

به نام خدا

نام: امیرمسعود

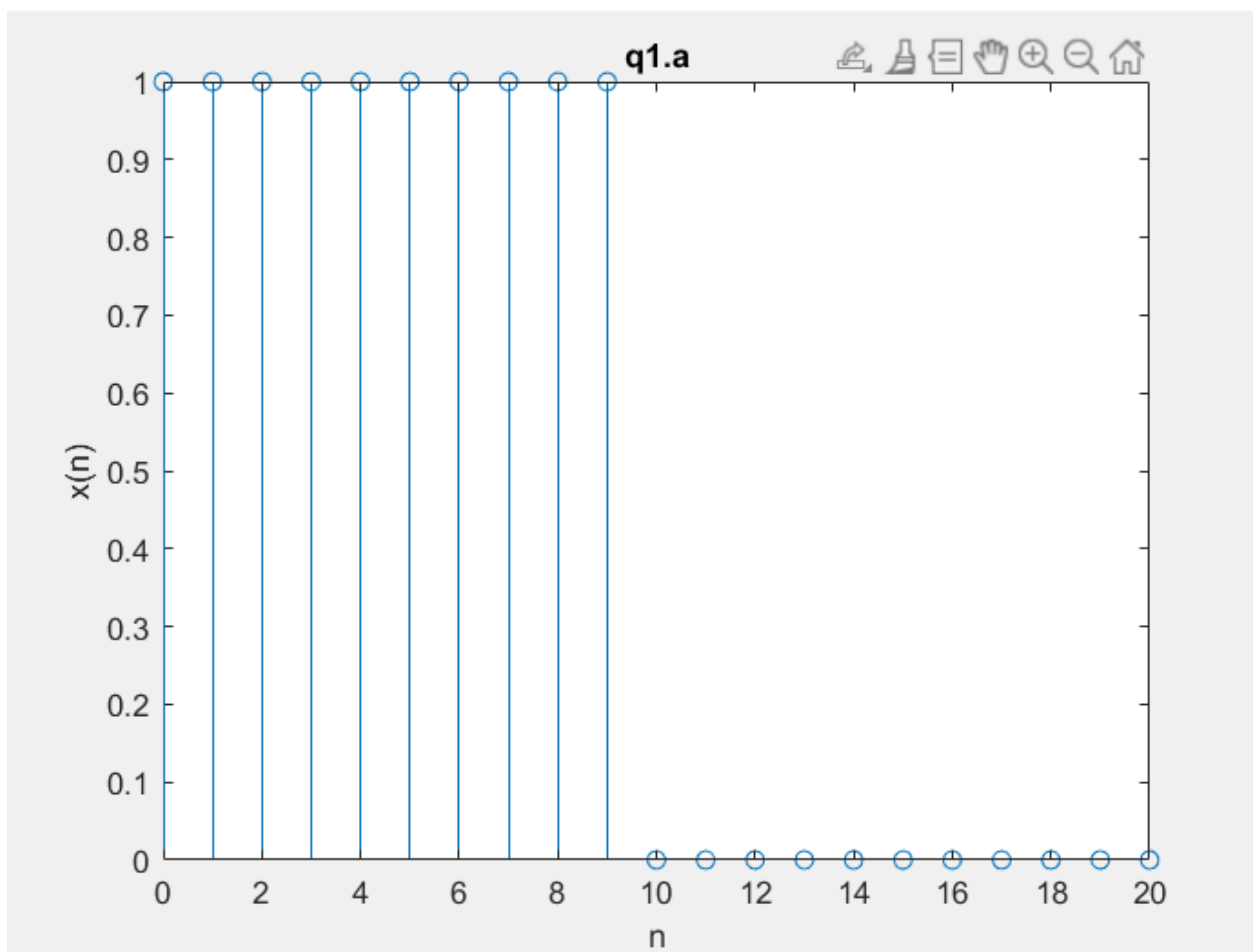
نام خانوادگی: شاکر

شماره دانشجویی: 97243081

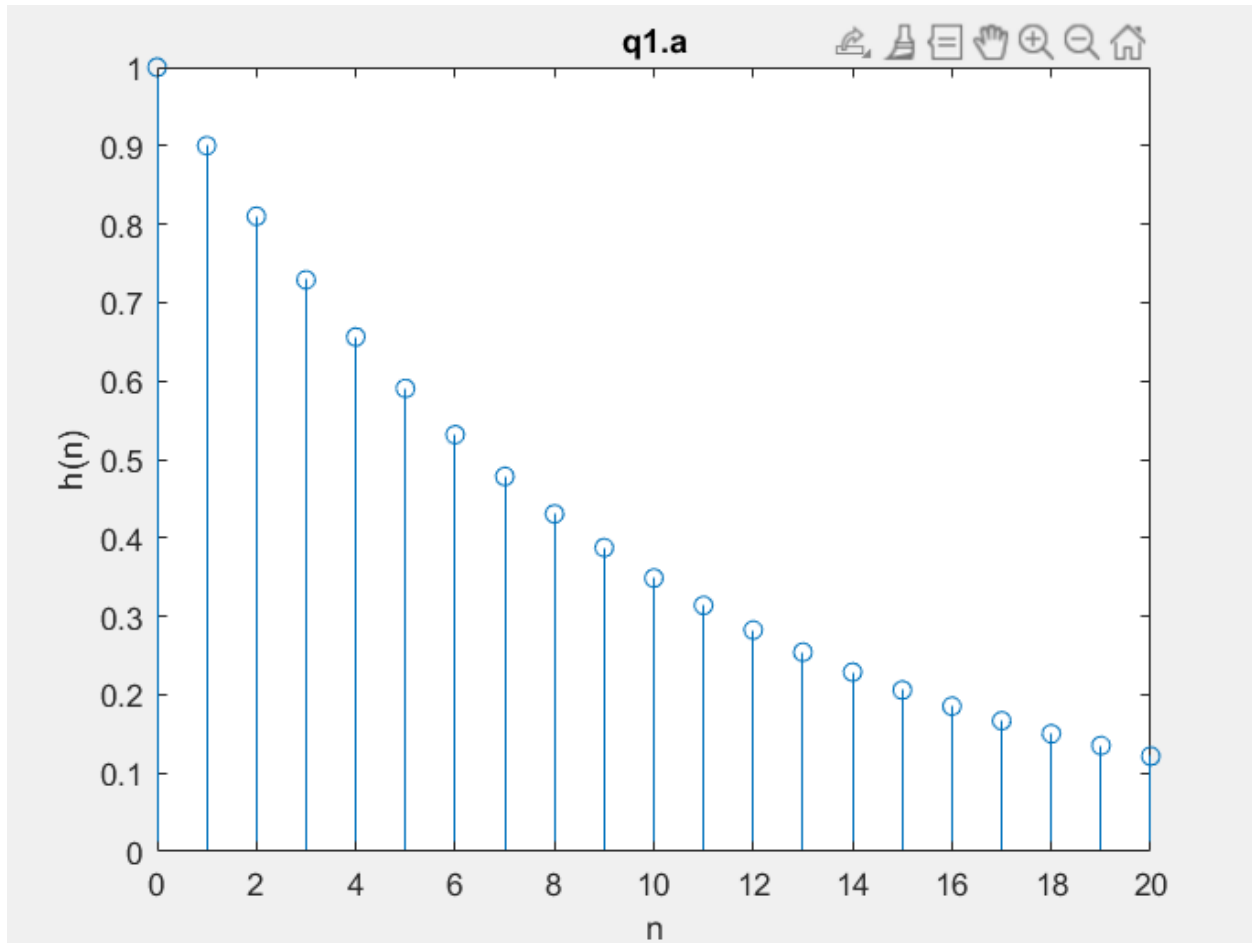
تمرین اول

(1)

الف) سیگنال  $x(n)$  که مطابق فرمول و شکل از 0 تا 9 مقدار 1 دارد و در بقیه  $n$  ها 0 است:

