EECS 16A Designing Information Devices and Systems I
Fall 2022 Discussion 14B

1. Polynomial Fitting

Let's try an example. Say we know that the output, y, is a quartic polynomial in x. This means that we know that y and x are related as follows:

$$y = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + a_4 x^4$$

We're also given the following observations:

x	y
0.0	24.0
0.5	6.61
1.0	0.0
1.5	-0.95
2.0	0.07
2.5	0.73
3.0	-0.12
3.5	-0.83
4.0	-0.04
4.5	6.42

(a) What are the unknowns in this question?

Answer:

The unknowns are a_0, a_1, a_2, a_3, and a_4.

(b) Can you write an equation corresponding to the first observation (x_0, y_0), in terms of a_0, a_1, a_2, a_3, and a_4? What does this equation look like? Is it linear in the unknowns?

Answer:

Plugging (x_0, y_0) into the expression for y in terms of x, we get

$$24 = a_0 + a_1 \cdot 0 + a_2 \cdot 0^2 + a_3 \cdot 0^3 + a_4 \cdot 0^4$$

You can see that this equation is linear in a_0, a_1, a_2, a_3, and a_4.

(c) Now, write a system of equations in terms of a_0, a_1, a_2, a_3, and a_4 using *all of the observations*.

Answer:

Write the next equation using the second observation. You will now get:

$$6.61 = a_0 + a_1 \cdot (0.5) + a_2 \cdot (0.5)^2 + a_3 \cdot (0.5)^3 + a_4 \cdot (0.5)^4$$

And for the third:

$$0.0 = a_0 + a_1 \cdot (1) + a_2 \cdot 1^2 + a_3 \cdot 1^3 + a_4 \cdot 1^4$$

Do you see a pattern? Let's write the entire system of equations in terms of a matrix now.

$$\begin{bmatrix} 1 & 0 & 0^2 & 0^3 & 0^4 \\ 1 & 0.5 & (0.5)^2 & (0.5)^3 & (0.5)^4 \\ 1 & 1 & 1^2 & 1^3 & 1^4 \\ 1 & 1.5 & (1.5)^2 & (1.5)^3 & (1.5)^4 \\ 1 & 2 & 2^2 & 2^3 & 2^4 \\ 1 & 2.5 & (2.5)^2 & (2.5)^3 & (2.5)^4 \\ 1 & 3 & 3^2 & 3^3 & 3^4 \\ 1 & 3.5 & (3.5)^2 & (3.5)^3 & (3.5)^4 \\ 1 & 4 & 4^2 & 4^3 & 4^4 \\ 1 & 4.5 & (4.5)^2 & (4.5)^3 & (4.5)^4 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix} = \begin{bmatrix} 24 \\ 6.61 \\ 0.0 \\ -0.95 \\ 0.07 \\ 0.73 \\ -0.12 \\ -0.83 \\ -0.04 \\ 6.42 \end{bmatrix}$$

(d) Finally, solve for a_0, a_1, a_2, a_3, and a_4 using IPython or any method you like. You have now found the quartic polynomial that best fits the data!

Answer:

Let $\mathbf{D}$ be the big matrix from the previous part.

$$\vec{a} = (\mathbf{D}^T\mathbf{D})^{-1}\mathbf{D}^T\vec{y} = \begin{bmatrix} 24.00958042 \\ -49.99515152 \\ 35.0039627 \\ -9.99561772 \\ 0.99841492 \end{bmatrix}$$

It turns out that the actual parameters for the polynomial equation were:

$$\vec{a} = \begin{bmatrix} 24 \\ -50 \\ 35 \\ -10 \\ 1 \end{bmatrix}$$

(Remember that our observations were noisy.)

Therefore, we have actually done pretty well with the least squares estimate!

(e) What if we didn't know the degree of the polynomial? Use the IPython Notebook to explore what happens when we choose a polynomial degree other 4 and explain what you see.

