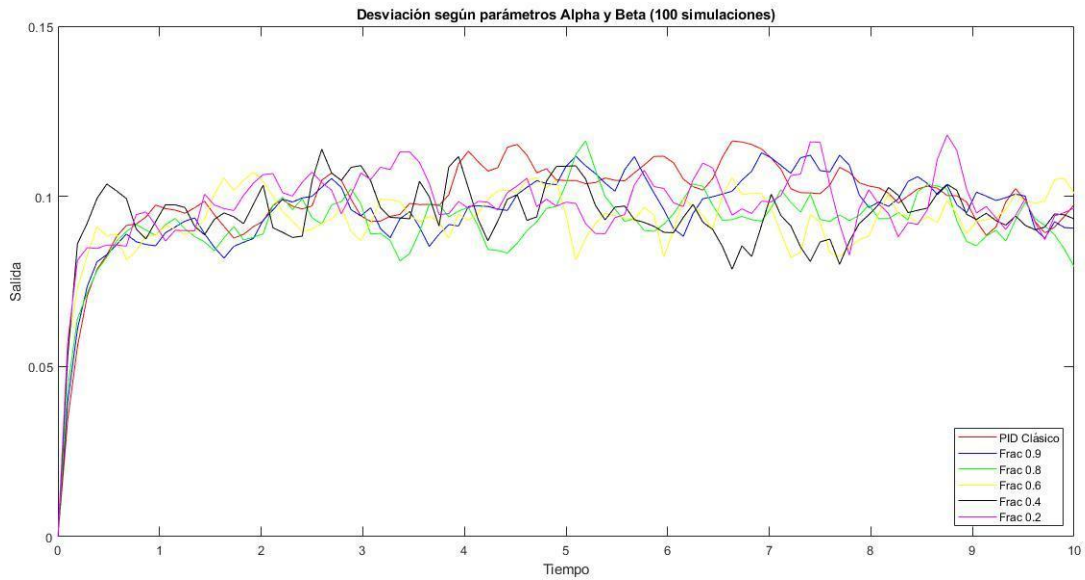


Figura 109.Desviación según parámetros Alpha y Beta (100 simulaciones).

Como podemos observar, la respuesta para el PID clásico y para los distintos valores aplicados al PID fraccionario obtenemos una desviación similar para cada caso, por lo que podemos concluir que cuando los valores de Alpha y Beta son iguales la desviación no se ve afectada.



10.2 Desviaciones Típicas con ruido de perturbación (500 simulaciones).

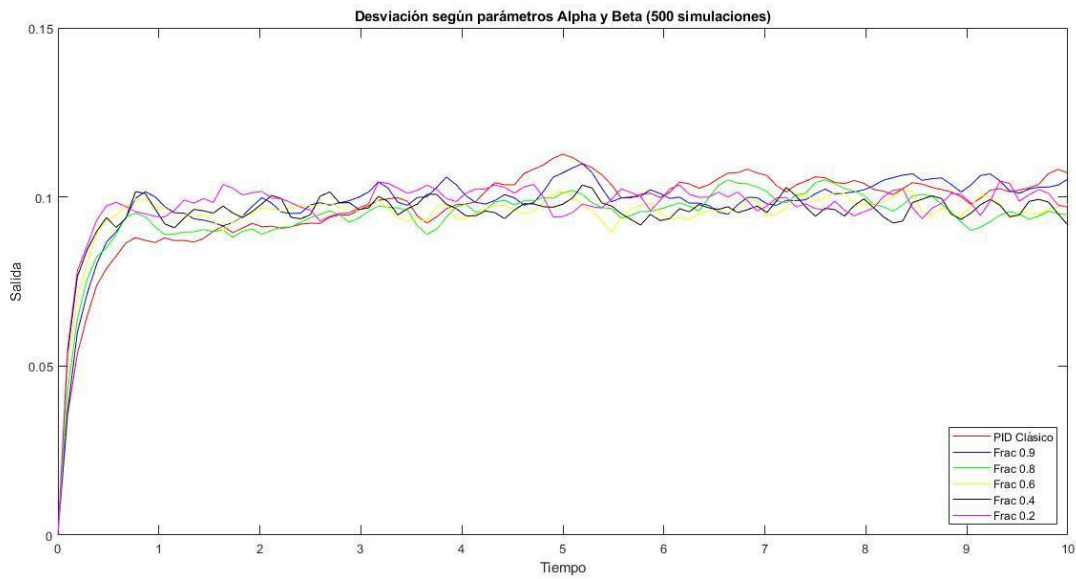


Figura 110.Desviación según parámetros Alpha y Beta (500 simulaciones).

En este caso, ocurre lo mismo que para las 100 simulaciones, aunque se varíen los valores de Alpha y Beta obtenemos que las gráficas de desviación típica para cada valor son muy similares y la desviación apenas se ve afectada.



11. Análisis Media Desviación Típica según parámetros Alpha y Beta.

Como no podemos sacar conclusiones de las gráficas anteriormente descritas, hemos procedido a realizar otro análisis de desviaciones típicas pero esta vez calculando las medias de las desviaciones cuando llegan al estacionario, cómo podemos observar es a partir del instante de tiempo 1 aproximadamente cuando esto ocurre. Una vez acotado el estudio para el tiempo mayor a 1 seg, se fueron variando los valores de Alpha y Beta de la siguiente manera, fijamos uno de los dos valores en 1 y el otro valor lo vamos modificando entre 0.2, 0.4, 0.6, 0.8 y 0.9.