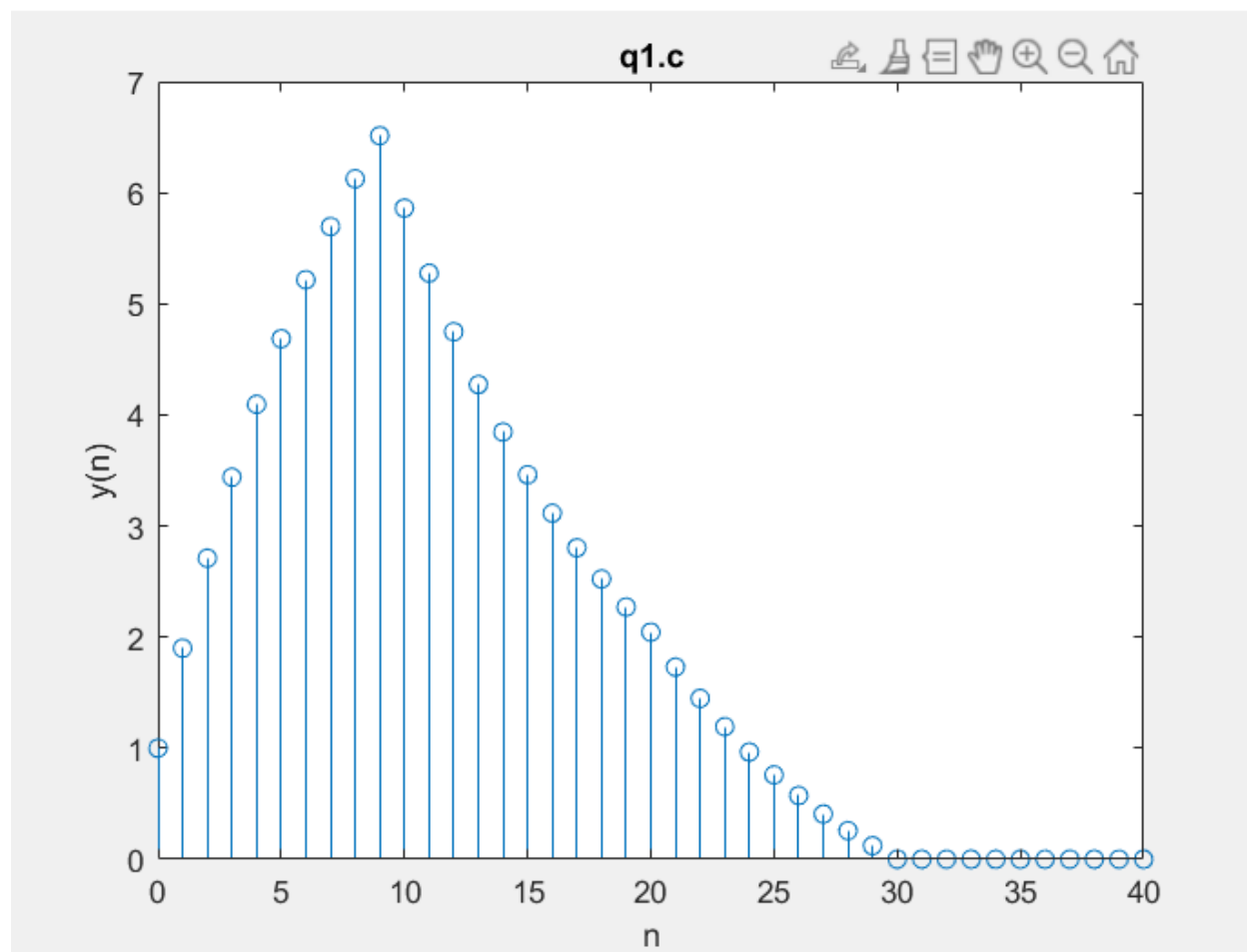


سیگنال  $h(n)$  که از 0 به بعد مقدار آن از 1 تا 0.12 کاهش میابد:



ب) خروجی  $y(n)$  برابر حاصل کانوولوشن سیگنال های  $x$  ,  $h$  است که میتوان به صورت  $\text{conv}(x, h)$  آن را نوشت.

