Σήματα και Συστήματα

Εβδομάδα 2: Βασικά Σήματα (Fundamental Signals)

Ιωάννης Στεφανής

ΕΛΛΗΝΙΚΟ ΜΕΣΟΓΕΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ

Περιεχόμενα (Table of Contents)

Συνολική Διάρκεια: ~90 λεπτά

Σήματα (Signals): Παντού γύρω μας

Σήμα (signal) = ποσότητα που μεταβάλλεται στο χρόνο/χώρο και μεταφέρει πληροφορία (information).

Ήχος
Πίεση αέρα = f(t)

WiFi / 4G
Ηλεκτρομαγνητικό κύμα

ECG
Ηλ. δραστηριότητα καρδιάς

Αισθητήρες
Μέτρηση ανά ώρα = x[n]

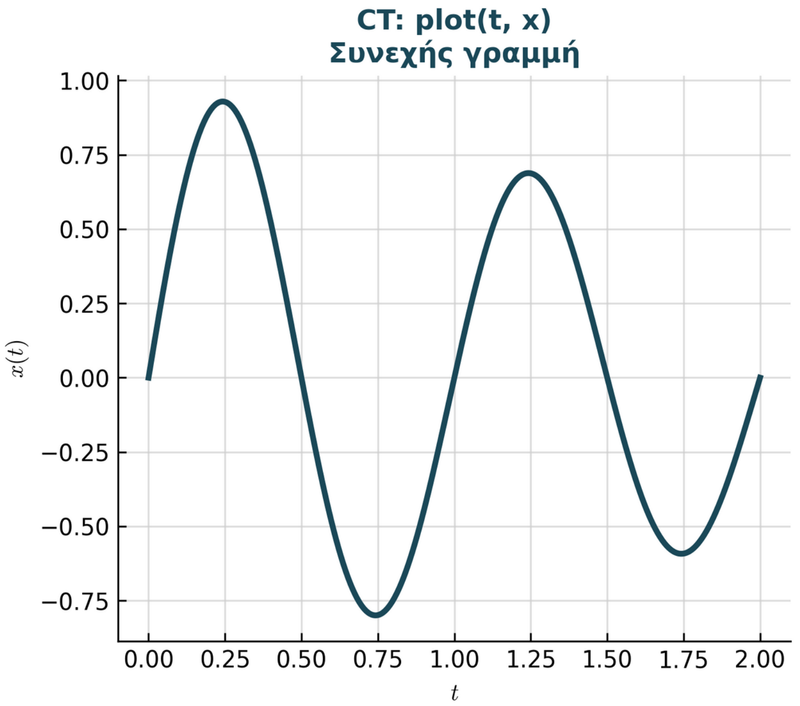
Χρηματιστήριο
Τιμή μετοχής = x[n]

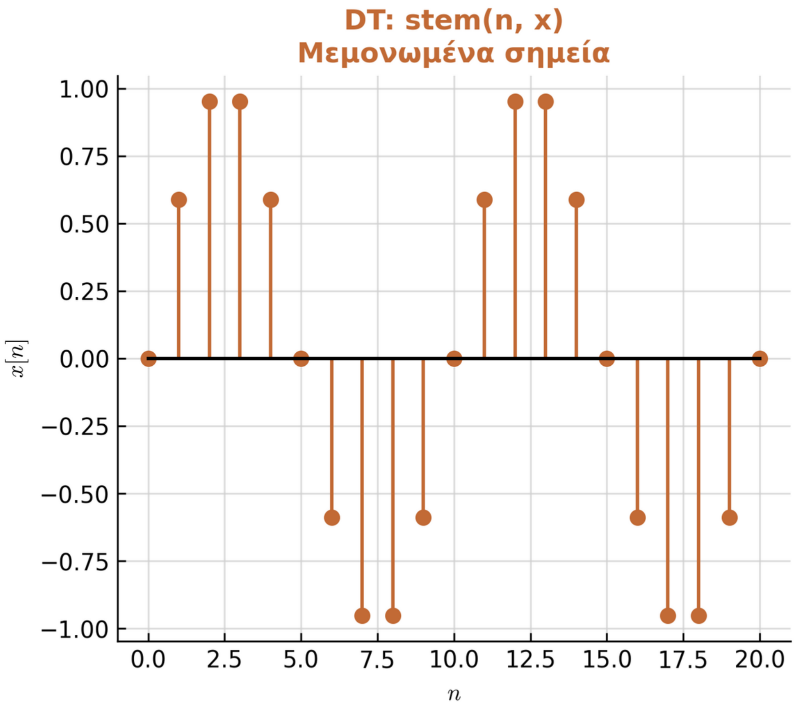
Εικόνα
Φωτεινότητα = f(x,y)

Στο εργαστήριο θα μάθουμε να αναλύουμε, σχεδιάζουμε και επεξεργαζόμαστε αυτά τα σήματα με GNU Octave. Κάθε σύνθετο σήμα χτίζεται από βασικά «δομικά στοιχεία» (building blocks).

Σήμα: ποσότητα που μεταβάλλεται και μεταφέρει πληροφορία. **Αυτή η εβδομάδα:** Μαθαίνουμε τα 8 βασικά σήματα = «LEGO» κάθε μηχανικού.

Συνεχούς (CT) vs Διακριτού (DT) Χρόνου





Δύο κόσμοι σημάτων:

CT — Continuous-Time: x(t)

Ορίζεται για κάθε t ∈ ℝ

Σχεδιάζεται: plot(t, x) → γραμμή

Παράδειγμα: μικρόφωνο, αισθητήρας

DT — Discrete-Time: x[n]

Ορίζεται μόνο σε n ∈ ℤ (ακέραιοι)

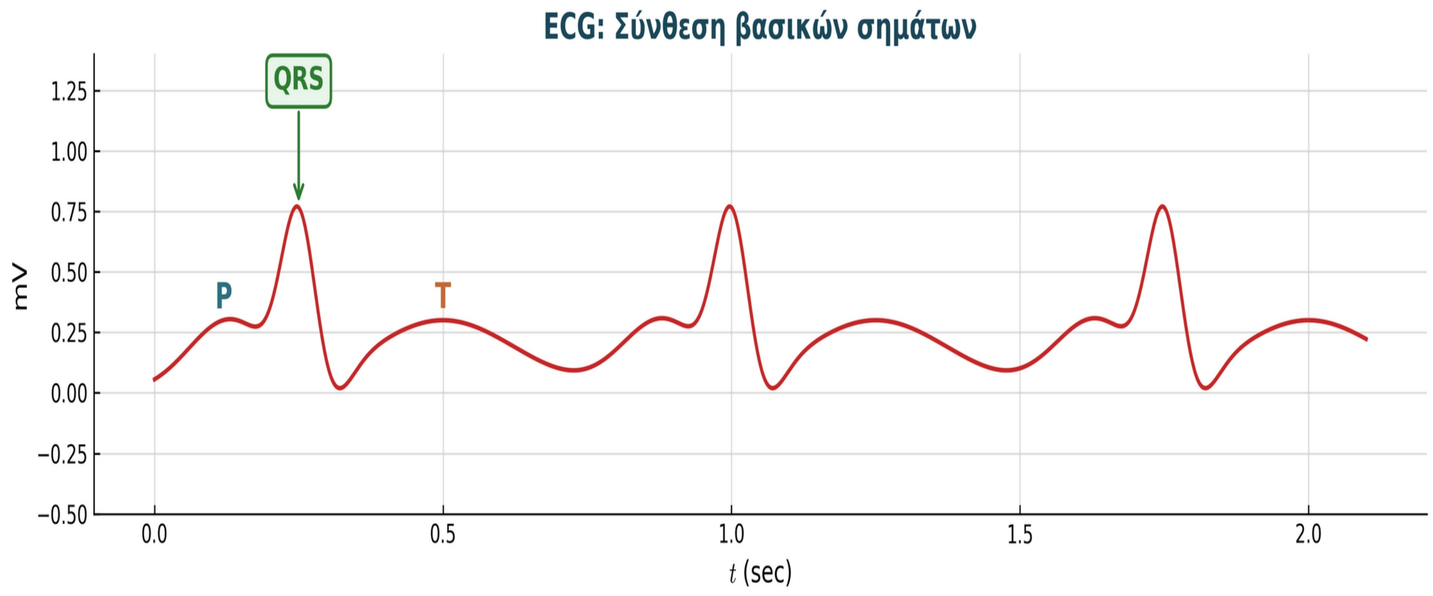
Σχεδιάζεται: stem(n, x) → σημεία

Παράδειγμα: CD ήχος, ψηφιακό βίντεο

Ο υπολογιστής δουλεύει ΜΟΝΟ με DT σήματα. Κάθε CT σήμα πρέπει πρώτα να μετατραπεί σε DT μέσω δειγματοληψίας (sampling): παίρνουμε "φωτογραφίες" σε ίσα χρονικά διαστήματα. Για σωστή αναπαράσταση, πρέπει αρκετά δείγματα ανά κύκλο (θεώρημα Nyquist, βλ. αργότερα).

Κανόνας: CT → plot() (γραμμή), DT → stem() (σημεία). ΠΟΤΕ plot σε DT: δείχνει ψεύτικη συνέχεια!

Βασικά Σήματα — Δομικά Στοιχεία Ανάλυσης



Κάθε σύνθετο σήμα χτίζεται
από απλά θεμελιώδη σήματα:

Το ECG αναλύεται σε κυματομορφές:

P = συστολή κόλπων (αργό κύμα)

QRS = συστολή κοιλιών (οξύ spike)

T = επαναπόλωση (αποκατάσταση)

→ Ένα σήμα, πολλές λειτουργίες μαζί!

Ήχος = άθροισμα ημιτόνων

Εικόνα = 2D Fourier σειρά

Τα 8 βασικά σήματα είναι τα
«LEGO» κάθε μηχανικού.
Αυτή η εβδομάδα τα μαθαίνουμε
και τα χτίζουμε στο Octave.

Στις επόμενες εβδομάδες θα δούμε πώς τα συνδυάζουμε: πράξεις, συνέλιξη, μετασχηματισμοί Fourier.
8 θεμελιώδη σήματα → κάθε κυματομορφή (όπως τα βασικά χρώματα → κάθε απόχρωση).

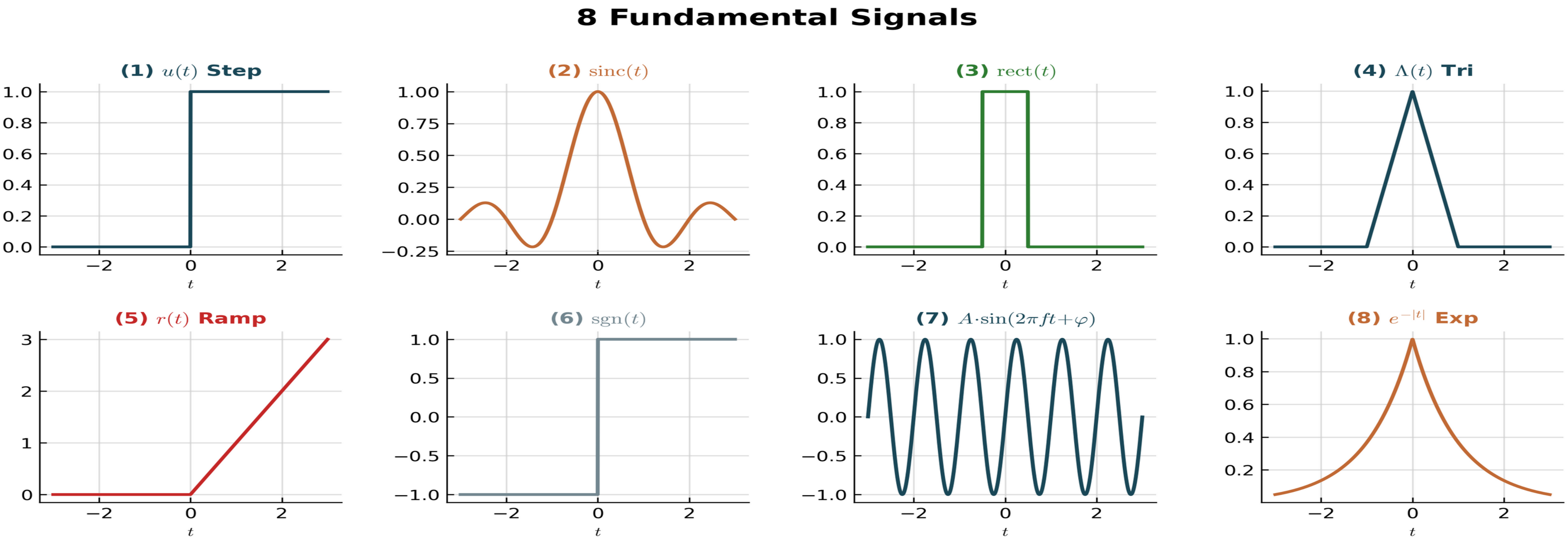
Ανάλυση vs Σύνθεση: Το ECG δεν ΚΑΤΑΣΚΕΥΑΖΕΤΑΙ από Gaussians — ΑΝΑΛΥΕΤΑΙ σε τμήματα που μοιάζουν με Gaussian. Ο Fourier (εβδ.5) κάνει αυτή την ανάλυση επίσημα. **Σειρά:** βασικά σήματα → εβδ.3: πράξεις → εβδ.4: συνέλιξη → εβδ.5+: Fourier.

Τι θα μάθουμε σήμερα

Βασικά Σήματα	u(t), δ(t), r(t), rect, sinc, sgn, Λ	25'
Εκθετικά & Ημιτονοειδή	Ae^(σt), Asin(2πft+φ), T=1/f	15'
Ενέργεια & Ισχύς	E, P, RMS, dB	10'
Μιγαδική Εκθετική	Euler formula, phasors	10'
Programming	Logical indexing, for, vectorization	15'
Ασκήσεις	Λυμένη + 5 εργαστηριακές (.m)	15'

Πριν ξεκινήσουμε: Πάντα τρέχουμε pkg load signal στην αρχή κάθε session. **Σχέδιο:** ~90 λεπτά. 6 ενότητες.

Κατάλογος Βασικών Σημάτων (Fundamental Signals)

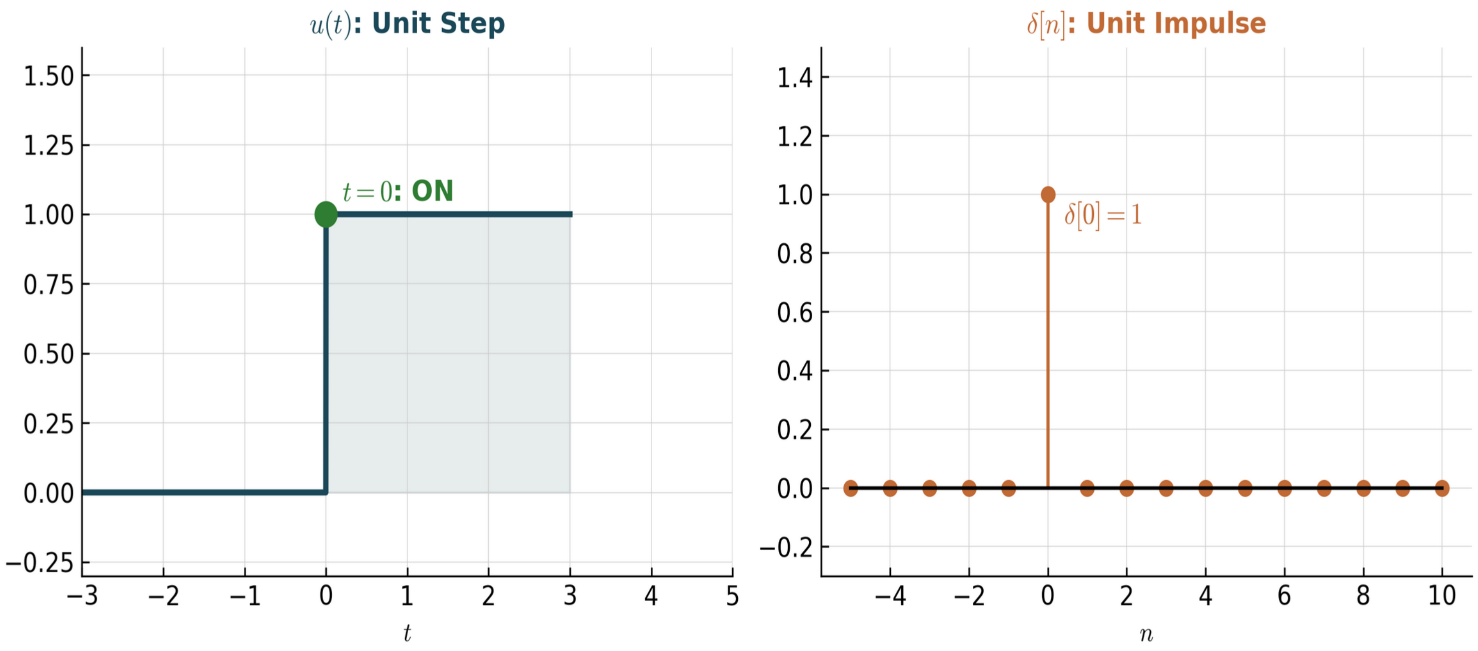


(1) u(t) Step (2) sinc(t) (3) rect(t) (4) Λ(t) Tri

(5) r(t) Ramp (6) sgn(t) Πρόσημο (7) A·sin(2πft+φ) Ημίτονο (8) e^(−|t|) Εκθετικό

8 θεμελιώδη σήματα: Αριθμημένα (1)-(8) + δ(t) Κρουστική (ειδική — δεν σχεδιάζεται, βλ. επόμενο slide). **Στόχος:** Κατασκευή και ανάλυση κάθε σήματος στο Octave.

