



Digital version

System response analysis

Mher Saribekyan A09210183

March 5, 2026

The systems

We have three systems:

a) $y[n] = 2 \cdot x[n]$

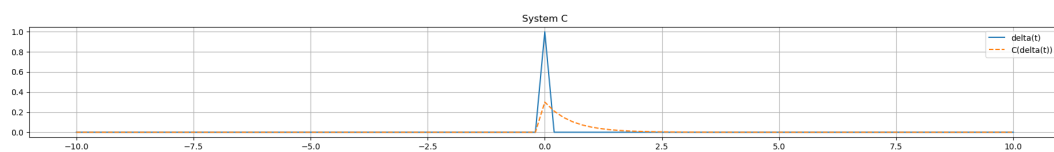
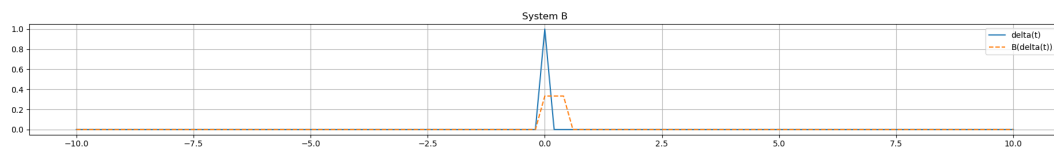
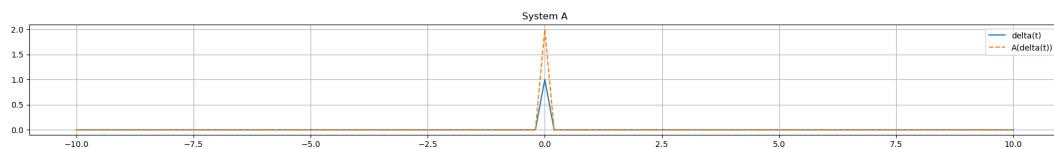
b) $y[n] = \frac{x[n] + x[n - 1] + x[n - 2]}{3}$

c) $y[n] = 0.3 \cdot x[n] + 0.7 \cdot y[n - 1]$

The first system amplifies the input signal by a factor of 2. The second system averages the last three points of the input signal. The last system a portion of the input signal and a portion of the output signal in the previous point, similar to a low pass filter.

Impulse response

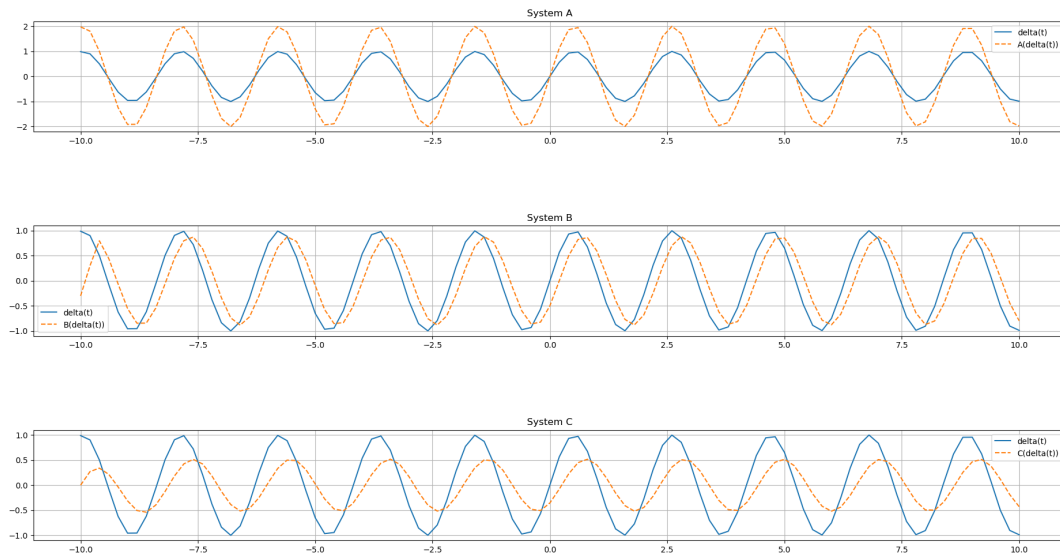
An impulse signal was given to the system, and the system response was analysed.



