Janne Koljonen
Vaasa yliopisto
AUTO1030 Signaalien käsittely
Harjoitus 6, tehtävät 3–4

12.3.2012

Konvoluutio

Konvoluutio lasketaan kaavalla:

(1)
$$y(n) = \sum_{m=-\infty}^{\infty} x(n-m)h(m) = \sum_{m=0}^{n} x(n-m)h(m) =$$
$$\sum_{m=0}^{n} h(m)x(n-m) = h(0)x(n) + h(1)x(n-1) + h(2)x(n-2) + ...$$

Kaavahan on sama kuin FIR-suotimen kaava:

(2)
$$y(n) = a_0 x(n) + a_1 x(n-1) + a_2 x(n-2) + ... = \sum_{i=0}^{n} a_0 x(n-i) \ .$$

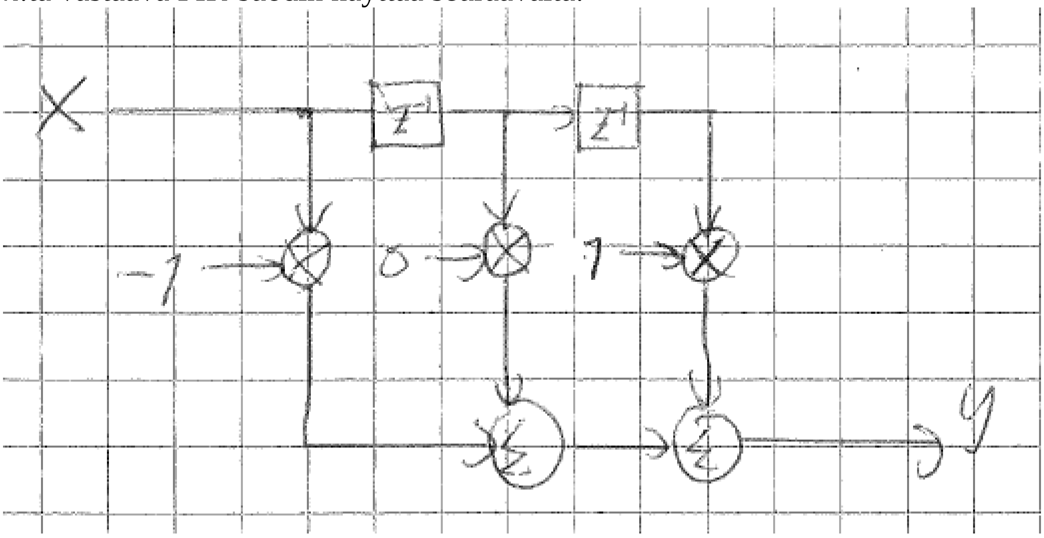
Kaavassa (1) aikasarja *h* vastaa FIR-suotimen kertoimia ja *x* suotimen syötesignaalia. Itse asiassa kaava ei ota kantaa, kumpi on kerroinvektori ja kumpi syötesignaali, konvoluutio on siis kommutatiivinen.

Ennen kuin ohjelmoidaan konvoluutio Javalla, käydään läpi esimerkki konvoluution laskemisesta.

Olkoon:

n	0	1	2	3	4	5	6	7	8	9
$x(n)$	5	5	6	7	8	8	8	6	4	2
$h(n)$	–1	0	1							

h:ta vastaava FIR-suodin näyttää seuraavalta:



Lasketaan ulostulo $y(n)$: liu'utetaan *käännettyä x*:ää *h*:n suhteen ja lasketaan kullakin siirtymällä tulojen summa. Näin saadaan konvoluutio:

n	0	1	2	3	4	5	6	7	8	9
$x(n)$	5	5	6	7	8	8	8	6	4	2

Janne Koljonen
Vaasa yliopisto
AUTO1030 Signaalien käsittely
Harjoitus 6, tehtävät 3–4