(1) Μοναδιαία Βηματική u(t) (Unit Step Function)




```
t = linspace(-3, 5, 2000);  u = (t >= 0);
subplot(1,2,1)                      % αριστερά: CT step
  plot(t, u, 'b', 'LineWidth', 3);
  plot(0,1,'go','MarkerFaceColor','g','MarkerSize',10)
  title('u(t): Unit Step');
  xlabel('t'); ylim([-0.3 1.5]); grid on
subplot(1,2,2)                      % δεξιά: DT impulse
  n = -5:10;  dn = (n == 0);
  stem(n, dn, 'filled');            % DT → stem()!
  title('delta[n]: Unit Impulse');
  xlabel('n'); grid on              % ▶ s08_unit_step_impulse.m
```

u(t) = Μονόπλευρο παράθυρο:

t < 0: τιμή 0 (OFF)

t ≥ 0: τιμή 1 (ON)

Σημείο κλειδί: t=0 → πράσινο ●

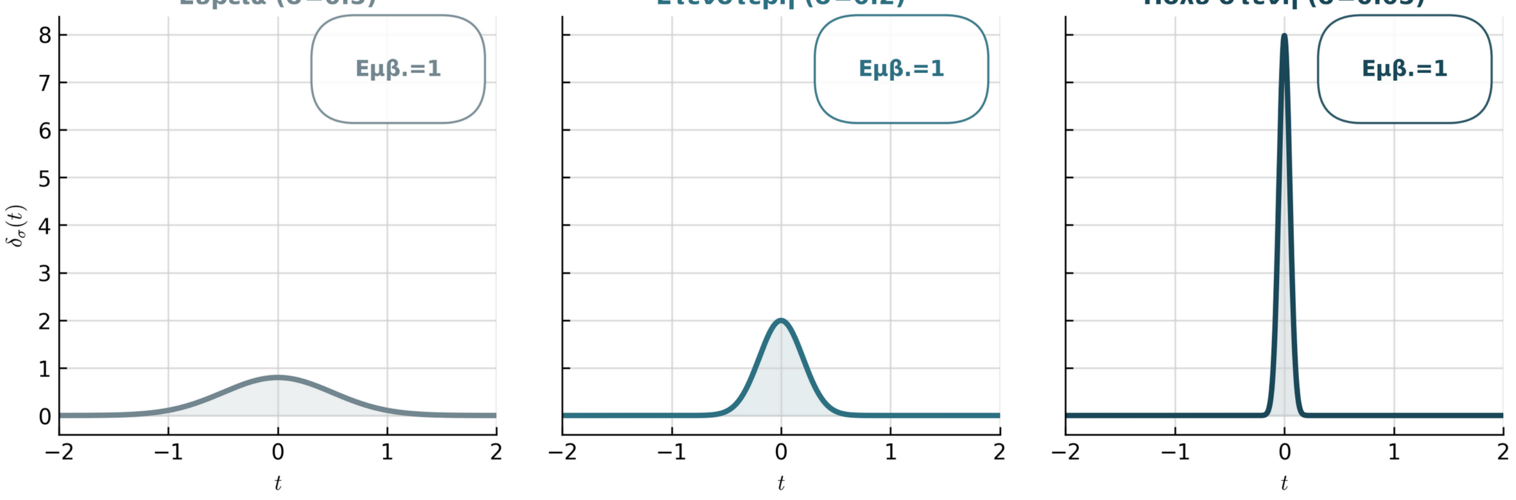
Χρήση: ενεργοποίηση κυκλωμάτων, αιτιατά (causal) συστήματα.

Octave: (t >= 0) → λογικός πίνακας

0 ή 1 σε κάθε θέση

Τι: Η u(t) «ανοίγει» στο t=0. Αριστερά plot(CT), δεξιά stem(DT). **Κώδικας=Plot:** Κάθε εντολή → αντίστοιχο γράφημα.

(2) Κρουστική δ(t): Προσεγγίσεις (Unit Impulse Approximations)



Τι είναι η δ(t);

ΔΕΝ είναι κανονική συνάρτηση!
Είναι κατανομή (distribution):
= το «όριο» μιας Gaussian
που στενεύει ενώ κρατάει
εμβαδόν = 1 ΠΑΝΤΑ.

Γιατί μας είναι χρήσιμη;

Εξάγει ΜΙΑ τιμή σήματος:
$\int x(t)\cdot\delta(t-t_0)\, dt = x(t_0)$
= Ιδιότητα Επιλογής (sifting)

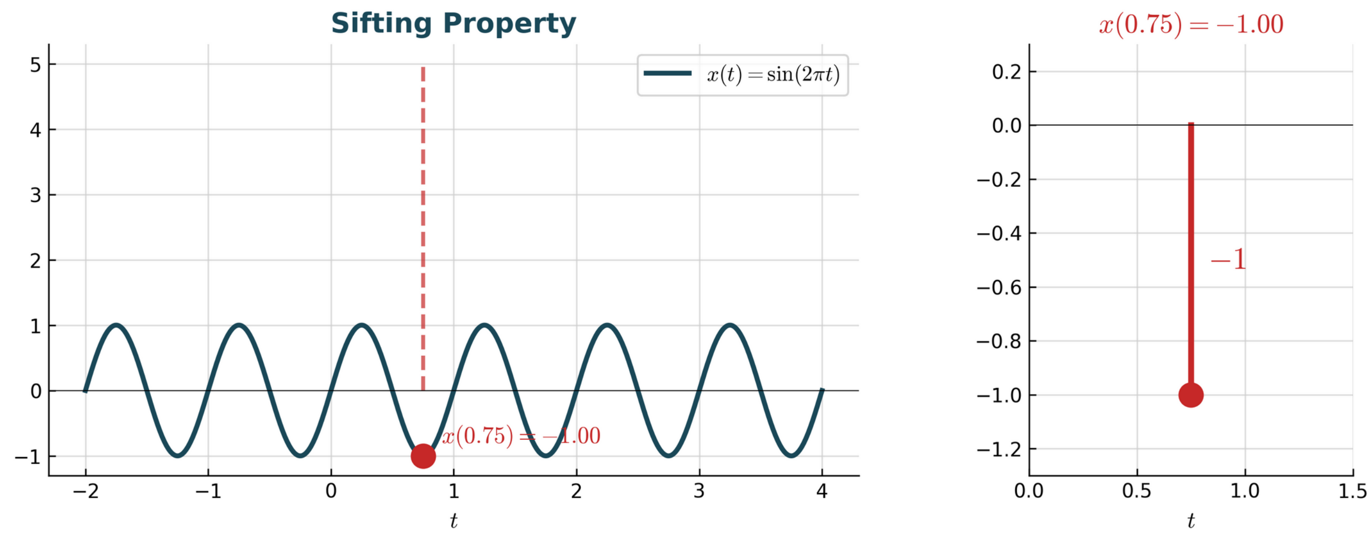
Στα plots:

Gaussian στενεύει (σ→0):
σ=0.5 → σ=0.2 → σ=0.05

```
t = linspace(-2, 2, 20001); dt = t(2)-t(1);
sigmas = [0.5  0.2  0.05];  % πλατιά→στενή
for k = 1:3
  sigma = sigmas(k);
  d = (1/(sigma*sqrt(2*pi)))*exp(-t.^2/(2*sigma^2));
  A = sum(d)*dt;              % εμβαδόν = 1 πάντα
  subplot(1,3,k)
    plot(t, d, 'LineWidth', 3); grid on
    title(sprintf('s=%g Emb=%.3f', sigma, A))
end                           % ► s09_gaussian_delta_approx.m
```

Κλειδί: δ(t) = όριο Gaussian που στενεύει. Εμβαδόν=1 πάντα. Στο όριο σ→0: «ψαρεύει» μία μόνο τιμή. **Πρακτικά:** Κρουστική απόκριση: τροφοδοτούμε σύστημα με δ(t) και βλέπουμε πώς αντιδρά.

δ(t): Ιδιότητες & Επιλογή (Sifting Property)



Βασικές Ιδιότητες:

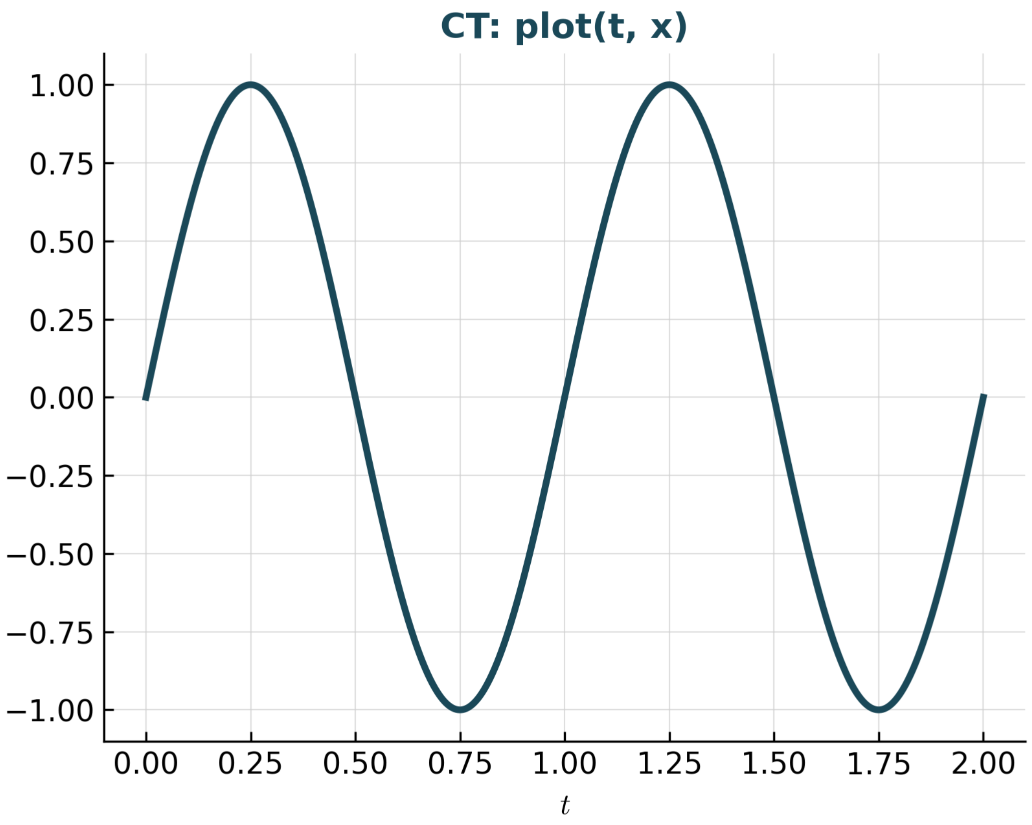
① $\int \delta(t)\,dt = 1 \rightarrow$ Εμβαδόν = 1
② $\delta(t) = 0$ για $t \neq 0$
③ $x(t)\cdot\delta(t-t_0) = x(t_0)\cdot\delta(t-t_0)$
→ **Ιδιότητα Επιλογής (Sifting)**
④ $\delta(at) = (1/|a|)\cdot\delta(t)$
⑤ $\delta(t) = d/dt\ u(t)$

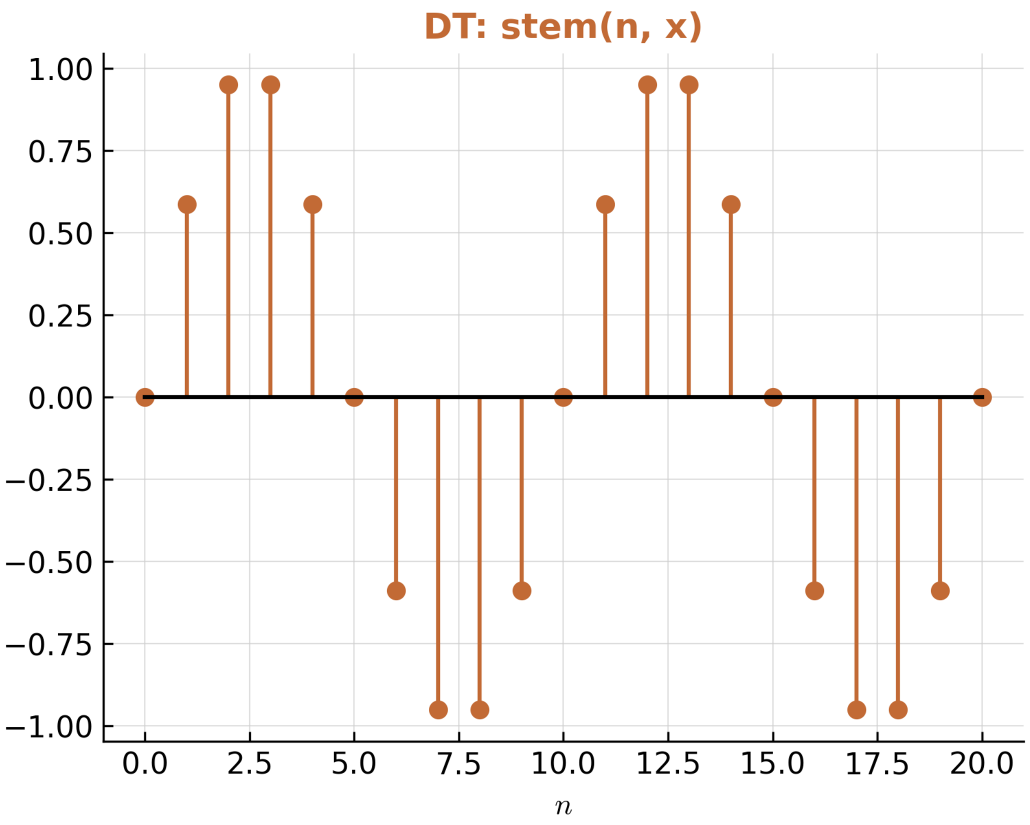
$\int x(t)\cdot\delta(t-t_0)dt = x(t_0)$ — «Ψαρεύει» ακριβώς μία τιμή.
Παράδ: t_0=0.75, x=sin(2πt) → x(0.75)=sin(1.5π)=−1.000
RC κύκλωμα: h(t)=(1/RC)·e^(−t/τ)·u(t) [κρουστική απόκριση]

```
t = linspace(-5,5,10000); dt = t(2)-t(1);
x = sin(2*pi*t); t0 = 0.75;
d = (abs(t-t0)<dt/2)/dt; % στενή δ στο t0
result = sum(x.*d)*dt;   % ≈ sin(1.5π)=-1
subplot(1,2,1); plot(t,x,'b','LineWidth',2); hold on; grid on
plot([t0 t0],[-1.5 5],'r--','LineWidth',2)
title('x(t)+δ στο t0'); xlabel('t')
subplot(1,2,2); bar(t0,result,0.3,'r'); grid on
title(sprintf('x(%.2f)=%.2f',t0,result))
```

Sifting: $\int x(t)\cdot\delta(t-t_0)dt = x(t_0)$. «Ψαρεύει» ακριβώς μία τιμή. **Κώδικας:** sin(2π×0.75) = sin(1.5π) = −1. Αριθμητικά: ≈−1.0000.

plot() vs stem(): Κανόνας Σχεδίασης



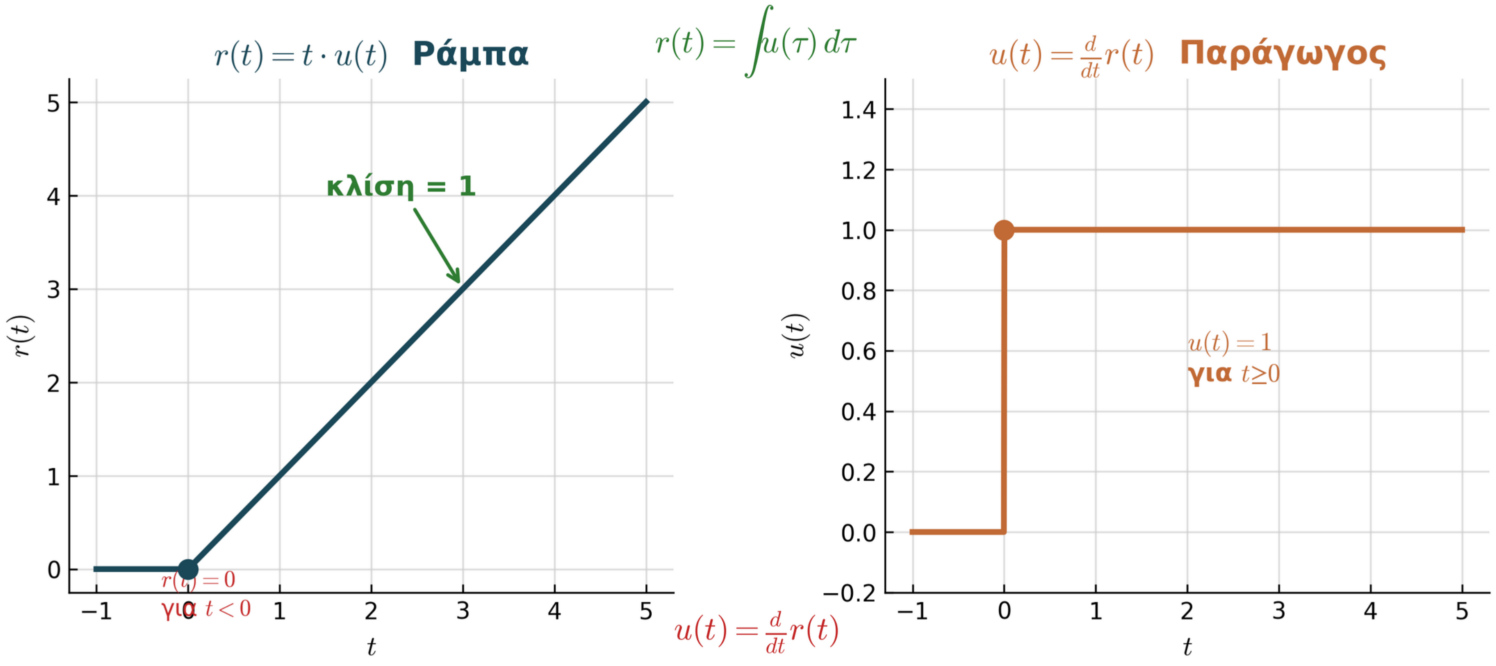



```
% CT σήμα → plot()
t=linspace(0,2,500); x_ct=sin(2*pi*t);
plot(t,x_ct,'b','LineWidth',3)
title('CT: plot()'); grid on
```

```
% DT σήμα → stem()  [ΠΟΤΕ plot!]
n=0:20; x_dt=sin(2*pi*n/10);
stem(n,x_dt,'filled')
title('DT: stem()'); grid on
```

Κανόνας: x(t) → plot() (γραμμή), x[n] → stem() (σημεία). ΠΟΤΕ plot σε DT, ΠΟΤΕ stem σε CT! **Γιατί:** plot σε DT «ενώνει» σημεία → ψεύτικη συνέχεια!