ج) حاصل کانولوشن  $x, h$  در شکل زیر مشخص است:



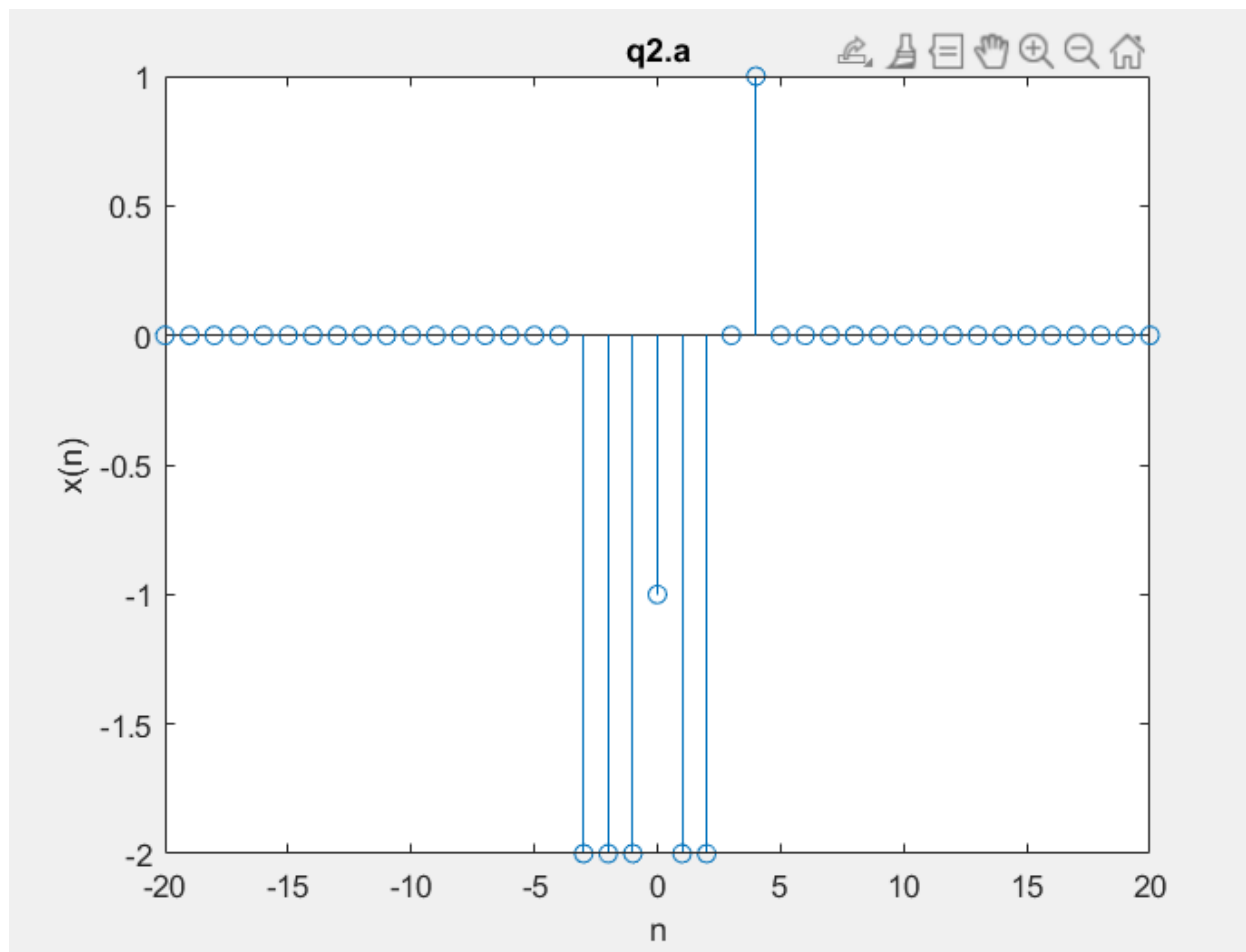
(2)

سیگنال  $x(n)$  با استفاده از توابع `Impseq`, `stepseq` پیاده سازی شده است.

همانطور که در شکل زیر مشخص است، سیگنال در نقاط  $2, 1, -1, -2, -3$  مقدار  $-2$  دارد و در نقطه  $0$  مقدار  $1$  دارد.

همچنین مقدار آن در نقطه  $4$  برابر  $1$  است و در باقی نقاط مقدار سیگنال  $0$  است.

با محاسبه دستی این سیگنال نیز به همین شکل می‌رسیم.



برای محاسبه سیگنال  $y(n)$  باید ابتدا  $x(2n+1)$  را به دست بیاوریم.

برای این کار، ابتدا سیگنال  $x(n)$  را به اندازه 1 واحد به چپ شیفت میدهیم و سپس اسکیل به اندازه 2 را انجام میدهیم.

در کد ابتدا سیگنال جدیدی به نام  $x\_shift\_by\_1$  تعریف شده که شیفت داده شده سیگنال  $x(n)$  است.

