بسم الله الرحمان الرحیم

گزارش پروژه کامپیوتری اول

سیگنال ها و سیستم ها

عرفان باقری سولا

شماره دانشجویی: ۸۱۰۱۹۸۳۶۱

لیست مطالب

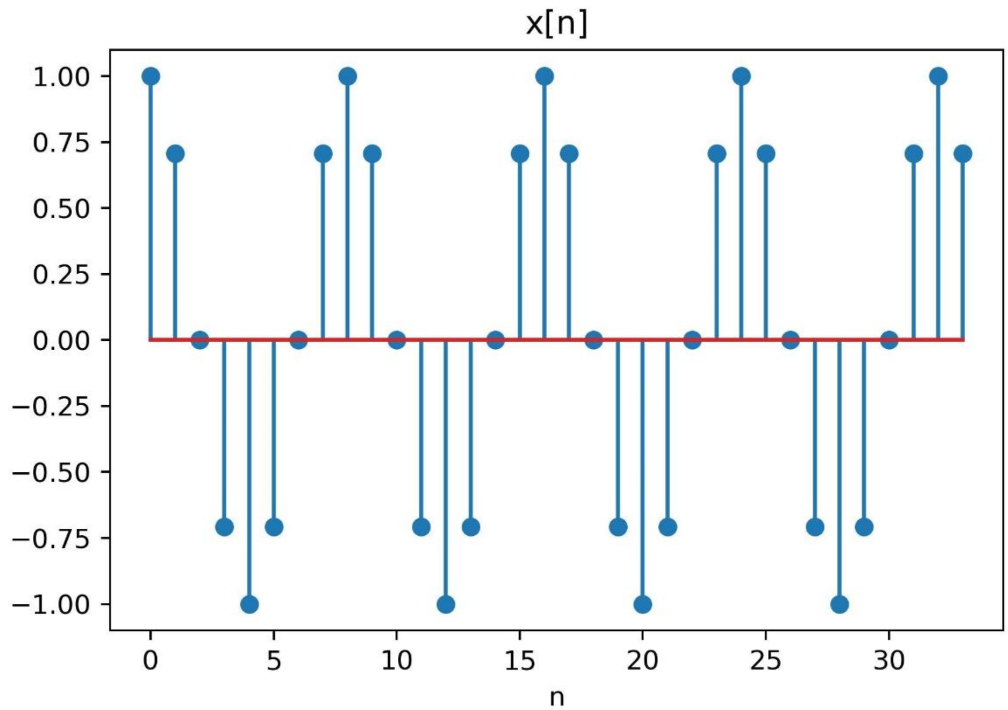
سوال ۱

در این سوال ضریب دلخواه k را ۰٫۱۲۵ در نظر میگیریم. به این ترتیب دوره تناوب (T) تابع برابر ۸ خواهد بود. البته k به صورت یک متغیر در نظر گرفته شده است و به راحتی قابل تغییر است. همچنین ضریب دلخواه A را برای راحتی کار در اینجا ۱ در نظر گرفته ایم.

برای این که بتوانیم خروجی سیستم های حافظه دار را به خوبی نمایش دهیم، در نهایت به اندازه یک دوره تناوب از هر دو سمت خروجی را برش میدهیم چون فقط میتوانیم دامنه محدودی را در نظر بگیریم و اگر این کار را نکنیم در لبه های آن دچار خطا می شویم.

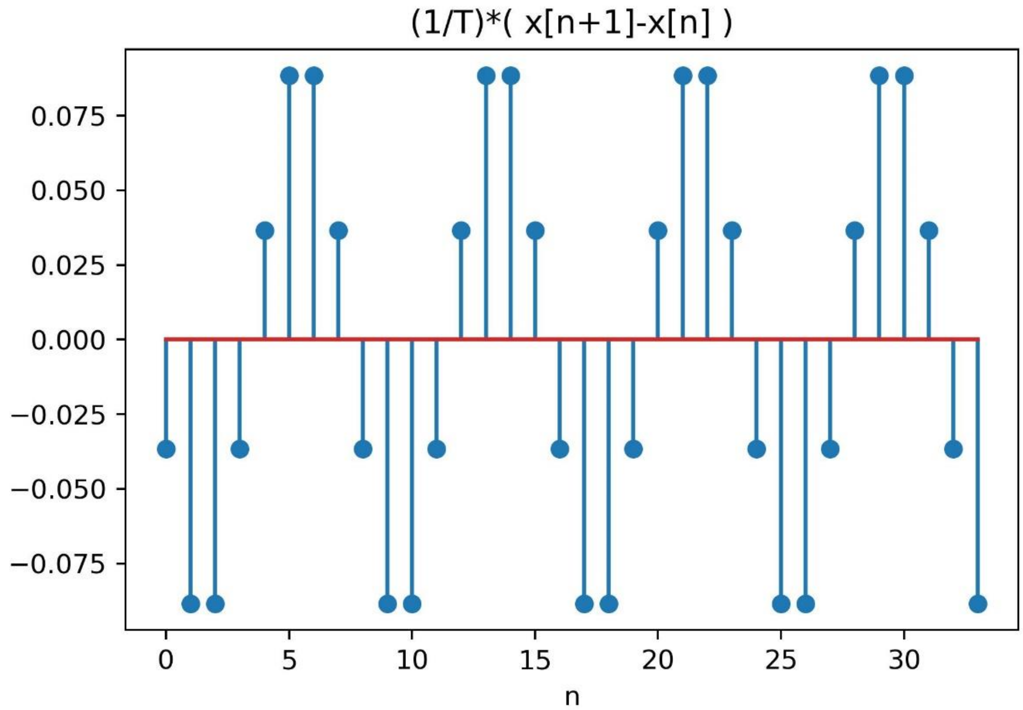
الف)

اینجا فرکانس نمونه برداری را برابر ۱ قرار می دهیم و عملا ورودی تابع تنها اعداد صحیح خواهند بود. تابع X را در شکل X-1-1 مشاهده می کنیم.

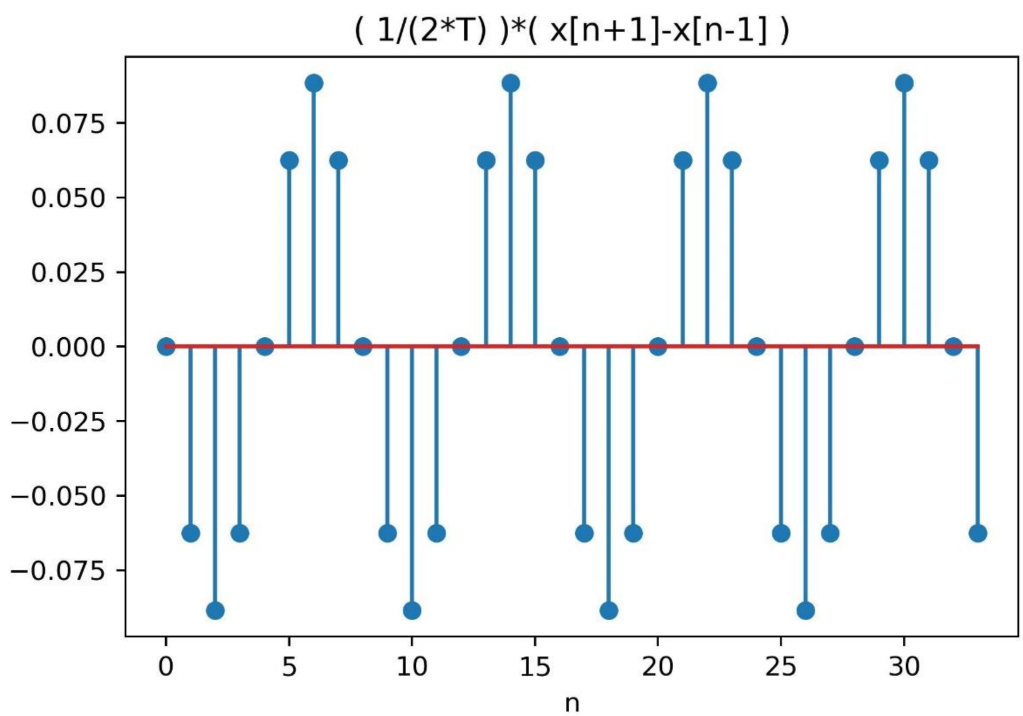


شکل X-1-1: تابع X بخش الف

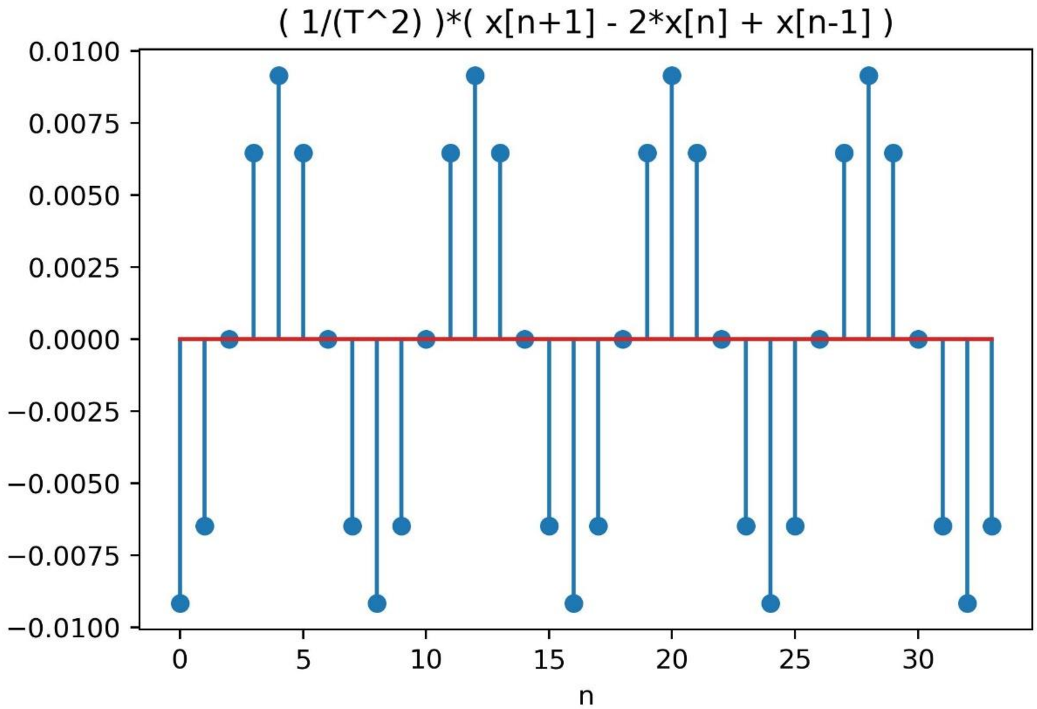
خروجی سیستم های این بخش در ادامه نمایش داده شده اند.