(3) Ράμπα r(t) = t·u(t) (Ramp Function)



r(t) = ∫u(τ)dτ → ολοκλήρωμα | u(t) = d/dt r(t) → παράγωγος

```
t=linspace(-1,5,2000); u=(t>=0);
r=t.*u;   % r(t)=t·u(t)
subplot(1,2,1)
  plot(t,r,'b','LineWidth',4); title('r(t)'); grid on
subplot(1,2,2)
  plot(t,u,'r','LineWidth',4); title('u=dr/dt'); grid on
```

Ορισμός:

r(t) = t·u(t)

= 0 για t<0, = t για t≥0

Σχέσεις (αριστ ↔ δεξί plot):

r(t) = ∫ u(τ) dτ (ολοκλήρωμα)

u(t) = d/dt r(t) (παράγωγος)

Ιδιότητες:

Κλίση = 1 (σταθερή)

Αρχίζει στο t=0 (η u(t) ενεργοποιεί)

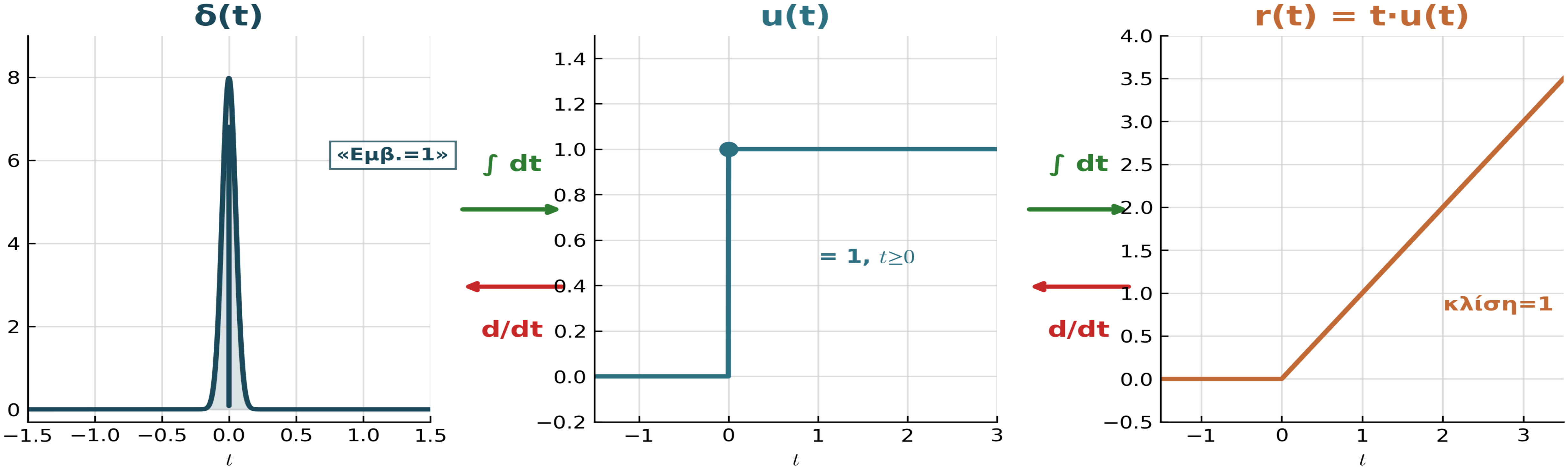
Πρακτικά:

Γραμμικά αυξανόμενη τάση/θέση/χρόνος

π.χ. σταθερή ταχύτητα → θέση = ράμπα

Αριστερά: r(t)=t·u(t) — γραμμικά αυξανόμενο μετά t=0, κλίση=1. **Δεξιά:** u(t) = παράγωγος r(t). Ράμπα ↔ step μέσω ολοκλήρωσης/παραγώγισης (ίδια σχέση με σ13).

Σχέσεις: δ(t) ↔ u(t) ↔ r(t)

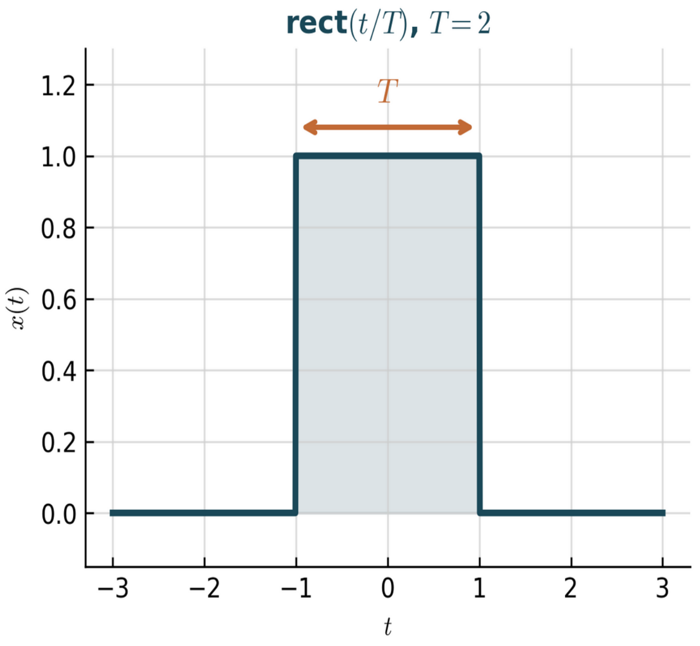


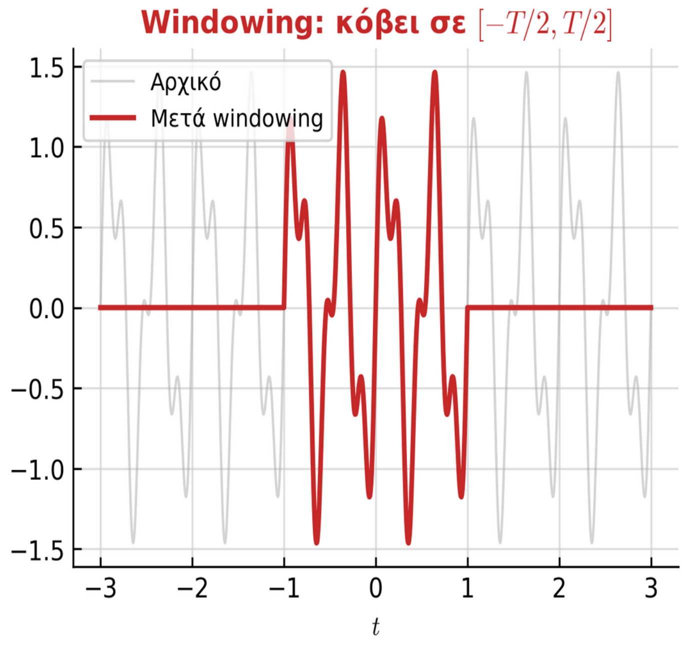
Ολοκλήρωση (Integration): δ → u → r. Κάθε βήμα συσσωρεύει εμβαδόν.
Παραγώγιση (Differentiation): r → u → δ. Κάθε βήμα δίνει ρυθμό αλλαγής.

Η δ(t) ΔΕΝ έχει πλάτος (τεχνικά Δ→0). Η u(t) έχει ύψος=1 (σταθερό). Η r(t) έχει κλίση=1.
Αναλογία: Θέση ↔ Ταχύτητα ↔ Επιτάχυνση (ίδιος μηχανισμός: παραγώγιση/ολοκλήρωση).

Αναλογία: Θέση→ταχύτητα→επιτάχυνση: η παραγώγιση δίνει ρυθμό αλλαγής. **Ερώτηση:** Αν r(t)=2t·u(t), ποια η παράγωγος; Απ: 2·u(t).

(4) Ορθογώνιος Παλμός rect(t/T) (Rectangular Pulse)






```
t = linspace(-3, 3, 1000);  T = 2;
rect_sig = (abs(t) <= T/2);
subplot(1,2,1);              % αριστερά: rect
plot(t, rect_sig, 'b', 'LineWidth', 2);
title('rect(t/T)'); grid on
% Παράθυρο (Windowing):
x = sin(2*pi*2*t) + 0.5*sin(2*pi*5*t);
y = x .* rect_sig;
subplot(1,2,2);              % δεξιά: windowed
plot(t, y, 'r', 'LineWidth', 2);
title('x · rect'); grid on
```

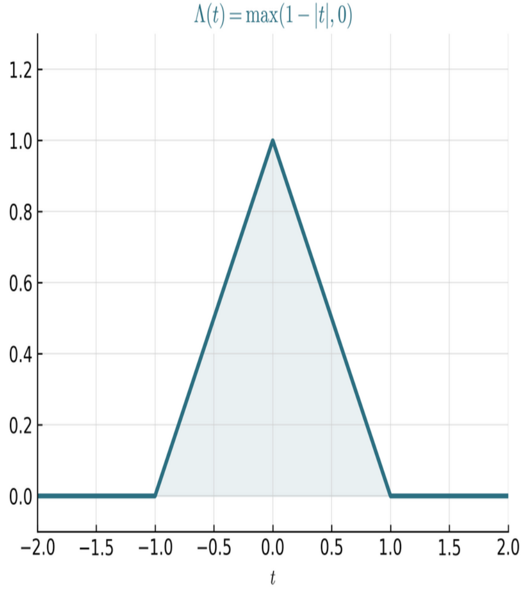
rect = αμφίπλευρο παράθυρο: 1 εντός [−T/2, T/2], 0 εκτός.

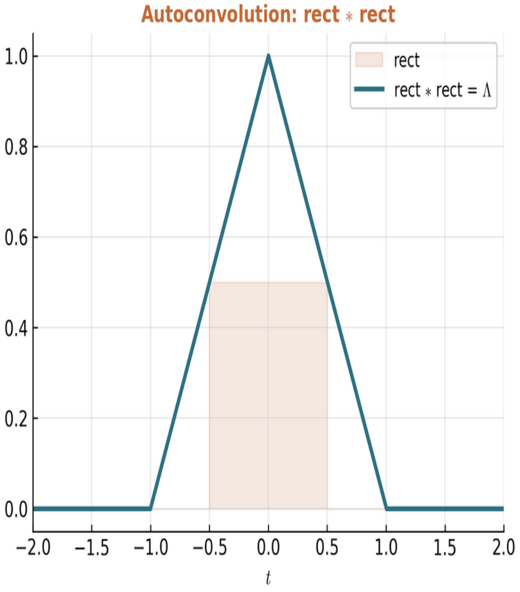
Αριστερά: καθαρός rect. Δεξιά: x·rect = κόβει χρονικό τμήμα (windowing).

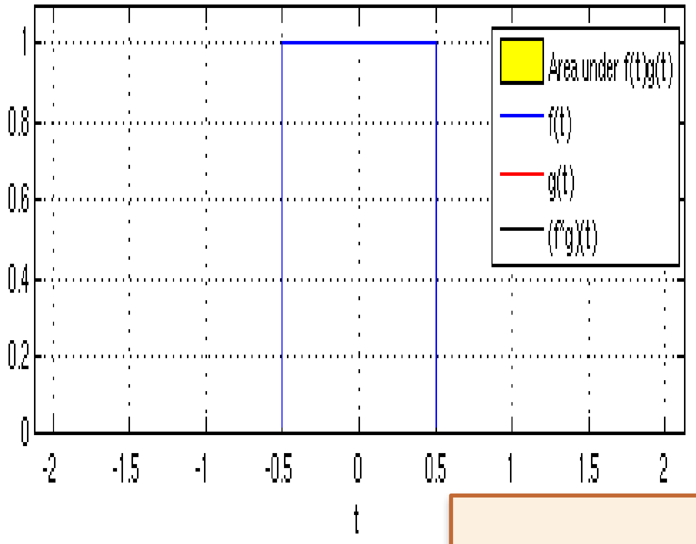
Σχέση: rect(t) = u(t+½)−u(t−½).

Windowing: x·rect = κρατάμε ΜΟΝΟ [−T/2, T/2], μηδενίζουμε τα υπόλοιπα. **Πρακτικά:** Παλμογράφος, μικρόφωνο, ADC: καταγράφουν σήμα σε πεπερασμένο χρονικό διάστημα — αυτό ΕΙΝΑΙ windowing!

(5) Τριγωνικός Παλμός Λ(t) (Triangular Pulse)







Ορισμός:

Λ(t) = max(1−|t|, 0)

Κορυφή = 1 στο t=0

Βάση = [−1, 1]

Τι είναι αυτοσυνέλιξη;

rect ∗ rect = Λ

«Σύρε-και-πολλαπλάσιασε»:

rect πάνω σε αντίγραφο → τρίγωνο!

(Αναλυτικά στην Εβδ. 4: Συνέλιξη)

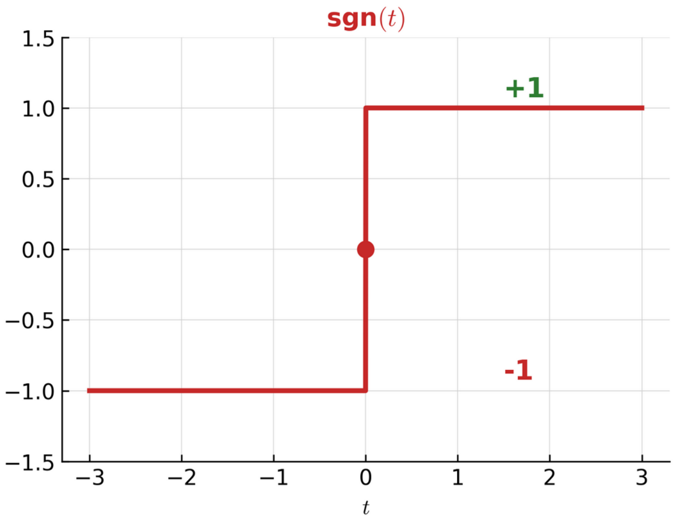
Χρήσεις:

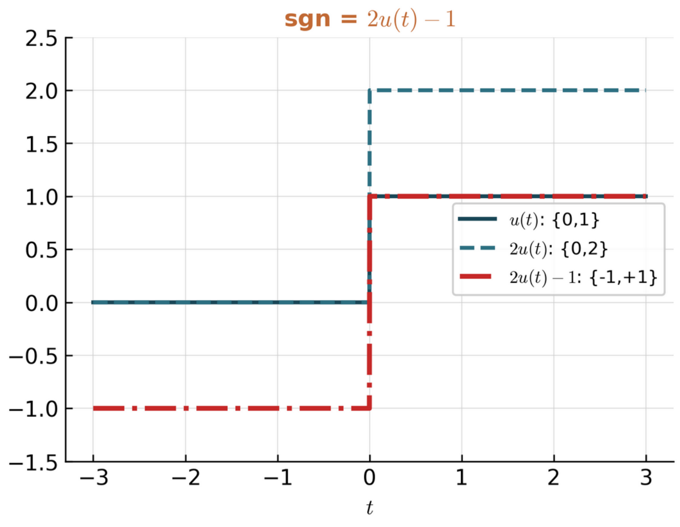
Ομαλή μετάβαση (fade-in/out)

Γραμμική παρεμβολή (interpolation)

Τι: Λ = «ομαλός» rect. Αυτοσυνέλιξη rect∗rect = τρίγωνο. **Animation:** Το GIF δείχνει τη συνέλιξη βήμα-βήμα — σε PowerPoint παίζει κατά τη παρουσίαση.

(6) Συνάρτηση Πρόσημου sgn(t) (Sign Function)






```
t=linspace(-3,3,2000); u=(t>=0);
subplot(1,2,1)  % sgn(t)
  plot(t,sign(t),'r','LineWidth',4)
  ylim([-1.5 1.5]); title('sgn(t)'); grid on
subplot(1,2,2)  % σχέση sgn=2u-1
  plot(t,u,'LineWidth',3); hold on; grid on
  plot(t,2*u,'--','LineWidth',3)
  plot(t,2*u-1,'r','LineWidth',4)
  title('sgn=2u-1'); legend('u','2u','2u-1')
```

Ορισμός sgn(t):

+1 αν t > 0

0 αν t = 0

−1 αν t < 0

Σχέση: sgn(t) = 2u(t) − 1

Βήμα-βήμα:

u(t) δίνει: 0 ή 1

2·u(t) δίνει: 0 ή 2

2u(t)−1 δίνει: −1 ή +1 ●

Μετατρέπει {0,1} → {−1,+1}!

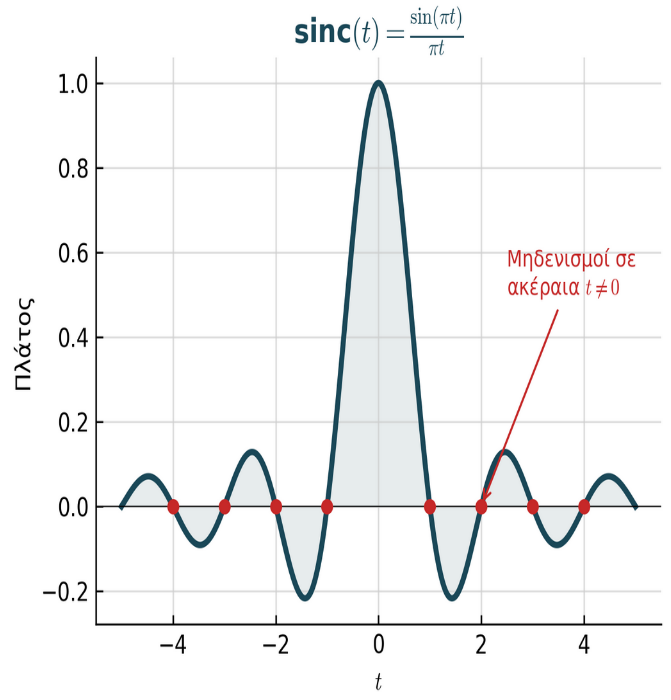
Χρήσεις (ELI5):

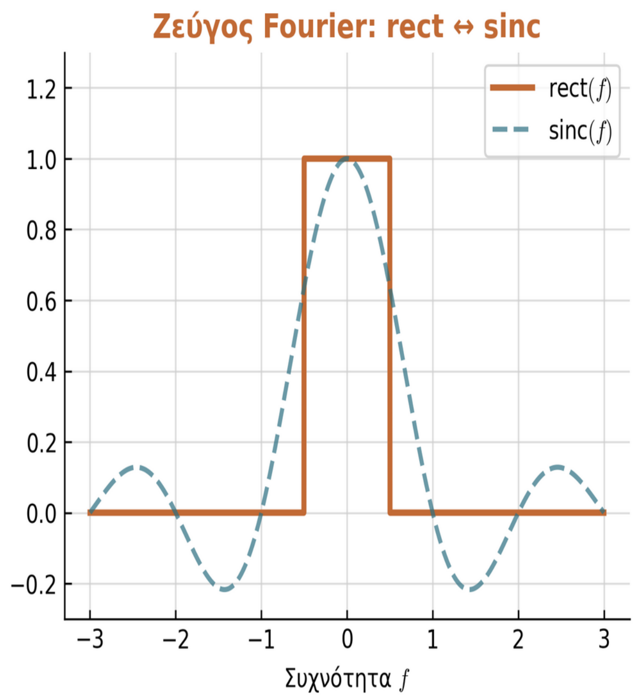
DC offset removal: σήμα 0~5V →
sgn→ {−1,+1} (κεντράρει)

Συγκριτής: x>0→+V, x<0→−V
(π.χ. ±12V ενισχυτής ήχου)

Τι: sgn = μόνο πρόσημο, χωρίς μέγεθος. 2u−1 μετατρέπει {0,1}→{−1,+1}. **Πρακτικά:** u(t)=ON/OFF, sgn(t)=πολικότητα +/−.

