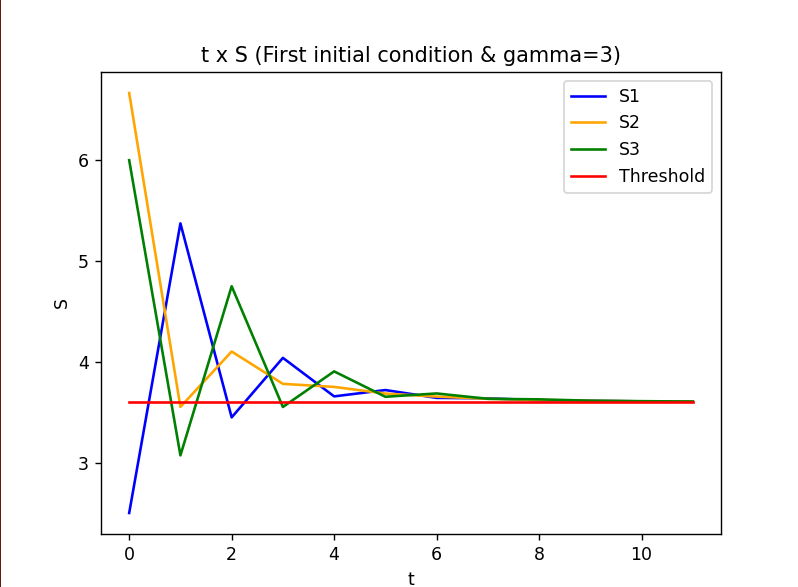
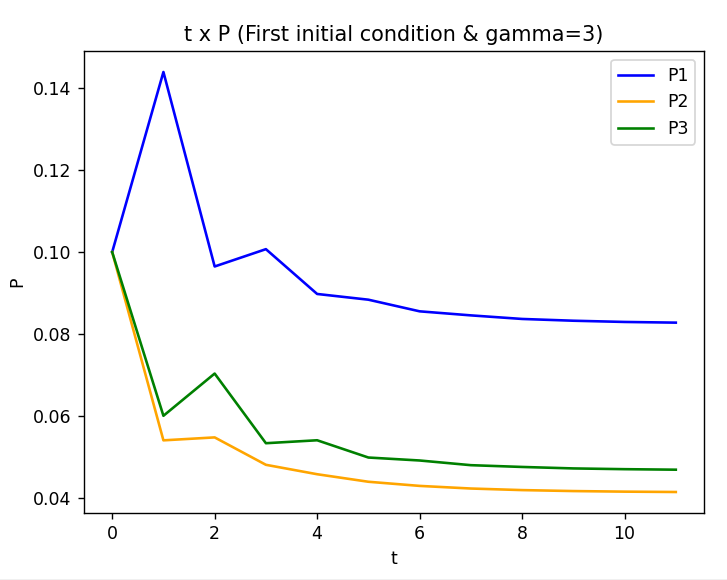
**EXERCISE 2:**