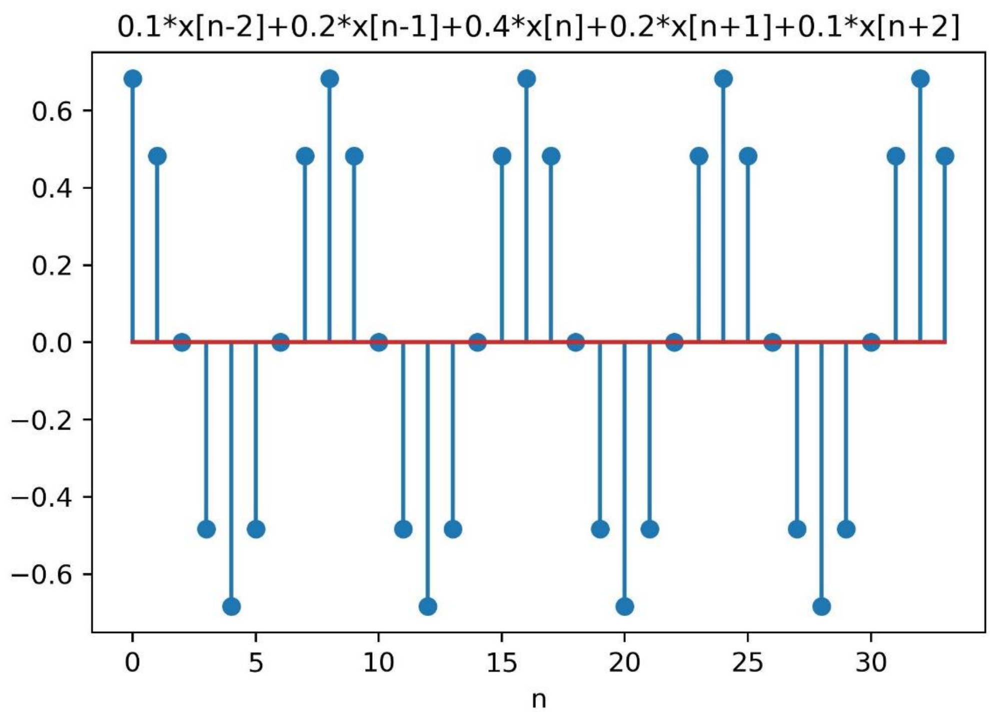
شکل A-1-1: خروجی سیستم LTI گسسته



شکل B-1-1: خروجی سیستم LTI گسسته



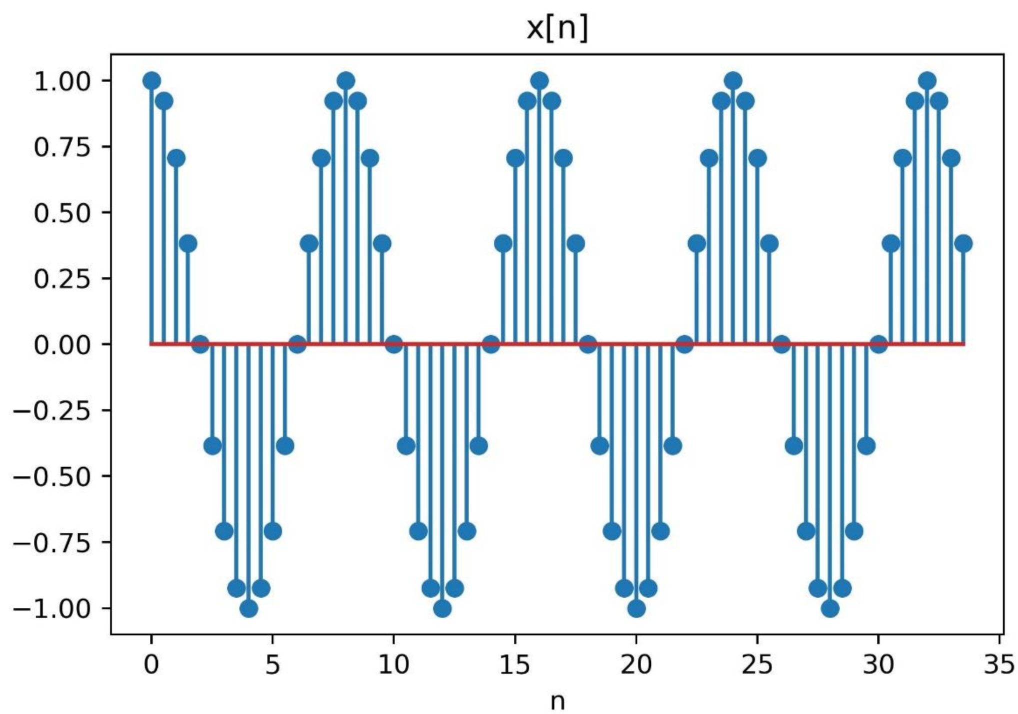
شکل 1-1-C: خروجی سیستم LTI گسسته



شکل 1-1-D: خروجی سیستم LTI گسسته

ب)

حالا فرکانس نمونه برداری بالا تر می بریم و دو برابر می کنیم تا تعداد نمونه ها افزایش یابد. با مشاهده شکل X-2-1 کاملا واضح است که تابع X جدید، به تابع پیوسته نزدیک تر شده است و منحنی های نرم تری دارد زیرا نمونه های بیشتری از تابع اصلی را در یک بازه برداشته ایم.

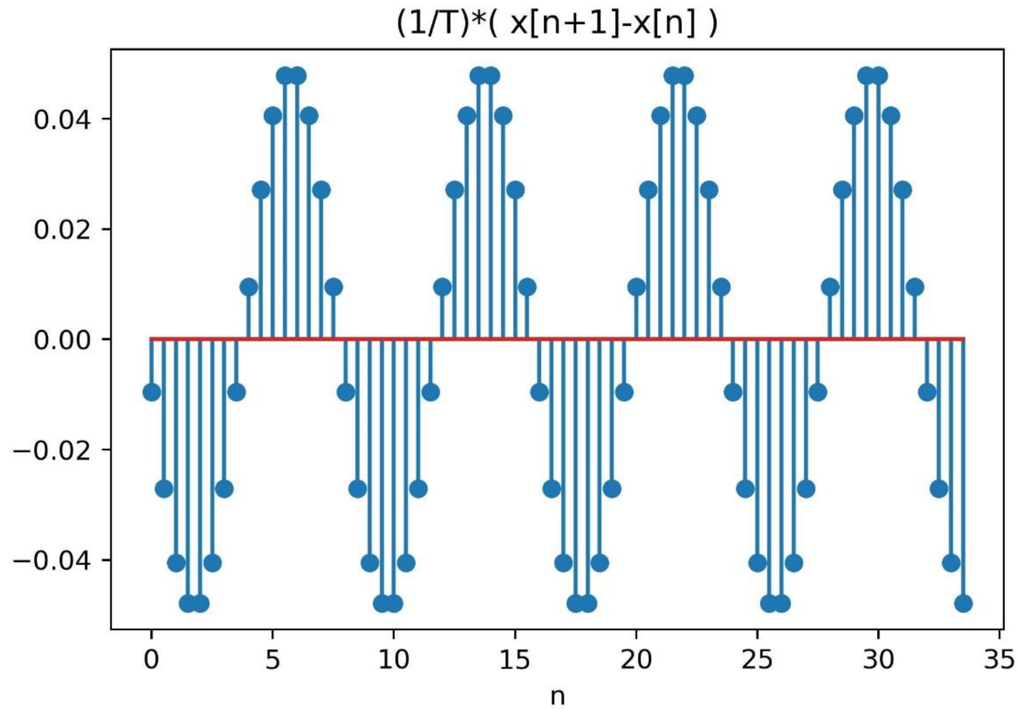


شکل X-2-1: تابع X بخش ب

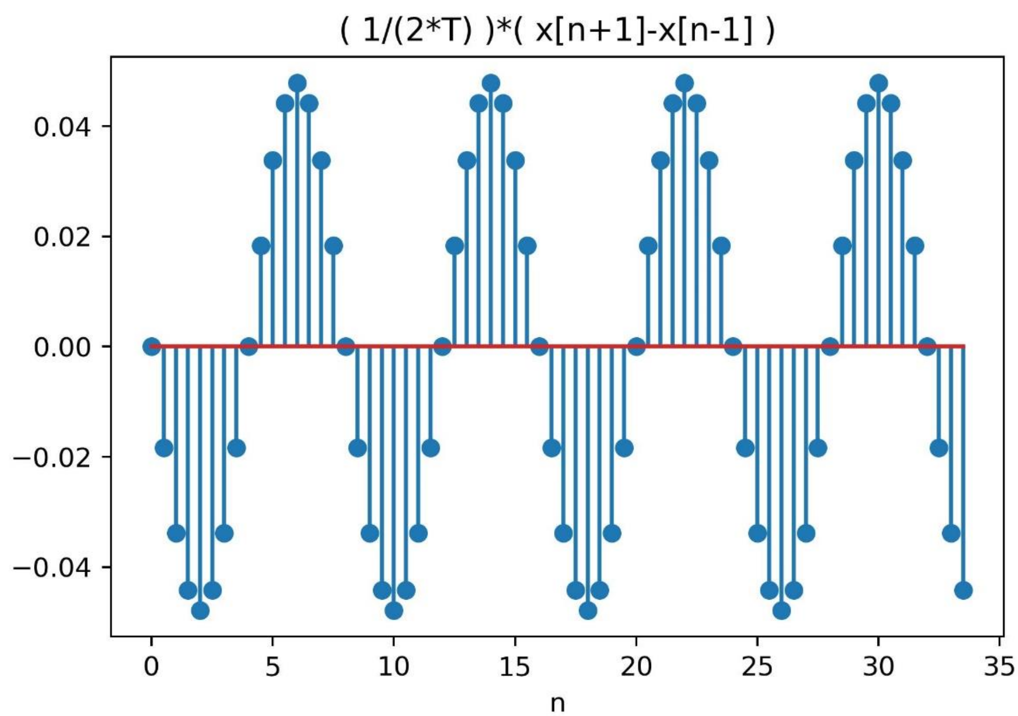
خروجی سیستم های این بخش را در ادامه مشاهده خواهیم کرد.

واضح است که چون این سیستم ها همگی گسسته هستند، نباید انتظار داشته باشیم که همانند اتفاقی که برای X افتاد، خروجی این سیستم ها نیز دقیقا همان جواب را فقط با دقت و نمونه های بیشتر داشته باشند. برای مثال سیستم گسسته A همواره با عنصر (n)اُم و عنصر (n+1)اُم ورودی خودش برای تولید عنصر (n)اُم خروجی خودش سر و کار دارد. در حالی که مثلا اگر فرض کنیم عنصر (n)اُم ورودی نشان گر t=1 باشد، در بخش اول که فرکانس نمونه برداری ما ۱ بود، عنصر کناری آن متناظر t=2 بود ولی اینجا که فرکانس نمونه برداری ما برابر ۲ می باشد، عنصر کناری متناظر با t=1.5 خواهد بود و کاملا واضح است که پاسخ کلی متفاوت خواهد بود.