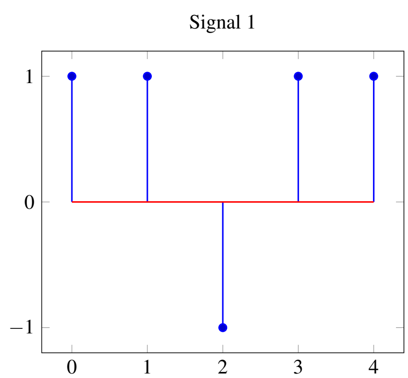
Answer: You'll notice that if you try to fit the data with a polynomial of degree < 4, your solution will be a worse fit than the quartic fit we got before. This makes sense; you're basically trying to fit the data with a *smaller set of parameters*, which means that you are trying to approximate the data by a coarser model. It is only natural that you are going to do worse, and our cost function tells us so. Now, what if you tried to fit the data with a polynomial of degree > 4? Your cost function will improve but ever so slightly. You're not gaining much, and furthermore, you are *overfitting* your data after some point of time. When you are overfitting, your model starts to describe the underlying noise in the data rather than the data itself. This means that it is actually a worse model for making future predictions! This is called the bias variance theorem in statistics and traditional ML.

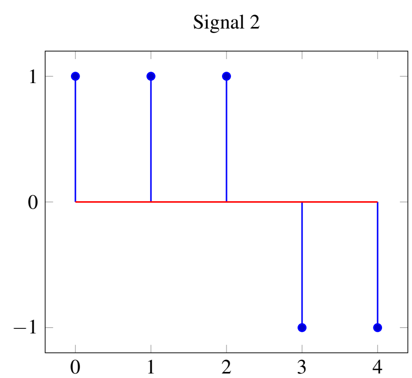
(f) OPTIONAL: Play around with what happens when you add more noise to the data or if you decide to drop data points on the IPython Notebook. Additionally, explore what you see when you change the degree of the polynomial alongside these factors.

Answer: You'll notice that if you increase the noise or you remove more data points, the polynomial fit becomes worse compared to the original data, with the errors becoming increasingly larger. However, if you play around with the degree of the polynomial, you might notice that if we choose our polynomial to be around degree 4-6, when we drop a few points or increase the noise moderately, it still gives us a polynomial fit that is very close to the original. If our degree was 10-15, dropping points or increasing our noise would wildly change our polynomial. This is another downside of overfitting our model, it is extremely susceptible to noisy or missing data.

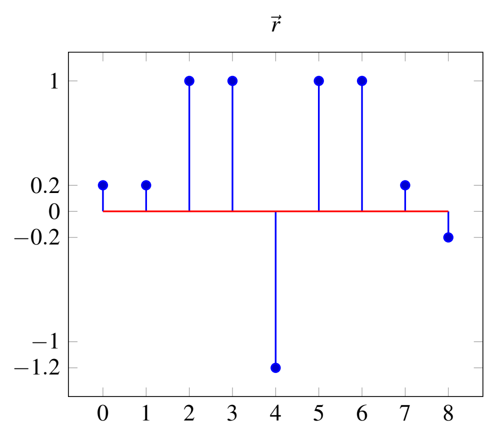
2. Identifying satellites and their delays

We are given the following two signals, $\vec{s}_1$ and $\vec{s}_2$ respectively, that are signatures for two satellites. Your cell phone recieves signals from these two satellites and given a recieved signal $r[n]$ you can identify which, if any, satellite sent the message based on their personal codes.





(a) Your cellphone antenna receives the following signal $r[n]$. You know that there may be some noise present in $r[n]$ in addition to the transmission from the satellite.



By computing the cross-correlations, can you identify which satellite(s) most likely sent the signal, and by what shift the code is identified relative to our received signal? You can use iPython to compute the cross-correlation. When using iPython to plot, think about the range of shifts k that we are interested in plotting based on the lengths of the signals.