11.1 Media Desviaciones típicas con ruido de perturbación manteniendo Alpha fija y variando Beta (500 simulaciones).

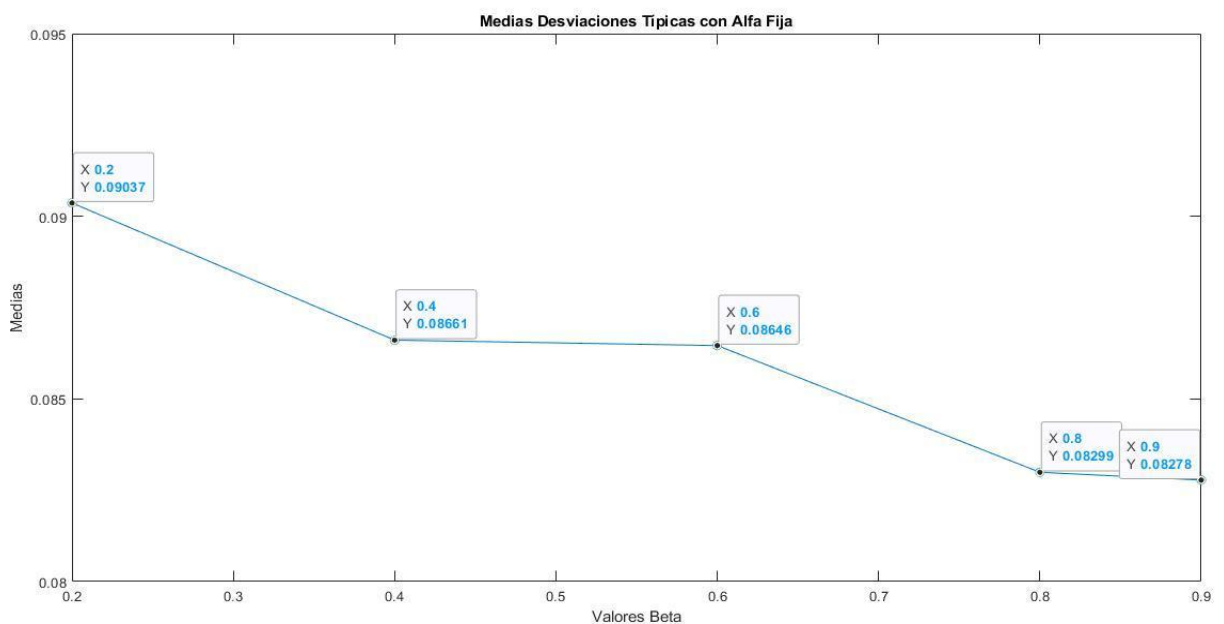


Figura 111. Media Desviaciones Típicas con Alfa Fija.

Tal y como podemos observar en la gráfica anterior, a medida que los valores de Beta (eje x) aumentan los valores de las medias de las desviaciones típicas (eje y) disminuyen.



11.2 Media Desviaciones típicas con ruido de perturbación manteniendo Beta fija y variando Alpha (500 simulaciones).

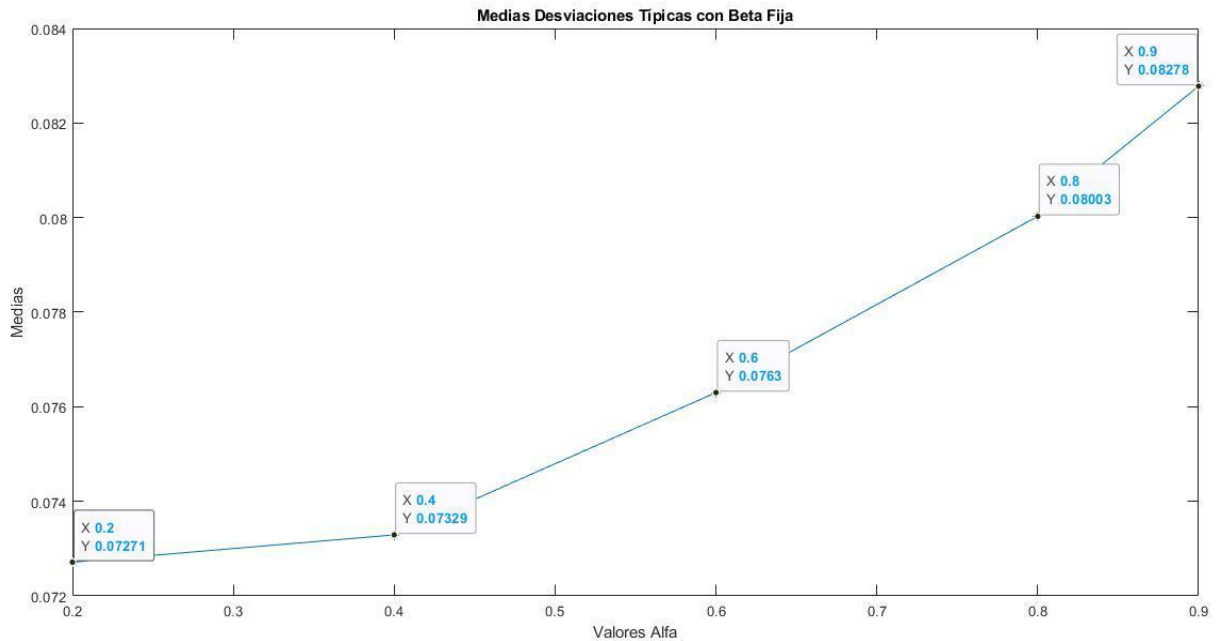


Figura 112. Media Desviaciones Típicas con Beta Fija.