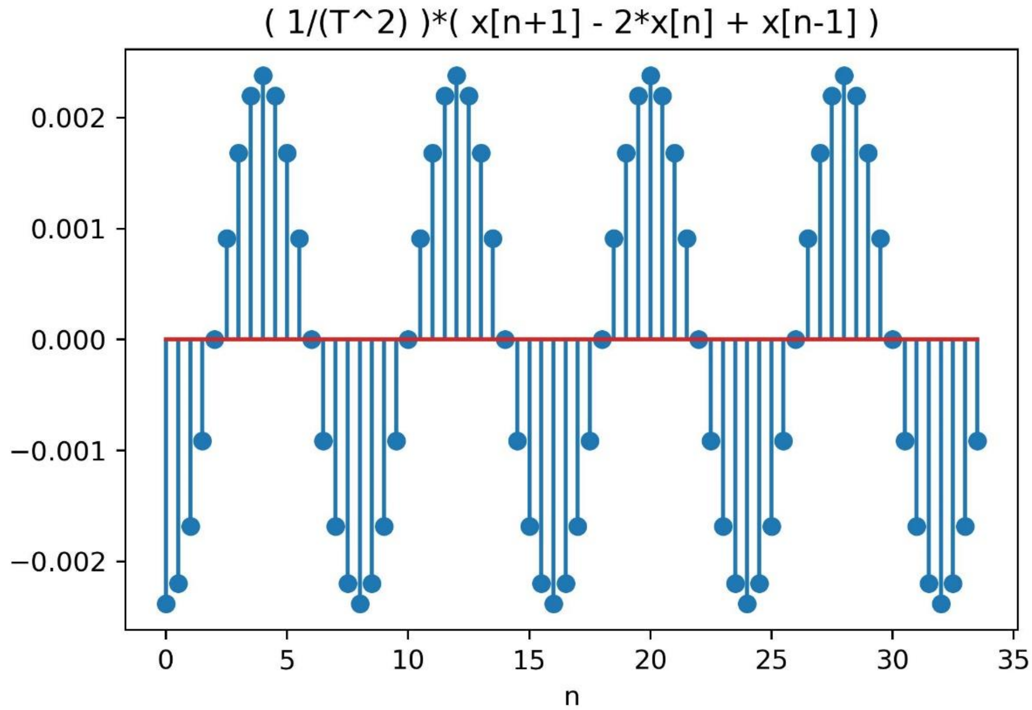
(البته ممکن است شکل کلی خروجی های متناظر با بخش قبل یکسان باشد و منحنی های نرم تر و دقیق تری را مشاهده کنیم ولی باید توجه کنیم که مقادیر عددی آنها طبق توضیحات قبل متفاوت هستند)



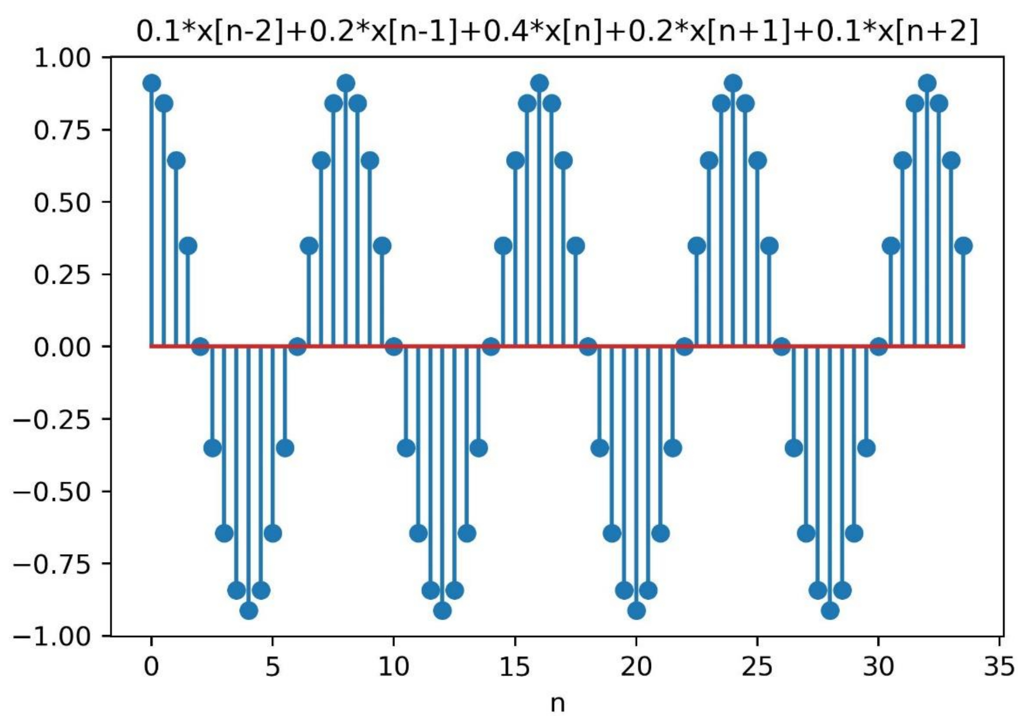
شکل A-2-1: خروجی سیستم LTI گسسته



شکل B-2-1: خروجی سیستم LTI گسسته



شکل 1-2-C: خروجی سیستم LTI گسسته



شکل 1-2-D: خروجی سیستم LTI گسسته

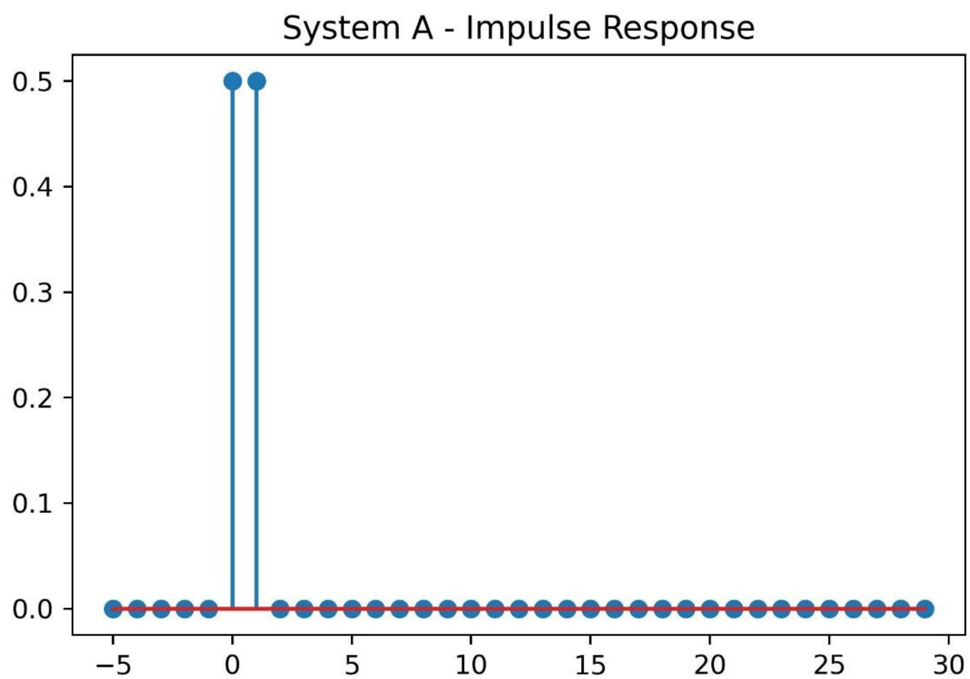
سوال ۲

در این سوال برای محاسبه پاسخ ضربه و پاسخ پله سیستم ها، به صورت functional یا تابعی عمل میکنیم. ابتدا تابع ضربه و تابع پله را به صورت یک function تعریف می کنیم. سپس سیستم ها را نیز به صورت یک function پیاده سازی میکنیم که یک تابع و مقدار یک نقطه دلخواه را گرفته و پاسخ سیستم را به ورودی، در آن نقطه برمی گردانند.

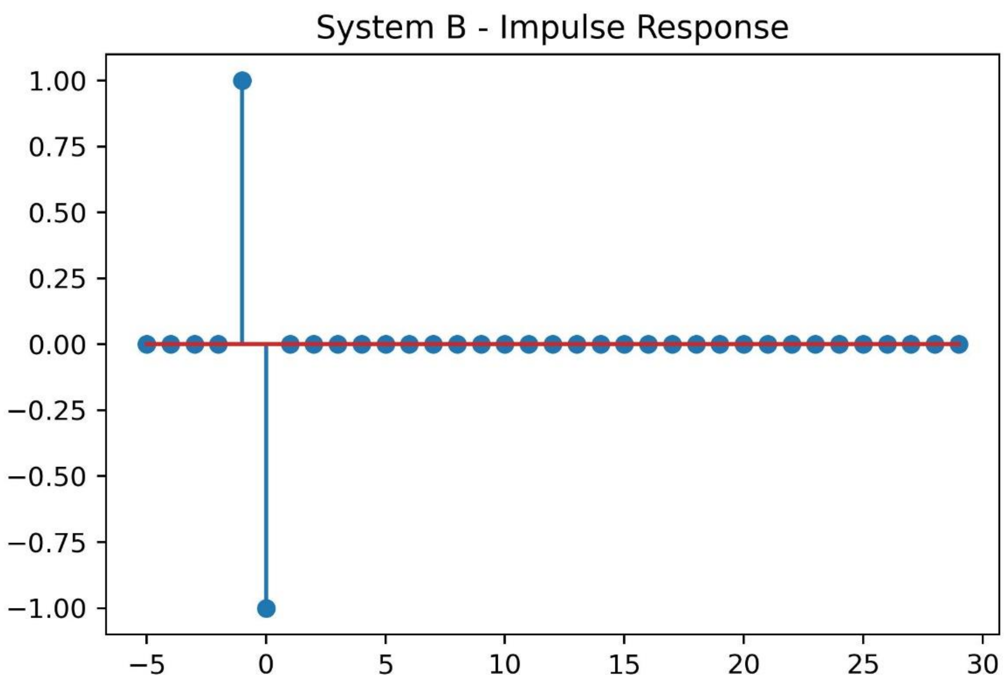
حالا به راحتی برای هر بازه دلخواهی می توانیم پاسخ ضربه و پله را به صورت دقیق محاسبه کنیم .

الف)

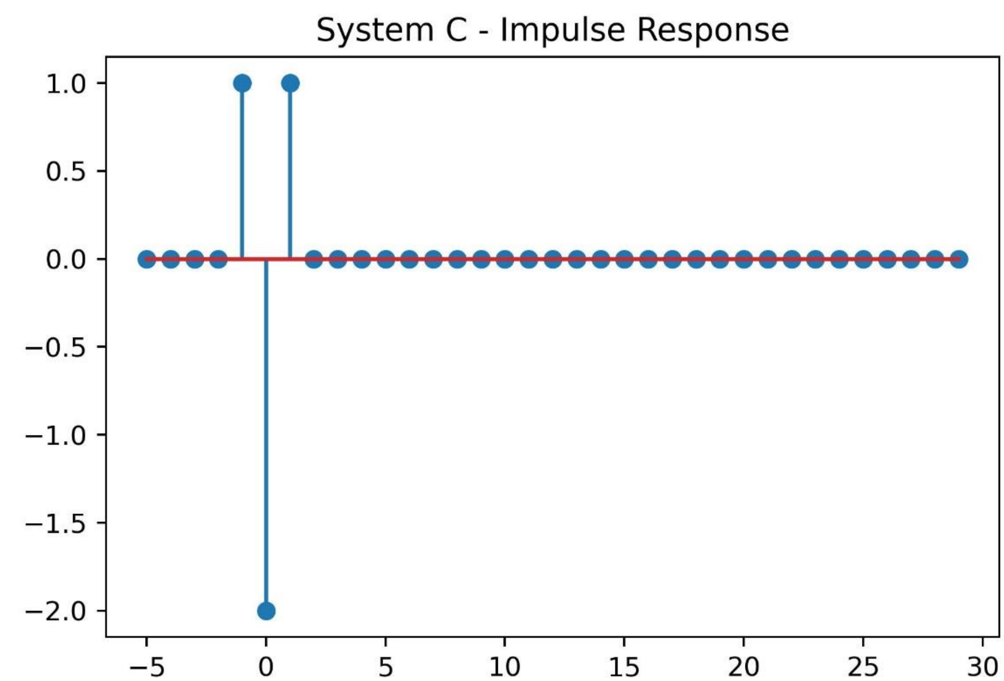
در ادامه پاسخ ضربه سیستم ها را مشاهده می کنیم.