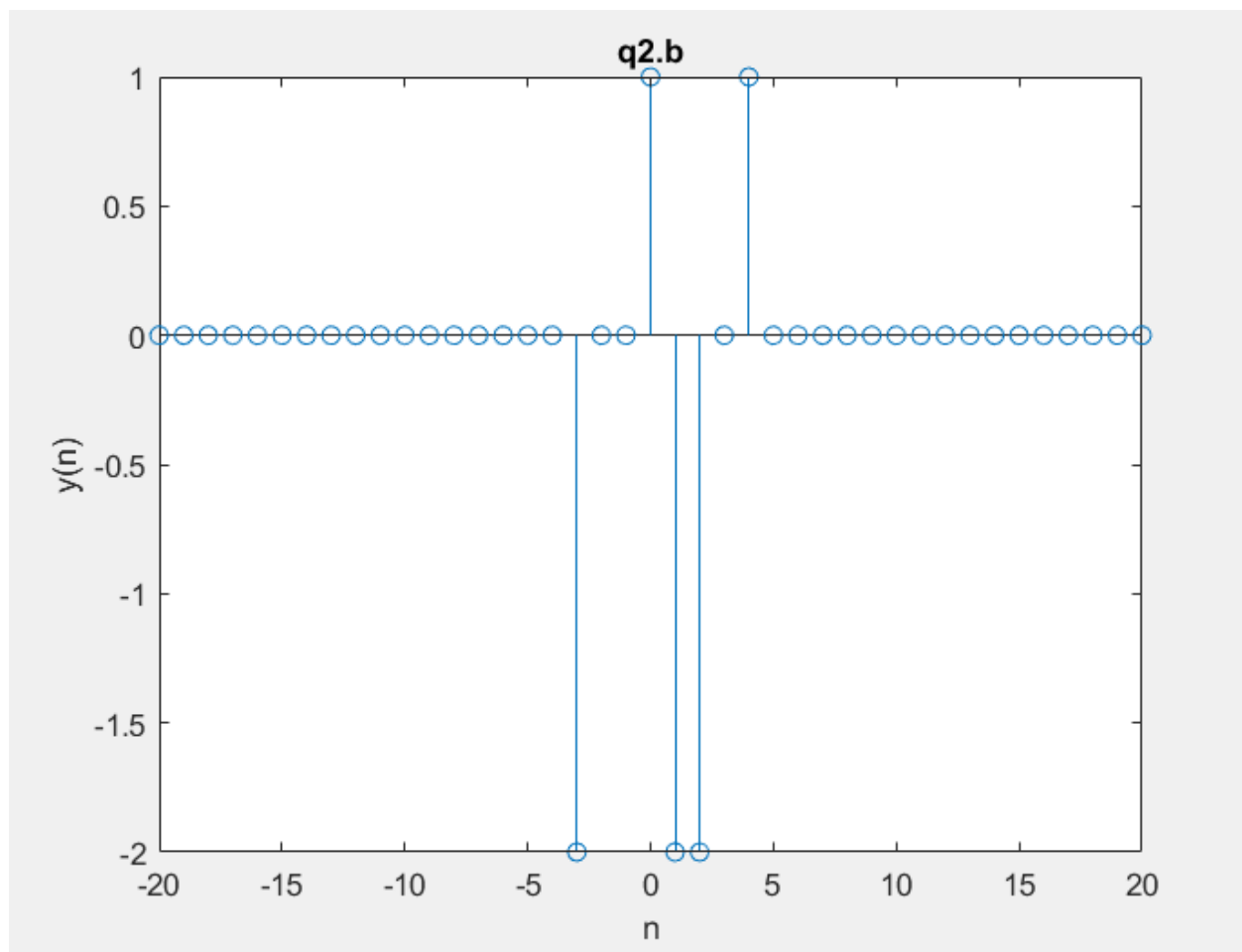
سپس با استفاده از تابع  $sigscale$  نوشته شده، سیگنال شیفت داده شده را به اندازه 2 واحد اسکیل میکنیم.

تابع  $sigscale$ ، در ورودی خود سیگنال و بازه آن و مقدار اسکیل (ضریب) را دریافت میکند.

سپس بازه و مقدار سیگنال را متناسب با مقدار اسکیل، تغییر میدهد.

پس از محاسبه  $x(2n+1)$ ، با استفاده از تابع  $sigadd$ ، سیگنال های  $x(n)$ ،  $-x(2n+1)$  را با هم جمع میکنیم.

