به نام خدا

گزارش پروژه سیگنال ها و سیستم ها

نگین فیروزیان

۹۴۳۱۰۱۸

در تابع showConvolution مقادیر دوتابع f1 و f2 و t0 که مقدار شیفت تابع دوم است گرفته می شود. و در حلقه ای به اندازه ی تناوب تابع کانوولوشن این دوتابع را با ضرب و انتگرال گرفتن تابع f1 و معکوس و شیفت یافته ی تابع f2 بدست می آورد.

و سپس توابع f1 و f2 و ضرب آن ها و کانوولوشن نهایی توسط ابزار کتابخانه MatplotLib رسم می شوند.

```
def showConvolution(f1, f2, t0):
    Fs = 50
    T = 5
    t = np.arange(-T, T, 1 / Fs)

    convolution = np.zeros(len(t))
    for n, t_ in enumerate(t):
        prod = lambda tau: f1(tau) * f2(t_ - tau)
        convolution[n] = scipy.integrate.simps(prod(t), t)

    f_shift = lambda t: f2(t0 - t)

    Heavy = lambda t: (t < t0) * 1
    convolution=convolution*Heavy(t)
    prod1 = lambda tau: f1(t) * f_shift(t)


    plt.subplot(311)
    plt.plot(t, f1(t), label=r'$f_1(\tau)$')
    plt.plot(t, f_shift(t), label=r'$f_2(t_0-\tau)$')


    plt.subplot(312)
    plt.plot(t, prod1(t), 'r-', label=r'$f_1(\tau)f_2(t_0-\tau)$')

    plt.subplot(313)
    plt.plot(t, convolution, 'g-', label='$(f_1*f_2)(t)$')
```

در خارج از تابع با استفاده از یک Slider یک ایونت ایجاد می کنیم که با هر تغییر این Slider تابع showConvolution با مقدار t0 ای که Slider مشخص

می شود فراخوانی می شود و همچنین نمودار آن ها دوباره با تغییرات اصلاح شده رسم می شود.

```
axis_color = 'lightgoldenrodyellow'
T_0 = 0

axT = plt.axes([0.35, 0.1, 0.45, 0.03], facecolor=axis_color)
sT = Slider(axT, 'T', -10, 10.0, valinit=T_0)

showConvolution(f1, f2, 4)

def sliders_on_changed(val):
    plt.subplot(312).cla()
    plt.subplot(311).cla()
    plt.subplot(313).cla()
    showConvolution(f1, f2,sT.val)
    fig.canvas.draw_idle()

sT.on_changed(sliders_on_changed)
```

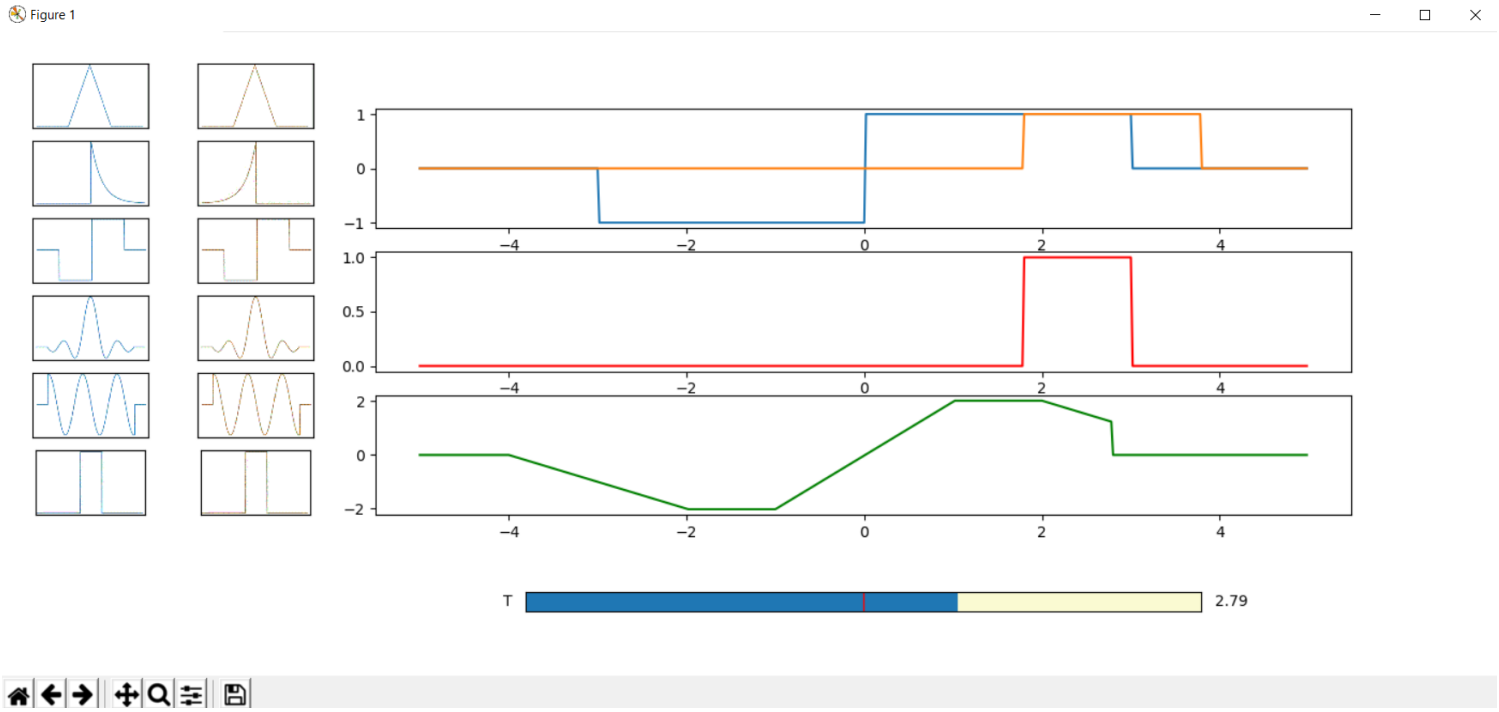
همچنین دکمه هایی برای انتخاب سیگنال های ورودی در نظر گرفته شده است که متناسب با انتخاب آن ها توابع f1 و f2 در showConvolution تغییر می کند و همچنین نمودار آن ها دوباره با تغییرات اصلاح شده رسم می شود.

نمونه ای از تولید یک دکمه در زیر آمده است:

```
PULSE = plt.imread("Rec.gif")
Pulse_button_ax = fig.add_axes([0.01, 0.25, 0.1, 0.1])
Pulse_button = Button(Pulse_button_ax, '', color=axis_color,
hovercolor='0.975',image=PULSE)
def Pulse_button_on_clicked(mouse_event):
    global f1
    f1 = lambda t: 1 * (abs(t - 0) < 3).astype(float)
    plt.subplot(312).cla()
    plt.subplot(311).cla()
    plt.subplot(313).cla()
    showConvolution(f1, f2,sT.val)
    fig.canvas.draw_idle()

Pulse_button.on_clicked(Pulse_button_on_clicked)
```

رابط کاربری این برنامه هم بصورت زیر است که نمودار نارنجی رنگ، توسط نوار متحرک پایین جابجا می شود و نتیجه ضرب و کانوولوشن آن در تابع دیگر، در دو نمودار پایین به نمایش در می آید. همچنین با انتخاب دکمه های موجود در سمت چپ مقدار توابع نارنجی و آبی را می توان به دلخواه تغییر داد.



سورس کد این برنامه در پوشه Source Code در فایل Project_9431018.py قرار دارد.