Answer: We calculate both $\text{corr}_r(\vec{s}_1)[k]$ and $\text{corr}_r(\vec{s}_2)[k]$:

$$\text{corr}_{\vec{r}}(\vec{s}_1)[k]$$

$\vec{r}$	0.2		0.2		1		1		−1.2		1		1		0.2		−0.2	
$\vec{s}_1[n+4]$	1		0		0		0		0		0		0		0		0	
$\langle\vec{r},\vec{s}_1[n+4]\rangle$	0.2	+	0	+	0	+	0	+	0	+	0	+	0	+	0	+	0	= 0.2

$\vec{r}$	0.2		0.2		1		1		−1.2		1		1		0.2		−0.2	
$\vec{s}_1[n+3]$	1		1		0		0		0		0		0		0		0	
$\langle\vec{r},\vec{s}_1[n+3]\rangle$	0.2	+	0.2	+	0	+	0	+	0	+	0	+	0	+	0	+	0	= 0.4

$\vec{r}$	0.2		0.2		1		1		−1.2		1		1		0.2		−0.2	
$\vec{s}_1[n+2]$	−1		1		1		0		0		0		0		0		0	
$\langle\vec{r},\vec{s}_1[n+2]\rangle$	−0.2	+	0.2	+	1	+	0	+	0	+	0	+	0	+	0	+	0	= 1

$\vec{r}$	0.2		0.2		1		1		−1.2		1		1		0.2		−0.2	
$\vec{s}_1[n+1]$	1		−1		1		1		0		0		0		0		0	
$\langle\vec{r},\vec{s}_1[n+1]\rangle$	0.2	+	−0.2	+	1	+	1	+	0	+	0	+	0	+	0	+	0	= 2

$\vec{r}$	0.2		0.2		1		1		−1.2		1		1		0.2		−0.2	
$\vec{s}_1[n]$	1		1		−1		1		1		0		0		0		0	
$\langle\vec{r},\vec{s}_1[n]\rangle$	0.2	+	0.2	+	−1	+	1	+	−1.2	+	0	+	0	+	0	+	0	= −0.8

$\vec{r}$	0.2		0.2		1		1		−1.2		1		1		0.2		−0.2	
$\vec{s}_1[n-1]$	0		1		1		−1		1		1		0		0		0	
$\langle\vec{r},\vec{s}_1[n-1]\rangle$	0	+	0.2	+	1	+	−1	+	−1.2	+	1	+	0	+	0	+	0	= 0

$\vec{r}$	0.2		0.2		1		1		−1.2		1		1		0.2		−0.2	
$\vec{s}_1[n-2]$	0		0		1		1		−1		1		1		0		0	
$\langle\vec{r},\vec{s}_1[n-2]\rangle$	0	+	0	+	1	+	1	+	1.2	+	1	+	1	+	0	+	0	= **5.2**

$\vec{r}$	0.2		0.2		1		1		−1.2		1		1		0.2		−0.2	
$\vec{s}_1[n-3]$	0		0		0		1		1		−1		1		1		0	
$\langle\vec{r},\vec{s}_1[n-3]\rangle$	0	+	0	+	0	+	1	+	−1.2	+	−1	+	1	+	0.2	+	0	= 0

$\vec{r}$	0.2		0.2		1		1		−1.2		1		1		0.2		−0.2	
$\vec{s}_1[n-4]$	0		0		0		0		1		1		−1		1		1	
$\langle\vec{r},\vec{s}_1[n-4]\rangle$	0	+	0	+	0	+	0	+	−1.2	+	1	+	−1	+	0.2	+	−0.2	= −1.2

$\vec{r}$	0.2		0.2		1		1		−1.2		1		1		0.2		−0.2	
$\vec{s}_1[n-5]$	0		0		0		0		0		1		1		−1		1	
$\langle\vec{r},\vec{s}_1[n-5]\rangle$	0	+	0	+	0	+	0	+	0	+	1	+	1	+	−0.2	+	−0.2	= 1.6

$\vec{r}$	0.2	0.2	1	1	−1.2	1	1	0.2	−0.2	
$\vec{s}_1[n-6]$	0	0	0	0	0	0	1	1	−1	
$\langle \vec{r}, \vec{s}_1[n-6] \rangle$	0 +	0 +	0 +	0 +	0 +	0 +	1 +	0.2 +	0.2	= 1.4