(7) Η Συνάρτηση sinc (Sinc Function)






```
subplot(1,2,1)              % αριστερά: sinc(t)
  t = linspace(-5,5,4000);
  plot(t, sinc(t), 'b', 'LineWidth', 4); hold on; grid on
  plot(-4:4, zeros(9,1), 'ro', 'MarkerSize', 8,'MarkerFaceColor','r')
  plot(0, 1, 'go', 'MarkerSize', 12,'MarkerFaceColor','g')
  title('sinc(t)=sin(pi*t)/(pi*t)'); xlabel('t')
subplot(1,2,2)              % δεξιά: rect↔sinc Fourier
  f = linspace(-3,3,2000);
  plot(f, abs(f)<=0.5, 'r', 'LineWidth', 4); hold on; grid on
  plot(f, sinc(f), 'b--', 'LineWidth', 3)
  legend('rect(f)','sinc(f)'); title('Ζεύγος Fourier') % ▶ s17.m
```

sinc(t) = sin(πt)/(πt)

Βασικές ιδιότητες:

sinc(0) = 1 (πράσινο ●)

sinc(n) = 0, n = ±1,±2,... (κόκκινα ●)

Κύριος λοβός (main lobe):

Πλάτος = 2 (από −1 ως +1)

Περιέχει ~90% της ενέργειας (∫sinc²)

Γιατί θεμελιώδες:

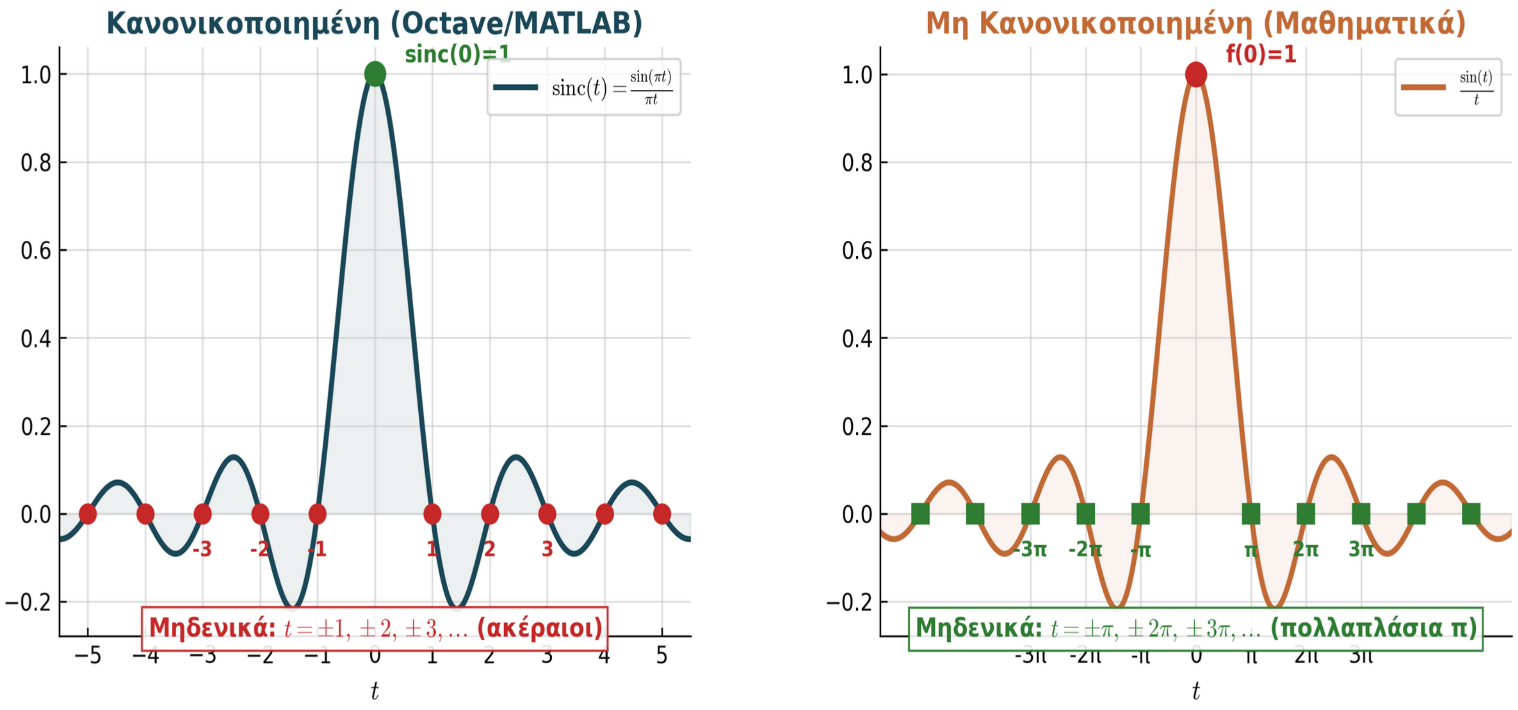
① rect ↔ sinc (ζεύγος Fourier)

② Ιδανική ανακατασκευή (Nyquist)

③ Ιδανικό low-pass φίλτρο

Αριστερά: sinc(t)=sin(πt)/(πt). Κεντρικός λοβός (main lobe) = μεγάλη κορυφή. Πλευρικοί λοβοί σβήνουν. Κόκκινα: μηδενισμοί σε κάθε ακέραιο. **Δεξιά:** Ζεύγος Fourier: rect(t) ↔ sinc(f). Παλμός στο χρόνο → sinc στη συχνότητα. Αυτό είναι η βάση φίλτρων & δειγματοληψίας (Εβδ.5+).

sinc: Κανονικοποιημένη vs Μη Κανονικοποιημένη



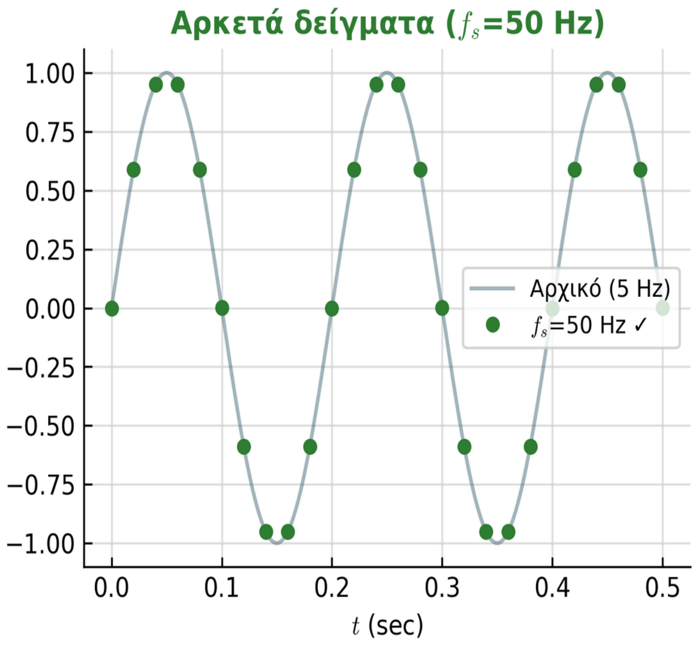

```
t = linspace(-6, 6, 2000);
subplot(1,2,1);                % κανονικοποιημένη
y_norm = sinc(t);              % zeros: ±1, ±2
plot(t, y_norm, 'b', 'LineWidth', 2); hold on
plot(-5:5, zeros(11,1), 'ro'); grid on
title('sinc(t)=sin(πt)/(πt)');
subplot(1,2,2);                % μη κανονικοποιημένη
t2 = linspace(-20, 20, 4000);
y_un = sin(t2)./(t2+eps);      % zeros: ±π, ±2π
plot(t2, y_un, 'Color', [.76 .42 .21]); grid on
title('sin(t)/t');
```

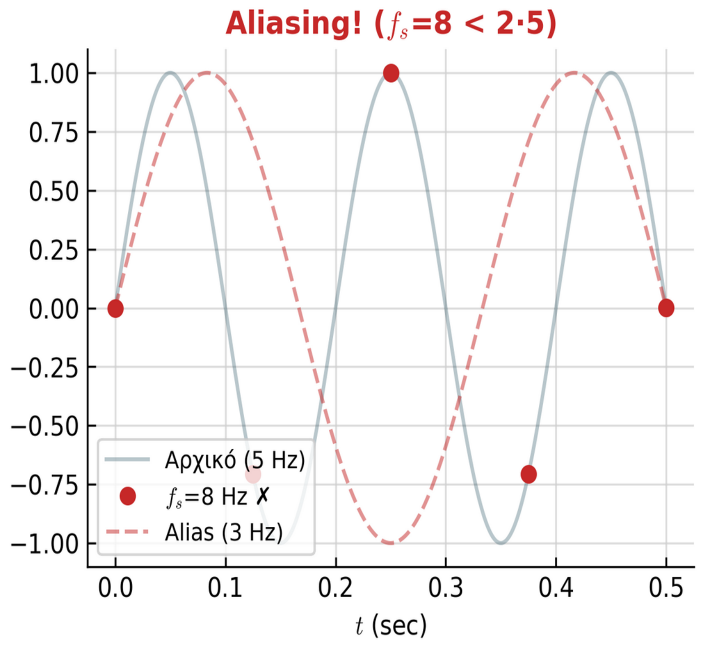
Κανονικοποιημένη sinc(t)=sin(πt)/(πt): μηδενίζεται στο t = ±1, ±2, ±3, ... (ΑΚΕΡΑΙΟΙ).
Μη κανονικοποιημένη sin(t)/t: μηδενίζεται στο t = ±π, ±2π, ±3π, ... (ΠΟΛΛΑΠΛΑΣΙΑ του π).

Octave/MATLAB: sinc(1)=0, sinc(2)=0, sinc(0.5)≠0. Βιβλία: sin(π)/π=0, sin(2π)/(2π)=0.
ΠΡΟΣΟΧΗ: Ελέγξτε πάντα ποια εκδοχή χρησιμοποιεί το βιβλίο σας!

Πρακτική σημασία: Η κανονικοποιημένη sinc(t)=sin(πt)/(πt) έχει zeros ΣΕ ΑΚΕΡΑΙΟΥΣ → ταιριάζει ακριβώς με δειγματοληψία (1 δείγμα/sec). Γι αυτό χρησιμοποιείται στην ψηφιακή επεξεργασία. **Προσοχή:** Πάντα ελέγξτε ποια εκδοχή χρησιμοποιεί το βιβλίο σας!

Δειγματοληψία & Nyquist (Sampling Theorem)






```
f_sig = 5;                          % σήμα 5 Hz
t = 0:0.001:0.5;                    % αναλυτικός χρόνος
x = sin(2*pi*f_sig*t);              % αρχικό σήμα

% Αρκετά δείγματα (fs=50Hz):
ts_good = 0:1/50:0.5;               % 50 δείγμ./sec
plot(t, x, 'b'); hold on
plot(ts_good, sin(2*pi*5*ts_good), 'go', 'MarkerSize', 5);

% Λίγα δείγματα (fs=8Hz) — ALIASING!
ts_bad = 0:1/8:0.5;
plot(ts_bad, sin(2*pi*5*ts_bad), 'ro', 'MarkerSize', 7);
```

Θεώρημα Nyquist-Shannon: fs ≥ 2·fmax (αλλιώς Aliasing!)

Αριστερά: fs=50Hz για σήμα 5Hz → αρκετά (πράσινα ●) → σωστή ανακατασκευή.
Δεξιά: fs=8Hz → ΌΧΙ αρκετά (κόκκινα ●) → η ανακατασκευή αποτυγχάνει (aliasing).

Πρακτικός κανόνας: CD ήχος → fmax=20kHz → fs=44.1kHz

Κανόνας: fs ≥ 2·fmax. Αν δειγματοληπτείς πολύ αραιά, χάνεις πληροφορία! **Πρακτικό:** CD: 44.1kHz, τηλέφωνο: 8kHz (μόνο φωνή).

Αρχή Αβεβαιότητας Χρόνου-Συχνότητας

«Κλικ» (σύντομο, σαν δ(t))

Δt μικρό → Δf ΜΕΓΑΛΟ
Ακούγεται σε ΟΛΕΣ τις συχνότητες
Ξέρεις ΠΟΤΕ, ΟΧΙ τι συχνότητα

Παράδειγμα: Ραντάρ με σύντομο παλμό
→ καλή θέση, κακή ταχύτητα

«Νότα» (μακρύ ημίτονο, π.χ. Λα=440Hz)

Δt ΜΕΓΑΛΟ → Δf μικρό
Ακούγεται ΜΙΑ καθαρή νότα
Ξέρεις ΤΙ συχνότητα, ΟΧΙ πότε

Παράδειγμα: Μακρύ beep
→ σαφής νότα, αβέβαιο «πότε»

Δt · Δf ≥ σταθερά → Δεν γίνεται ΚΑΙ τα δύο μικρά ταυτόχρονα!

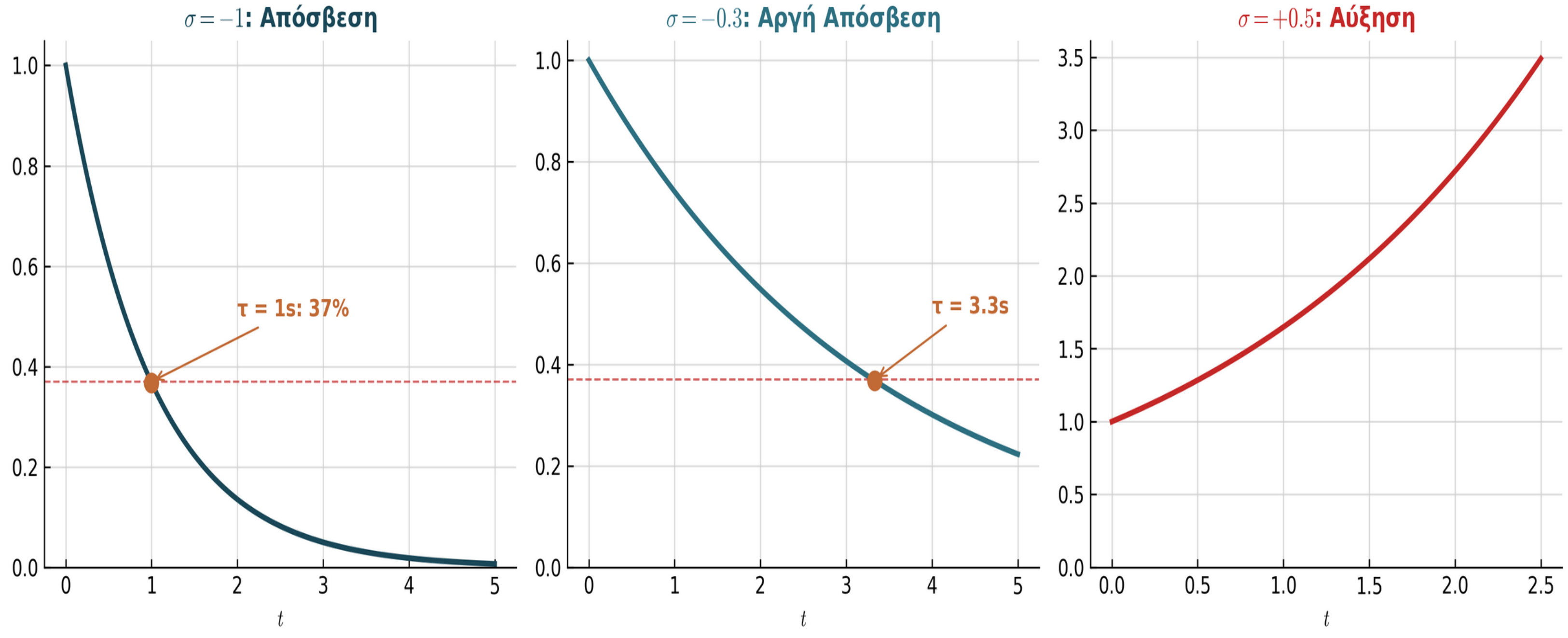
Παράδειγμα — Χτύπημα τυμπάνου (broadband signal):
Θεμελιώδης συχνότητα = η χαμηλότερη (βασική) συχνότητα ενός ήχου (π.χ. 100Hz σε τύμπανο).
Αρμονικές = ακέραια πολλαπλάσια: 2f, 3f, 4f… (δίνουν τη «χροιά» του ήχου).
Τύμπανο: πολλές αρμονικές + μη-αρμονικοί υπερτόνοι = πλατύ φάσμα → ξέρεις ΠΟΤΕ αλλά ΟΧΙ σε ποια f.

Θα δούμε αναλυτικά στις Σειρές Fourier (Εβδ. 5+).