Para este caso en el que el parámetro que variamos es Alpha se puede comprobar que a medida que los valores de este aumentan (eje x) la media de las desviaciones (eje y) también aumenta.

12. Conclusiones de los datos obtenidos.

Una vez hecho todos los análisis, podemos sacar conclusiones tanto de los datos como de las gráficas. Primero, al ver las salidas según los parámetros Alfa y Beta, se puede notar que, aunque el objetivo de un controlador PID es llegar al valor de consigna establecido, hay un tipo de PID (en este caso el fraccionario) que no llega a dicho valor. Nótese también que el transitorio de dicho PID es mucho más “suave” que el del Clásico por lo que podemos intuir que una combinación adecuada de los parámetros puede llegar a darnos una salida suave y eficiente.



El desfase que logramos ver de los PID fraccionarios con respecto al PID clásico, se puede evidenciar también en las gráficas de la densidad de probabilidad, cuando vemos que a lo largo del tiempo la media (amplitud de la gráfica) de los valores de salida del PID Clásico se va acercando y desplazando hacia el valor de consigna, mientras que en el caso fraccionario se alejan más del valor de consigna según disminuyen los parámetros Alfa y Beta.

Por otro lado, al comparar las desviaciones típicas, se pudo observar, que independientemente de si se llega o no al valor de consigna, la distribución de los valores con respecto a la media es simétrica. Esto también se puede ver en las gráficas de densidad de probabilidad, comprobándose que los anchos de las gráficas son muy parecidos a lo largo del tiempo.

Cabe destacar que al introducir valores Gaussianos (Distribución Normal) en el ruido, se obtuvieron salidas de la misma forma, por lo que la distribución de los valores no se vio afectada.

Se analizaron los siguientes tres casos:

- Se varía Alpha y se mantiene fija en un valor Beta.
- Se varía Beta y se mantiene fija en un valor Alpha.
- Se varían Alpha y Beta pero siempre son iguales.

concluyéndose lo siguiente:

Si Alpha se modifica obtenemos una gráfica ascendente, lo que nos indica que cuanto mayor es el valor, más desviación tenemos.

Si Beta se modifica obtenemos una gráfica descendente, lo que nos indica que cuanto mayor es el valor, menos desviación tenemos.

Si los valores de los parámetros son iguales la desviación no se ve afectada aunque aumentemos o disminuyamos los valores, por lo que en este caso se produce una compensación de las desviaciones.



13. Final Degree Project's General Conclusion.

Throughout this project we have developed knowledge that complements what we have learnt in the degree, these being part of the field of Control Engineering. Although in the degree we covered studies related to PID controllers, in the TFG we were able to increase our learning by investigating fractional PIDs. Although this study was only a first contact between the two controllers, we were able to see the differences and similarities between their behaviour and thereby give rise to further research in this field. Having another alternative when using a type of controller, and knowing that the fractional PID has a wide range of possibilities, opens the door for industries to carry out improvement, updating or creation works that contain this PID.

On the other hand, we increased our knowledge in the implementation and development of Matlab, working with new tools and functions that may be useful in the future. Finally, the fact that we did not obtain optimal results at first, made us investigate further into the coding and tuning of the controllers, which was beneficial for the development of the work.

14. Referencias.

- PID clásico y fraccionario:

[1] Fernández Presmanes, Á. (2017). *Diseño de un algoritmo PID aplicado al Páncreas Artificial*[Trabajo fin de grado, Universidad Politécnica de Madrid]. <http://oa.upm.es/52309/>

[2] Blas M. Vinagre, & Concepción A. Monje. (2006). Introducción al Control Fraccionario. *Revista Iberoamericana de Automática e Informática Industrial*, 3, 5–23. https://www.researchgate.net/publication/28141974_Introduccion_al_Control_Fraccionario

[3] *Controlador PID - Control Automático - Picuino*. (s. f.). Picuino. <https://www.picuino.com/es/arduprog/control-pid.html>

[4] Viola Villamizar, J. B. (2012). *Control Fraccionario al diseño de controladores*[Trabajo fin de grado, Universidad Pontificia Bolivariana]. https://repository.upb.edu.co/bitstream/handle/20.500.11912/5076/digital_24700.pdf?sequence=3



[5] Martín Reyes, G. (2012). La función de probabilidad normal: Características y aplicaciones. *eXtoicos*, 6, 107–110. <http://www.extoikos.es/n6/pdf/16.pdf>

[6] Estévez Pérez, S., León Gil, J. and Matesanz García, A., (2015). *Aplicación de técnicas de control fraccionario y comparación de resultados con respecto al control clásico*. [Trabajo fin de Grado. Universidad de La Laguna]. <https://riull.ull.es/xmlui/bitstream/handle/915/1118/Aplicacion+de+tecnicas+de+control+fraccionario+y+comparacion+de+resultados+con+respecto+al+control+clasico.pdf;jsessionid=0108F82D41487BE1A102BB2F20B85903?sequence=1>

- Imágenes:

[7] Blas M. Vinagre, & Concepción A. Monje. (2006). Introducción al Control Fraccionario. *Revista Iberoamericana de Automática e Informática Industrial*, 3, 5–23. https://www.researchgate.net/publication/28141974_Introduccion_al_Control_Fraccionario

- Códigos de MATLAB y Esquemas Simulink: Empresa: MathWorks.