$\vec{r}$	0.2	0.2	1	1	−1.2	1	1	0.2	−0.2	
$\vec{s}_1[n-7]$	0	0	0	0	0	0	0	1	1	
$\langle \vec{r}, \vec{s}_1[n-7] \rangle$	0 +	0 +	0 +	0 +	0 +	0 +	0 +	0.2 +	−0.2	= 0

$\vec{r}$	0.2	0.2	1	1	−1.2	1	1	0.2	−0.2	
$\vec{s}_1[n-8]$	0	0	0	0	0	0	0	0	1	
$\langle \vec{r}, \vec{s}_1[n-8] \rangle$	0 +	0 +	0 +	0 +	0 +	0 +	0 +	0 +	−0.2	= −0.2

$\mathrm{corr}_{\vec{r}}(\vec{s}_2)[k]$

$\vec{r}$	0.2	0.2	1	1	−1.2	1	1	0.2	−0.2	
$\vec{s}_2[n+4]$	−1	0	0	0	0	0	0	0	0	
$\langle \vec{r}, \vec{s}_2[n+4] \rangle$	−0.2 +	0 +	0 +	0 +	0 +	0 +	0 +	0 +	0	= −0.2

$\vec{r}$	0.2	0.2	1	1	−1.2	1	1	0.2	−0.2	
$\vec{s}_2[n+3]$	−1	−1	0	0	0	0	0	0	0	
$\langle \vec{r}, \vec{s}_2[n+3] \rangle$	−0.2 +	−0.2 +	0 +	0 +	0 +	0 +	0 +	0 +	0	= −0.4

$\vec{r}$	0.2	0.2	1	1	−1.2	1	1	0.2	−0.2	
$\vec{s}_2[n+2]$	1	−1	−1	0	0	0	0	0	0	
$\langle \vec{r}, \vec{s}_2[n+2] \rangle$	0.2 +	−0.2 +	−1 +	0 +	0 +	0 +	0 +	0 +	0	= −1

$\vec{r}$	0.2	0.2	1	1	−1.2	1	1	0.2	−0.2	
$\vec{s}_2[n+1]$	1	1	−1	−1	0	0	0	0	0	
$\langle \vec{r}, \vec{s}_2[n+1] \rangle$	0.2 +	0.2 +	−1 +	−1 +	0 +	0 +	0 +	0 +	0	= −1.6

$\vec{r}$	0.2	0.2	1	1	−1.2	1	1	0.2	−0.2	
$\vec{s}_2[n]$	1	1	1	−1	−1	0	0	0	0	
$\langle \vec{r}, \vec{s}_2[n] \rangle$	0.2 +	0.2 +	1 +	−1 +	1.2 +	0 +	0 +	0 +	0	= 1.6

$\vec{r}$	0.2	0.2	1	1	−1.2	1	1	0.2	−0.2	
$\vec{s}_2[n-1]$	0	1	1	1	−1	−1	0	0	0	
$\langle \vec{r}, \vec{s}_2[n-1] \rangle$	0 +	0.2 +	1 +	1 +	1.2 +	−1 +	0 +	0 +	0	= 2.4

$\vec{r}$	0.2	0.2	1	1	−1.2	1	1	0.2	−0.2	
$\vec{s}_2[n-2]$	0	0	1	1	1	−1	−1	0	0	
$\langle \vec{r}, \vec{s}_2[n-2] \rangle$	0 +	0 +	1 +	1 +	−1.2 +	−1 +	−1 +	0 +	0	= −1.2

$\vec{r}$	0.2	0.2	1	1	-1.2	1	1	0.2	-0.2	
$\vec{s}_2[n-3]$	0	0	0	1	1	1	-1	-1	0	
$\langle \vec{r}, \vec{s}_2[n-3] \rangle$	$0\ +$	$0\ +$	$0\ +$	$1\ +$	$-1.2\ +$	$1\ +$	$-1\ +$	$-0.2\ +$	0	$= -0.4$