Αρχή: Δt·Δf ≥ σταθερά: δεν γίνεται και τα δύο μικρά. **Πρακτικά:** Παλαμάκια = σύντομος, πλατύ φάσμα. Σταθερό Λα πιάνου = μακρύ, στενή συχνότητα.

(8) Εκθετικά Σήματα Ae^(σt) (Exponential Signals)

$Ae^{\sigma t}$: **ο σ ελέγχει απόσβεση/αύξηση,** $\tau = -1/\sigma$



Παράμετροι:

σ < 0: Απόσβεση (Decay)

τ = −1/σ = «χρόνος ζωής»

Φανταστείτε μπάλα που αναπηδά:
κάθε αναπήδηση πέφτει στο 37%.
Μετά 5 αναπηδήσεις: σχεδόν 0.

t=τ → 37%, t=5τ → 99% σβηστό

σ = 0: Σταθερό DC

Ae^0 = A (σταθερή τιμή πάντα)

σ > 0: Εκθετική αύξηση

Αυξάνεται χωρίς όριο!

= Ασταθές σύστημα

Π.χ. μικρόφωνο μπροστά σε ηχείο:
ήχος→μικρόφωνο→ηχείο→μικρόφωνο→ΔΥΝΑΤΟΤΕΡΑ!
(ακουστική ανάδραση — Larsen effect)

Τι: Ae^(σt): σ<0 decay, σ=0 DC, σ>0 αύξηση. Σταθερά χρόνου τ=−1/σ. **Πρακτικά:** Πυκνωτές, φίλτρα, απόσβεση ταλαντώσεων.

Εκθετικά: Κώδικας Octave (3 Περιπτώσεις)

Δημιουργήστε αρχείο:

→ Ανοίξτε editor, γράψτε τον κώδικα

→ Αποθηκεύστε ως ask2_exponentials.m

→ octave ask2_exponentials.m

→ print -dpng ... (αποθηκεύει εικόνα)

```
% ask2_exponentials.m
t = linspace(0, 5, 500);
subplot(1,3,1); plot(t, exp(-2*t), 'b', 'LineWidth', 2); grid on
title('Απόσβεση: σ=-2');
subplot(1,3,2); plot(t, ones(size(t)), 'g', 'LineWidth', 2); grid on
title('DC: σ=0');
subplot(1,3,3); plot(t(1:300), exp(t(1:300)), 'r', 'LineWidth', 2);
grid on; title('Αύξηση: σ=+1');
print -dpng ask2_exponentials.png   % αποθήκευση
```

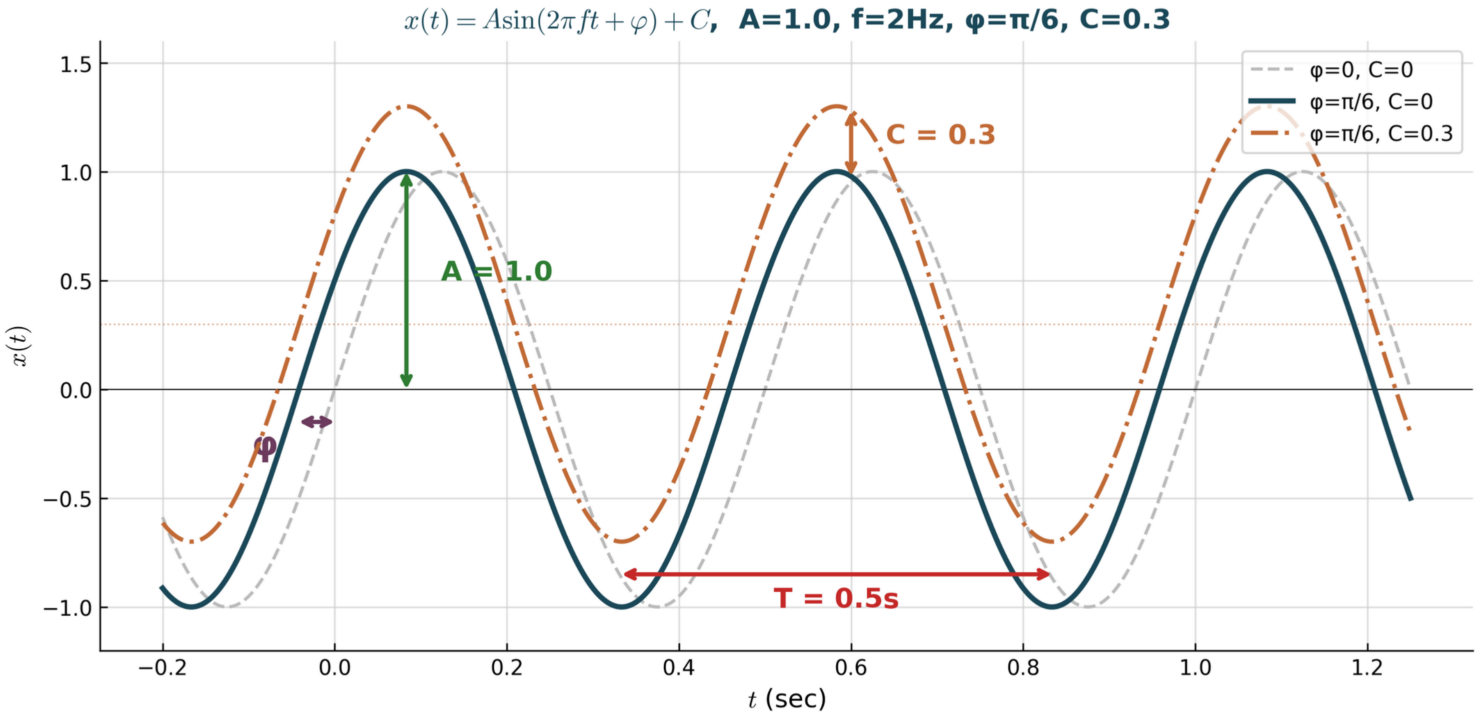
Κάθε subplot = μία περίπτωση: αριστερά απόσβεση (μπλε), μέση DC (πράσινο), δεξιά αύξηση (κόκκινο).

Η σταθερά χρόνου τ = −1/σ δίνει: πόσο γρήγορα αποσβένεται ή αυξάνεται.

Αρχείο: ask2_exponentials.m → τρέξτε με: octave ask2_exponentials.m

Δοκιμάστε: Αλλάξτε τις τιμές sigma και παρατηρήστε πώς αλλάζει η ταχύτητα decay/growth. **Hint:** Αποθηκεύστε σε .m αρχείο: exponentials_demo.m

Ημιτονοειδή Σήματα: Εισαγωγή



x(t) = A · sin(2πft + φ) + C

3 παράμετροι:

A = Πλάτος (Amplitude)
Μέγιστη απόκλιση από 0

f = Συχνότητα (Hz)
Πόσοι κύκλοι / δευτερόλεπτο
T = 1/f (περίοδος σε sec)

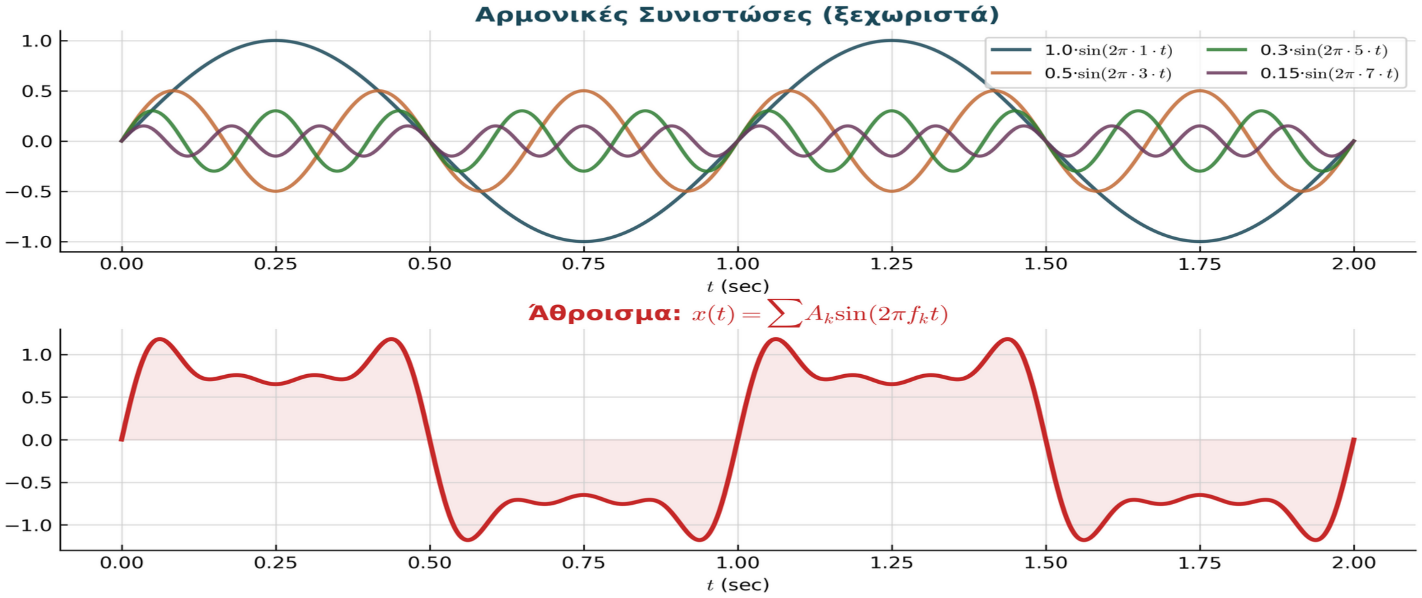
φ = Φάση (Phase, rad)
Μετατόπιση στο χρόνο
Γκρι --: φ=0 (αρχικό)
Μπλε: φ=π/6 (μετατοπισμένο)

C = DC Offset (μετατόπιση)
Ανεβάζει/κατεβάζει όλη την κυματομορφή κατακόρυφα
φ>0: αριστερά, φ<0: δεξιά

```
A=3; f=5; phi=pi/6;              % παράμετροι: A=3, f=5Hz, φ=30°
t  = linspace(0, 2/f, 500);      % χρόνος: 2 περίοδοι
x0 = A*sin(2*pi*f*t);            % αρχικό (φ=0, γκρι)
x1 = A*sin(2*pi*f*t + phi);      % μετατοπισμένο (φ=π/6, μπλε)
plot(t*1000, x0, 'color',[.7 .7 .7], 'LineWidth',2); hold on; grid on
plot(t*1000, x1, 'b', 'LineWidth',2.5)    % x άξονας σε ms!
xlabel('t  (ms)'); ylabel('x(t)')
legend('phi=0','phi=pi/6'); title('Επίδραση φάσης')
```

Τι: Κάθε ημίτονο ορίζεται πλήρως από 3 αριθμούς: A, f, φ. **Fourier (Εβδ.5+):** ΚΑΘΕ σήμα μπορεί να αναλυθεί σε άθροισμα ημιτόνων — γι'αυτό αυτά τα 3 είναι κλειδιά.

Ημιτονοειδή: Πλάτος, Συχνότητα, Φάση




```
t  = linspace(0, 1, 1000);              % 1 sec
% --- 4 ημιτονοειδή συνιστώσες ---
x1 = 1.0*sin(2*pi*1*t);                % f=1Hz, A=1.0
x2 = 0.5*sin(2*pi*3*t + pi/4);        % f=3Hz, A=0.5, φ=45°
x3 = 0.3*sin(2*pi*5*t);                % f=5Hz, A=0.3
x4 = 0.2*sin(2*pi*7*t + pi/3);        % f=7Hz, A=0.2, φ=60°
xs = x1 + x2 + x3 + x4;               % ΑΘΡΟΙΣΜΑ
subplot(2,1,1); plot(t,[x1;x2;x3;x4],'LineWidth',1.5); grid on
legend('f=1','f=3','f=5','f=7'); title('Συνιστώσες')
subplot(2,1,2); plot(t,xs,'k','LineWidth',2.5); grid on
title('Άθροισμα = σύνθετο σήμα'); xlabel('t (sec)')
```

Πάνω: 4 αρμονικές (διαφ. f)

Κάτω: το ΑΘΡΟΙΣΜΑ τους.

ω = 2πf (γωνιακή, rad/s)

Fourier:

$x(t) = \Sigma A_k \cdot \sin(2\pi f_k t + \varphi_k)$

Κάθε σήμα = άθροισμα ημιτόνων

(εβδ. 5+)

3 παράμετροι: A=πλάτος (πόσο ψηλά φτάνει), f=συχνότητα (πόσο γρήγορα ταλαντώνεται), φ=φάση (πόσο μετατοπισμένο). **Κλειδί:** Αλλαγή A→μόνο ύψος. Αλλαγή f→μόνο ταχύτητα/πλάτος T. Αλλαγή φ→μόνο μετατόπιση δεξιά/αριστερά. Ανεξάρτητες!

Ημιτονοειδή: Κώδικας Octave

```
t = linspace(0, 2, 1000);        % 2 δευτερόλεπτα, 1000 σημεία

% --- Παράμετροι ---
A = 1.5;                          % πλάτος
f = 2;                            % συχνότητα 2 Hz
phi = pi/4;                       % φάση π/4 rad = 45°

% --- Σήμα ---
x = A * sin(2*pi*f*t + phi);     % x(t) = A·sin(2πft + φ)

% --- Σχεδίαση ---
plot(t, x, 'b', 'LineWidth', 2); grid on
xlabel('t (sec)'); ylabel('x(t)');
title(sprintf('A=%.1f, f=%d Hz, φ=π/4', A, f));

% --- Μετρήσεις ---
T_period = 1/f;                   % T = 1/f = 0.5 sec
omega = 2*pi*f;                   % ω = 2πf = 4π rad/s
fprintf('T = %.2f sec, ω = %.2f rad/s\n', T_period, omega);
```

Ο κώδικας δημιουργεί ένα ημίτονο $x(t) = 1.5 \cdot \sin(2\pi \cdot 2 \cdot t + \pi/4)$ και το σχεδιάζει.
Αλλάξτε A, f, phi για να δείτε πώς αλλάζει κάθε παράμετρος.

Βασικές σχέσεις:

$T = 1/f$ (περίοδος)

$f=2\text{Hz} \rightarrow T=0.5\text{sec}$

$\omega = 2\pi f$ (γωνιακή)

$f=2\text{Hz} \rightarrow \omega=4\pi$ rad/s

Μετατροπή μοιρών→rad:

$\varphi_{rad} = \varphi_{deg} \times \pi/180$

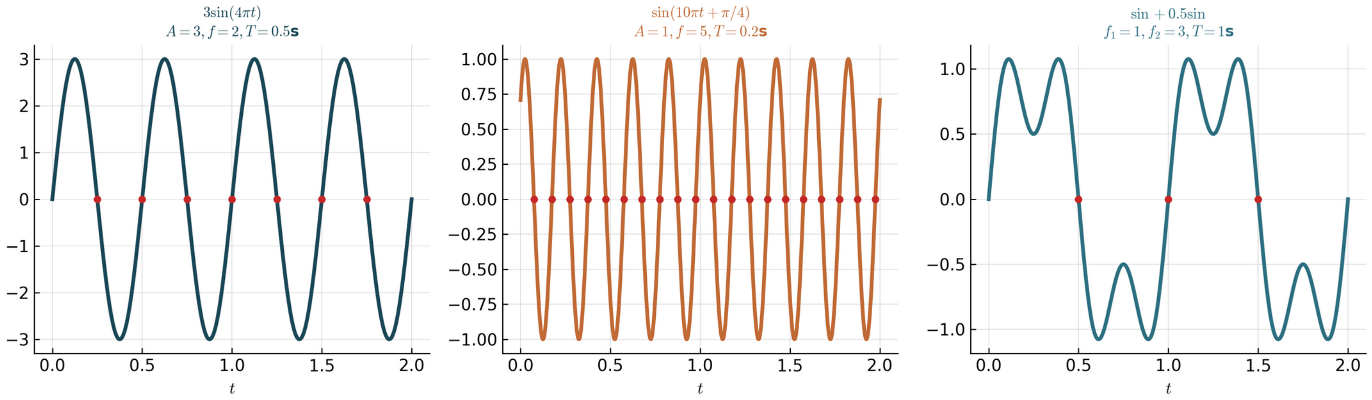
$45° = \pi/4$ rad

Δημιουργήστε αρχείο:

→ sinusoid_demo.m

Δοκιμάστε: Αλλάξτε A, f, phi και παρατηρήστε τις αλλαγές. **Ερώτηση:** Αν f=50Hz (ρεύμα AC), πόση η περίοδος T;

Λυμένο Παράδειγμα: Ανάλυση Ημιτονοειδούς




```
% x(t) = 3sin(100πt + π/6) — Octave:
t = linspace(0, 0.05, 1000);   % 2.5 περίοδοι (50Hz→T=20ms)
A=3; f=50; phi=pi/6;           % εξαχθέντες παράμετροι
x = A*sin(2*pi*f*t + phi);     % ανακατασκευή!
plot(t*1000, x, 'b', 'LineWidth', 2.5); grid on
xlabel('t (ms)'); title('3sin(100πt+π/6) — AC 50Hz')
yline(A,'r--'); yline(-A,'r--') % ±πλάτος
```

Δίνεται: $x(t) = 3\sin(100\pi t + \pi/6)$

Βρείτε A, f, T, ω, φ:

$A = 3$ (πλάτος)

$\omega = 100\pi$ rad/s

$f = \omega/(2\pi) = 50$ Hz

$T = 1/f = 0.02$ sec $= 20$ ms

$\varphi = \pi/6$ rad $= 30°$

Αυτή είναι η AC τάση 50Hz!

(Ελληνικό ηλεκτρικό δίκτυο)

Μέθοδος: Αντιστοιχίστε με $A\sin(\omega t+\varphi)$, εξάγετε A, ω, $f=\omega/(2\pi)$, $T=1/f$, φ. **Κλειδί:** $\omega = 2\pi f$ πάντα!

ΕΞΑΣΚΗΣΗ

Περιοδικότητα (Periodicity): T, f, ω

Ένα σήμα x(t) είναι ΠΕΡΙΟΔΙΚΟ αν υπάρχει T > 0 τέτοιο ώστε:

$x(t + T) = x(t)$ για κάθε t

Η μικρότερη τέτοια T λέγεται θεμελιώδης περίοδος (fundamental period).