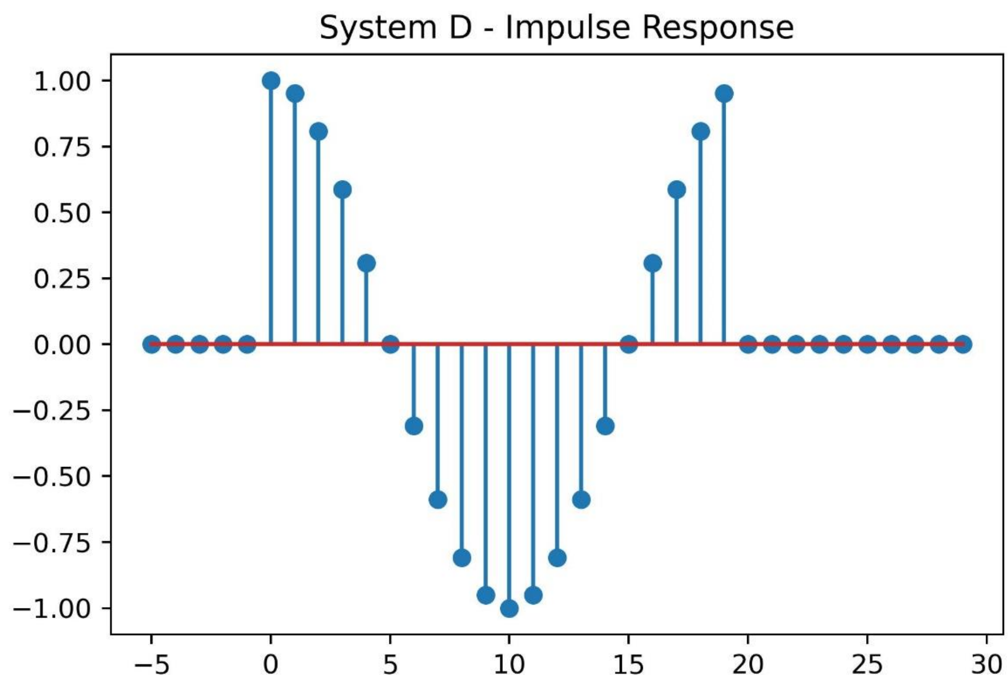
شکل A-1-2: پاسخ ضربه سیستم LTI گسسته



شکل 2-1-B: پاسخ ضربه سیستم LTI گسسته



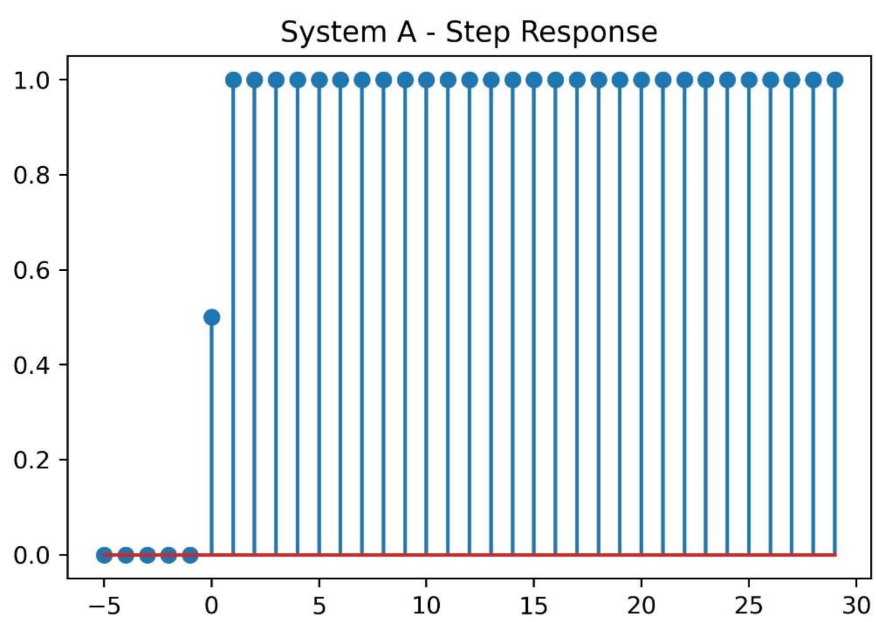
شکل 2-1-C: پاسخ ضربه سیستم LTI گسسته



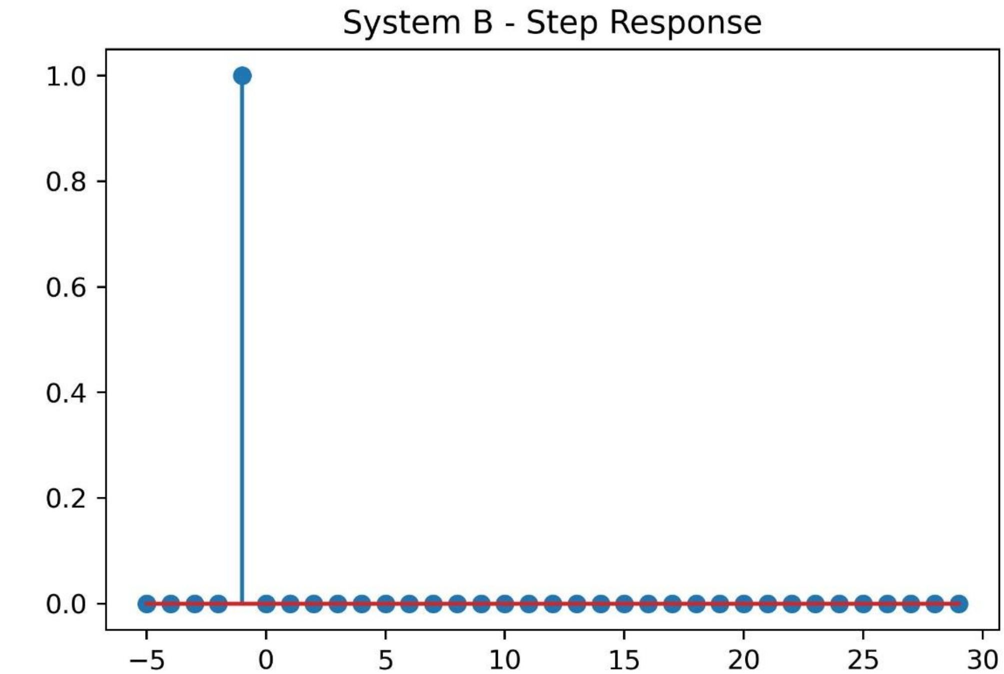
شکل D-1-2: پاسخ ضربه سیستم LTI گسسته

ب)

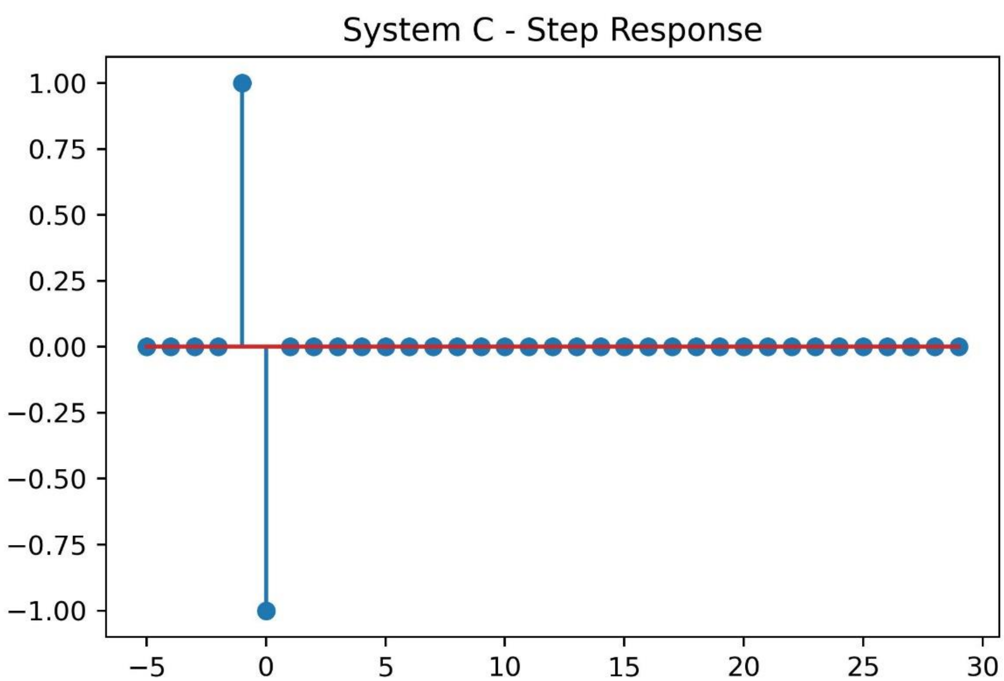
در ادامه پاسخ پله سیستم ها را مشاهده می کنیم.



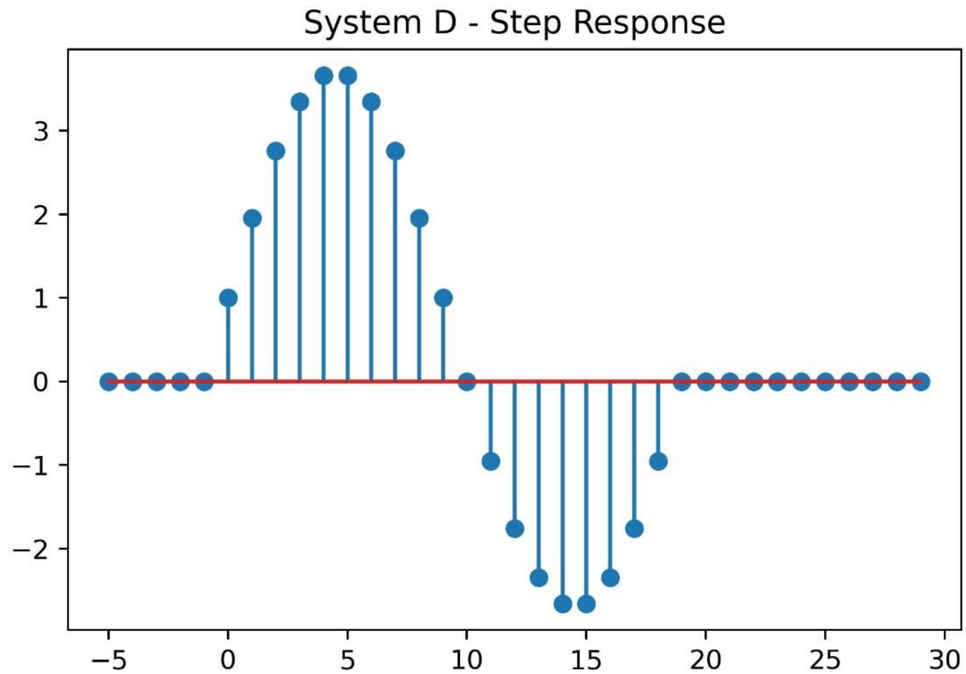
شکل A-2-2: پاسخ پله سیستم LTI گسسته



شکل 2-2-B: پاسخ پله سیستم LTI گسسته



شکل 2-2-C: پاسخ پله سیستم LTI گسسته



شکل D-2-2: پاسخ پله سیستم LTI گسسته

سوال ۳

الف)

در اینجا n را به ۵ بازه تقسیم می کنیم زیرا X[n] فقط در پنج نقطه مقدار غیر صفر دارد. سپس به صورت پارامتری پاسخ سیستم را برای این ورودی محاسبه می کنیم. محاسبات دستی را در ادامه مشاهده خواهیم کرد.