$\vec{r}$	0.2	0.2	1	1	-1.2	1	1	0.2	-0.2	
$\vec{s}_2[n-4]$	0	0	0	0	1	1	1	-1	-1	
$\langle \vec{r}, \vec{s}_2[n-4] \rangle$	$0\ +$	$0\ +$	$0\ +$	$0\ +$	$-1.2\ +$	$1\ +$	$1\ +$	$-0.2\ +$	0.2	$= 0.8$

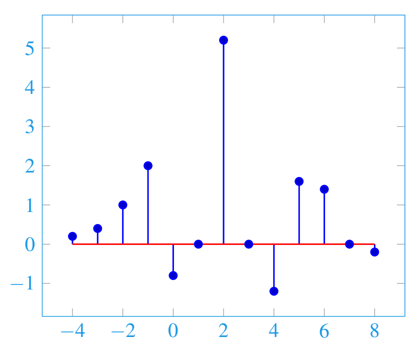
$\vec{r}$	0.2	0.2	1	1	-1.2	1	1	0.2	-0.2	
$\vec{s}_2[n-5]$	0	0	0	0	0	1	1	1	-1	
$\langle \vec{r}, \vec{s}_2[n-5] \rangle$	$0\ +$	$0\ +$	$0\ +$	$0\ +$	$0\ +$	$1\ +$	$1\ +$	$0.2\ +$	0.2	$= 2.4$

$\vec{r}$	0.2	0.2	1	1	-1.2	1	1	0.2	-0.2	
$\vec{s}_2[n-6]$	0	0	0	0	0	0	1	1	1	
$\langle \vec{r}, \vec{s}_2[n-6] \rangle$	$0\ +$	$0\ +$	$0\ +$	$0\ +$	$0\ +$	$0\ +$	$1\ +$	$0.2\ +$	-0.2	$= 1$

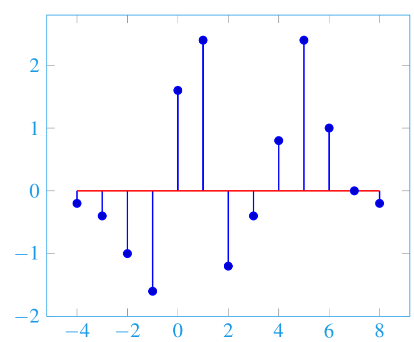
$\vec{r}$	0.2	0.2	1	1	-1.2	1	1	0.2	-0.2	
$\vec{s}_2[n-7]$	0	0	0	0	0	0	0	1	1	
$\langle \vec{r}, \vec{s}_2[n-7] \rangle$	$0\ +$	$0\ +$	$0\ +$	$0\ +$	$0\ +$	$0\ +$	$0\ +$	$0.2\ +$	-0.2	$= 0$

$\vec{r}$	0.2	0.2	1	1	-1.2	1	1	0.2	-0.2	
$\vec{s}_2[n-8]$	0	0	0	0	0	0	0	0	1	
$\langle \vec{r}, \vec{s}_2[n-8] \rangle$	$0\ +$	$0\ +$	$0\ +$	$0\ +$	$0\ +$	$0\ +$	$0\ +$	$0\ +$	-0.2	$= -0.2$

Cross-correlation of $\vec{r}$ and $\vec{s}_1$

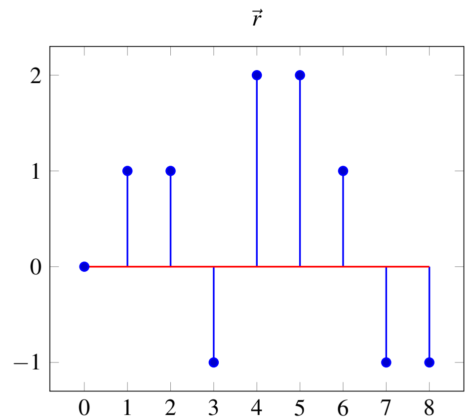


Cross-correlation of $\vec{r}$ and $\vec{s}_2$



The maximum correlation value is 5.2 at $k = 2$ from satellite 1. Therefore, the transmission likely comes from satellite 1.