**2.**

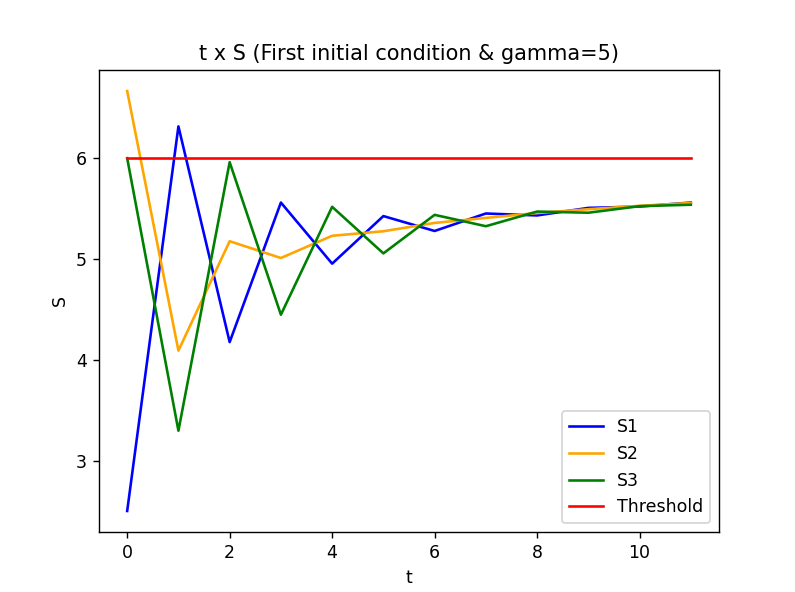
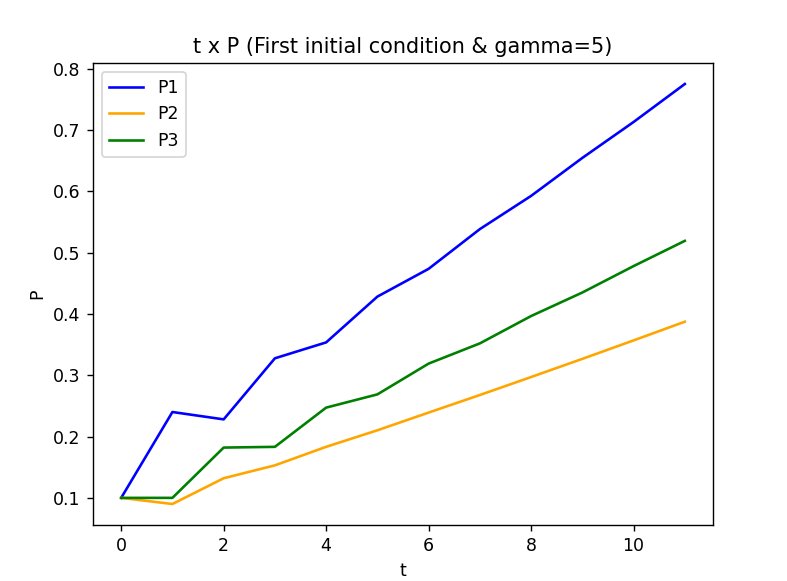
* p1 = p2 = p3 = 0.1 and gamma = 3

1. import numpy as np
2. import matplotlib.pyplot as plt
3. # With gamma = 3
4. G = np.array([[1, 0.2, 0.1],[0.1, 2, 0.1],[0.3, 0.1, 3]])
5. gamma = 3
6. alpha = 1.2
7. sigma = 0.1\*\*2
8. t = np.arange(0,12,1)
9. # First initial condition
10. p\_init = np.array([[0.1],[0.1],[0.1]])
11. A = alpha\*gamma\*np.array([[0, G[0][1]/G[0][0], G[0][2]/G[0][0]],
12. [G[1][0]/G[1][1], 0, G[1][2]/G[1][1]],
13. [G[2][0]/G[2][2], G[2][1]/G[2][2], 0]])
14. B = alpha\*gamma\*np.array([[1/G[0][0]],
15. [1/G[1][1]],
16. [1/G[2][2]]])
17. P1arr = []
18. P2arr = []
19. P3arr = []
20. S1arr = []
21. S2arr = []
22. S3arr = []
23. p=p\_init
24. for x in range(0,12):
25. q1 = sigma + (0.2\*p[1]) + (0.1\*p[2])
26. q2 = sigma + (0.1\*p[0]) + (0.1\*p[2])
27. q3 = sigma + (0.3\*p[0]) + (0.1\*p[1])
28. S = np.array([G[0][0]\*p[0]/q1, G[1][1]\*p[1]/q2, G[2][2]\*p[2]/q3])
29. P1arr.append(p[0])
30. P2arr.append(p[1])
31. P3arr.append(p[2])
32. S1arr.append(S[0])
33. S2arr.append(S[1])
34. S3arr.append(S[2])
35. p = np.dot(A,p) + np.dot(B,sigma)
37. target = [alpha\*gamma]\*12
38. plt.plot(t,S1arr, color='b', label='S1')
39. plt.plot(t,S2arr, color='orange', label='S2')
40. plt.plot(t,S3arr, color='g', label='S3')
41. plt.plot(t,target, color='r', label='Threshold')
42. plt.xlabel("t")
43. plt.ylabel("S")
44. plt.title("t x S (First initial condition & gamma=3)")
45. plt.legend()
46. plt.show()
47. plt.plot(t,P1arr, color='b', label='P1')
48. plt.plot(t,P2arr, color='orange', label='P2')
49. plt.plot(t,P3arr, color='g', label='P3')
50. plt.xlabel("t")
51. plt.ylabel("P")
52. plt.title("t x P (First initial condition & gamma=3)")
53. plt.legend()
54. plt.show()