Βασικές σχέσεις:

$f = 1/T$ (θεμελιώδης συχνότητα, Hz)

$\omega = 2\pi f = 2\pi/T$ (γωνιακή συχνότητα, rad/s)

Παράδειγμα: $\sin(2\pi\cdot 50\cdot t)$ → $\omega=100\pi$, f=50Hz, T=20ms

Άθροισμα ημιτόνων:

$x(t) = \sin(2\pi f_1 t) + \sin(2\pi f_2 t)$

Περιοδικό αν f_1/f_2 = ρητός (p/q)

Αλλιώς (√2, π, e): ΟΧΙ περιοδικό!

$T_0 = LCM(T_1, T_2)$ (θεμ. περίοδος)

$f_0 = GCD(f_1, f_2)$ (θεμ. συχνότητα)

Παράδειγμα:

f_1=6Hz, f_2=4Hz

f_0 = GCD(6,4) = 2Hz ← θεμελιώδης

$T_0 = 1/f_0$ = 0.5s

Δηλαδή: και τα 6Hz και τα 4Hz

είναι ΑΡΜΟΝΙΚΕΣ των 2Hz!

6 = 3×2, 4 = 2×2 → 3η & 2η αρμ.

Γιατί GCD = θεμελιώδης;

Φανταστείτε μετρονόμο στα 2Hz:

Κάθε 3ο χτύπημα = 6Hz ✓

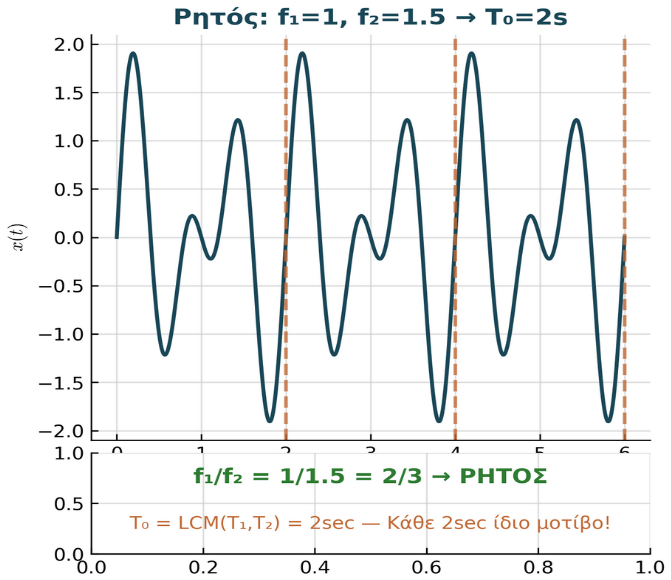
Κάθε 2ο χτύπημα = 4Hz ✓

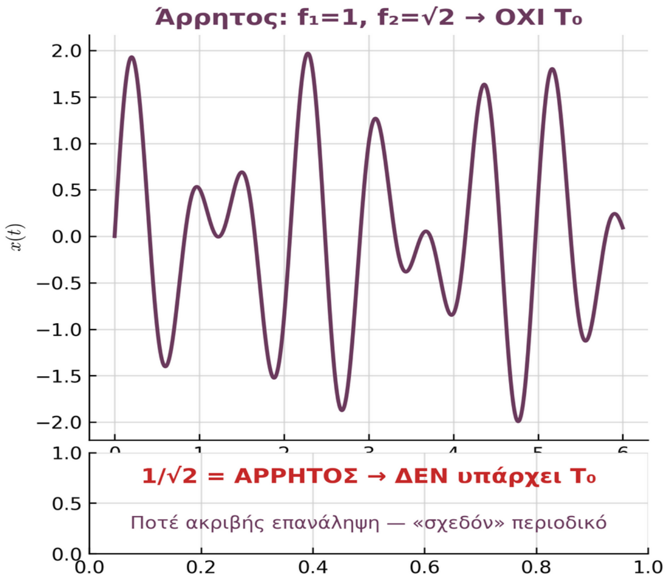
→ Η πιο αργή f (=GCD) που

«χωράει» τις f_1 ΚΑΙ f_2 ακριβώς.

GCD=θεμελιώδης: f_1=6, f_2=4 → GCD=2Hz. Τα 6 & 4 είναι αρμονικές (3η & 2η) του 2Hz. $T_0=1/f_0$=0.5s. **LCM=περίοδος:** T_1=1/6, T_2=1/4 → T_0=LCM=0.5s. Κάθε 0.5s ξαναρχίζουν μαζί.

Περιοδικότητα: Ρητά & Άρρητα Παραδείγματα





Αριστερό plot: οι κάθετες πορτοκαλί γραμμές δείχνουν κάθε T_0=2sec — το σήμα επαναλαμβάνεται ακριβώς.
Δεξί plot: δεν υπάρχει T_0. Φαίνεται «σχεδόν» περιοδικό αλλά ποτέ δεν ταυτίζεται.

Λυμένο παράδειγμα:

Αριστερά: f_1=1Hz, f_2=1.5Hz

f_1/f_2 = 1/1.5 = 2/3 → ΡΗΤΟΣ
T_1=1s, T_2=2/3s
T_0 = LCM(1, 2/3) = 2sec
Κάθετες γραμμές: κάθε 2sec ίδιο μοτίβο!

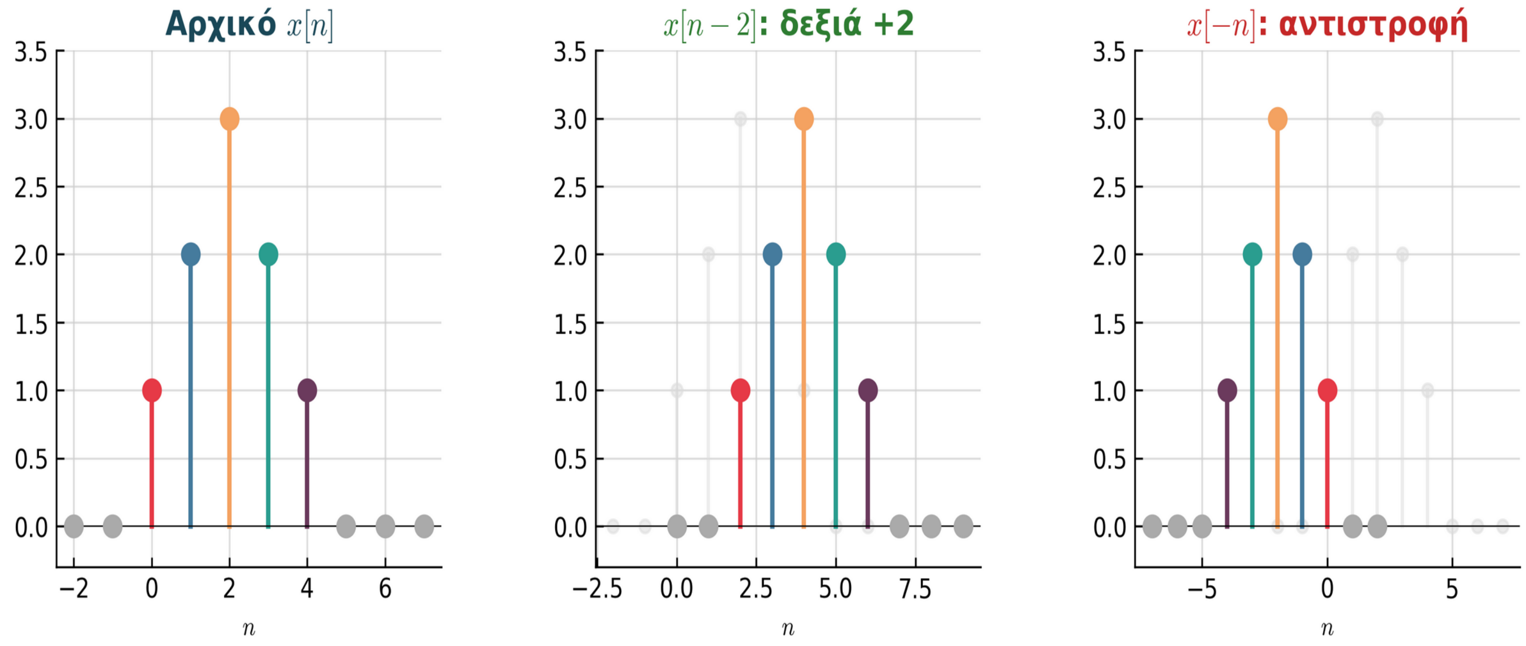
Δεξιά: f_1=1Hz, f_2=√2 Hz

1/√2 = ΑΡΡΗΤΟΣ → ΟΧΙ T_0!
Δεν ταιριάζει ΠΟΤΕ ακριβώς.

```
t = linspace(0, 6, 2000);
x_rat = sin(2*pi*1*t) + sin(2*pi*1.5*t);
x_irr = sin(2*pi*1*t) + sin(2*pi*sqrt(2)*t);
subplot(1,2,1); plot(t, x_rat); grid on
title('Ρητός: T_0=2s'); xlabel('t');
subplot(1,2,2); plot(t, x_irr); grid on
title('Άρρητος: ΟΧΙ T_0'); xlabel('t');
```

Ρητός λόγος: f_1/f_2 = p/q → ΠΕΡΙΟΔΙΚΟ, T_0 = LCM(T_1,T_2). **Άρρητος:** f_1/f_2 = άρρητος → ΜΗ περιοδικό. Ποτέ ακριβής επανάληψη.

Πράξεις DT Σημάτων (DT Signal Operations)



Μέσο panel: x[n−2] → κάθε δείγμα μετακινείται 2 θέσεις δεξιά (γκρι=πριν, πράσινο=μετά).
Δεξί panel: x[−n] → αντικατοπτρισμός γύρω από n=0 (γκρι=πριν, κόκκινο=μετά).
ΚΑΝΟΝΑΣ: x[n−k] ΔΕΞΙΑ, x[n+k] ΑΡΙΣΤΕΡΑ (αντίθετο πρόσημο!).

3 βασικές πράξεις:

① Καθυστέρηση: x[n−k]
k>0 → ΔΕΞΙΑ κατά k θέσεις
(n−k: αφαιρώ → «αργεί»)

② Μετατόπιση αριστερά: x[n+k]
k>0 → ΑΡΙΣΤΕΡΑ κατά k θέσεις
(n+k: προσθέτω → «σπεύδει»)

③ Αντιστροφή: x[−n]
Καθρέπτης ως προς n=0

Γκρι = αρχικό (σύγκριση)

3 πράξεις: Αντιστροφή x[−n], Μετατόπιση x[n−k], Κλιμάκωση A·x[n]. **Εβδ.3:** Θα δούμε αναλυτικά + συνδυασμένες πράξεις.

Ενέργεια & Ισχύς: Γιατί μας ενδιαφέρουν;

Ενέργεια (Energy) E

Πόσο «δυνατό» είναι ΣΥΝΟΛΙΚΑ;
$E = \int |x(t)|^2 \, dt$ (άθροισμα τετραγώνων)

Αν $E < \infty$ → Σήμα Ενέργειας
Παράδειγμα: παλμός, κρούση

Ισχύς (Power) P

Πόσο «δυνατό» είναι ΑΝΑ ΔΕΥΤΕΡΟΛΕΠΤΟ;
$P = \lim (1/2T) \int |x(t)|^2 \, dt$

Αν $0 < P < \infty$ → Σήμα Ισχύος
Παράδειγμα: ημίτονο, AC τάση

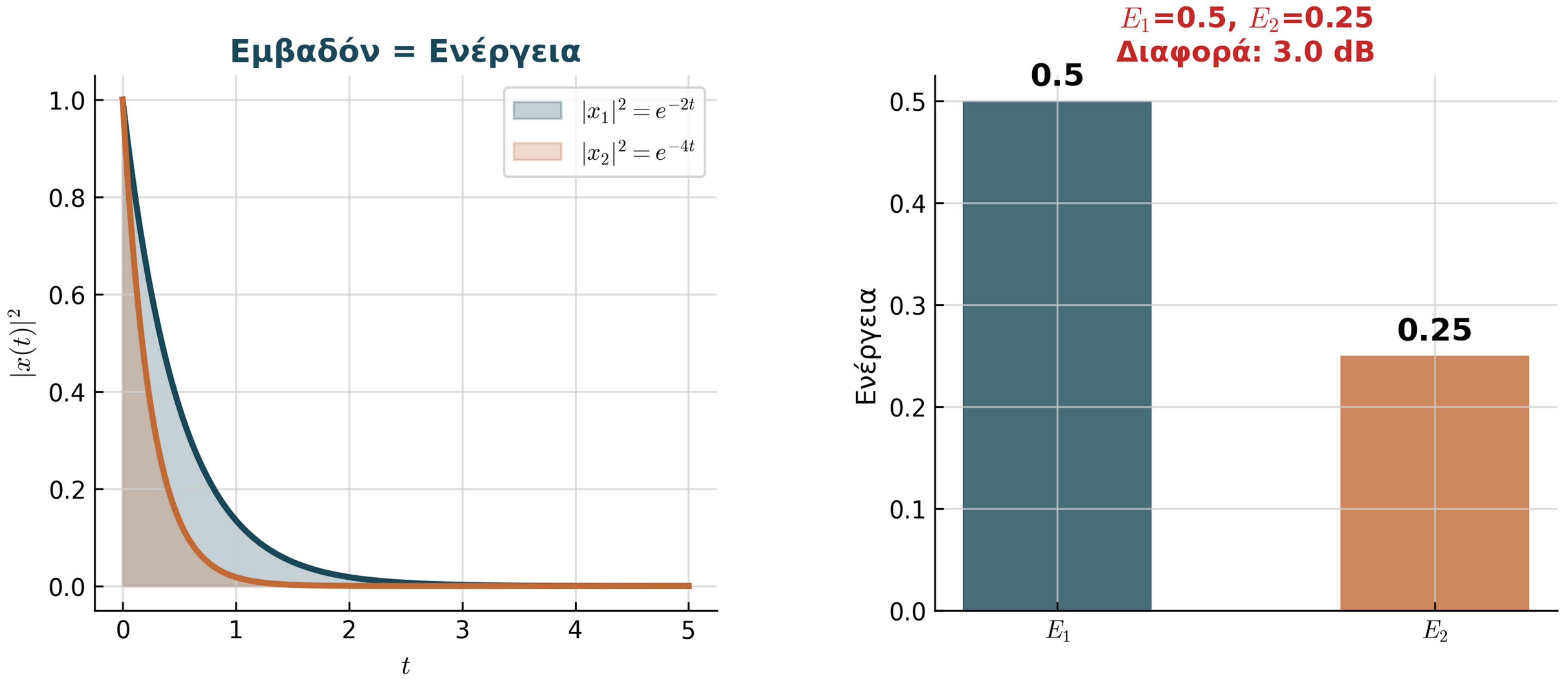
ΓΙΑΤΙ μας ενδιαφέρουν;
- Μπαταρία κινητού: πόση ενέργεια καταναλώνει ένα σήμα WiFi;
- Ηχείο: πόση ισχύ χρειάζεται για να ακουστεί ο ήχος;
- Κεραία: αρκεί η ισχύς του σήματος ή χάνεται στο θόρυβο;

Αναλογία: Μπαλόνι νερού που σκάει (παλμός) → πεπερασμένο νερό = ενέργεια.
Βρύση (συνεχές) → ατελείωτο νερό αλλά σταθερή ροή = ισχύς.

RMS = $\sqrt{P}$: Η πρίζα δίνει 325V κορυφή αλλά 230V RMS — αυτό μετράει η λάμπα.

Κανόνας: Παλμός → ενέργεια ($E<\infty$). Περιοδικό → ισχύς ($P<\infty$). **RMS:** $\sqrt{P}$ = η ισοδύναμη σταθερή τιμή.

Ενέργεια & Ισχύς: Τύποι και Υπολογισμός



Αριστερά: x_1=e^(−t), x_2=e^(−2t) — σκιασμένες περιοχές = |x|² = ΕΝΕΡΓΕΙΑ (πεπερασμένη, φθίνει).
Δεξιά: μπάρες E_1=0.5, E_2=0.25. Διαφορά 3dB = η x_1 έχει διπλάσια ενέργεια.
Μονάδες: αν x=τάση → E σε [V²·s], P σε [V²]. Αδιάστατο σήμα → αδιάστατα.

Τύποι:

CT Ενέργεια:
E = ∫ |x(t)|² dt
(= εμβαδόν κάτω από |x|²)

CT Ισχύς:
P = lim (1/2T) ∫ |x(t)|² dt
(= μέσος όρος ενέργειας/sec)

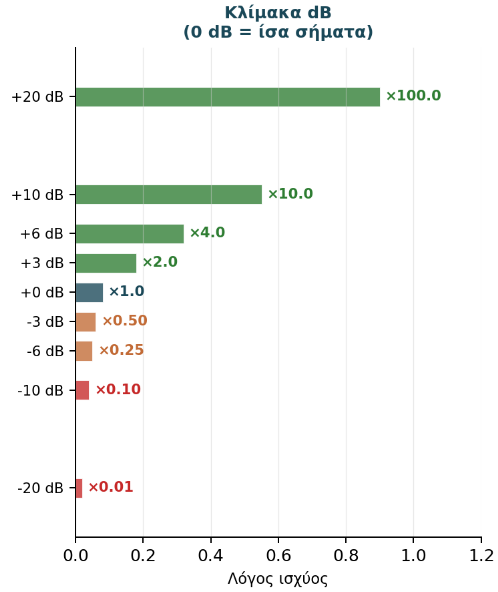
DT Ενέργεια:
E = Σ |x[n]|²
(= άθροισμα τετραγώνων)

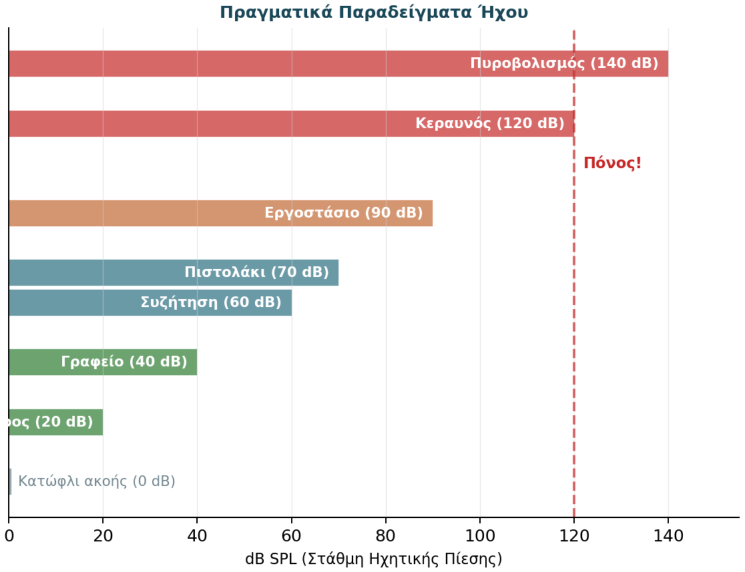
DT Ισχύς:
P = lim (1/2N+1) Σ |x[n]|²
(= μέσος όρος ανά δείγμα)

RMS = √P (Root Mean Square)
(= η DC τιμή με ίδια ισχύ)

Πρακτικά: Octave: E = sum(abs(x).^2)*dt, P = mean(abs(x).^2). **AC ρεύμα:** Vpp=325V, RMS=230V (αυτό μετράμε).