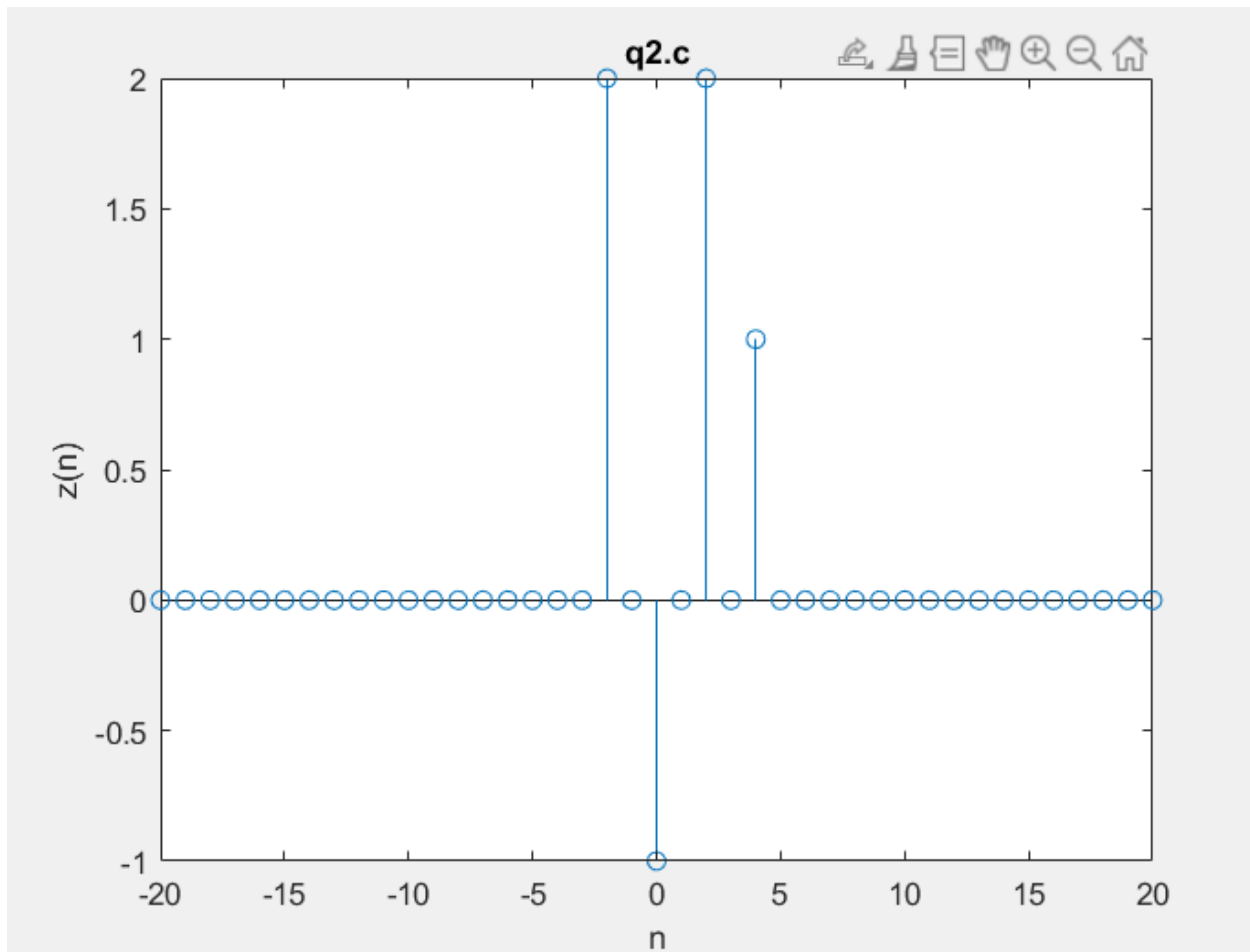
سیگنال خروجی را در شکل زیر میبینیم:



برای محاسبه سیگنال  $z$ ، تابع  $e^{j\pi n}$  را در سیگنال  $x(n)$  به صورت نقطه ای ضرب میکنیم.

تابع  $e^{j\pi n}$  به فرم رابطه اوایلر نوشته شده است. با توجه به اینکه  $\sin(j\pi n)$  همواره برابر صفر است، میتوان آن را در نظر نگرفت.

در آخر برای رسم سیگنال، همانطور که خواسته شده بود، قسمت حقیقی سیگنال رسم شده است (هر چند قسمت موهومی سیگنال برابر صفر است):



(3)

الف) تابع `myconv` در ورودی خود دو سیگنال `func1, func2` را دریافت میکند.

طول آنها را محاسبه میکند و مقادیر `x, h` را ابتدا برابر صفر قرار میدهد.

سپس با استفاده از دو حلقه `for`، ابتدا مقادیر عناصر بردار `y` (خروجی) را برابر صفر قرار میدهد و در حلقه `for` بعدی، عملیات ضرب و جمع لازم برای محاسبه کانولوشن را انجام میدهد.

ب) سیگنال  $x_1$  با توجه به فرمول آن پیاده سازی شده است.

تنها نکته دارای اهمیت آن عملیات ضرب و توان و تقسیم نقطه ای است و اینکه از تابع `ones` به منظور تطبیق سائز صورت و مخرج در عبارت  $1/2^{(n-1)}$  استفاده شده است.

برای محاسبه سیگنال  $x_2$ ، از دو حلقه `for` تو در تو استفاده شده است.