[8] *MathWorks - Creadores de MATLAB y Simulink - MATLAB y Simulink*. (s. f.). MATLAB & Simulink. <https://es.mathworks.com>

15. Anexos.

15.1. Códigos.

100 SIMULACIONES SIN RUIDO

-Definicion.m

```
%----- PID CLÁSICO -----  
-----  
  
Z=zeros(105,100); %defino la matriz donde voy a almacenar las salidas  
tiempo=linspace(0,10,105); %tiempo  
tiempodef=linspace(0,10,105); %tiempo de referencia para la  
interpolación  
W=zeros(length(tiempo),2); %el ruido tendrá la misma dimension que el  
tiempo  
W(:,1)=tiempo; %la primera columna será el tiempo y la segunda los  
datos  
entrada=zeros(length(tiempo),100); %creo la matriz de donde se cogeran  
los ruidos que entran al sistema  
sreal=zeros(length(tiempo),1); %definimos el tamaño de la salida  
inicial  
srealdef=zeros(length(tiempo),1); %sreal luego de la interpolacion
```

-BucleRuidoCero.m

```
%----- PID CLÁSICO -----  
-----  
  
for j=1:100  
W(:,2)=entrada(:,j); %almacenamos cada columna de la entrada en la  
segunda columna del ruido  
sim('der_fracv2') %enviamos datos al simulink  
srealdef=interp1(tiempo,sreal,tiempodef) %las dimensiones van  
cambiando debido a ajustes que hace  
%el simulink por lo tanto interpolamos y llevamos las salidas al rango  
de  
%tiempo que queremos  
srealdef=srealdef'; %traspuesta  
Z(:,j)=srealdef; %llenamos la matriz de salida Z con cada vector de  
salida  
  
size(sreal)  
size(srealdef)  
end
```

-MaxMinZ.m

```
%----- PID CLÁSICO -----  
-----  
  
%DENSIDAD DE PROBABILIDAD  
  
MinZ = min(min(Z)) %hallamos el valor max y min de la matrix de salida  
MaxZ = max(max(Z))  
  
N = 25; %valor para hacer las divisiones
```

```

Div = (MaxZ - MinZ)/N; %Divisiones
Cuenta=zeros(107,26); %creamos el vector donde se acumularan los
valores
Cuenta(107,:)=(MinZ:Div:MaxZ); %en la fila 107 almacenamos las
divisiones
%para luego hacer la cuenta en la primera fila

```

-BucleCuenta.m

```

%----- PID CLÁSICO -----
-----

j=1
x=1
p=1

while j<=105
    for i=1:100 %recorremos columnas
        for x=1:25
            if Z(j,i)>=Cuenta(107,x) && Z(j,i)<=Cuenta(107,x+1) %esta
entre esos dos valores
                Cuenta(106,x+1)=Cuenta(106,x+1)+1 %suma 1 al valor
correspondiente de la primera fila y los acumulamos en la fila 106
            else
                end
            end
        end
        if p<=105
            Cuenta(p,:)=Cuenta(106,:) %una vez acumulados, los vamos
copiando en todo el vector y así tener
                                %en cada fila cada salida acumulada
            p=p+1
            Cuenta(106,:)=0
        else
            p=1
        end
    end

j=j+1 %una vez recorras toda la fila, pasa a la siguiente
end

```

-DesviaciónRuidoCero.m

```

%----- PID CLÁSICO -----
-----

DesviacionZ=std(Z'); %sacamos la desviación de cada columna y se
almacenan en una matriz de una fila por 105 columnas

```

-MediaRuidoCero.m

```

%----- PID CLÁSICO -----
-----

```

```
MediaZ=mean(Z'); %sacamos la media de cada columna y se almacenan en
una matriz de una fila por 105 columnas
```

-Gráficos.m

```
%----- PID CLÁSICO -----
-----

%Salida sin ruido de perturbación
plot(tiempodef,Z)
axis([-0.1 10 0 1.5])

%Densidad de Probabilidad sin Ruido de Perturbación
for i=1:10
subplot(2,5,i),plot(Cuenta(107,:),Cuenta(i*5,+)/100)
axis([0 1.5 0 1.2])
end

%Desviación sin Ruido de Perturbación
plot(tiempodef,DesviacionZ)
axis([0 10 -0.5 0.5])

%Media sin Ruido de Perturbación
plot(tiempodef,MediaZ)
axis([-0.1 10 0 1.5])
```

500 SIMULACIONES CON RUIDO

-Definición.m

```
%----- PID FRACCIONARIO -----
-----

Zfrac=zeros(105,500); %defino la matriz donde voy a almacenar las
salidas
tiempo=linspace(0,10,105); %tiempo
tiempodef=linspace(0,10,105); %tiempo de referencia para la
interpolación
Wfrac=zeros(length(tiempo),2); %el ruido tendrá la misma dimension que
el tiempo
Wfrac(:,1)=tiempo; %la primera columna será el tiempo y la segunda los
datos
entradafrac=zeros(length(tiempo),500); %creo la matriz de donde se
cogoran los ruidos que entran al sistema
for i=1:500
aleatoriosfrac=randn(105,1); %creo una variables que contenga números
aleatorios gaussianos de media cero y desviación típica 1
entradafrac(:,i)=aleatoriosfrac;%la entrada será una matriz donde
hacemos 100 simulaciones
%con ruidos gaussianos de media 0 y desviación típica 1
end
sfrac=zeros(length(tiempo),1); %definimos el tamaño de la salida
inicial
sfracdef=zeros(length(tiempo),1); %sreal luego de la interpolacion
```


