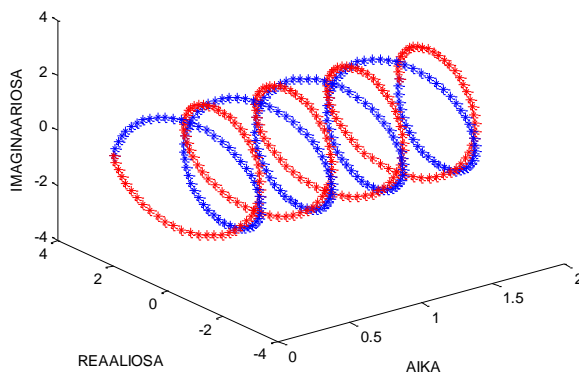


Octave-harjoitus: kompleksisen eksponenttifunktion visualisointi 3D-koordinaatistossa

Tehtävänanto: Toteuta Octave-ohjelma, joka muodostaa kompleksiset signaalit $x_1(t)$ ja $x_2(t)$ sekä esittää ne 3D-koordinaatistossa. Lopuksi ohjelma laskee signaalit yhteen ja esittää summasignaalin omassa ikkunassa 2D-koordinaatistossa.

$$x_1(t) = Ae^{i\omega t}$$

$$x_2(t) = Ae^{-i\omega t}, \quad \text{missä } \omega = 2\pi f$$



Tehtävän suoritusohje:

- 1) Tyhjennä muuttujat muistista clear-komennolla
- 2) Alusta muuttujat: signaalin ajalliseksi pituudeksi 2 s, näytepisteiden väliseksi ajaksi 0.01 s, taajuudeksi 2 Hz ja amplitudiksi 2. Nimeä muuttujat itse.
- 3) Luo aikavektori, joka sisältää tasavälein arvoja välillä 0...(signaalin ajallinen pituus). Käytä näytearvojen välisenä aikana kohdassa 2 määrittelemääsi lukua.
- 4) Alusta signaaleille $x_1(t)$ ja $x_2(t)$ nollavektorit komennolla `zeros`. Signaalin pituus (näytteiden lukumäärä) on sama kuin aikavektorin pituus (näytteiden lkm.)
- 5) Muodosta signaalit $x_1(t)$ ja $x_2(t)$
- 6) Nyt sinulla on kaksi kompleksista signaalia omissa vektoreissaan. Toisin sanoen signaalien arvot ovat kompleksilukuja.
- 7) Erotta seuraavaksi molempien signaalien arvoista reaalisosat ja imaginaariosat erillisiin vektoreihin. Tämän toimenpiteen jälkeen sinulla on neljä vektoria, kahdessa signaaliarvojen reaalisosat, kahdessa imaginaariosat. Käytä komentoja *real* ja *imag*.
- 8) Nyt voit siirtyä signaalien visualisointiin
- 9) Piirrä signaali $x_1(t)$ 3D-koordinaatiostoon piste kerrallaan.
`for i = 1:signaalin näytteiden lkm.`

`axis([0 2 -4 4 -4 4])` % Määritellään akseliston skaala: 0...2 x-akseli, -3...3 y-akseli, -3...3 z-akseli

`xlabel('AIKA')`

`ylabel('REAAIOSA')`

`zlabel('IMAGINAARIOSA')`

`plot3(t(i),re_x1(i),im_x1(i), '*')` % Yhden pisteen muodostamiseen tarvitaan 3 koordinaattia: aika, signaalin arvon reaalisosa ko- ajanhetkellä ja signaalin arvon imaginaariosa ko. ajanhetkellä

`hold on` % lukitaan kuvaaja seuraavaa silmukan suorituskierron varten

`pause(0.05)` % Määritetään silmukoiden suoritusten välinen viive/odotusaika

`end`

- 10) Toista proseduuri signaalille $x_2(t)$
- 11) Tallenna koodi ja kokeile sen toimivuutta
- 12) Muodosta summasignaali $x_1(t)$ ja $x_2(t)$ ja piirrä tulos omaan ikkunaan normaalilla plot-komennolla