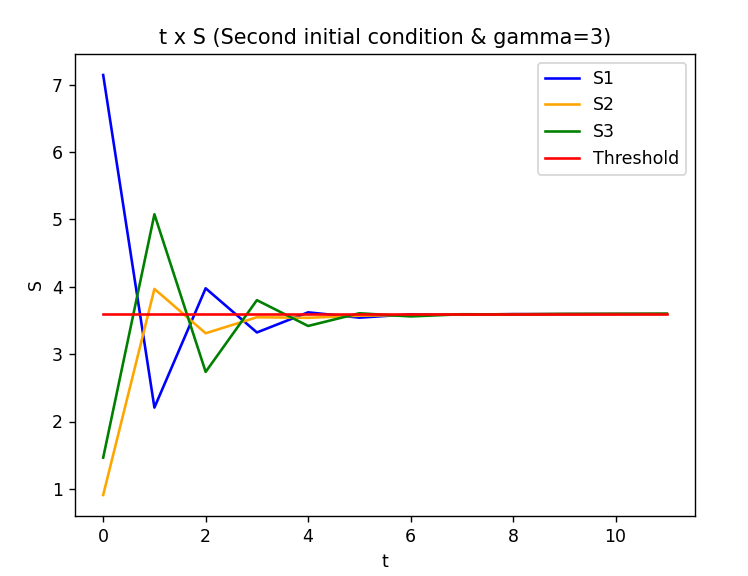
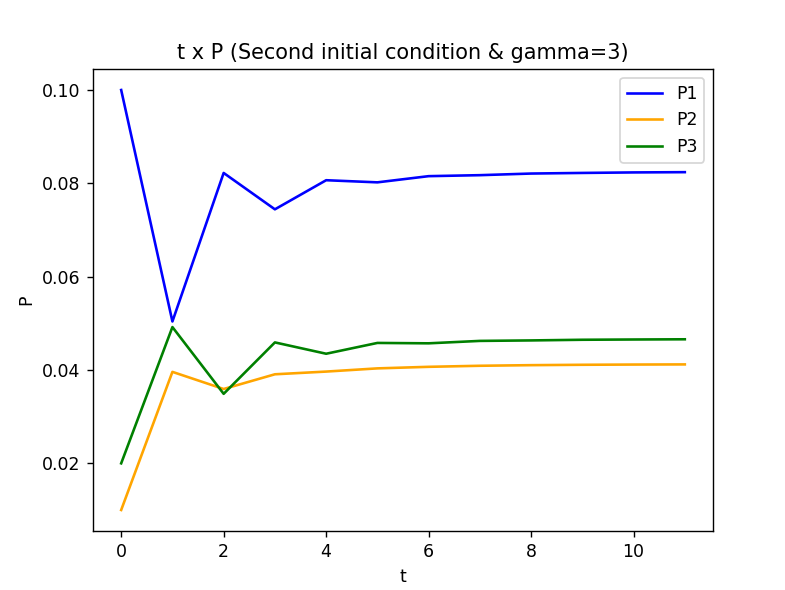
* p1 = p2 = p3 = 0.1 and gamma = 5

1. import numpy as np
2. import matplotlib.pyplot as plt
3. # With gamma = 5
4. G = np.array([[1, 0.2, 0.1],[0.1, 2, 0.1],[0.3, 0.1, 3]])
5. gamma = 5
6. alpha = 1.2
7. sigma = 0.1\*\*2
8. t = np.arange(0,12,1)
9. # First initial condition
10. p\_init = np.array([[0.1],[0.1],[0.1]])
11. A = alpha\*gamma\*np.array([[0, G[0][1]/G[0][0], G[0][2]/G[0][0]],
12. [G[1][0]/G[1][1], 0, G[1][2]/G[1][1]],
13. [G[2][0]/G[2][2], G[2][1]/G[2][2], 0]])
14. B = alpha\*gamma\*np.array([[1/G[0][0]],
15. [1/G[1][1]],
16. [1/G[2][2]]])
17. P1arr = []
18. P2arr = []
19. P3arr = []
20. S1arr = []
21. S2arr = []
22. S3arr = []
23. p=p\_init
24. for x in range(0,12):
25. q1 = sigma + (0.2\*p[1]) + (0.1\*p[2])
26. q2 = sigma + (0.1\*p[0]) + (0.1\*p[2])
27. q3 = sigma + (0.3\*p[0]) + (0.1\*p[1])
28. S = np.array([G[0][0]\*p[0]/q1, G[1][1]\*p[1]/q2, G[2][2]\*p[2]/q3])
29. P1arr.append(p[0])
30. P2arr.append(p[1])
31. P3arr.append(p[2])
32. S1arr.append(S[0])
33. S2arr.append(S[1])
34. S3arr.append(S[2])
35. p = np.dot(A,p) + np.dot(B,sigma)
37. target = [alpha\*gamma]\*12
38. plt.plot(t,S1arr, color='b', label='S1')
39. plt.plot(t,S2arr, color='orange', label='S2')
40. plt.plot(t,S3arr, color='g', label='S3')
41. plt.plot(t,target, color='r', label='Threshold')
42. plt.xlabel("t")
43. plt.ylabel("S")
44. plt.title("t x S (First initial condition & gamma=5)")
45. plt.legend()
46. plt.show()
47. plt.plot(t,P1arr, color='b', label='P1')
48. plt.plot(t,P2arr, color='orange', label='P2')
49. plt.plot(t,P3arr, color='g', label='P3')
50. plt.xlabel("t")
51. plt.ylabel("P")
52. plt.title("t x P (First initial condition & gamma=5)")
53. plt.legend()
54. plt.show()