Κλίμακα Decibel (dB)





► P: $10\cdot\log_{10}(100W/1W) = 10\cdot\log_{10}(100) = 10\cdot 2 = 20$ dB → ενισχυτής 20dB κέρδος

► A: $20\cdot\log_{10}(2V/1V) = 20\cdot\log_{10}(2) \approx 20\cdot 0.301 \approx 6$ dB → +6dB = ×2 πλάτος = ×4 ισχύς

► Κεραία WiFi: Gain=12dBi, BW=65°, −3dB@±32.5°

Στο κέντρο: +12dBi ενίσχυση. Σε ±32.5°: ισχύς = ½ (half-power point). Εκτός ±32.5°: ραγδαία πτώση.

dB = Αδιάστατος ΛΟΓΟΣ

(σχετική μονάδα, ΟΧΙ απόλυτη!)

Ισχύς: $dB = 10\cdot\log_{10}(P_1/P_2)$

P = ισχύς σε Watt (W)

Πλάτος: $dB = 20\cdot\log_{10}(A_1/A_2)$

A = πλάτος σε Volt (V) ή Pa

Γιατί 20; → $P \propto A^2$:

Ηλεκτρικά: $P = V^2/R$ (αν R ίδια)

ή $P = I^2\cdot R$ (αν R ίδια)

→ $10\cdot\log(A^2/A_0^2) = 20\cdot\log(A/A_0)$

Κλειδιά:

0 dB = ίσα (λόγος=1)

+3 dB = 2× ισχύς

+6 dB = 2× πλάτος (4× ισχύς)

−3 dB = μισή ισχύς

Αρνητικά = ασθενέστερο σήμα

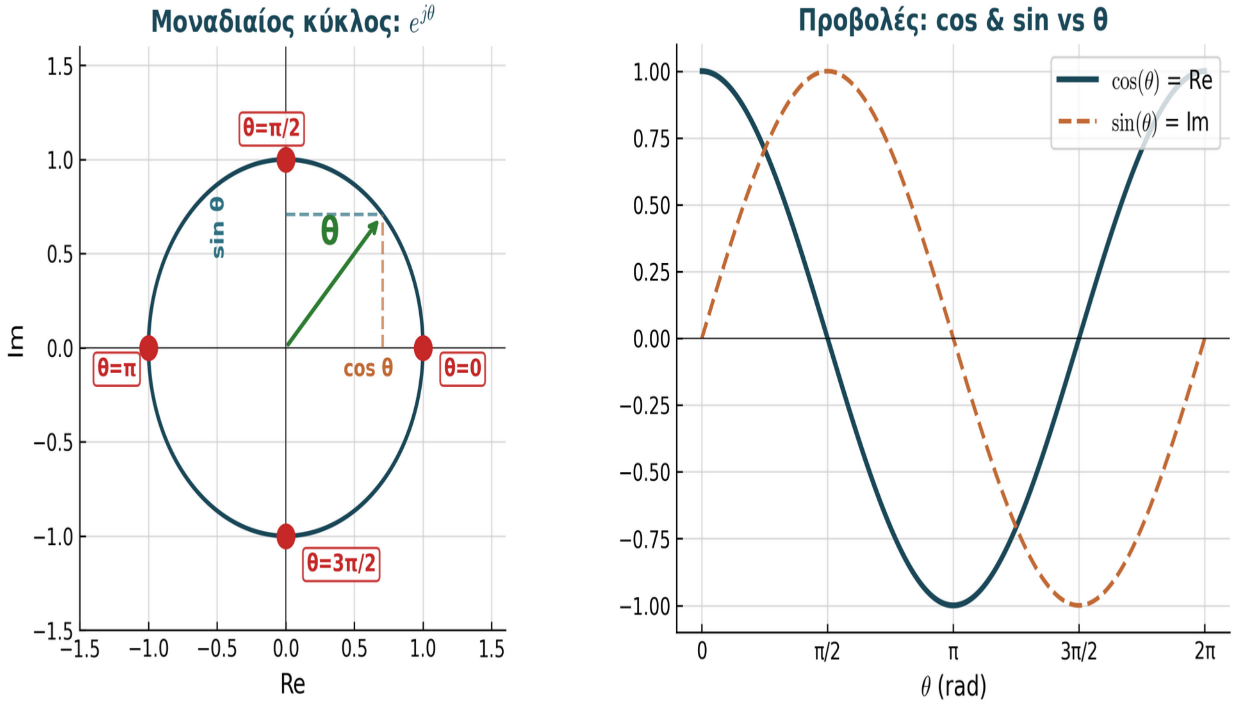
Κεραίες: −3dB = half-power point

π.χ. WiFi κεραία: ±30° = −3dB

SPL: αναφορά = 20μPa (ακοή)

Γιατί 10 vs 20; $P \propto A^2$: $10\cdot\log(A^2) = 20\cdot\log(A)$. Πάντα ρώτα: «ισχύς ή πλάτος;» **Κεραία specs:** Gain=12dBi, BW=65°, −3dB@±32.5°: στο κέντρο 12dB ενίσχυση. Σε ±32.5° η ισχύς=½. Εκτός 65° εξασθενεί.

Τύπος Euler: e^(jθ) = cos(θ) + j·sin(θ)



Αριστερά: Μοναδιαίος κύκλος. Κόκκινες τελείες = 4 ειδικές γωνίες (0, π/2, π, 3π/2).
Πράσινο βέλος = γωνία θ. Διακεκομμένες = προβολές cos θ (Re) και sin θ (Im).
Δεξιά: cos(θ) (μπλε) και sin(θ) (πορτοκαλί) καθώς η γωνία θ αυξάνεται.

Τύπος Euler:

e^(jθ) = cos(θ) + j·sin(θ)

Τι σημαίνει ΑΠΛΑ:

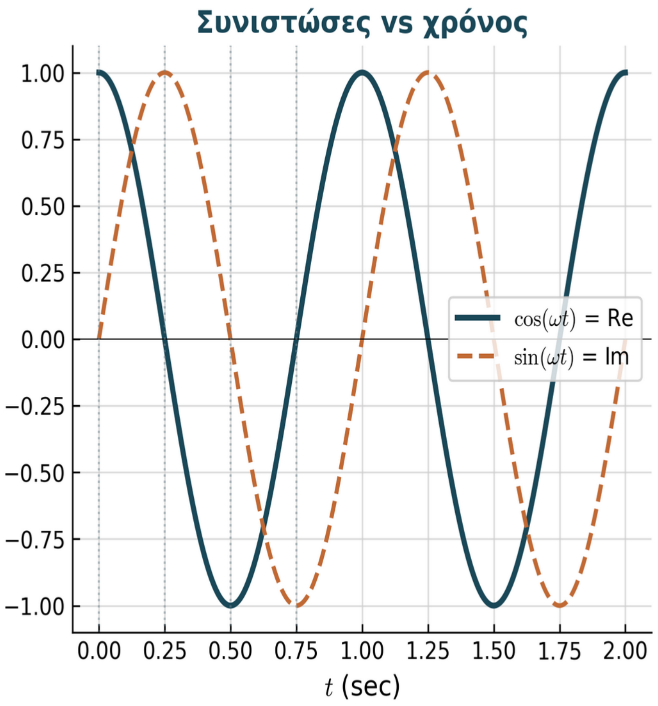
Αντί να γράφω ξεχωριστά cos ΚΑΙ sin,
τα «πακετάρω» σε ΕΝΑ e^(jθ).
Ο αριθμός e^(jθ) = ένα σημείο
στον κύκλο ακτίνας 1.

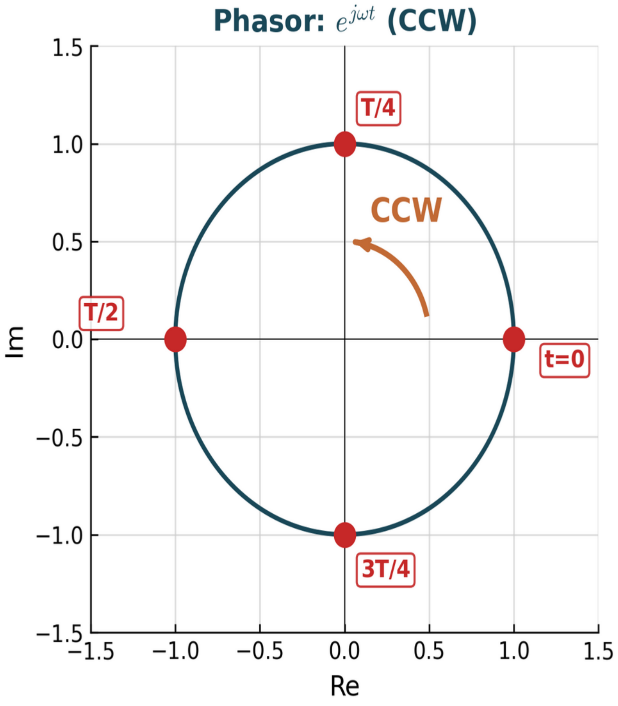
ΓΙΑΤΙ είναι χρήσιμος:

① cos+sin → ΕΝΑ εκθετικό
② e^(jα)·e^(jβ) = e^(j(α+β))
(γωνίες ΠΡΟΣΤΙΘΕΝΤΑΙ!)
③ AC: V = V_o·e^(j(ωt+φ))

Γιατί; Αντί cos+sin → ΕΝΑ e^(jθ). Πολ/σμός→πρόσθεση γωνιών. AC: V_oe^(j(ωt+φ)). **Κλειδί:** Γι' αυτό οι ηλ. μηχανικοί χρησιμοποιούν phasors!

Phasors: Περιστρεφόμενο Βέλος (Rotating Vector)





Αριστερά: cos(ωt) και sin(ωt) vs χρόνος. Δεξιά: phasor στον μοναδιαίο κύκλο.
Στο t=0 το βέλος ξεκινά στο (1,0). Γυρίζει counter-clockwise (CCW) — πορτοκαλί τόξο.
Κάθε T/4 = 90°: (1,0)→(0,j)→(−1,0)→(0,−j)→(1,0).

Phasor = σταθερό βέλος:

Μήκος = Πλάτος A
Γωνία = Φάση φ

Πώς δουλεύει ΑΠΛΑ:

Τροχός ποδηλάτου γυρίζει.
Από μπροστά: βαλβίδα πάνω-κάτω = cos(ωt)
Από πλάι: βαλβίδα αριστ-δεξ = sin(ωt)

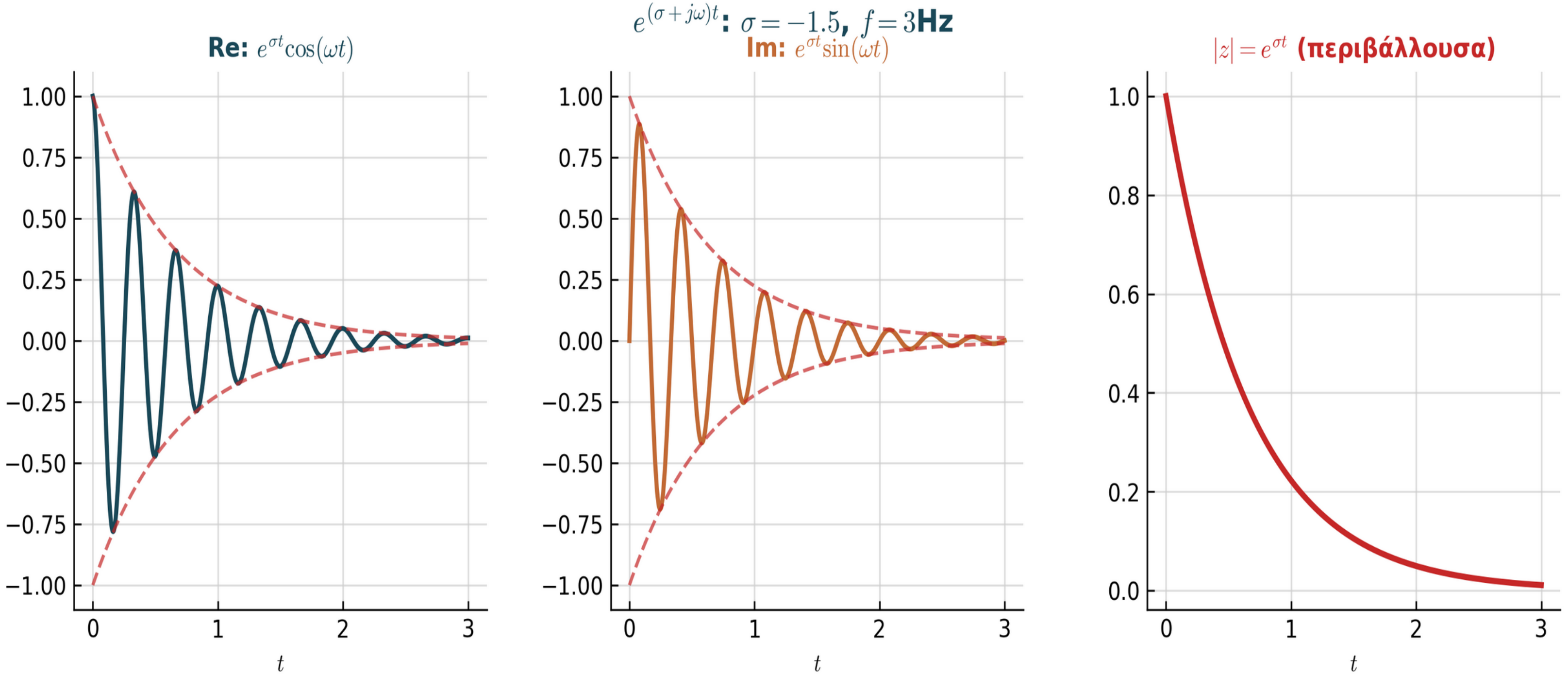
AC Παράδειγμα:

V(t) = 325·cos(100πt + π/6)
Phasor: V = 325∠30°
RMS: 325/√2 = 230V

Phasor: Σταθερό βέλος: μήκος=A, γωνία=φ. Η ωt περιστροφή υπονοείται. **AC:** 325∠30° → V(t) = 325cos(100πt+30°), RMS=230V.

ΕΞΑΣΚΗΣΗ

Μιγαδική Εκθετική e^((σ+jω)t)



Αριστερά: Re{e^(st)} = e^(σt)·cos(ωt) — ταλάντωση που σβήνει (σ<0).
Μεσαία: Im{e^(st)} = e^(σt)·sin(ωt) — ίδιο αλλά μετατοπισμένο 90°.
Δεξιά: |e^(st)| = e^(σt) — η «περιβάλλουσα» (envelope), μόνο η απόσβεση.

e^(st) όπου s = σ + jω:

σ = πραγματικό μέρος:

Ελέγχει το «σβήσιμο»:
σ<0: φθίνει (σαν κρούση τυμπάνου)
σ=0: σταθερό πλάτος
σ>0: αυξάνεται (ασταθές!)
→ Ακουστική ανάδραση (Larsen):
ήχος→μικρόφωνο→ηχείο→δυνατότερα!

ω = φανταστικό μέρος:

Ελέγχει την ταλάντωση:
f = ω/(2π) Hz

Μαζί:

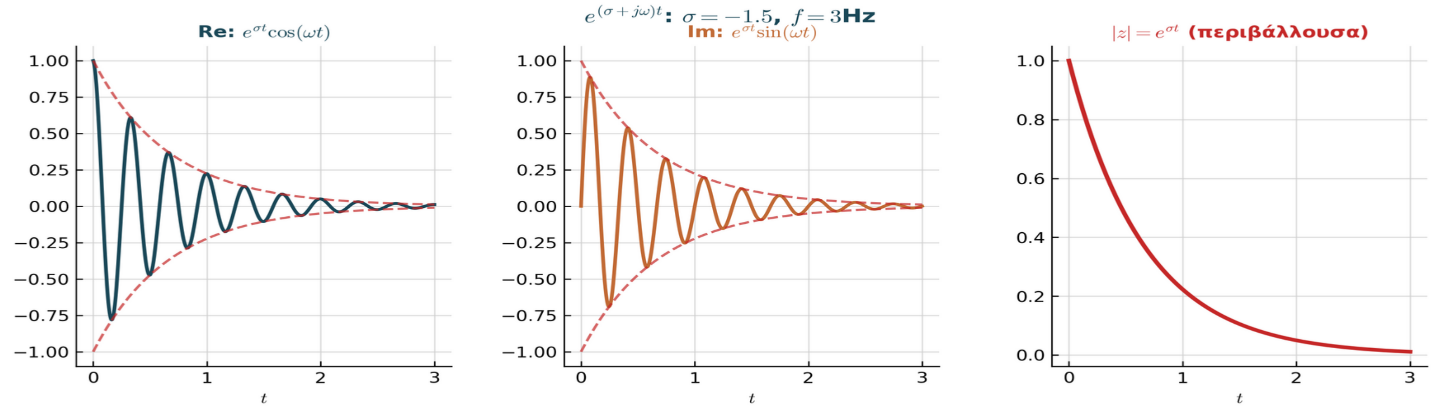
e^(st) = e^(σt) × e^(jωt)
= περιβάλλουσα × ταλάντωση

Περιβάλλουσα (envelope):

= η καμπύλη που «αγκαλιάζει»
τις κορυφές της ταλάντωσης.
Δηλαδή: e^(σt) = η γραμμή
που ακολουθούν τα maxima.