$$h[n] = \alpha^n u[n] \quad 0 < \alpha < 1$$

$$y[n] = \sum_{k=-\infty}^{+\infty} x[k]h[n-k]$$

$$y[n] = \sum_{k=-\infty}^{n} x[k]\alpha^{(n-k)}$$

$$x[n] = \begin{cases} 1 & 0 \leqslant n \leqslant 4 \\ 0 & o.w \end{cases}$$

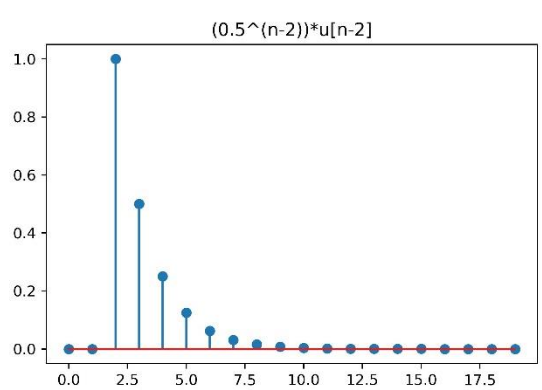
$$\Rightarrow y[n] = \alpha^n u[n] + \alpha^{n-1} u[n-1] + \alpha^{n-2} u[n-2] + \alpha^{n-3} u[n-3] + \alpha^{n-4} u[n-4]$$

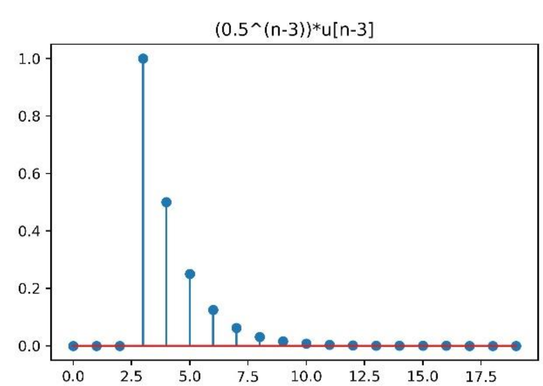
$$\alpha = \frac{1}{2}$$

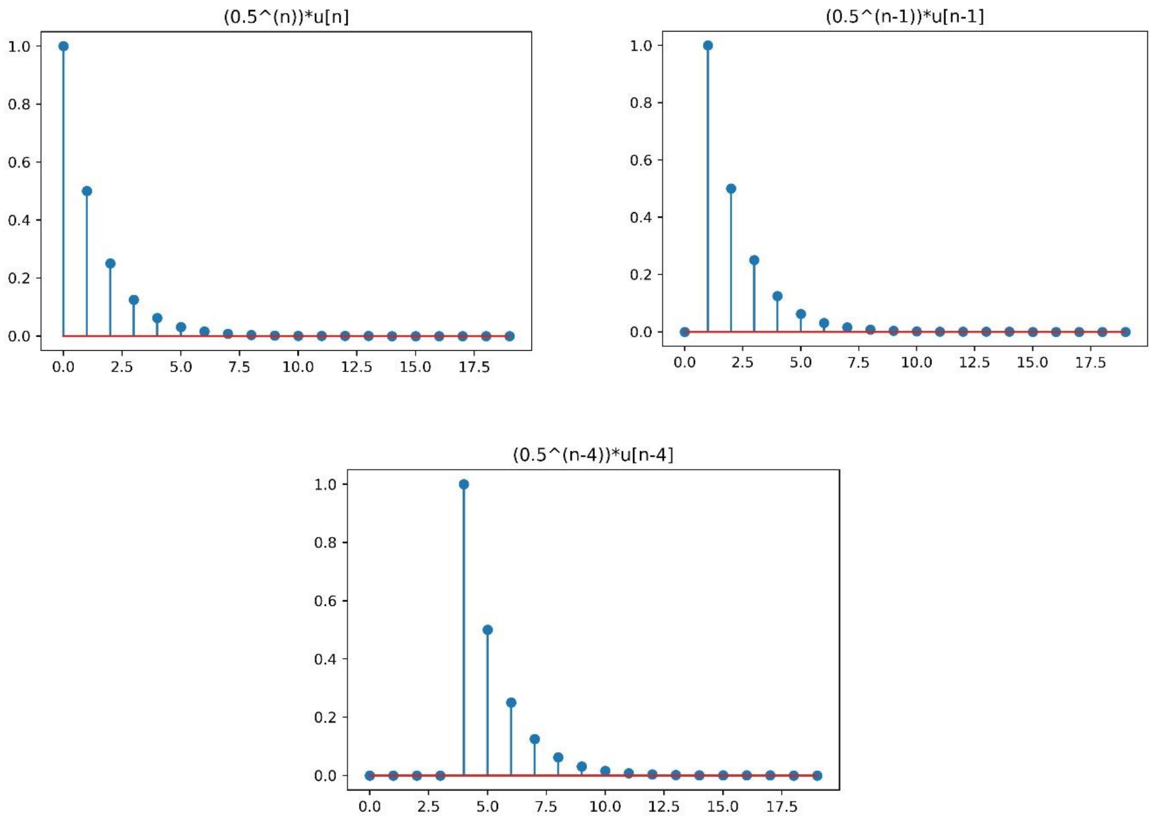
ب)

همانند محاسبات بالا، با تقسیم بندی n به پنج بازه مجزا، ابتدا پاسخ را برای هر بازه محاسبه می کنیم و چون سیستم LTI می باشد، می توانیم در آخر با جمع کردن این پاسخ ها، پاسخ کلی سیستم را به ورودی بیابیم.

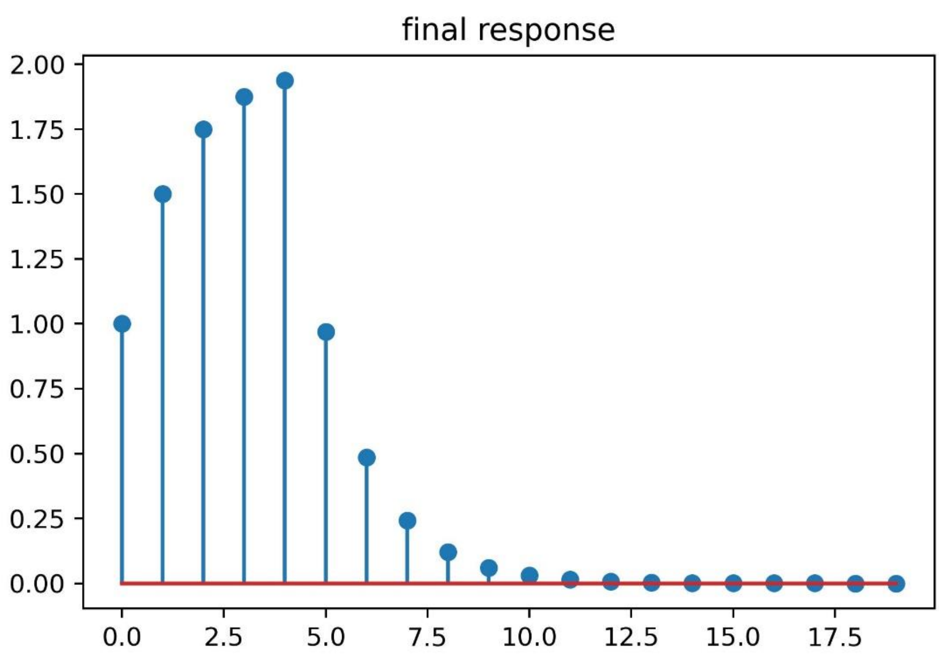
نمودار پاسخ برای هر بازه را شکل 1-3 مشاهده می کنیم. پاسخ نهایی سیستم به ورودی در شکل 2-3 نمایش داده شده است.







شکل 3-1: پاسخ سیستم برای هر بازه



شکل 3-2: پاسخ سیستم به ورودی

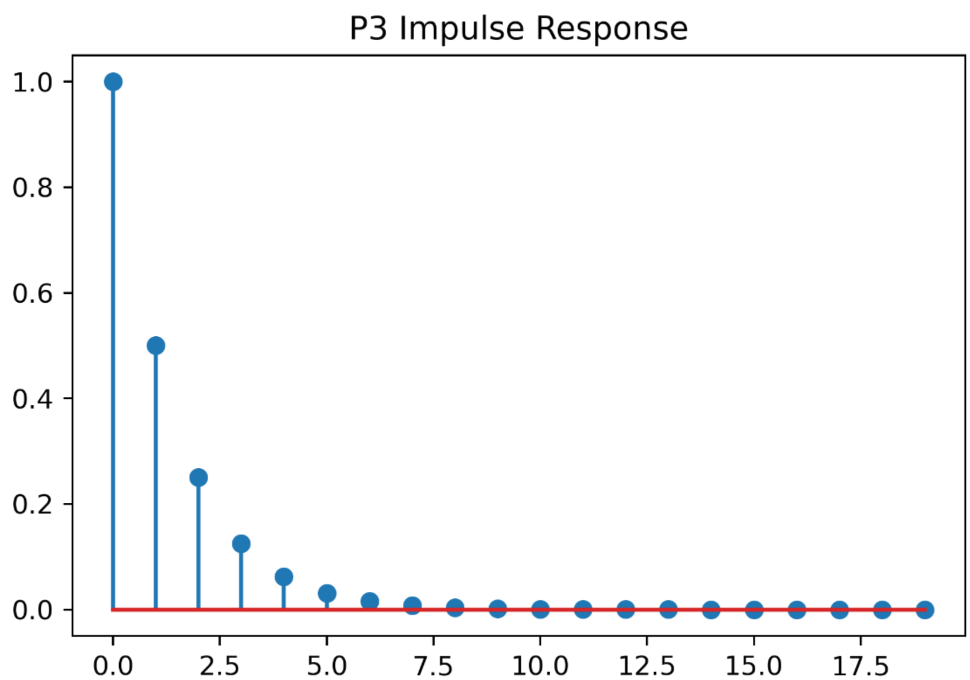
سوال ۴

گسسته زمان – Convolution

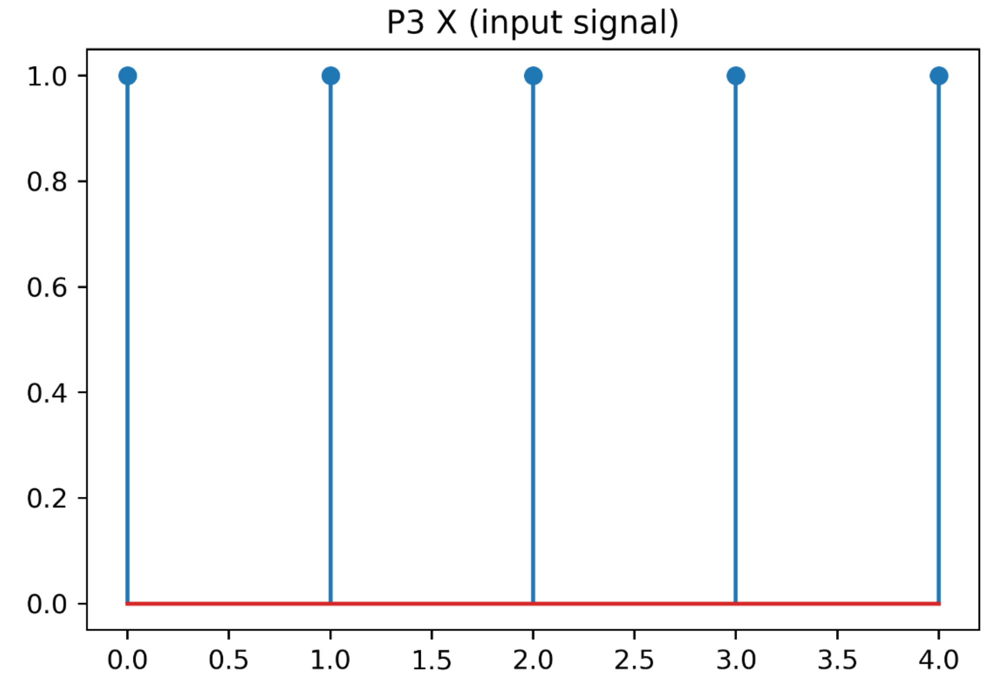
تابعی برای محاسبه convolution می نویسیم. (فایل های کد پایتون همگی ضمیمه شده اند)