12.3.2012

$h(n)$ –1 0 1

$y(0)$: –1 · 5 = –5

–1 0 1

2 4 6 8 8 8 7 6 5 5

$y(1)$: –1 · 5 + 0 · 5 = –5

–1 0 1

2 4 6 8 8 8 7 6 5 5

$y(2)$: –1 · 6 + 0 · 5 + 1 · 5 = –1

–1 0 1

2 4 6 8 8 8 7 6 5 5

$y(3)$: –1 · 7 + 0 · 6 + 1 · 5 = –2

–1 0 1

2 4 6 8 8 8 7 6 5 5

$y(4)$: –1 · 8 + 0 · 7 + 1 · 6 = –2

–1 0 1

2 4 6 8 8 8 7 6 5 5

$y(5)$: –1 · 8 + 0 · 8 + 1 · 7 = –1

–1 0 1

2 4 6 8 8 8 7 6 5

$y(6)$: –1 · 8 + 0 · 8 + 1 · 8 = 0

–1 0 1

2 4 6 8 8 8 7 6

$y(7)$: –1 · 6 + 0 · 8 + 1 · 8 = 2

–1 0 1

2 4 6 8 8 8 7

$y(8)$: –1 · 4 + 0 · 6 + 1 · 8 = 4

–1 0 1

2 4 6 8 8 8

$y(9)$: –1 · 2 + 0 · 4 + 1 · 6 = 4

–1 0 1

2 4 6 8 8

$y(10)$: 0 · 2 + 1 · 4 = 4

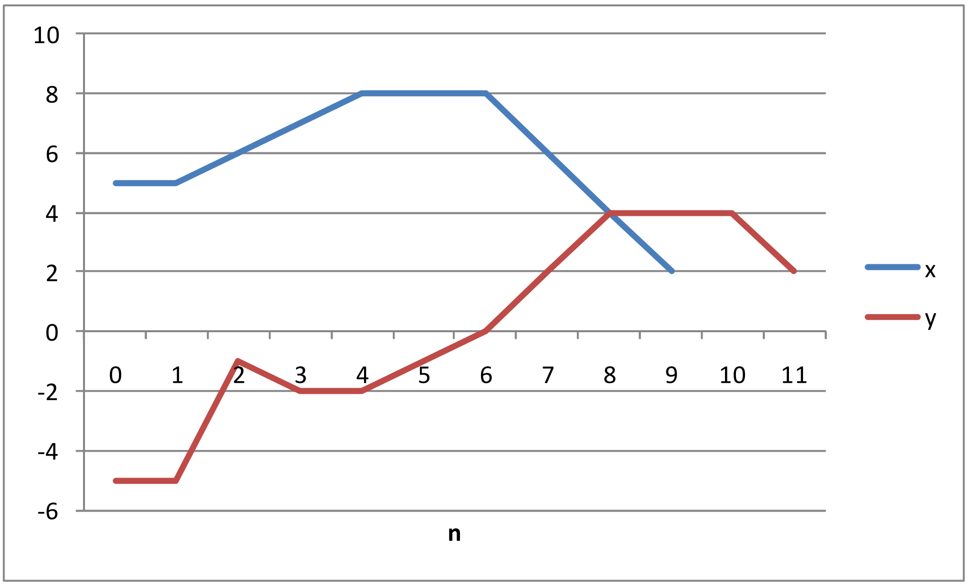
–1 0 1

2 4 6 8

$y(11)$: 1 · 2 = 2

–1 0 1

2 4 6



kuva. syöte ja FIR-suotimen ulostulo.

Kyseisen FIR-suotimen ulostulo on verrannollinen *x*:n derivaatan vastalukuun.

Sitten Javalla. Oleellista on taas huomata, että konvoluutiossa kullakin *n*:n arvolla lasketaan pistetulo *h*:n ja joidenkin *x*:n arvojen välillä.

```
/*****************************************************************************************
Janne Koljonen
Vaasan yliopisto
AUTO1030
9.3.2012
Konvoluution laskeminen
*****************************************************************************************/

import java.text.*;

public class Konvoluutio
{
	// Konvoluutio kaikilla n:n arvoilla
	public static double[] konvoluutio(double[] x1, double[] x2)
	{
		double[] tulos = new double[x1.length+x2.length-1];

		// Konvoluutio lasketaan samantapaisesti kuin korrelaatio, mutta
		// jompi kumpi aikasarjoista käännetään toisinpäin.
		// Mitään normalisointeja ei tosin tehdä.
		// Oletetaan nyt, että x2 vastaa FIR-suotimen kertoimia.

		// Apuvektori, joka vastaa FIR-suotimen siirtorekisteriä (z(-1) lohkot)
		double[] temp = new double[x2.length];

		// Konvoluution laskemisen voi toteuttaa monella tapaa. Tehdään nyt siten,
		// että oletetaan signaalin x1 olevan nolla ennen ensimmäistä ja
		// jälkeen viimeisen näytteen.
		// Käydään läpi tulos-muuttujan (ulostulo y) kaikki indeksit
		for (int n=0; n<tulos.length; n++)
		{
			// Siirretään näytteitä eteenpäin temp-vektorin näytteitä.
			// Lisätään uusi näyte kohdasta x1[n]
			for (int i=temp.length-1; i>0; i--)
			{
				temp[i] = temp[i-1];
			}

			if (n<x1.length)
```

```
			{
				temp[0] = x1[n];
			}
			else
			{
				temp[0] = 0;
			}

			// Sitten lasketaan pistetulo
			tulos[n] = pistetulo(temp, x2);
		}
		return tulos;
	}

	// Pistetulo olettaen, että x1 ja x2 ovat yhtä pitkiä
	private static double pistetulo(double[] x1, double[] x2)
	{
		double temp=0;
		for (int i=0; i<x1.length; i++)
		{
			temp += x1[i]*x2[i];
		}
		return temp;
	}


	/** double-taulukoiden tulostus **/
	public static void printVector(double[] v)
	{
		NumberFormat nf=NumberFormat.getInstance();
		nf.setMinimumFractionDigits(2);
		nf.setMaximumFractionDigits(2);
		for(int i=0;i<v.length;i++)
		{
			System.out.print(nf.format(v[i])+" ");
		}
		System.out.println();
	}

	/** double-taulukoiden tulostus **/
	public static void printVector(double[] v,String otsikko)
	{
		System.out.print(otsikko+": ");
		printVector(v);
	}

	// Testataan konvoluutiota (tehtävä 4)
	public static void main(String[] sArgs)
	{
		System.out.println("Testi alkaa");

		// Testi 1
		double[] x1 = {4.0, 4.0, 4.0, 0.0, 0.0, 0.0, -4.0, -4.0, -4.0, -4.0};
		double[] x2 = {1.0, 2.0, 4.0, 2.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0};

		double[] tulos1 = konvoluutio(x1, x2);
		printVector(tulos1, "Konvoluutio");
	}
}
```

Tulos: {4, 12, 28, 32, 28, 12, 0, –12, –28, –36, –36, –28, –12, –4, 0, 0, 0, 0, 0}.