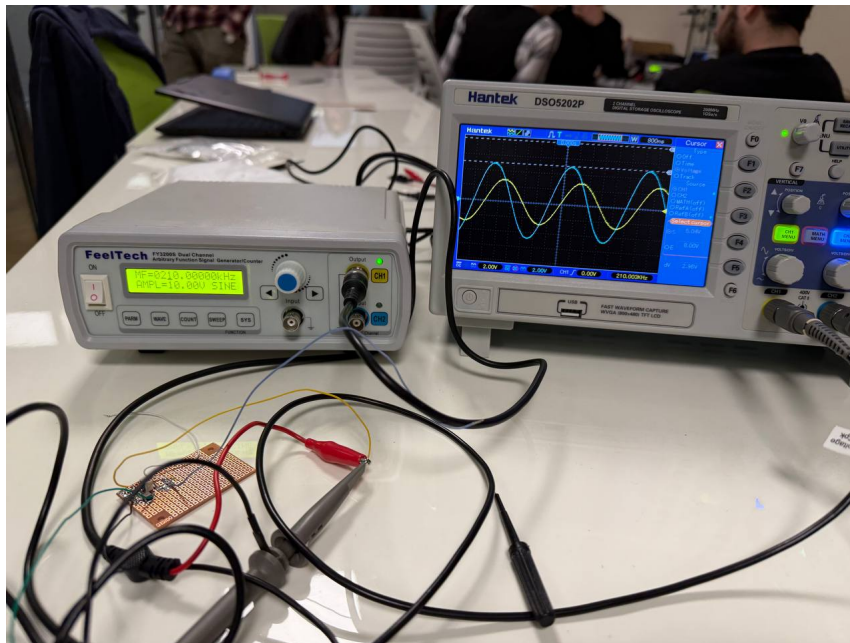
We see that the first system amplifies, the second signal averages and the last signal spreads the quick impulse over a longer period of time.

Sine wave response

A sine wave signal was given to the system, and the response was analysed.

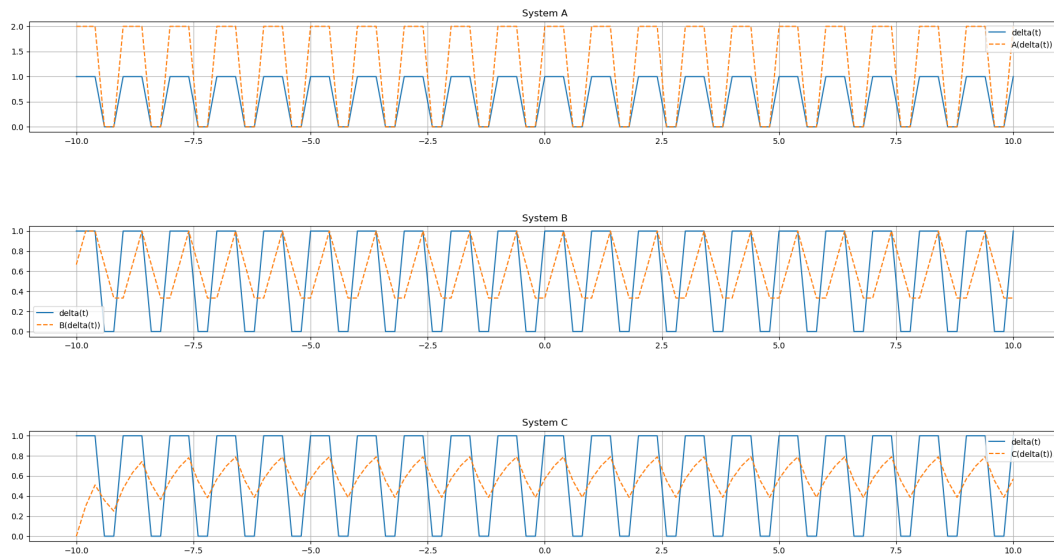


We see that for the first case, the sine wave has double the amplitude. For the second system, it is shifted a bit. The last system it's both shifted and attenuated, similar to an RC filter used in Lab 2.



Square wave response

A square wave signal was given to the system, and the response was analysed.



Same as the previous cases. For the third system we see that the output is similar to an RC capacitor charge/discharge curves.

Simulation code

```
import numpy as np
import matplotlib.pyplot as plt

t = np.linspace(-10, 10, 101)
# delta = np.where(t == 0, 1, 0) # Delta
delta = np.sin(3 * t) # Sine wave
# delta = np.where(t - np.floor(t) < 0.5, 1, 0) # Square Wave

def system_A(x):
    return 2 * x

def system_B(x):
    return (x + np.roll(x, 1) + np.roll(x, 2))/3

def system_C(x):
    y = np.zeros_like(x, dtype=float)
    for n in range(1, len(y)):
        y[n] = (0.3 * x[n]) + (0.7 * y[n - 1])
    return y

A_vals = system_A(delta)
B_vals = system_B(delta)
C_vals = system_C(delta)

fig, (ax1, ax2, ax3) = plt.subplots(3, 1, figsize=(10, 4))
```

```
ax1.plot(t, delta, label="delta(t)")
ax1.plot(t, A_vals, '--', label="A(delta(t))")
ax1.legend()
ax1.set_title("System A")
ax1.grid()

ax2.plot(t, delta, label="delta(t)")
ax2.plot(t, B_vals, '--', label="B(delta(t))")
ax2.legend()
ax2.set_title("System B")
ax2.grid()

ax3.plot(t, delta, label="delta(t)")
ax3.plot(t, C_vals, '--', label="C(delta(t))")
ax3.legend()
ax3.set_title("System C")
ax3.grid()

plt.tight_layout()
plt.show()
```