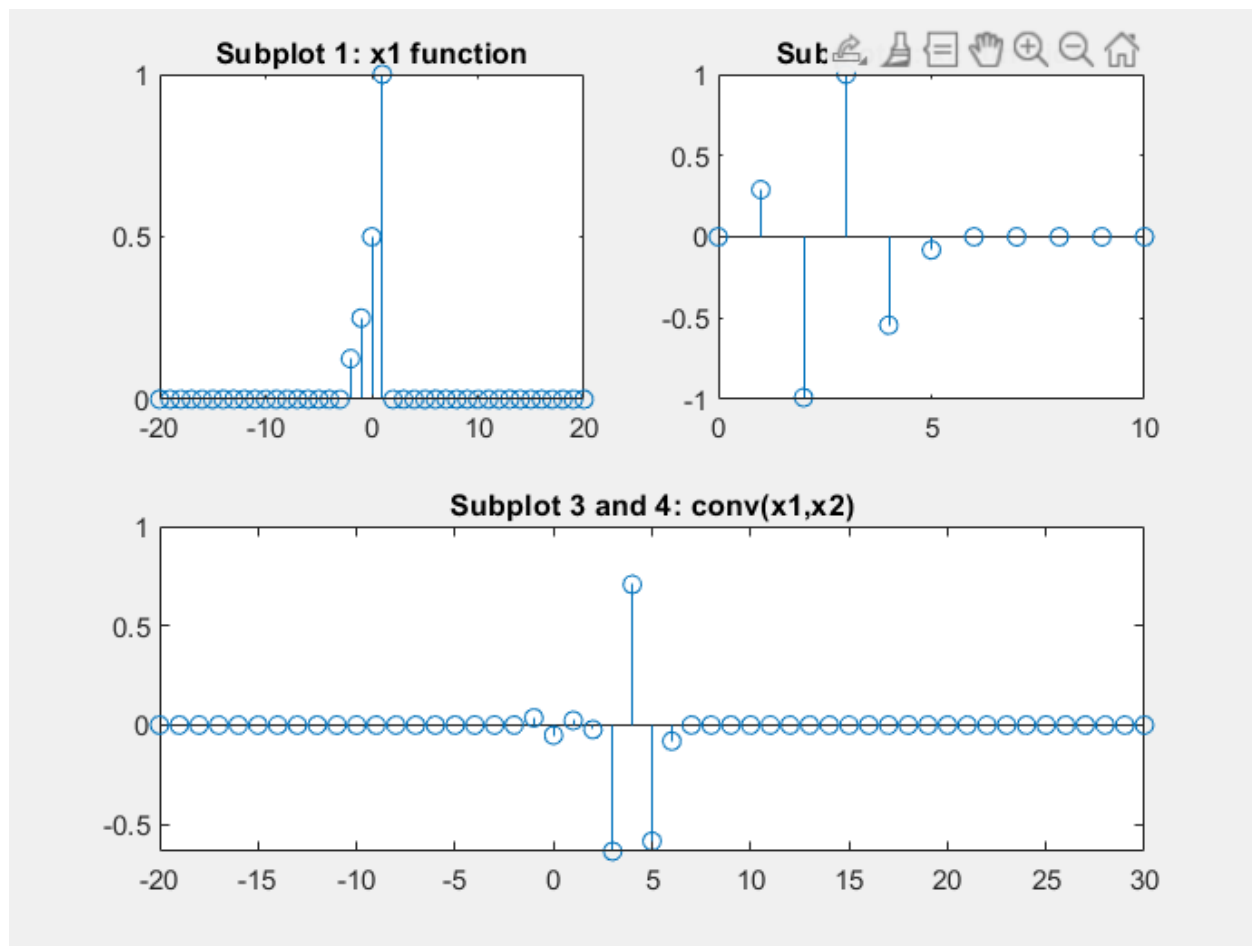
اولی از 1 تا 6 و دومی از -4 تا  $n$ .

با توجه به اینکه برای محاسبه سیگما نیازمند مقدار سیگنال های پله واحد هستیم و نه خود سیگنال، در حلقه `for` داخلی مقادیر به صورت  $u_1(m)$ ,  $u_2(m)$  نوشته شده است.

از متغیر کمکی `sum` برای جمع زدن مقادیر به ازای هر  $n$  استفاده شده است.

در پایان هر بار اجرای حلقه `for` داخلی، مقدار `sum` در  $x_2(n)$  قرار داده شده است.

شکل زیر سیگنال های  $x_1$ ,  $x_2$ ,  $\text{conv}(x_1, x_2)$  را نشان میدهد:

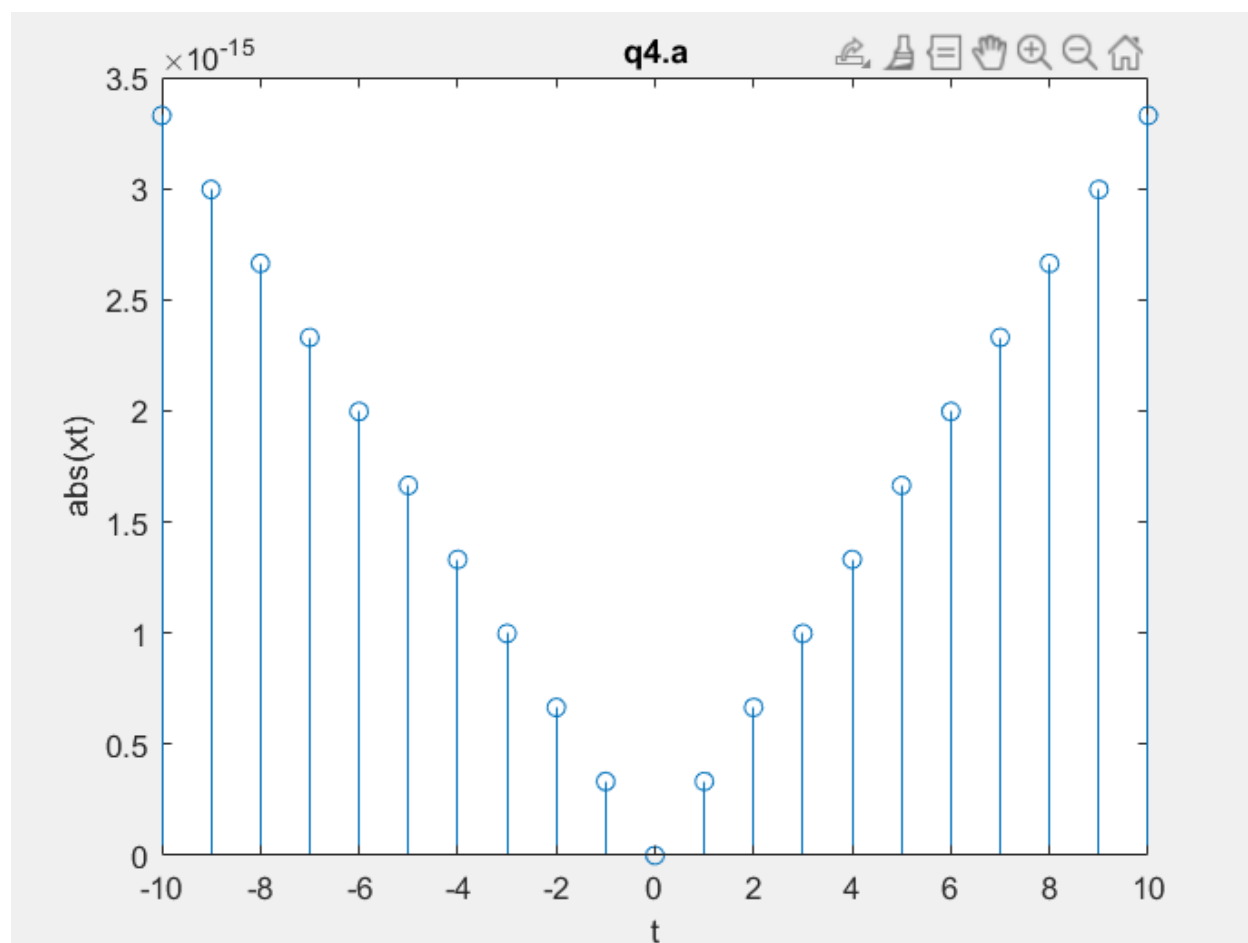


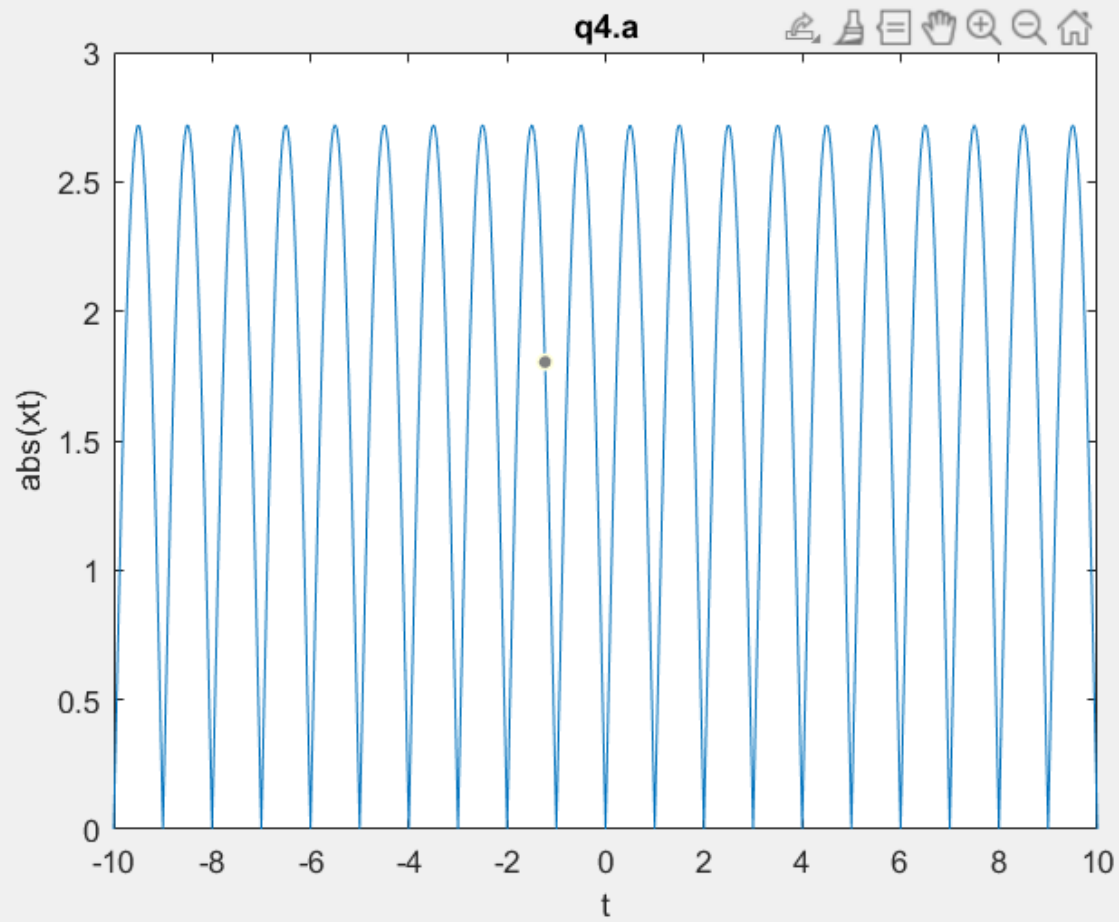


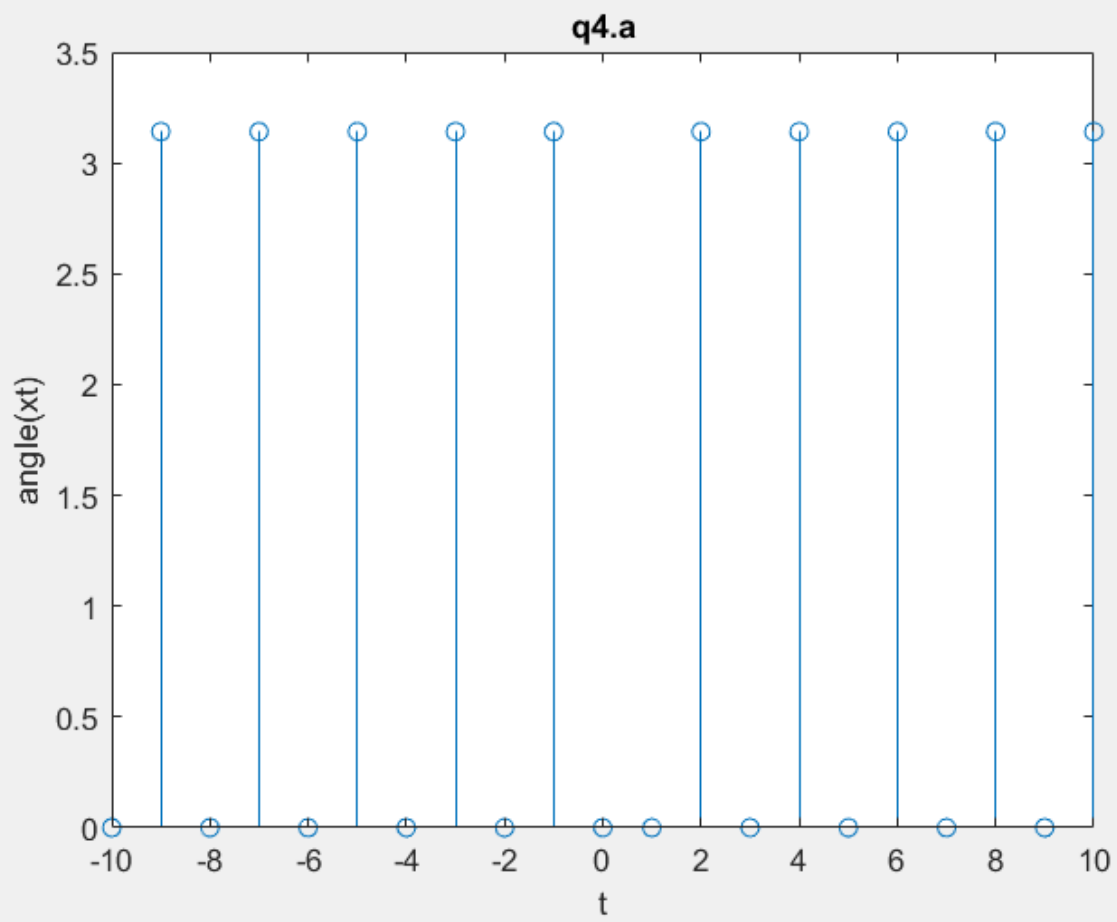
(4)

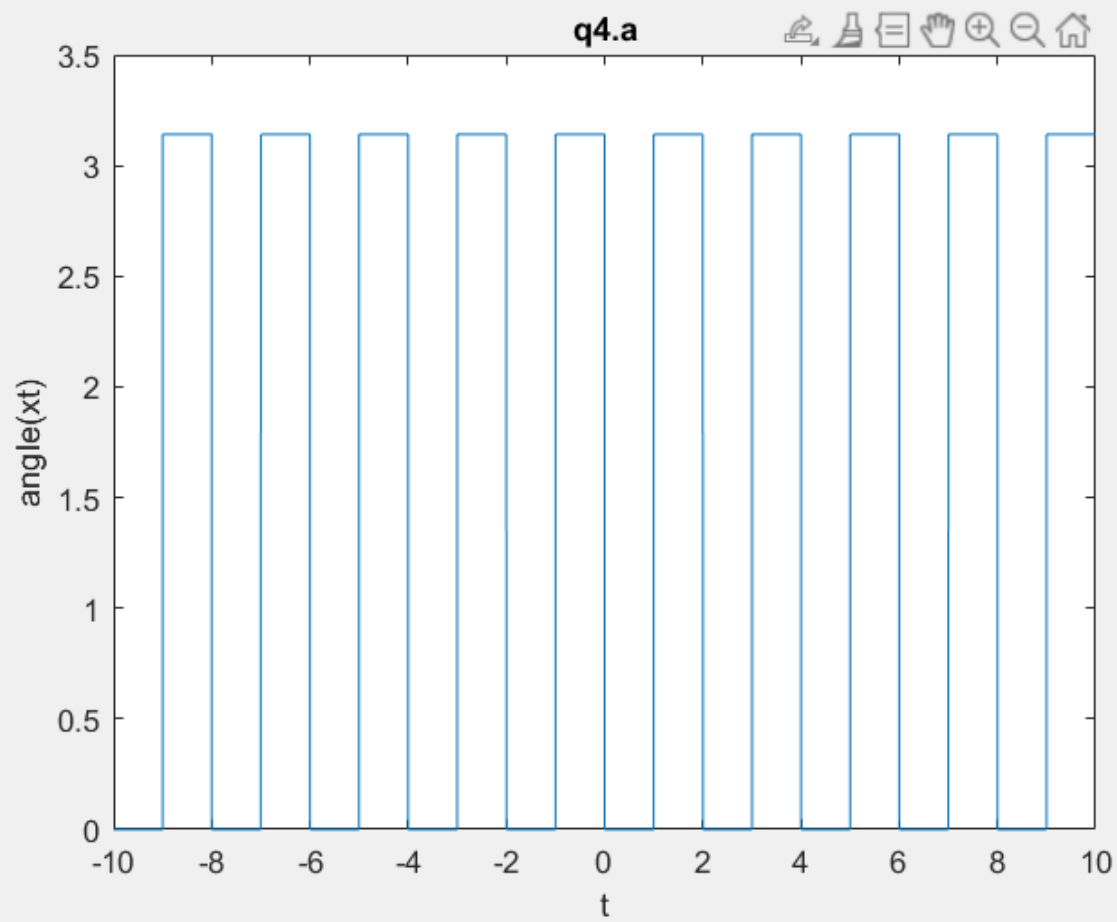
اندازه و فاز سیگنال های  $x(t)$ ,  $X(w)$  با استفاده از دستور های  $\text{abs}$ ,  $\text{angle}$  محاسبه شده و با استفاده از دستور های  $\text{stem}$ ,  $\text{plot}$  رسم شده اند.

شکل های مربوط به  $x(t)$ :









شکل های مربوط به  $X(w)$ :