الف)

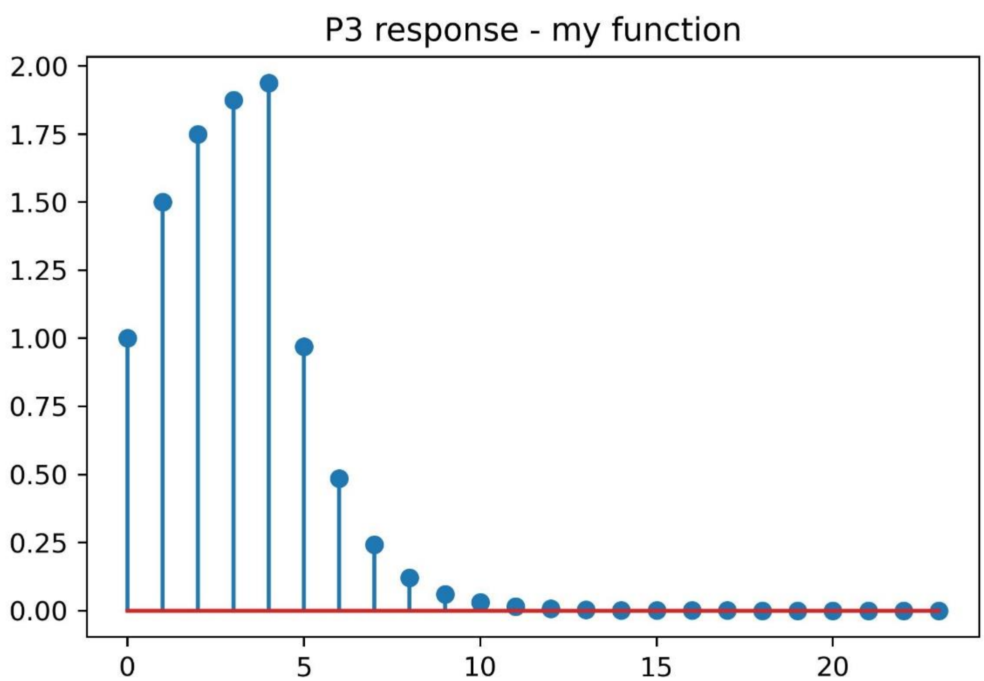
خواسته های سوال به ترتیب در ادامه نمایش داده شده اند.



شکل 1-1-4: پاسخ ضربه سیستم سوال ۳



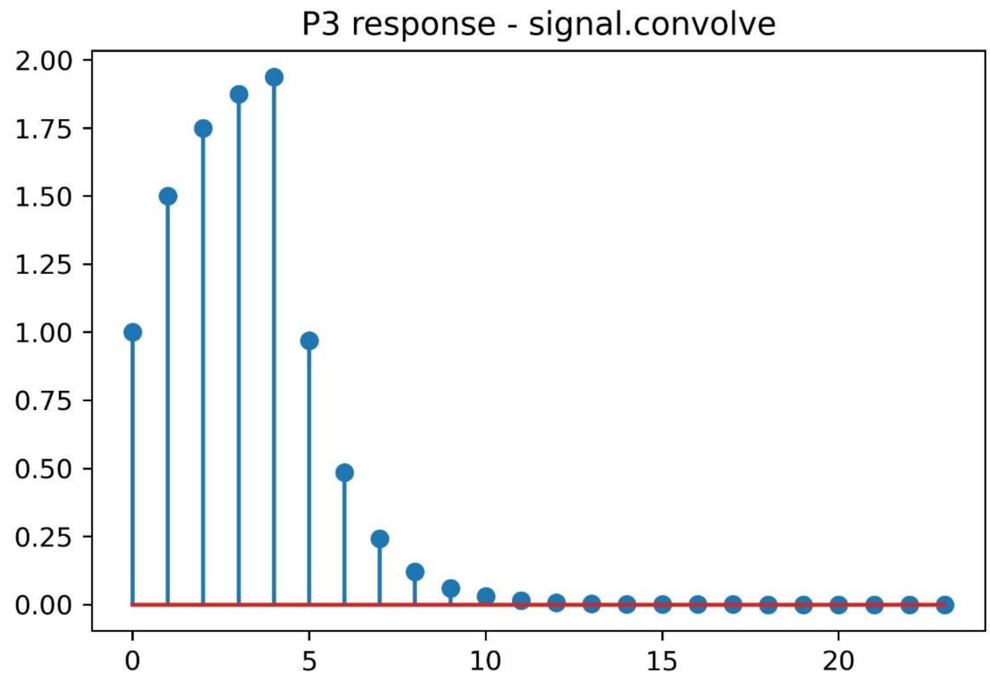
شکل 2-1-4: سیگنال ورودی در سوال ۳



شکل 3-1-4: پاسخ سیستم سوال ۳ به ورودی X با استفاده از تابع کانولوشن نوشته شده در این سوال

(ب

حالا با استفاده از تابع آماده signal.convolve در کتابخانه scipy در پایتون، سعی میکنیم پاسخ سیستم سوال ۳ را ورودی X آن سوال محاسبه کنیم. می بینیم که نتیجه همانند نتیجه تابعی بود که خودمان نوشتیم و هردو با نتیجه سوال ۳ همخوانی دارند.



شکل 4-1-4: پاسخ سیستم سوال ۳ به ورودی X با استفاده از تابع آماده signal.convolve

(ج

در حالت full به صورت کامل و تاجایی که امکان دارد، کانولوشن را محاسبه می کند و طول آرایه خروجی برابر مجموع طول دو آرایه ورودی می باشد. در صورتی که در حالت same، طول آرایه خروجی به اندازه مینیمم اندازه دو آرایه ورودی می باشد و به نسبت حالت full فقط نقاط مرکزی باز گردانده می شود.

با انتخاب بازه مناسب برای نمایش می توان خروجی حالت same را با خروجی حالت full نیز نمایش داد.

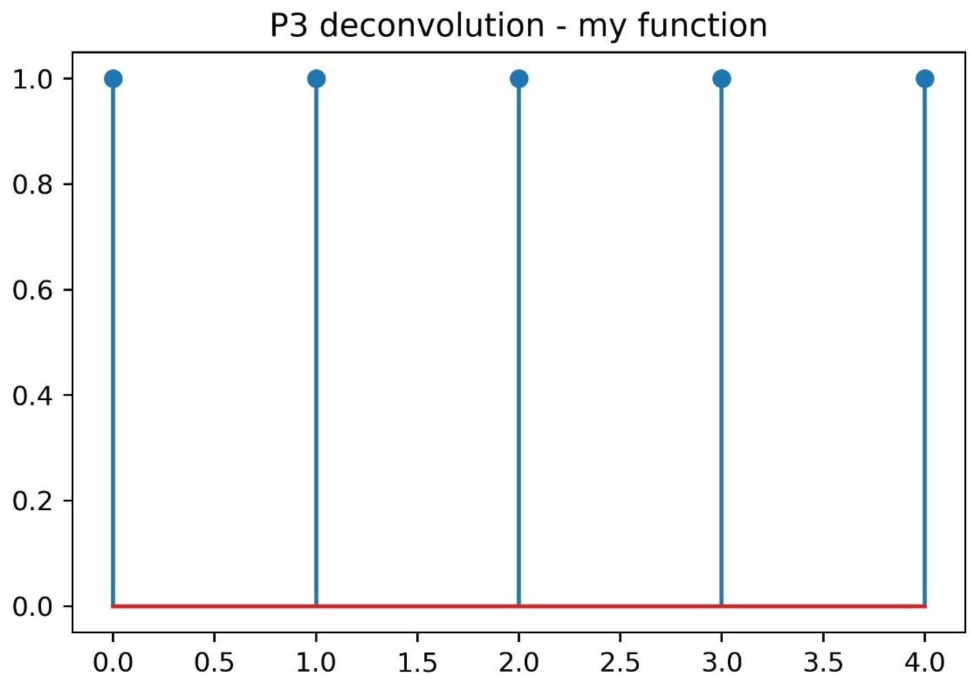
گسسته زمان – Deconvolution

الف)

تابع deconvolve را در پایتون پیاده سازی می کنیم که کد مربوط به آن نیز ضمیمه شده است.

با دادن خروجی سیستم سوال ۳ و همچنین پاسخ ضربه آن، ورودی را با این تابع محاسبه می کنیم.

نتیجه را در شکل 1-2-4 می توان مشاهده کرد.



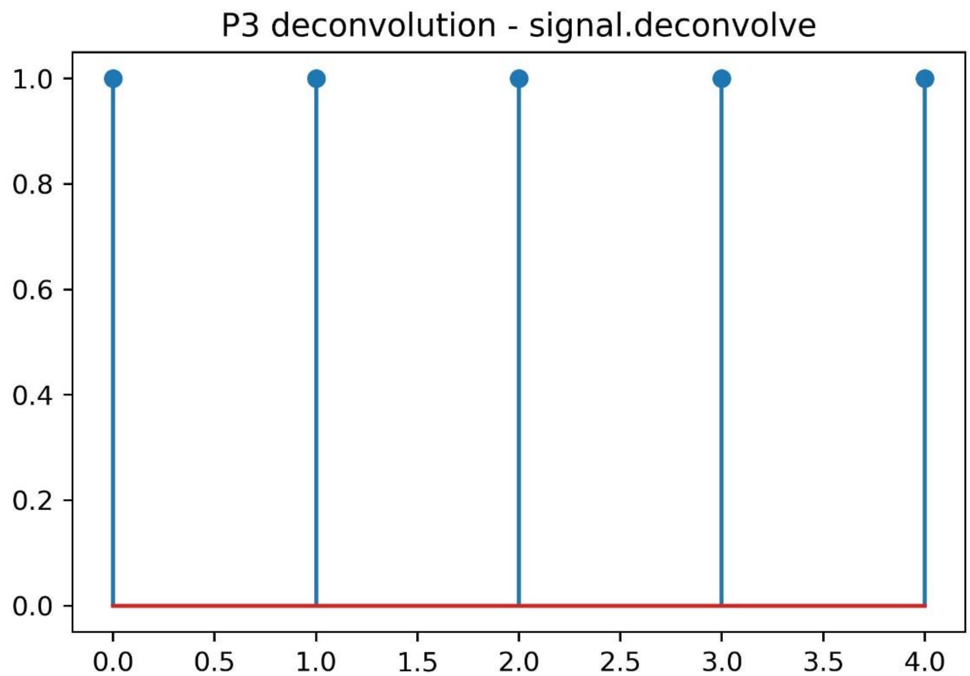
شکل 1-2-4: حاصل deconvolution با تابع نوشته شده در این سوال

ب)

خیر، به طور کلی عمل deconvolution پاسخ های یکتایی ندارد زیرا عمل convolution در حالت کلی برگشت پذیر نیست. ولی اینجا چون دامنه ما محدود است، با داشتن حاصل convolution در حالت full، به پاسخ های یکتایی دست پیدا کرد و تابع اولیه را به میزان خوبی ساخت.