**** ****

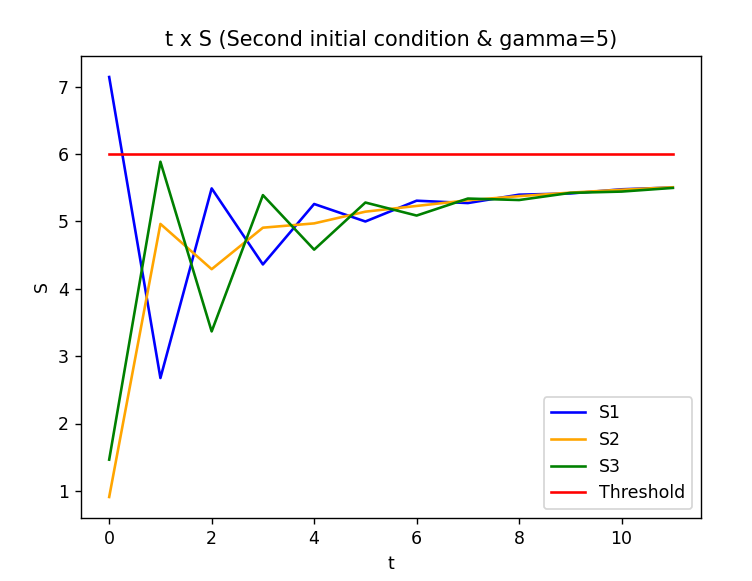
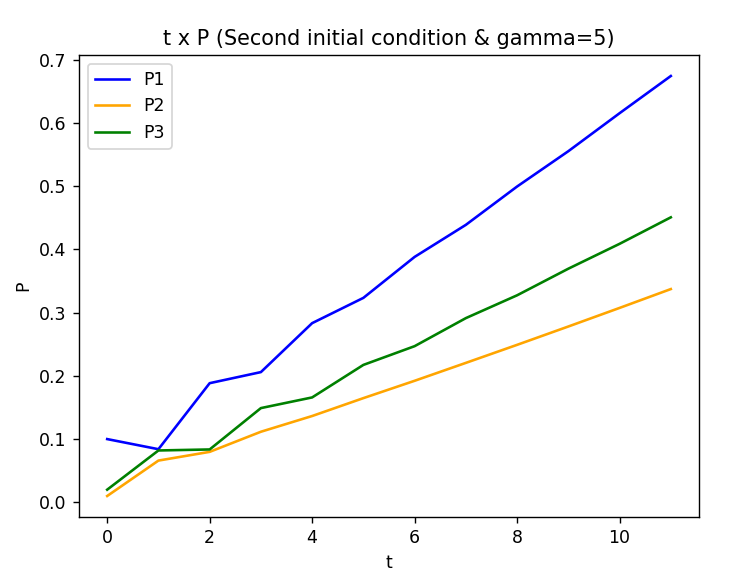
* p1 = 0.1, p2 = 0.01, p3 = 0.02 and gamma = 3

