دانشگاه تهران
دانشکدگان فارابی
دانشکده مهندسی

**سیگنال‌ها و سیستم‌ها**
**نیمسال دوم ۰۳-۱۴۰۲**
http://courses.fouladi.ir/sigsys

## تکلیف کامپیوتری شماره‌ی ۷

فصل هفتم

# نمونه‌برداری

Sampling

تمرین‌های زیر را در MATLAB انجام بدهید و کدهای مربوطه را در قالب یک فایل m تحویل بدهید.

۱) **(آشنایی با نمونه‌برداری)** سیگنال $s(t) = \sin(\pi t)$ را با نرخ‌های 1، 1.5 و 2.5 هرتز نمونه‌برداری کنید $(-3 < t < 3)$.

(الف) سیگنال‌های نمونه‌برداری شده را در سه شکل جداگانه رسم کنید.

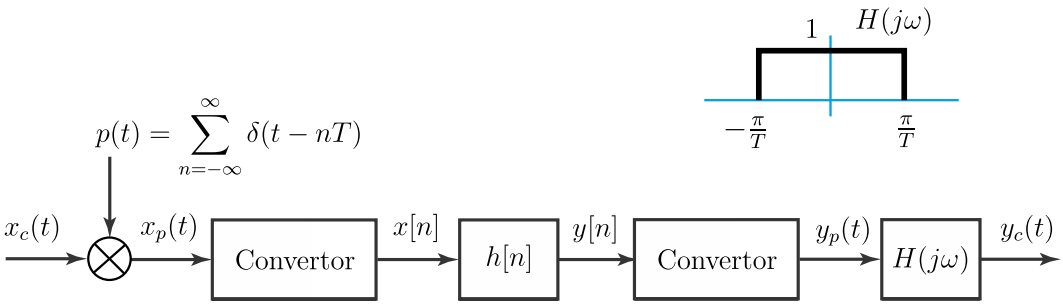
(ب) با اتصال نمونه‌ها به‌یکدیگر آنها را به‌صورت پیوسته بازیابی کنید.

(ج) در کدام موارد aliasing اتفاق می‌افتد؟

(د) فرکانس نمونه‌برداری که از aliasing جلوگیری می‌کند، چند هرتز است؟

۲) **(نمونه‌برداری و فیلتر کردن سیگنال)** برای سیگنال $x_c(t)$ سیگنال‌های $y[n]$ و $y_c(t)$ را به‌ازای نرخ‌های نمونه‌برداری $T = 0.05, 0.125, 0.25$ در بازه‌ی زمانی $-3 < t < 3$ به‌دست آورده و رسم کنید.

$$x(t) = \sin(6\pi t) + 3\cos(15\pi t) + noise$$




۳) **(آشنایی با Down-Sampling و Up-Sampling)** تصویر رنگی ضمیمه شده (`flowers.bmp`) را با استفاده از تابع `rgb2gray` به یک تصویر grayscale تبدیل کنید و سپس:

(الف) ابتدا این تصویر را با نرخ 2، down-sample کنید و تصویر به‌دست آمده را نمایش دهید. (برای این کار از تابع `downsample` استفاده کنید.)

(ب) تصویر حاصل از مرحله‌ی قبل را با نرخ 2، up-sample کنید و تصویر به‌دست آمده را نمایش دهید.

(ج) تفاوت تصویر به‌دست آمده در مرحله‌ی (ب) با تصویر اولیه را با استفاده از MSE محاسبه کنید.

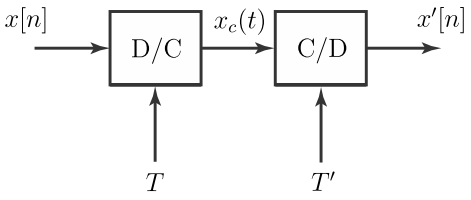
(د) بر روی تصویر جدید، هر یک از دو فیلتر بالاگذر و پایین‌گذر را اعمال نموده، نتیجه را نمایش داده و برای هر حالت، MSE را محاسبه کنید. (برای اعمال فیلتر از تابع `filter` استفاده کنید.)


Prepared by Kazim Fouladi <kazim@fouladi.ir>, Spring 2024 (۲۸ اسفند ۱۴۰۲)

# آشنایی با Down-Sampling و Up-Sampling

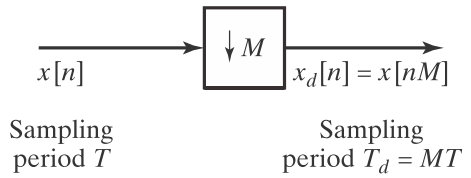
همان‌طورکه می‌دانید می‌توان سیگنال‌های پیوسته-زمان را به‌کمک نمونه‌برداری به سیگنال‌هایی دیجیتال و قابل پردازش توسط پردازنده‌های گسسته-زمان تبدیل کرد. می‌دانیم که نرخ نمونه‌برداری مشخص‌کننده‌ی آن است که در یک بازه‌ی زمانی چه تعداد نمونه از یک سیگنال پیوسته-زمان برداشته‌ایم. حال فرض کنید بنا بر دلایلی نیاز داریم که نرخ نمونه‌برداری تغییر کند (کم و یا زیاد بشود). چه راهکارهایی برای این‌کار وجود دارد؟

۱) یک راهکارکه در ابتدا به ذهن می‌رسد آن است که به کمک نمونه هایی که از سیگنال اصلی داریم، آن سیگنال را در حوزه زمان پیوسته بازسازی کنیم و سپس سیگنال بازسازی شده را با نرخ دلخواه نمونه‌برداری کنیم. بلوک دیاگرام زیر این پردازش را نشان می‌دهد.