ج)

این بار از تابع آماده signal.deconvolve استفاده می کنیم که نتیجه آن در شکل 2-2-4 قابل مشاهده است. می بینیم که حاصل این تابع با چیزی که با تابع نوشته شده خودمان به دست آوردیم یکسان است.



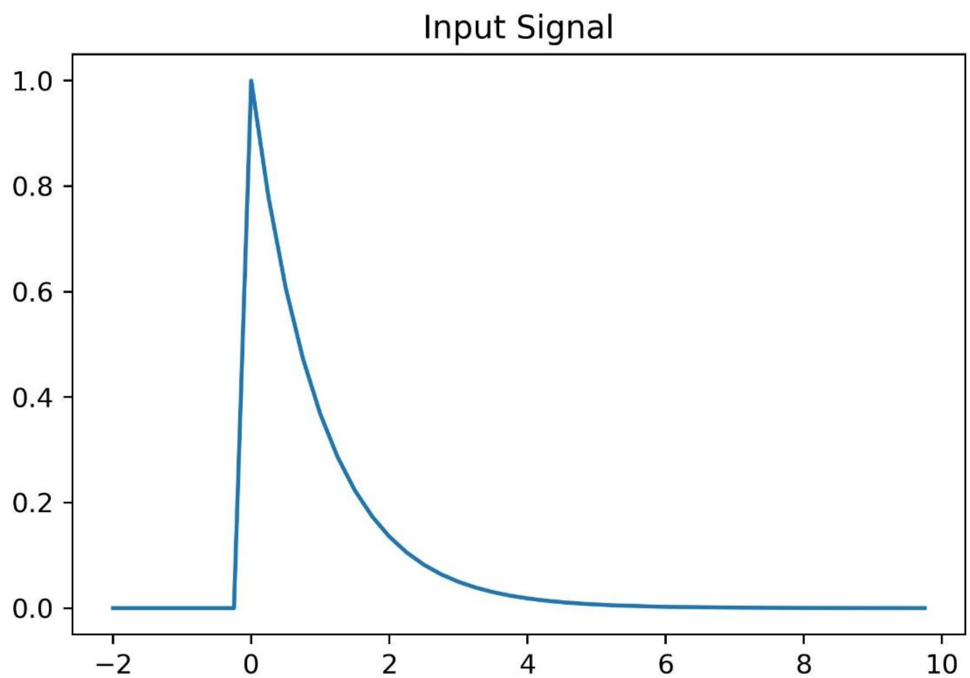
شکل 2-2-4: حاصل تابع آماده signal.deconvolve

پیوسته زمان – ۱

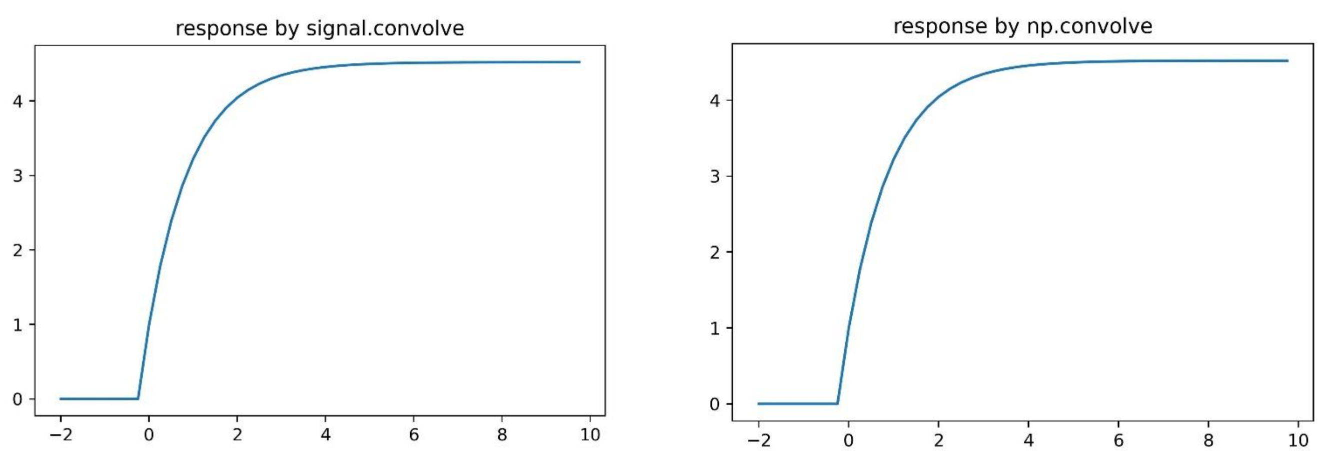
برای راحتی مقدار دلخواه آلفا را ۱ در نظر می گیریم.

الف)

با هر دو تابع، کانولوشن سیگنال ورودی و تابع ضربه را محاسبه می کنیم. ملاحظه می کنیم که خروجی هر دو تابع یکسان است.



شکل 1-3-4: سیگنال ورودی X



شکل 2-3-4: مقایسه تابع convolve در دو کتابخانه numpy و scipy.signal

ب)

حل دستی و محاسبه scaling-factor

$$x(t) = e^{-t}\,u(t)$$

$$h(t) = u(t)$$

$$y(t) = \int_{-\infty}^{+\infty} x(\tau)\,h(t-\tau)\,d\tau = \int_0^t e^{-\tau}\,d\tau = \left[-e^{-\tau}\right]_0^t$$

$$y(t) = (1 - e^{-t})\,u(t) \longrightarrow \lim_{t\to\infty} y(t) = 1$$

در حالت گسسته:

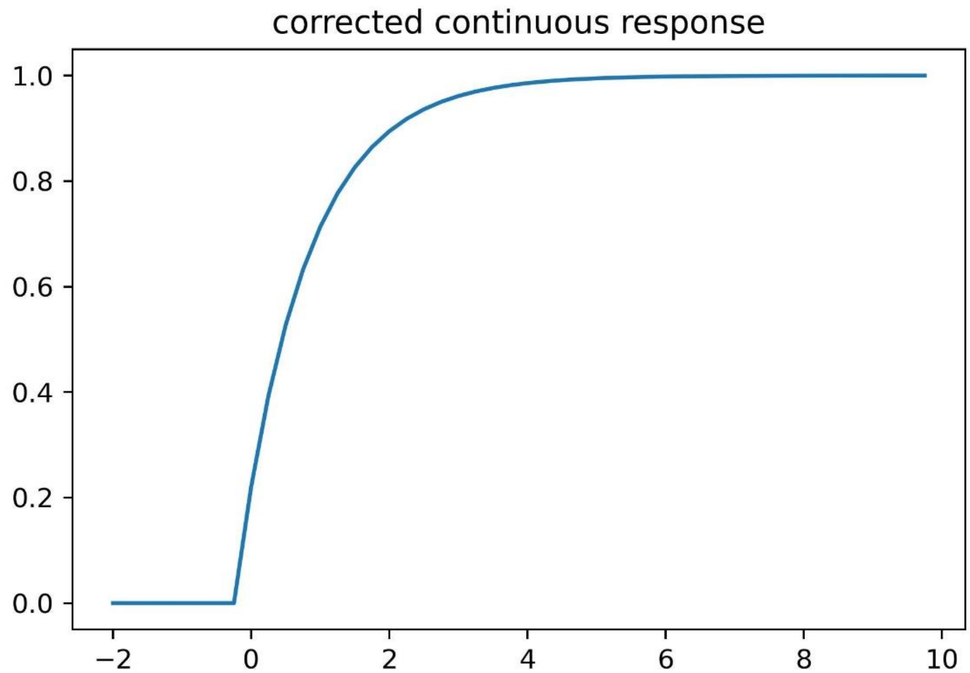
$$e^{0} + e^{-\frac{1}{4}} + e^{-\frac{1}{8}} + \cdots$$

$$\text{Sampling-step} = \frac{1}{4} \Rightarrow \text{Scaling-factor} = \frac{1}{1-e^{-\frac{1}{4}}} \simeq 4.521$$

ج)

سیگنال به دست آمده در کانولوشن های بخش الف، با جواب تحلیلی یکسان نیستند، گرچه شکل کلی آنها مانند هم است. این تفاوت به این دلیل است که ما اینجا به جای انتگرال گیری داریم این عمل را جمع های متوالی مدل می کنیم. در نتیجه جواب ما بیشتر از مقدار واقعی انتگرال است.

اینجا تابع ورودی به صورت نمایی است و پاسخ ضربه سیستم به صورت تابع پله است، در نتیجه به راحتی می توان این scaling را با محاسبه scaling_factor رفع کرد. پاسخ صحیح این سیستم به ورودی را پس از اصلاح در شکل 3-3-4 مشاهده می کنیم.



شکل 3-3-4: پاسخ اصلاح شده به فرم پیوسته

پیوسته زمان – 2

ابتدا در بخش الف، پاسخ سیستم را طبق سیگنال های داده شده به صورت تحلیلی محاسبه می کنیم.

الف)

$$h(t) = u(t) \qquad x(t) = \begin{cases} 0 & t \leqslant -2 \\ t+2 & -2 < t \leqslant -1 \\ 1 & -1 < t \leqslant 1 \\ 2-t & 1 < t \leqslant 2 \\ 0 & 2 < t \end{cases}$$

$$y(t) = \int_{-\infty}^{+\infty} x(\tau)\, h(t-\tau)\, d\tau = \int_{-\infty}^{t} x(\tau)\, d\tau$$

$t \leqslant -2$: $y(t) = 0$

$-2 < t \leqslant -1$: $y(t) = \int_{-2}^{t} (\tau+2)\, d\tau = \left[\frac{\tau^2}{2} + 2\tau\right]_{-2}^{t}$

$$= \frac{t^2}{2} + 2t + 2$$

$-1 < t \leqslant 1$: $y(t) = \frac{1}{2} + \int_{-1}^{t} d\tau = \frac{1}{2} + \left[\tau\right]_{-1}^{t}$

$$= t + \frac{3}{2}$$

$1 < t \leqslant 2$: $y(t) = \frac{5}{2} + \int_{1}^{t} (2-\tau)\, d\tau = \frac{5}{2} + \left[2\tau - \frac{\tau^2}{2}\right]_{1}^{t}$

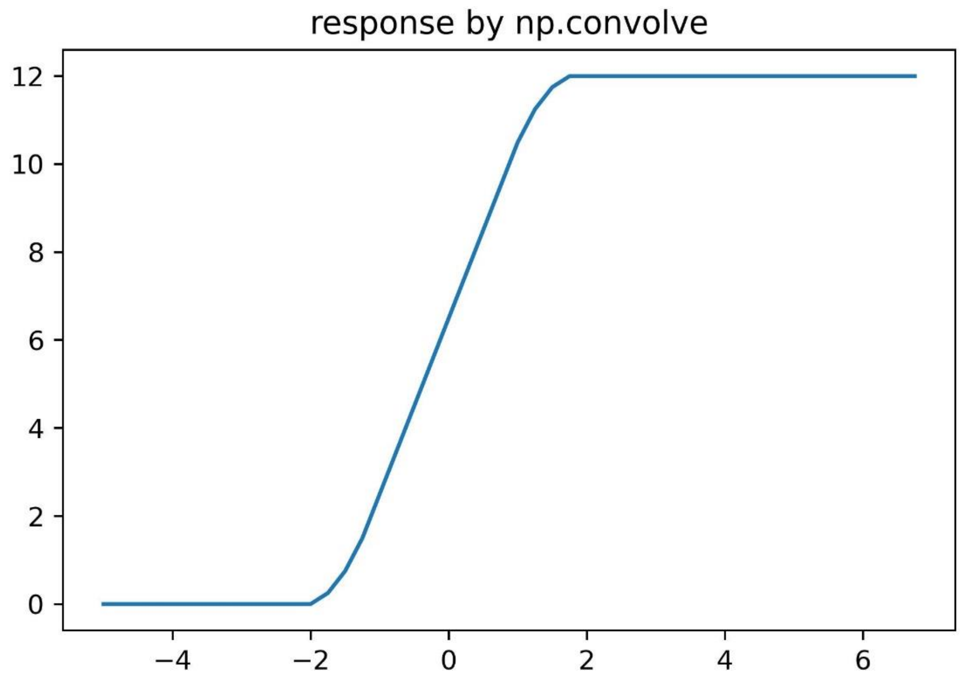
$$= \frac{-t^2}{2} + 2t + 1$$

$2 < t$: $y(t) = 3$

for $2 < t$: scaling-factor $= \frac{12}{3} = 4$

ب)

حالا خروجی سیستم را به کمک تابع np.convolve محاسبه می کنیم. شکل کلی پاسخ با جواب تحلیلی همخوانی دارد (قسمتی سهمی و قسمتی خطی) ولی همانند توضیحات بخش قبل در اینجا هم scaling رخ داده است که بسته به بازه t مقدار آن متفاوت است. البته اصلاح خروجی کانولوشن برای به دست آوردن پاسخ پیوسته صحیح در صورت سوال خواسته نشده است.