(b) Now your cellphone receives a new signal $r[n]$ as below. Can you identify which satellite(s) most likely sent the signal, and by what shift the code is identified relative to our received signal?



Answer: We want to find shifts k_1 and k_2 such that: $\vec{r}[n] = \vec{s}_1[n-k_1] + \vec{s}_2[n-k_2]$.
We calculate both $\mathrm{corr}_{\vec{r}}(\vec{s}_1)[k]$ and $\mathrm{corr}_{\vec{r}}(\vec{s}_1)[k]$ for different shifts k. The index where the maximum correlation value is achieved will tell us the shift indices (delays).

$\mathrm{corr}_{\vec{r}}(\vec{s}_1)[k]$

$\vec{r}$	0		1		1		−1		2		2		1		−1		−1	
$\vec{s}_1[n+4]$	1		0		0		0		0		0		0		0		0	
$\langle \vec{r}, \vec{s}_1[n+4] \rangle$	0	+	0	+	0	+	0	+	0	+	0	+	0	+	0	+	0	$=0$

$\vec{r}$	0		1		1		−1		2		2		1		−1		−1	
$\vec{s}_1[n+3]$	1		1		0		0		0		0		0		0		0	
$\langle \vec{r}, \vec{s}_1[n+3] \rangle$	0	+	1	+	0	+	0	+	0	+	0	+	0	+	0	+	0	$=1$

$\vec{r}$	0		1		1		−1		2		2		1		−1		−1	
$\vec{s}_1[n+2]$	−1		1		1		0		0		0		0		0		0	
$\langle \vec{r}, \vec{s}_1[n+2] \rangle$	0	+	1	+	1	+	0	+	0	+	0	+	0	+	0	+	0	$=2$

$\vec{r}$	0		1		1		−1		2		2		1		−1		−1	
$\vec{s}_1[n+1]$	1		−1		1		1		0		0		0		0		0	
$\langle \vec{r}, \vec{s}_1[n+1] \rangle$	0	+	−1	+	1	+	−1	+	0	+	0	+	0	+	0	+	0	$=-1$

$\vec{r}$	0		1		1		−1		2		2		1		−1		−1	
$\vec{s}_1[n]$	1		1		−1		1		1		0		0		0		0	
$\langle \vec{r}, \vec{s}_1[n] \rangle$	0	+	1	+	−1	+	−1	+	2	+	0	+	0	+	0	+	0	$=1$

$\vec{r}$	0		1		1		−1		2		2		1		−1		−1	
$\vec{s}_1[n-1]$	0		1		1		−1		1		1		0		0		0	
$\langle \vec{r}, \vec{s}_1[n-1] \rangle$	0	+	1	+	1	+	1	+	2	+	2	+	0	+	0	+	0	$=\mathbf{7}$

$\vec{r}$	0		1		1		−1		2		2		1		−1		−1	
$\vec{s}_1[n-2]$	0		0		1		1		−1		1		1		0		0	
$\langle\vec{r},\vec{s}_1[n-2]\rangle$	0	+	0	+	1	+	−1	+	−2	+	2	+	1	+	0	+	0	= 1

$\vec{r}$	0		1		1		−1		2		2		1		−1		−1	
$\vec{s}_1[n-3]$	0		0		0		1		1		−1		1		1		0	
$\langle\vec{r},\vec{s}_1[n-3]\rangle$	0	+	0	+	0	+	−1	+	2	+	−2	+	1	+	−1	+	0	= −1

$\vec{r}$	0		1		1		−1		2		2		1		−1		−1	
$\vec{s}_1[n-4]$	0		0		0		0		1		1		−1		1		1	
$\langle\vec{r},\vec{s}_1[n-4]\rangle$	0	+	0	+	0	+	0	+	2	+	2	+	−1	+	−1	+	−1	= 1