این روش از دیدگاه تئوری کاملاً صحیح است اما در عمل کارایی لازم را ندارد. به این دلیل که در عمل برای پیاده‌سازی تبدیل آنالوگ به دیجیتال از بلوک A/D استفاده می‌شود. اما از آنجایی که بلوک A/D موجب ایجاد نویز چندی‌سازی (quantization-noise) در سیستم می‌شود و این نویز اصلاً مطلوب نیست. پس در عمل به دنبال راهکاری هستیم که با پردازش کاملاً گسسته بتوان نرخ نمونه‌برداری را تغییر بدهیم. بدین منظور باید با چند بلوک پردازش گسسته آشنا بشویم.

۲) **Donw-Sampler**: فرض کنید سیگنال اولیه با نرخ $T$ نمونه‌برداری شده باشد. می‌خواهیم این نرخ را به $T_d$ تغییر بدهیم به‌صورتی که $T_d = MT$ که در آن $M$ یک عدد صحیح مثبت است. به عمل کاهش نرخ نمونه‌برداری down-sampling می‌گویند. عملاً برای رسیدن به نمونه برداری با نرخ $T_d$ تنها کافی است که تعدادی از نمونه‌هایی را که داریم دور بریزیم. این کار توسط بلوکی به نام فشرده‌ساز (Compressor) انجام می‌شود. این بلوک یک سیگنال گسسته‌ی ورودی مانند $x[n]$ را به‌عنوان ورودی گرفته، از هر $M$ نمونه، $M-1$ نمونه را دور می‌ریزد و سیگنال $x_d[n]$ را با نمونه‌های باقیمانده می‌سازد و آن‌را به‌عنوان خروجی تحویل می‌دهد. این فرآیند به صورت دیاگرام بلوکی زیر نمایش داده می‌شود.




بازنمایی حوزه‌ی زمان و فرکانس فشرده‌ساز به‌صورت زیر می‌باشد:

- حوزه‌ی زمان:

$$x_d[n] = x[nM]$$

- حوزه‌ی فرکانس:

$$X_d(e^{j\omega}) = \frac{1}{M}\sum_{i=0}^{M-1} X(e^{j(\frac{\omega}{M}-\frac{2\pi i}{M})})$$

از آنجایی که رابطه‌ی حوزه‌ی فرکانس پیچیده‌تر است، بهتر است اثر این رابطه را به‌صورت نموداری درک کنیم:

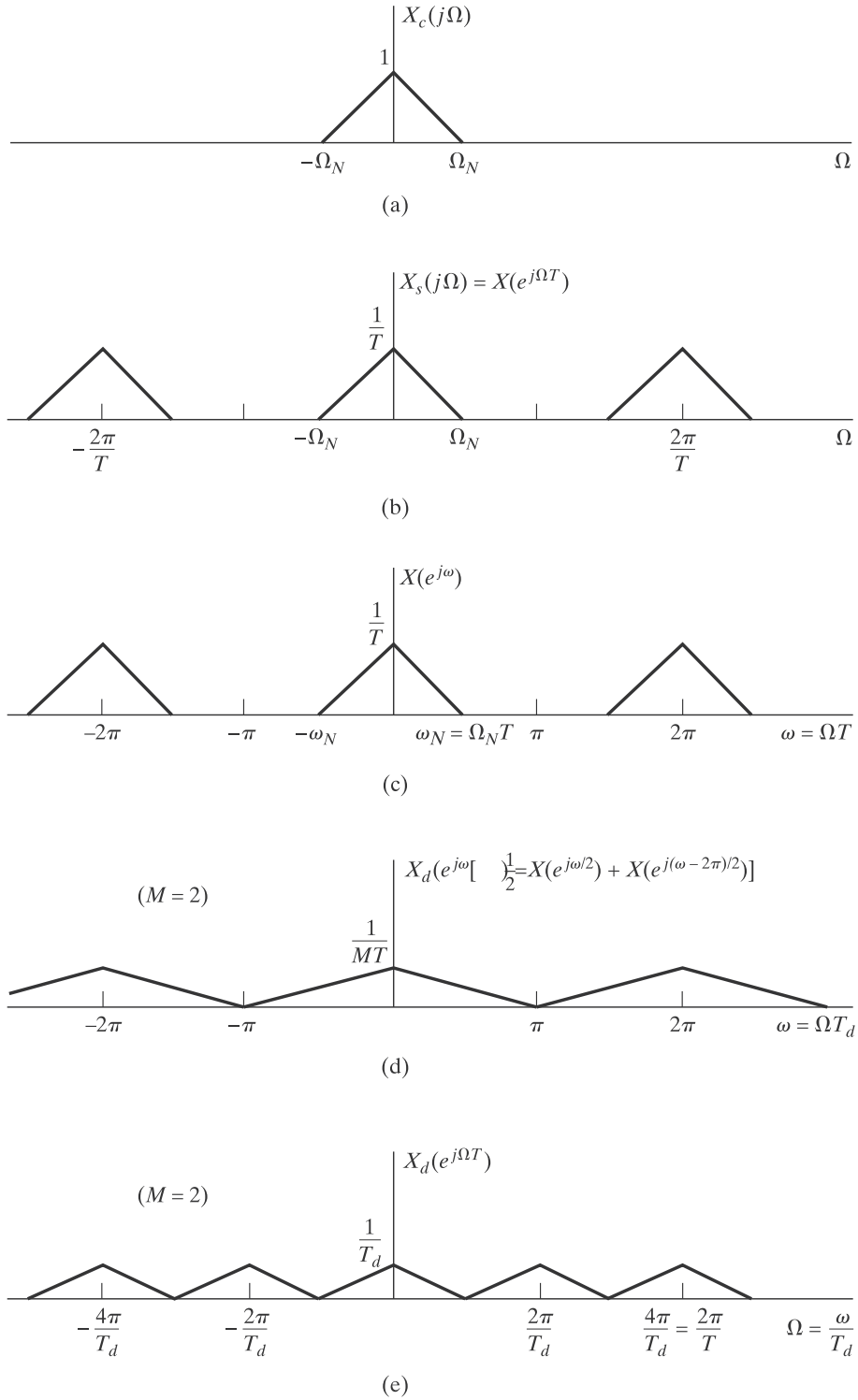
(الف) بازکردن محدوده‌ی فرکانسی $|\omega| \leq \pi$ با ضریب $M$

(ب) متناوب‌سازی با دوره‌ی تناوب $2\pi$ و جمع


Prepared by Kazim Fouladi <kazim@fouladi.ir>, Spring 2024 (۲۸ اسفند ۱۴۰۲)

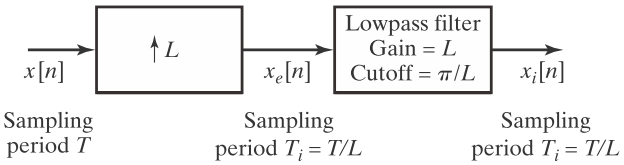
(ج) ضرب دامنه در $1/M$

برای درک بهتر، سیگنال زیر را که در حوزه‌ی فرکانس نمایش داده شده است را به یک فشرده‌ساز با $M = 2$ می‌دهیم.




۳) **Up-Sampler**: مجدداً فرض کنید که سیگنال اولیه با نرخ $T$ نمونه‌برداری شده است. می‌خواهیم این نرخ را به $T_i$ تغییر بدهیم به‌صورتی‌که $T_i = T/L$ که در آن $L$ یک عدد صحیح مثبت است. به عمل افزایش نرخ نمونه‌برداری up-sampling می‌گویند. برای رسیدن به نرخ $T_i$ کافی است به ازای هر نمونه، $L-1$ نمونه تولید کنیم. اما این نمونه‌ها باید به شکل مناسبی مقداردهی شوند تا سیگنال اصلی را خراب نکنیم. نسبت دادن مقادیر مناسب به این نمونه‌ها در واقع عمل درون‌یابی (interpolation) است. به

کمک فیلتر پایین‌گذر می‌توان درون‌یابی ایده‌آل را پیاده‌سازی کرد. روش کار بدین صورت است که به سیگنال ورودی به تعداد کافی نمونه‌ی صفر اضافه می‌کنیم و سپس این نمونه‌ها را با مقادیر درون‌یابی شده پر می‌کنیم. بلوکی که عمل اضافه کردن نمونه‌های صفر را انجام می‌دهد، بسط‌دهنده (expander) نام دارد. دیاگرام بلوکی یک up-sampler در زیر آورده شده است:




بازنمایی حوزه‌ی زمان و فرکانس بسط‌دهنده به‌صورت زیر می‌باشد:

- حوزه‌ی زمان:

$$x_e[n] = \begin{cases} x\left[\frac{n}{L}\right] & , n = \cdots, -2L, -L, 0, L, 2L, \cdots \\ 0 & , \text{otherwise} \end{cases}$$

- حوزه‌ی فرکانس:

$$X_e(e^{j\omega}) = X(e^{j\omega L})$$

# مراجع

[1] M.N.O. Sadiku, W.H. Ali, **Signals and Systems: A Primer with Matlab**, CRC Press, 2016.

[2] A.V. Oppenheim and R.W. Schafer, **Discrete-Time Signal Processing**, 3rd edition, Prentice Hall, 2010.