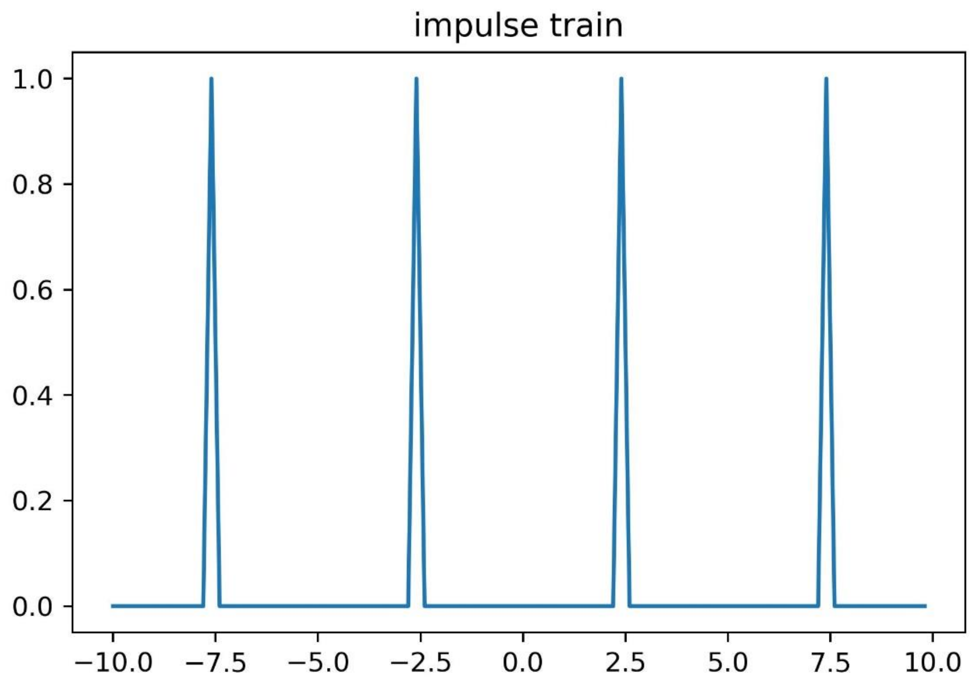


شکل 1-4-4: پاسخ اصلاح نشده سیستم با استفاده از np.convolve

پیوسته زمان – 3

الف)

قطار ضربه را پیاده سازی کرده و در شکل 1-5-4 نمودار آن را برای T=5 مشاهده می کنیم.



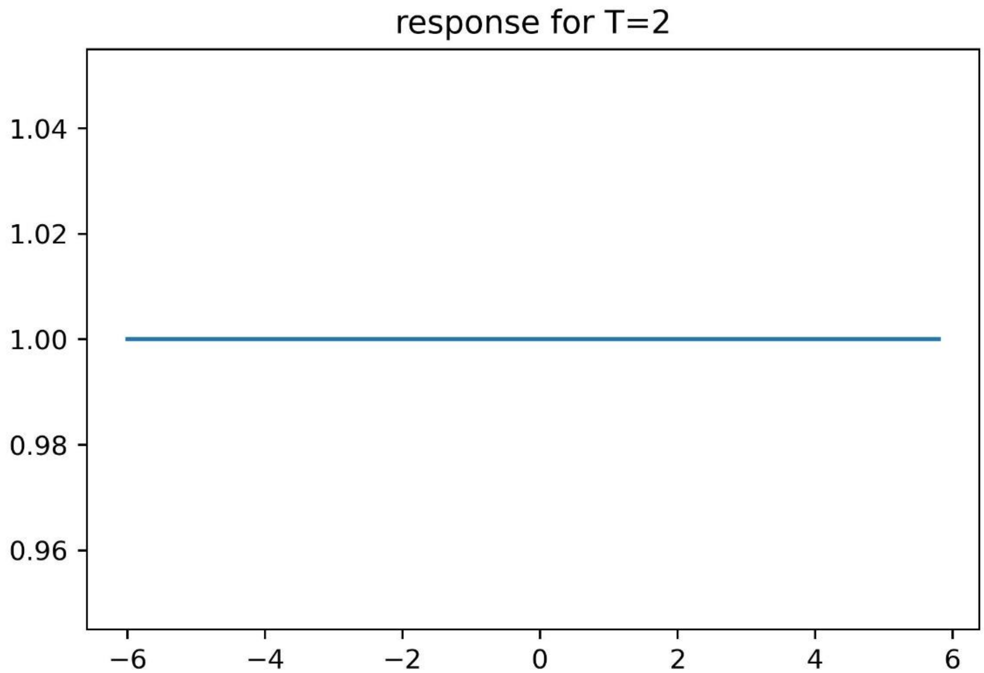
شکل 1-5-4: قطار ضربه برای T=5

ب)

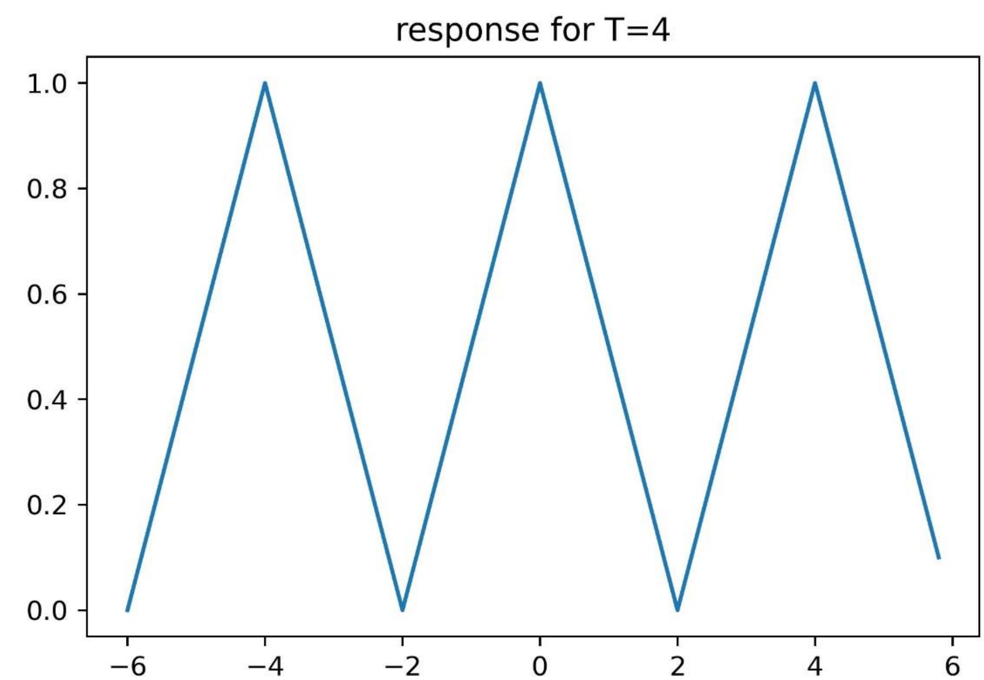
مقدار مرزی T که از روی خروجی بتوان ورودی را بازیابی کرد، برابر ۴ می باشد. زیرا اگر T کمتر از ۴ باشد، در بازه غیر صفر سیگنال ورودی بیش از یک تابع ضربه در هنگام انتگرال گیری وجود خواهد داشت و این باعث می شود که به کل پاسخ یک مقدار DC افزوده شود. برای مثال اگر مقدار T برابر ۲ باشد، سیگنال خروجی یک DC خالص خواهد بود که پیدا کردن ورودی از آن در حالت دامنه نامحدود برای قطار ضربه، غیر ممکن خواهد بود.

ج)

طبق توضیحات بالا، خروجی سیستم را برای چند مقدار مختلف T در ادامه مشاهده می کنیم.

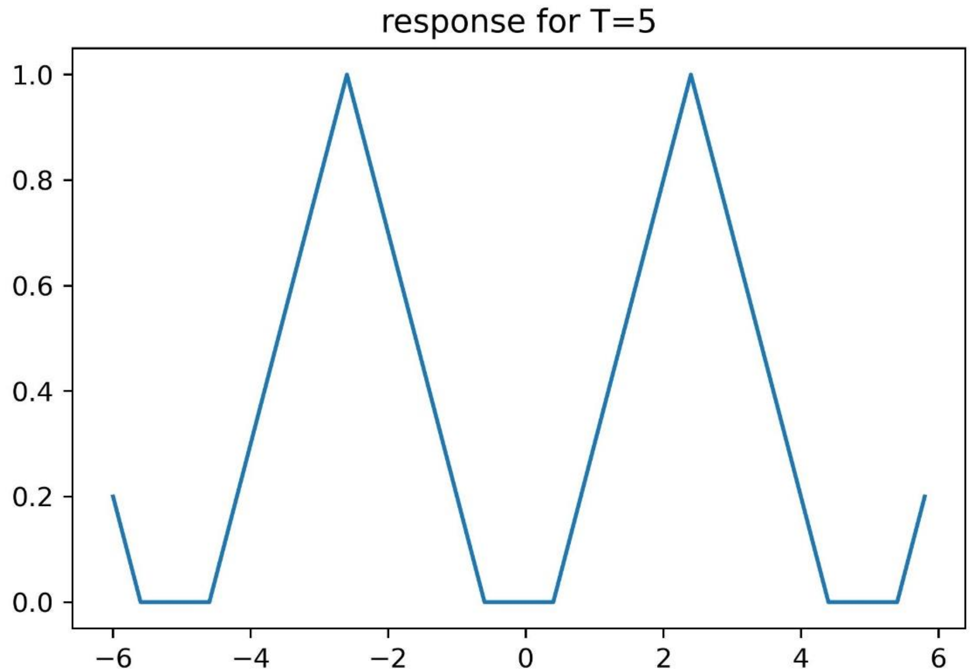


شکل 2-5-4: خروجی سیستم برای T=2



شکل 3-5-4: خروجی سیستم برای T=4

مشاهده می کنیم که با افزایش T و رسیدن به مرز ۴، خروجی شبیه ورودی می شود و به راحتی می توان ورودی را بازیابی کرد.



شکل 4-5-4: خروجی سیستم برای T=5