$\vec{r}$	0		1		1		−1		2		2		1		−1		−1	
$\vec{s}_1[n-5]$	0		0		0		0		0		1		1		−1		1	
$\langle\vec{r},\vec{s}_1[n-5]\rangle$	0	+	0	+	0	+	0	+	0	+	2	+	1	+	1	+	−1	= 3

$\vec{r}$	0		1		1		−1		2		2		1		−1		−1	
$\vec{s}_1[n-6]$	0		0		0		0		0		0		1		1		−1	
$\langle\vec{r},\vec{s}_1[n-6]\rangle$	0	+	0	+	0	+	0	+	0	+	0	+	1	+	−1	+	1	= 1

$\vec{r}$	0		1		1		−1		2		2		1		−1		−1	
$\vec{s}_1[n-7]$	0		0		0		0		0		0		0		1		1	
$\langle\vec{r},\vec{s}_1[n-7]\rangle$	0	+	0	+	0	+	0	+	0	+	0	+	0	+	−1	+	−1	= −2

$\vec{r}$	0		1		1		−1		2		2		1		−1		−1	
$\vec{s}_1[n-8]$	0		0		0		0		0		0		0		0		1	
$\langle\vec{r},\vec{s}_1[n-8]\rangle$	0	+	0	+	0	+	0	+	0	+	0	+	0	+	0	+	−1	= −1

$$\mathrm{corr}_{\vec{r}}(\vec{s}_2)[k]$$

$\vec{r}$	0		1		1		−1		2		2		1		−1		−1	
$\vec{s}_2[n+4]$	−1		0		0		0		0		0		0		0		0	
$\langle\vec{r},\vec{s}_2[n+4]\rangle$	0	+	0	+	0	+	0	+	0	+	0	+	0	+	0	+	0	= 0

$\vec{r}$	0		1		1		−1		2		2		1		−1		−1	
$\vec{s}_2[n+3]$	−1		−1		0		0		0		0		0		0		0	
$\langle\vec{r},\vec{s}_2[n+3]\rangle$	0	+	−1	+	0	+	0	+	0	+	0	+	0	+	0	+	0	= −1

$\vec{r}$	0		1		1		−1		2		2		1		−1		−1	
$\vec{s}_2[n+2]$	1		−1		−1		0		0		0		0		0		0	
$\langle\vec{r},\vec{s}_2[n+2]\rangle$	0	+	−1	+	−1	+	0	+	0	+	0	+	0	+	0	+	0	= −2

$\vec{r}$	0	1	1	-1	2	2	1	-1	-1	
$\vec{s}_2[n+1]$	1	1	-1	-1	0	0	0	0	0	
$\langle \vec{r}, \vec{s}_2[n+1] \rangle$	$0\ +$	$1\ +$	$-1\ +$	$1\ +$	$0\ +$	$0\ +$	$0\ +$	$0\ +$	0	$= 1$

$\vec{r}$	0	1	1	-1	2	2	1	-1	-1	
$\vec{s}_2[n]$	1	1	1	-1	-1	0	0	0	0	
$\langle \vec{r}, \vec{s}_2[n] \rangle$	$0\ +$	$1\ +$	$1\ +$	$1\ +$	$-2\ +$	$0\ +$	$0\ +$	$0\ +$	0	$= 1$

$\vec{r}$	0	1	1	-1	2	2	1	-1	-1	
$\vec{s}_2[n-1]$	0	1	1	1	-1	-1	0	0	0	
$\langle \vec{r}, \vec{s}_2[n-1] \rangle$	$0\ +$	$1\ +$	$1\ +$	$-1\ +$	$-2\ +$	$-2\ +$	$0\ +$	$0\ +$	0	$= -3$