-BucleRuido.m

```
%----- PID FRACCIONARIO -----  
-----  
  
for j=1:500  
Wfrac(:,2)=entradafrac(:,j); %almacenamos cada columna de la entrada  
en la segunda columna del ruido  
sim('der_fracv3') %enviamos datos al simulink  
sfracdef=interp1(tiempo,sfrac,tiempodef) %las dimensiones van  
cambiando debido a ajustes que hace  
%el simulink por lo tanto interpolamos y llevamos las salidas al rango  
de  
%tiempo que queremos  
sfracdef=sfracdef'; %traspuesta  
Zfrac(:,j)=sfracdef; %llenamos la matriz de salida Z con cada vector  
de salida  
  
size(sfrac)  
size(sfracdef)  
end
```

-MaxMinZ.m

```
%----- PID FRACCIONARIO -----  
-----  
  
%DENSIDAD DE PROBABILIDAD  
  
MinZfrac = min(min(Zfrac)) %hallamos el valor max y min de la matrix  
de salida  
MaxZfrac = max(max(Zfrac))  
  
Nfrac = 125; %valor para hacer las divisiones  
Divfrac = (MaxZfrac - MinZfrac)/Nfrac; %Divisiones  
Cuentafrac=zeros(107,126); %creamos el vector donde se acumularan los  
valores  
Cuentafrac(107,:)=(MinZfrac:Divfrac:MaxZfrac); %en la segunda fila  
almacenamos las divisiones  
%para luego hacer la cuenta en la primera fila
```

-BucleCuenta.m

```
%----- PID FRACCIONARIO -----  
-----  
  
j=1  
xfrac=1  
pfrac=1  
  
while j<=105
```

```

        for i=1:500 %recorremos columnas
            for xfrac=1:125
                if Zfrac(j,i)>=Cuentafrac(107,xfrac) &&
Zfrac(j,i)<=Cuentafrac(107,xfrac+1) %esta entre esos dos valores
                    Cuentafrac(106,xfrac+1)=Cuentafrac(106,xfrac+1)+1
%suma 1 al valor correspondiente de la primera fila y los acumulamos
en la fila 101
                    else
                        end
                end
            end
        end
        if pfrac<=105
            Cuentafrac(pfrac,:)=Cuentafrac(106,:) %una vez acumulados, los
vamos copiando en todo el vector y así tener
                                %en cada fila cada salida acumulada
            pfrac=pfrac+1
            Cuentafrac(106,:)=0
        else
            pfrac=1
        end

j=j+1 %una vez recorras toda la fila, pasa a la siguiente
end

```

-DesviaciónRuido.m

```

%----- PID FRACCIONARIO -----
-----

DesviacionZfrac=std(Zfrac'); %sacamos la desviación de cada columna y
se almacenan en una matriz de una fila por 105 columnas

```

-MediaRuido.m

```

%----- PID FRACCIONARIO -----
-----

MediaZfrac=mean(Zfrac'); %sacamos la media de cada columna y se
almacenan en una matriz de una fila por 105 columnas

```

-Gráficos.m

```

%----- PID FRACCIONARIO -----
-----

%Salida sin ruido de perturbación
for i=1:10
    subplot(2,5,i),plot(tiempodef,Zfrac(:,i*25))
    axis([-0.1 10 0 1.5])
end

for i=11:20

```

```

subplot(2,5,(i-10)),plot(tiempodef,Zfrac(:,i*25))
axis([-0.1 10 0 1.5])
end

%Densidad de Probabilidad sin Ruido de Perturbación
for i=1:10
subplot(2,5,i),plot(Cuentafrac(107,:),Cuentafrac(i*5,)/500)
axis([-0.1 1.5 0 0.4])
end

for i=11:20
subplot(2,5,i-10),plot(Cuentafrac(107,:),Cuentafrac(i*5,)/500)
axis([-0.1 1.5 0 0.4])
end

%Desviación sin Ruido de Perturbación
plot(tiempodef,DesviacionZfrac)
axis([0 10 -0.1 0.2])

%Media sin Ruido de Perturbación
plot(tiempodef,MediaZfrac)
axis([-0.1 10 0 1.5])

```

GRÁFICOS COMBINADOS

-GC-100 SIN RUIDO.m

```

%----- 100 SIMULACIONES SIN RUIDO -----

%Salida sin ruido de perturbación
plot(tiempodef,Z(:,1),'r')
axis([-0.1 10 0 1.5])
title('Salida Sin Ruido de Perturbación (100 simulaciones)')
xlabel('Tiempo')
ylabel('Salida')

hold on

plot(tiempodef,Zfrac(:,1),'b')
axis([-0.1 10 0 1.5])
legend({'PID Clásico','PID Fraccionario'],'location','southeast')

hold off

%Densidad de probabilidad sin ruido de perturbación
for i=1:10
subplot(2,5,i),plot(Cuenta(107,:),Cuenta(i*5,)/100,'r')
axis([0 1.5 0 1.2])
xlabel('Intervalos')
ylabel('Cuentas')

hold on

subplot(2,5,i),plot(Cuentafrac(107,:),Cuentafrac(i*5,)/100,'b')
axis([0 1.5 0 1.2])