1. import numpy as np
2. import matplotlib.pyplot as plt
3. # With gamma = 3
4. G = np.array([[1, 0.2, 0.1],[0.1, 2, 0.1],[0.3, 0.1, 3]])
5. gamma = 3
6. alpha = 1.2
7. sigma = 0.1\*\*2
8. t = np.arange(0,12,1)
9. # Second initial condition
10. p\_init = np.array([[0.1],[0.01],[0.02]])
11. A = alpha\*gamma\*np.array([[0, G[0][1]/G[0][0], G[0][2]/G[0][0]],
12. [G[1][0]/G[1][1], 0, G[1][2]/G[1][1]],
13. [G[2][0]/G[2][2], G[2][1]/G[2][2], 0]])
14. B = alpha\*gamma\*np.array([[1/G[0][0]],
15. [1/G[1][1]],
16. [1/G[2][2]]])
17. P1arr = []
18. P2arr = []
19. P3arr = []
20. S1arr = []
21. S2arr = []
22. S3arr = []
23. p=p\_init
24. for x in range(0,12):
25. q1 = sigma + (0.2\*p[1]) + (0.1\*p[2])
26. q2 = sigma + (0.1\*p[0]) + (0.1\*p[2])
27. q3 = sigma + (0.3\*p[0]) + (0.1\*p[1])
28. S = np.array([G[0][0]\*p[0]/q1, G[1][1]\*p[1]/q2, G[2][2]\*p[2]/q3])
29. P1arr.append(p[0])
30. P2arr.append(p[1])
31. P3arr.append(p[2])
32. S1arr.append(S[0])
33. S2arr.append(S[1])
34. S3arr.append(S[2])
35. p = np.dot(A,p) + np.dot(B,sigma)
37. target = [alpha\*gamma]\*12
38. plt.plot(t,S1arr, color='b', label='S1')
39. plt.plot(t,S2arr, color='orange', label='S2')
40. plt.plot(t,S3arr, color='g', label='S3')
41. plt.plot(t,target, color='r', label='Threshold')
42. plt.xlabel("t")
43. plt.ylabel("S")
44. plt.title("t x S (Second initial condition & gamma=3)")
45. plt.legend()
46. plt.show()
47. plt.plot(t,P1arr, color='b', label='P1')
48. plt.plot(t,P2arr, color='orange', label='P2')
49. plt.plot(t,P3arr, color='g', label='P3')
50. plt.xlabel("t")
51. plt.ylabel("P")
52. plt.title("t x P (Second initial condition & gamma=3)")
53. plt.legend()
54. plt.show()

* p1 = 0.1, p2 = 0.01, p3 = 0.02 and gamma = 5

1. import numpy as np
2. import matplotlib.pyplot as plt
3. # With gamma = 5
4. G = np.array([[1, 0.2, 0.1],[0.1, 2, 0.1],[0.3, 0.1, 3]])
5. gamma = 5
6. alpha = 1.2
7. sigma = 0.1\*\*2
8. t = np.arange(0,12,1)
9. # Second initial condition
10. p\_init = np.array([[0.1],[0.01],[0.02]])
11. A = alpha\*gamma\*np.array([[0, G[0][1]/G[0][0], G[0][2]/G[0][0]],
12. [G[1][0]/G[1][1], 0, G[1][2]/G[1][1]],
13. [G[2][0]/G[2][2], G[2][1]/G[2][2], 0]])
14. B = alpha\*gamma\*np.array([[1/G[0][0]],
15. [1/G[1][1]],
16. [1/G[2][2]]])
17. P1arr = []
18. P2arr = []
19. P3arr = []
20. S1arr = []
21. S2arr = []
22. S3arr = []
23. p=p\_init
24. for x in range(0,12):
25. q1 = sigma + (0.2\*p[1]) + (0.1\*p[2])
26. q2 = sigma + (0.1\*p[0]) + (0.1\*p[2])
27. q3 = sigma + (0.3\*p[0]) + (0.1\*p[1])
28. S = np.array([G[0][0]\*p[0]/q1, G[1][1]\*p[1]/q2, G[2][2]\*p[2]/q3])
29. P1arr.append(p[0])
30. P2arr.append(p[1])
31. P3arr.append(p[2])
32. S1arr.append(S[0])
33. S2arr.append(S[1])
34. S3arr.append(S[2])
35. p = np.dot(A,p) + np.dot(B,sigma)
37. target = [alpha\*gamma]\*12
38. plt.plot(t,S1arr, color='b', label='S1')
39. plt.plot(t,S2arr, color='orange', label='S2')
40. plt.plot(t,S3arr, color='g', label='S3')
41. plt.plot(t,target, color='r', label='Threshold')
42. plt.xlabel("t")
43. plt.ylabel("S")
44. plt.title("t x S (Second initial condition & gamma=5)")
45. plt.legend()
46. plt.show()
47. plt.plot(t,P1arr, color='b', label='P1')
48. plt.plot(t,P2arr, color='orange', label='P2')
49. plt.plot(t,P3arr, color='g', label='P3')
50. plt.xlabel("t")
51. plt.ylabel("P")
52. plt.title("t x P (Second initial condition & gamma=5)")
53. plt.legend()
54. plt.show()

**HENCE, THE CONTROLLER CAN ACHIEVE THE GOAL OF REACHING ALPHA=GAMMA FOR GAMMA = 3, BUT NOT FOR GAMMA = 5.**