$\vec{r}$	0	1	1	-1	2	2	1	-1	-1	
$\vec{s}_2[n-2]$	0	0	1	1	1	-1	-1	0	0	
$\langle \vec{r}, \vec{s}_2[n-2] \rangle$	$0\ +$	$0\ +$	$1\ +$	$-1\ +$	$2\ +$	$-2\ +$	$-1\ +$	$0\ +$	0	$= -1$

$\vec{r}$	0	1	1	-1	2	2	1	-1	-1	
$\vec{s}_2[n-3]$	0	0	0	1	1	1	-1	-1	0	
$\langle \vec{r}, \vec{s}_2[n-3] \rangle$	$0\ +$	$0\ +$	$0\ +$	$-1\ +$	$2\ +$	$2\ +$	$-1\ +$	$1\ +$	0	$= 3$

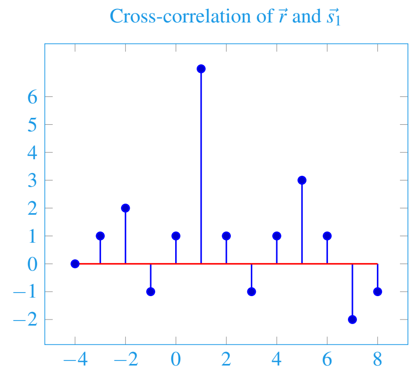
$\vec{r}$	0	1	1	-1	2	2	1	-1	-1	
$\vec{s}_2[n-4]$	0	0	0	0	1	1	1	-1	-1	
$\langle \vec{r}, \vec{s}_2[n-4] \rangle$	$0\ +$	$0\ +$	$0\ +$	$0\ +$	$2\ +$	$2\ +$	$1\ +$	$1\ +$	1	$= \mathbf{7}$

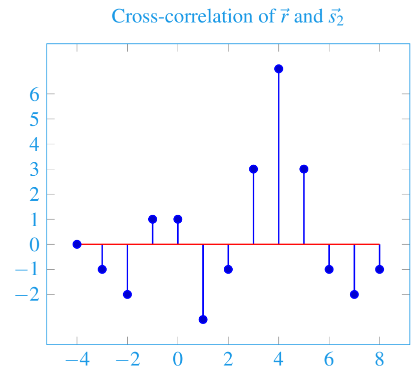
$\vec{r}$	0	1	1	-1	2	2	1	-1	-1	
$\vec{s}_2[n-5]$	0	0	0	0	0	1	1	1	-1	
$\langle \vec{r}, \vec{s}_2[n-5] \rangle$	$0\ +$	$0\ +$	$0\ +$	$0\ +$	$0\ +$	$2\ +$	$1\ +$	$-1\ +$	1	$= 3$

$\vec{r}$	0	1	1	-1	2	2	1	-1	-1	
$\vec{s}_2[n-6]$	0	0	0	0	0	0	1	1	1	
$\langle \vec{r}, \vec{s}_2[n-6] \rangle$	$0\ +$	$0\ +$	$0\ +$	$0\ +$	$0\ +$	$0\ +$	$1\ +$	$-1\ +$	-1	$= -1$

$\vec{r}$	0	1	1	-1	2	2	1	-1	-1	
$\vec{s}_2[n-7]$	0	0	0	0	0	0	0	1	1	
$\langle \vec{r}, \vec{s}_2[n-7] \rangle$	$0\ +$	$0\ +$	$0\ +$	$0\ +$	$0\ +$	$0\ +$	$0\ +$	$-1\ +$	-1	$= -2$

$\vec{r}$	0	1	1	-1	2	2	1	-1	-1	
$\vec{s}_2[n-8]$	0	0	0	0	0	0	0	0	1	
$\langle \vec{r}, \vec{s}_2[n-8] \rangle$	$0\ +$	$0\ +$	$0\ +$	$0\ +$	$0\ +$	$0\ +$	$0\ +$	$0\ +$	-1	$= -1$





The maximum correlation between signals $\vec{r}$ and $\vec{s}_1$ was achieved at $k_1 = 1$, and the maximum correlation between signals $\vec{r}$ and $\vec{s}_2$ was achieved at $k_2 = 4$.