```

```

hold off
end

title('Densidad de Probabilidad Sin Ruido de Perturbación (100
simulaciones)', 'position', [-3 3 0])
legend({'PID Clásico', 'PID Fraccionario'}, 'location', 'southeast')

%Media sin ruido de perturbación
plot(tiempodef, MediaZ, 'r')
axis([-0.1 10 0 1.5])
title('Media Sin Ruido de Perturbación (100 simulaciones)')
xlabel('Tiempo')
ylabel('Salida')

hold on

plot(tiempodef, MediaZfrac, 'b')
axis([-0.1 10 0 1.5])
legend({'PID Clásico', 'PID Fraccionario'}, 'location', 'southeast')

hold off

%Desviación sin ruido de perturbación
plot(tiempodef, DesviacionZ, 'r')
axis([0 10 -0.5 0.5])
title('Desviación Sin Ruido de Perturbación (100 simulaciones)')
xlabel('Tiempo')
ylabel('Salida')

hold on

plot(tiempodef, DesviacionZfrac, 'b')
axis([0 10 -0.5 0.5])
legend({'PID Clásico', 'PID Fraccionario'}, 'location', 'southeast')

hold off

```

-GC-100 CON RUIDO.m

```

%----- 100 SIMULACIONES CON RUIDO -----

%Salida con ruido de perturbación
for i=1:10
subplot(2,5,i), plot(tiempodef, Z(:,i*5), 'r')
axis([-0.1 10 0 1.5])
xlabel('Tiempo')
ylabel('Salida')

hold on

subplot(2,5,i), plot(tiempodef, Zfrac(:,i*5), 'b')
axis([-0.1 10 0 1.5])

```

```

hold off
end

title('Salida con Ruido de Perturbación (100 simulaciones)
1', 'position', [-22 3.75 0])
legend({'PID Clásico', 'PID Fraccionario'}, 'location', 'southeast')

for i=11:20
subplot(2,5,(i-10)),plot(tiempodef,Z(:,i*5),'r')
axis([-0.1 10 0 1.5])
xlabel('Tiempo')
ylabel('Salida')

hold on

subplot(2,5,(i-10)),plot(tiempodef,Zfrac(:,i*5),'b')
axis([-0.1 10 0 1.5])

hold off
end

title('Salida con Ruido de Perturbación (100 simulaciones)
2', 'position', [-22 3.75 0])
legend({'PID Clásico', 'PID Fraccionario'}, 'location', 'southeast')

%Densidad de probabilidad con ruido de perturbación
for i=1:10
subplot(2,5,i),plot(Cuenta(107,:),Cuenta(i*5,)/100,'r')
axis([0 1.5 0 0.4])
xlabel('Intervalos')
ylabel('Cuentas')

hold on

subplot(2,5,i),plot(Cuentafrac(107,:),Cuentafrac(i*5,)/100,'b')
axis([0 1.5 0 0.4])

hold off
end

title('Densidad de Probabilidad Con Ruido de Perturbación (100
simulaciones) 1', 'position', [-3 1 0])
legend({'PID Clásico', 'PID Fraccionario'}, 'location', 'southeast')

for i=11:20
subplot(2,5,i-10),plot(Cuenta(107,:),Cuenta(i*5,)/100,'r')
axis([0 1.5 0 0.4])
xlabel('Intervalos')
ylabel('Cuentas')

hold on

subplot(2,5,i-10),plot(Cuentafrac(107,:),Cuentafrac(i*5,)/100,'b')
axis([0 1.5 0 0.4])

```

```

hold off
end

title('Densidad de Probabilidad Con Ruido de Perturbación (100
simulaciones) 2','position',[-3 1 0])
legend({'PID Clásico','PID Fraccionario'},'location','southeast')

%Media con ruido de perturbación
plot(tiempodef,MediaZ,'r')
axis([-0.1 10 0 1.5])
title('Media con Ruido de Perturbación (100 simulaciones)')
xlabel('Tiempo')
ylabel('Salida')

hold on

plot(tiempodef,MediaZfrac,'b')
axis([-0.1 10 0 1.5])
legend({'PID Clásico','PID Fraccionario'},'location','southeast')

hold off

%Desviación con ruido de perturbación
plot(tiempodef,DesviacionZ,'r')
axis([0 10 -0.5 0.5])
title('Desviación con Ruido de Perturbación (100 simulaciones)')
xlabel('Tiempo')
ylabel('Salida')

hold on

plot(tiempodef,DesviacionZfrac,'b')
axis([0 10 -0.5 0.5])
legend({'PID Clásico','PID Fraccionario'},'location','southeast')

hold off

```

-GC-500 SIN RUIDO.m

```

%----- 500 SIMULACIONES SIN RUIDO -----

%Salida sin ruido de perturbación
plot(tiempodef,Z(:,1),'r')
axis([-0.1 10 0 1.5])
title('Salida Sin Ruido de Perturbación (500 simulaciones)')
xlabel('Tiempo')
ylabel('Salida')

hold on

plot(tiempodef,Zfrac(:,1),'b')
axis([-0.1 10 0 1.5])
legend({'PID Clásico','PID Fraccionario'},'location','southeast')

```

```

hold off

%Densidad de probabilidad sin ruido de perturbación
for i=1:10
subplot(2,5,i),plot(Cuenta(107,:),Cuenta(i*5,+)/500,'r')
axis([0 1.2 0 1.2])
xlabel('Intervalos')
ylabel('Cuentas')

hold on

subplot(2,5,i),plot(Cuentafrac(107,:),Cuentafrac(i*5,+)/500,'b')
axis([0 1.2 0 1.2])

hold off
end

title('Densidad de Probabilidad Sin Ruido de Perturbación (500
simulaciones)','position',[-3 3 0])
legend({'PID Clásico','PID Fraccionario'],'location','southeast')

%Media sin ruido de perturbación
plot(tiempodef,MediaZ,'r')
axis([-0.1 10 0 1.5])
title('Media Sin Ruido de Perturbación (500 simulaciones)')
xlabel('Tiempo')
ylabel('Salida')

hold on

plot(tiempodef,MediaZfrac,'b')
axis([-0.1 10 0 1.5])
legend({'PID Clásico','PID Fraccionario'],'location','southeast')

hold off

%Desviación sin ruido de perturbación
plot(tiempodef,DesviacionZ,'r')
axis([0 10 -0.5 0.5])
title('Desviación Sin Ruido de Perturbación (500 simulaciones)')
xlabel('Tiempo')
ylabel('Salida')

hold on

plot(tiempodef,DesviacionZfrac,'b')
axis([0 10 -0.5 0.5])
legend({'PID Clásico','PID Fraccionario'],'location','southeast')

hold off

```

-GC-500 CON RUIDO.m

```

%----- 500 SIMULACIONES CON RUIDO -----

```

```

%Salida con ruido de perturbación
for i=1:10
subplot(2,5,i),plot(tiempodef,Z(:,i*25),'r')
axis([-0.1 10 0 1.5])
xlabel('Tiempo')
ylabel('Salida')

hold on

subplot(2,5,i),plot(tiempodef,Zfrac(:,i*25),'b')
axis([-0.1 10 0 1.5])

hold off
end

title('Salida con Ruido de Perturbación (500 simulaciones)
1','position',[-22 3.75 0])
legend({'PID Clásico','PID Fraccionario'},'location','southeast')

for i=11:20
subplot(2,5,(i-10)),plot(tiempodef,Z(:,i*5),'r')
axis([-0.1 10 0 1.5])
xlabel('Tiempo')
ylabel('Salida')

hold on

subplot(2,5,(i-10)),plot(tiempodef,Zfrac(:,i*5),'b')
axis([-0.1 10 0 1.5])

hold off
end

title('Salida con Ruido de Perturbación (500 simulaciones)
2','position',[-22 3.75 0])
legend({'PID Clásico','PID Fraccionario'},'location','southeast')

%Densidad de probabilidad con ruido de perturbación
for i=1:10
subplot(2,5,i),plot(Cuenta(107,:),Cuenta(i*5,+)/500,'r')
axis([0 1.5 0 0.1])
xlabel('Intervalos')
ylabel('Cuentas')

hold on

subplot(2,5,i),plot(Cuentafrac(107,:),Cuentafrac(i*5,+)/500,'b')
axis([0 1.5 0 0.1])

hold off
end

title('Densidad de Probabilidad Con Ruido de Perturbación (500
simulaciones) 1','position',[-3 0.25 0])
legend({'PID Clásico','PID Fraccionario'},'location','southeast')

```



```

for i=11:20
subplot(2,5,i-10),plot(Cuenta(107,:),Cuenta(i*5,+)/500,'r')
axis([0 1.5 0 0.1])
xlabel('Intervalos')
ylabel('Cuentas')

hold on

subplot(2,5,i-10),plot(Cuentafrac(107,:),Cuentafrac(i*5,+)/500,'b')
axis([0 1.5 0 0.1])

hold off
end

title('Densidad de Probabilidad Con Ruido de Perturbación (500
simulaciones) 2','position',[-3 0.25 0])
legend({'PID Clásico','PID Fraccionario'},'location','southeast')

%Media con ruido de perturbación
plot(tiempodef,MediaZ,'r')
axis([-0.1 10 0 1.5])
title('Media con Ruido de Perturbación (500 simulaciones)')
xlabel('Tiempo')
ylabel('Salida')

hold on

plot(tiempodef,MediaZfrac,'b')
axis([-0.1 10 0 1.5])
legend({'PID Clásico','PID Fraccionario'},'location','southeast')

hold off

%Desviación con ruido de perturbación
plot(tiempodef,DesviacionZ,'r')
axis([0 10 -0.5 0.5])
title('Desviación con Ruido de Perturbación (500 simulaciones)')
xlabel('Tiempo')
ylabel('Salida')

hold on

plot(tiempodef,DesviacionZfrac,'b')
axis([0 10 -0.5 0.5])
legend({'PID Clásico','PID Fraccionario'},'location','southeast')

hold off

```

ARCHIVOS COMPLEMENTARIOS

-AlphaFija.m

```
%MEDIA DE LA DESVIACIÓN TÍPICA

MediaDesv1=0 %v3
MediaDesv2=0 %108
MediaDesv3=0 %106
MediaDesv4=0 %104
MediaDesv5=0 %102

AcotDesv=zeros(85,1)

for i=20:105
    AcotDesv(i,1)=DesviacionZfrac(1,i)
end

MediaDesv1=mean(AcotDesv)
MediaDesv2=mean(AcotDesv)
MediaDesv3=mean(AcotDesv)
MediaDesv4=mean(AcotDesv)
MediaDesv5=mean(AcotDesv)

MatrizMediaDesv=zeros(2,5)

MatrizMediaDesv(2,1)=0.9
MatrizMediaDesv(2,2)=0.8
MatrizMediaDesv(2,3)=0.6
MatrizMediaDesv(2,4)=0.4
MatrizMediaDesv(2,5)=0.2

MatrizMediaDesv(1,1)=MediaDesv1
MatrizMediaDesv(1,2)=MediaDesv2
MatrizMediaDesv(1,3)=MediaDesv3
MatrizMediaDesv(1,4)=MediaDesv4
MatrizMediaDesv(1,5)=MediaDesv5

plot(MatrizMediaDesv(2,:),MatrizMediaDesv(1,:), '-o')
title('Medias Desviaciones Típicas con Alfa Fija')
xlabel('Valores Beta')
ylabel('Medias')
ylim([0.08,0.095])
```

-BetaFija.m

```
%MEDIA DE LA DESVIACIÓN TÍPICA

MediaDesv1=0 %v3
MediaDesv2=0 %081
MediaDesv3=0 %061
MediaDesv4=0 %041
```

```

MediaDesv5=0 %021

AcotDesv=zeros(85,1)

for i=20:105
    AcotDesv(i,1)=DesviacionZfrac(1,i)
end

MediaDesv1=mean(AcotDesv)
MediaDesv2=mean(AcotDesv)
MediaDesv3=mean(AcotDesv)
MediaDesv4=mean(AcotDesv)
MediaDesv5=mean(AcotDesv)

MatrizMediaDesv=zeros(2,5)

MatrizMediaDesv(2,1)=0.9
MatrizMediaDesv(2,2)=0.8
MatrizMediaDesv(2,3)=0.6
MatrizMediaDesv(2,4)=0.4
MatrizMediaDesv(2,5)=0.2

MatrizMediaDesv(1,1)=MediaDesv1
MatrizMediaDesv(1,2)=MediaDesv2
MatrizMediaDesv(1,3)=MediaDesv3
MatrizMediaDesv(1,4)=MediaDesv4
MatrizMediaDesv(1,5)=MediaDesv5

plot(MatrizMediaDesv(2,:),MatrizMediaDesv(1,:), '-*')
title('Medias Desviaciones Típicas con Beta Fija')
xlabel('Valores Alfa')
ylabel('Medias')

```

-Gráficos Desviación 500 SIM.m

```

%Desviación según parámetros Alpha y Beta
plot(tiempodef,DesviacionZ,'r')
axis([4 7 0.06 0.12])
title('Desviación según parámetros Alpha y Beta (500 simulaciones)')
xlabel('Tiempo')
ylabel('Salida')

hold on

plot(tiempodef,DesviacionZfrac09,'b')
axis([4 7 0.06 0.12])

plot(tiempodef,DesviacionZfrac08,'g')
axis([4 7 0.06 0.12])

plot(tiempodef,DesviacionZfrac06,'y')
axis([4 7 0.06 0.12])

plot(tiempodef,DesviacionZfrac04,'k')
axis([4 7 0.06 0.12])

```

```

plot(tiempodef,DesviacionZfrac02,'m')
axis([4 7 0.06 0.12])
legend({'PID Clásico','Frac 0.9','Frac 0.8','Frac 0.6','Frac
0.4','Frac 0.2'},'location','southeast')

hold off

```

-Gráficos Salidas 500 SIM.m

```

%Salidas según parámetros Alpha y Beta (sin ruido)
plot(tiempodef,Z,'r')
axis([0 10 0 1.2])
title('Salidas según parámetros Alpha y Beta (500 simulaciones)')
xlabel('Tiempo')
ylabel('Salida')

```

```
hold on
```

```

plot(tiempodef,Zfrac09,'b')
axis([0 10 0 1.2])

```

```

plot(tiempodef,Zfrac08,'g')
axis([0 10 0 1.2])

```

```

plot(tiempodef,Zfrac06,'y')
axis([0 10 0 1.2])

```

```

plot(tiempodef,Zfrac04,'k')
axis([0 10 0 1.2])

```

```

plot(tiempodef,Zfrac02,'m')
axis([0 10 0 1.2])
legend({'PID Clásico','Frac 0.9','Frac 0.8','Frac 0.6','Frac
0.4','Frac 0.2'},'location','southeast')

```

```
hold off
```

-Gráficos Medias Salidas 500 SIM.m

```

%Media de salidas según parámetros Alpha y Beta (con ruido)
plot(tiempodef,MediaZ,'r')
axis([0 10 0 1.2])
title('Medias de Salidas según parámetros Alpha y Beta (500
simulaciones)')
xlabel('Tiempo')
ylabel('Salida')

```

```
hold on
```

```

plot(tiempodef,MediaZfrac09,'b')
axis([0 10 0 1.2])

```

```

plot(tiempodef,MediaZfrac08,'g')
axis([0 10 0 1.2])

```

```
plot(tiempodef,MediaZfrac06,'y')
axis([0 10 0 1.2])

plot(tiempodef,MediaZfrac04,'k')
axis([0 10 0 1.2])

plot(tiempodef,MediaZfrac02,'m')
axis([0 10 0 1.2])
legend({'PID Clásico','Frac 0.9','Frac 0.8','Frac 0.6','Frac
0.4','Frac 0.2'},'location','southeast')

hold off
```