Σύνθεση: e^(σt) = «πόσο σβήνει» (περιβάλλουσα). e^(jωt) = «πόσο γρήγορα ταλαντώνεται». Μαζί: ταλάντωση ΜΕ σβήσιμο. **Πρακτικά:** Κρούση τυμπάνου, αντήχηση χορδής, απόσβεση ταλάντωσης γέφυρας — ΟΛΑ e^(σt)cos(ωt) με σ<0.

Μιγαδική Εκθετική: Κώδικας Octave



Αυτό παράγει ο κώδικας:

Μπλε = Re{e^(st)} = cos × decay

Κόκκινη -- = περιβάλλουσα e^(σt)

Αρχείο: ask2_complex_exp.m

```
t = linspace(0,3,2000);
sigma=-1.5; omega=2*pi*3;
z = exp((sigma+1j*omega)*t);  % μιγαδικό
env = exp(sigma*t);           % περιβάλλουσα
subplot(1,3,1)                % Re: e^{σt}cos(ωt)
  plot(t,real(z),'b','LineWidth',3); hold on; grid on
  plot(t,env,'r--',t,-env,'r--','LineWidth',2)
  title('Re: e^{σt}cos(ωt)'); xlabel('t')
subplot(1,3,2)                % Im: e^{σt}sin(ωt)
  plot(t,imag(z),'LineWidth',3); grid on
  title('Im: e^{σt}sin(ωt)'); xlabel('t')
subplot(1,3,3)                % |z| = περιβάλλουσα
  plot(t,abs(z),'r','LineWidth',4); grid on
  title('|z| = e^{σt}'); xlabel('t')  % ▶ s36.m
```

exp(s*t) δίνει μιγαδικές τιμές.

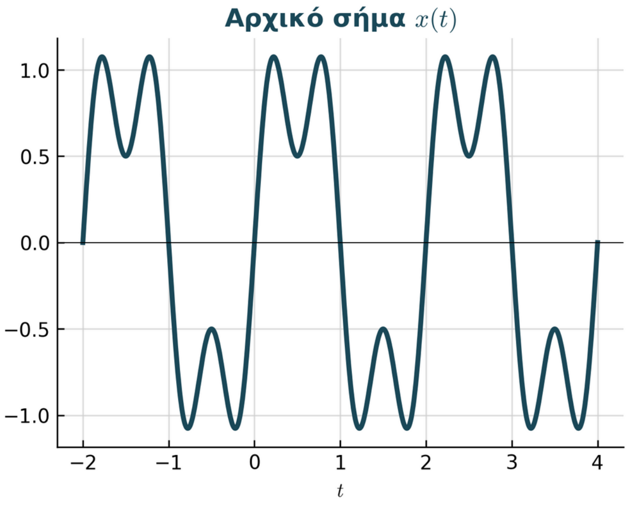
real() = e^(σt)·cos(ωt).

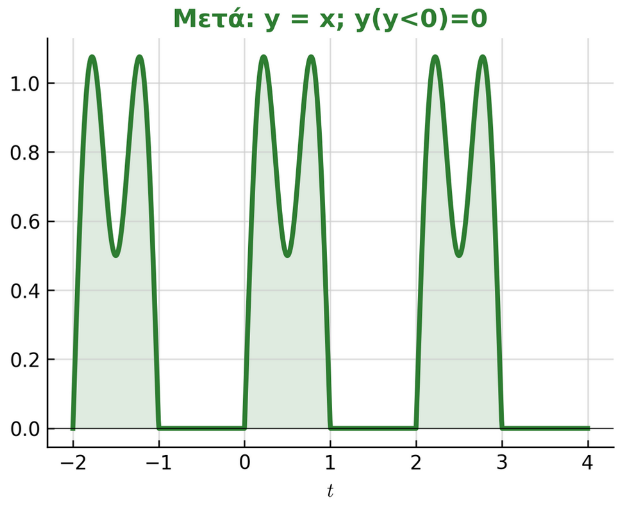
Κόκκινη -- = ±e^(σt).

Δοκιμάστε: sigma = 0.

Octave: exp(s*t) με s=σ+jω. real() = cos, imag() = sin. **Αποθηκεύστε:** complex_exp_demo.m

Logical Indexing: Η Δύναμη του Octave






```
t = linspace(-3,3,1000);  % χρονική βάση
u   = (t >= 0);           % u(t): step
rc  = (abs(t) <= 0.5);    % rect(t)
msk = (t>=0) & (t<=2);    % παράθυρο [0,2]
y   = sin(2*pi*t).*msk;   % windowed sin
z   = y; z(z<0) = 0;      % κόψε αρνητικά
% clamp σε [-0.5, +0.5]:
y(y>0.5)=-0.5; y(y<-0.5)=-0.5;
```

Vectorized: u=(t>=0), rect=(abs(t)<=0.5), mask=(t>=0)&(t<=2)
y.*mask = window · y.*msk · z(z<0)=0 — ΟΛΑ χωρίς for!

Τι είναι:

Λογική συνθήκη → πίνακας true/false

Παραδείγματα:

u = (t >= 0);   % step
rect = (abs(t) <= 0.5); % rect
mask = (t>=0) & (t<=2); % [0,2]
y = x .* mask;   % window

Αντικατάσταση τιμών:

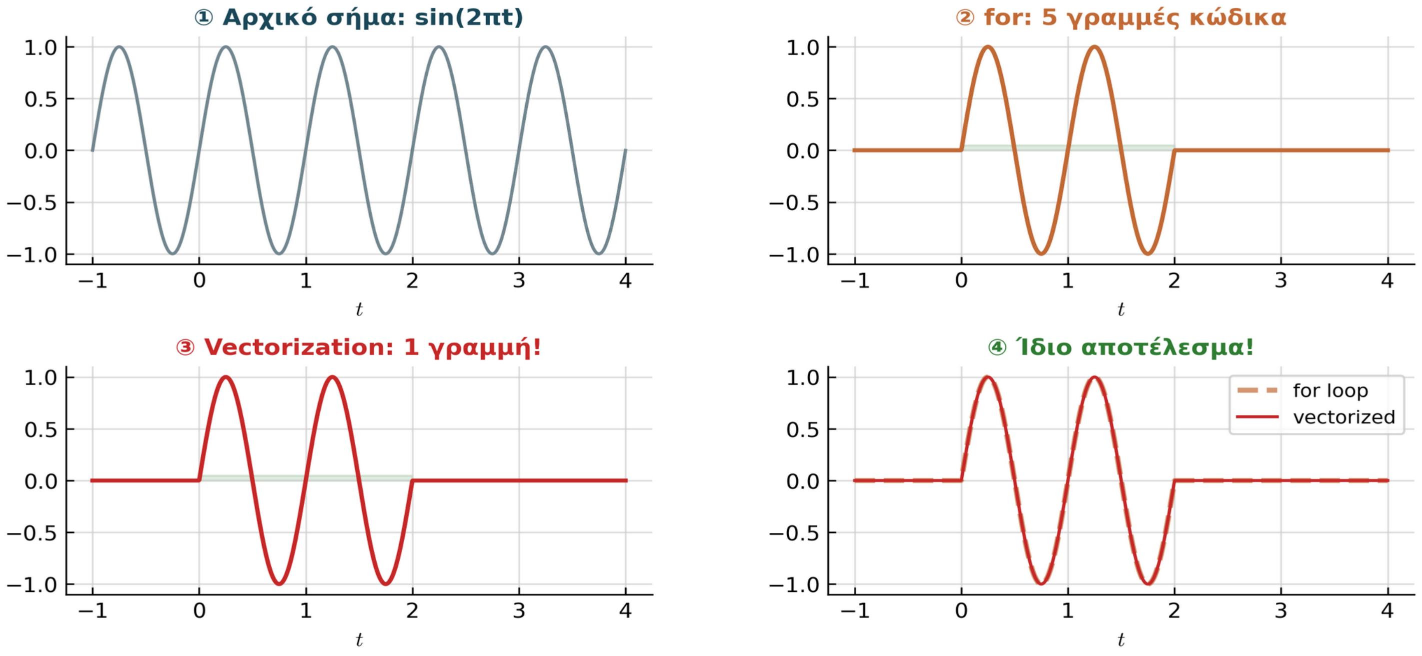
y = x; y(y < 0) = 0; % κόψε <0
y(abs(y) > 1) = 1;   % clamp
y(y > 0.5) = NaN;   % αφαίρεσε

Σύνθετο:

(t>=0) | (t<=-3)   % ένωση

Κλειδί: mask=(t>=0) → [0 0 0 1 1 1]. x.*mask → σήμα μόνο όπου mask=1. **Vectorize:** u=(t>=0), rect=(abs(t)<=.5), clamp: y=x; y(y>1)=1. ΑΠΟΦΥΓΕΤΕ for αν δεν χρειάζεται!

Τεχνικές Octave: for vs Vectorization



for loop:

5 γραμμές: for+if+end+end

Vectorization:

mask·sin = 1 γραμμή! Ίδιο!

Ταχύτητα:

Vectorized: ~10-100× πιο γρήγορο!

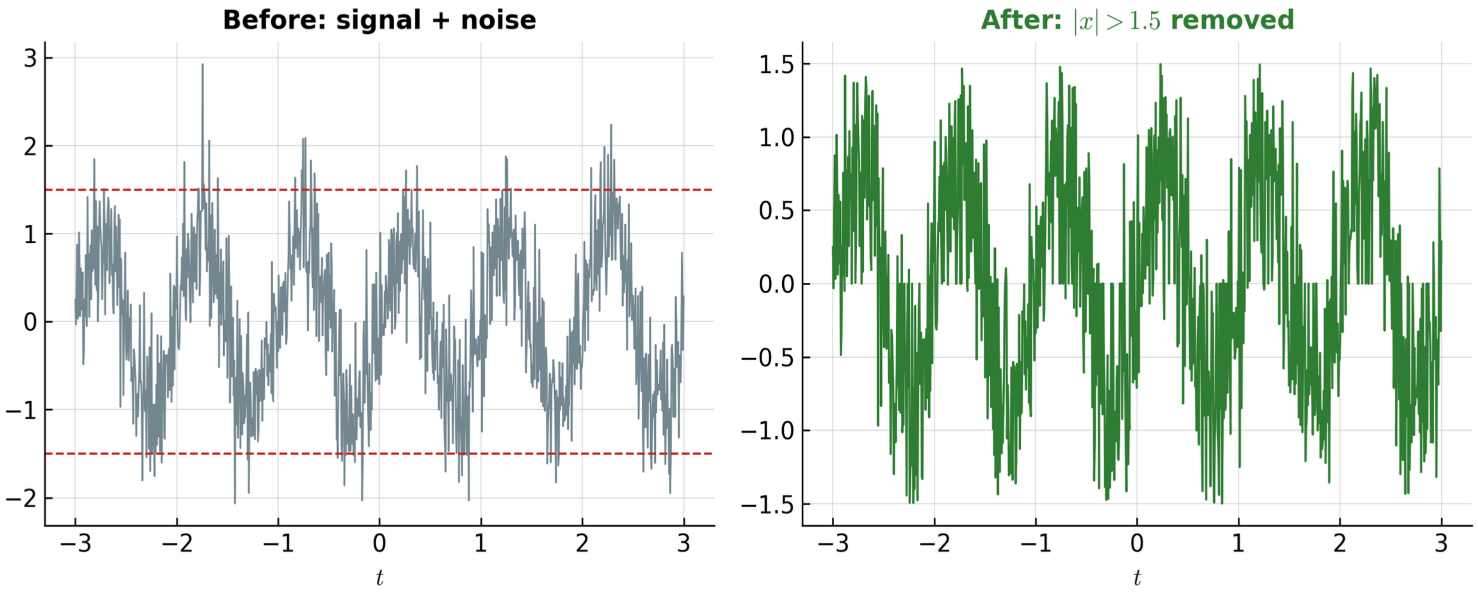
Octave βελτιστοποιεί πράξεις σε ολόκληρα arrays ταυτόχρονα (BLAS).

Κανόνας: vectorize πρώτα!

```
t=linspace(-1,4,1000); x=sin(2*pi*t);
% ΑΡΓΟ: for loop (5 γραμμές)
y=zeros(size(t));
for i=1:length(t)
  if t(i)>=0 && t(i)<=2; y(i)=sin(2*pi*t(i)); end
end
% ΓΡΗΓΟΡΟ: 1 γραμμή!
mask=(t>=0)&(t<=2); y2=sin(2*pi*t).*mask;
subplot(2,2,1); plot(t,x,'color',[.7.7.7],'LineWidth',3); grid on; title('① Αρχ.')
subplot(2,2,2); plot(t,y,'LineWidth',3);   grid on; title('② for')
subplot(2,2,3); plot(t,y2,'r','LineWidth',3); grid on; title('③ Vect.')
subplot(2,2,4); plot(t,y,'--'); hold on; plot(t,y2,'r'); grid on; title('④ Ίδιο')
```

Σύγκριση: for: 5 γραμμές, αργό (στοιχείο-στοιχείο). Vectorized: 1 γραμμή, 10-100× ταχύτερο! **Κανόνας:** Πάντα vectorize πρώτα. for μόνο αν η λογική δεν γίνεται με mask (π.χ. εξαρτημένοι υπολογισμοί).

Λυμένη Άσκηση: Threshold & Clipping




```
randn('seed', 1)                            % αναπαραγώγιμο
t=linspace(-3,3,1000);
x=sin(2*pi*t)+0.5*randn(size(t)); thr=1.5;
y=x; y(abs(y)>thr)=0;  % logical indexing
subplot(1,2,1); plot(t,x,'LineWidth',1.5); hold on; grid on
  line([min(t) max(t)],[thr
thr],'Color','r','LineStyle','--','LineWidth',2)
  line([min(t) max(t)],[-thr -
thr],'Color','r','LineStyle','--','LineWidth',2)
  title('Πριν: σήμα+θόρυβος')
subplot(1,2,2); plot(t,y,'g','LineWidth',1.8); grid on
  title(sprintf('Μετά: |x|>%.1f → 0',thr)) % ▶ s39.m
```

Βήματα λύσης:

① sin(2πt) + θόρυβος randn → σήμα με spikes

② abs(y) > 1.5 → λογικός πίνακας: true στα outliers

③ y(...) = 0 → μηδενίζει ΜΟΝΟ αυτά (logical indexing)

Εφαρμογή: ECG/EEG artifact removal, audio clipping, noise gate.

Τι: abs(y)>1.5 βρίσκει outliers, y(...)=0 τα αφαιρεί. ΜΙΑ γραμμή! **Εφαρμογή:** ECG: αφαίρεση artifacts, Audio: noise gate.

Συχνά Λάθη

① plot αντί stem σε DT

x[n] θέλει stem(), όχι plot()!
plot → ψεύτικη συνέχεια

② Ξεχνάμε .* και .^

x*y → πίνακες, x.*y → element-wise
ΠΑΝΤΑ τελεία πριν * / ^ σε σήματα

③ rad vs Hz σύγχυση

ω (rad/s) ≠ f (Hz), ω = 2πf
sin() θέλει ΠΑΝΤΑ radians

④ sinc: ποια εκδοχή;

Octave/MATLAB: sinc(t) = sin(πt)/(πt)
Μερικά βιβλία: sin(t)/t. Ελέγξτε ποια!

⑤ Λίγα δείγματα

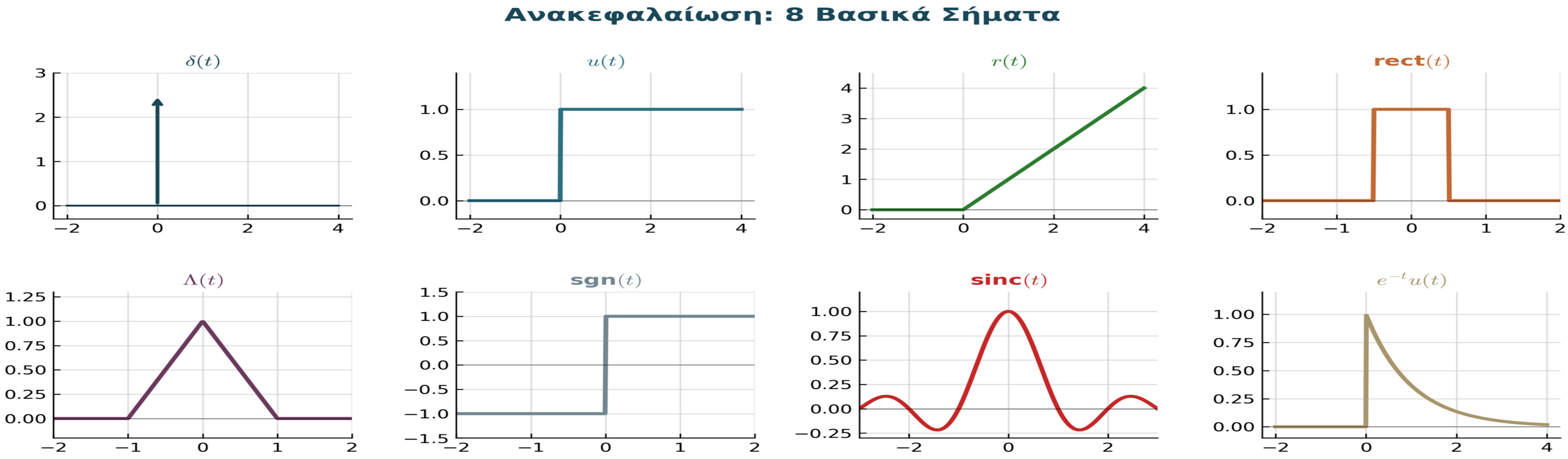
linspace(-5,5,10) → τραχιά
linspace(-5,5,1000) → λεία. Πάντα ≥500!

⑥ Ξεχνάμε pkg load signal

Χωρίς αυτό: sinc(), fft() κλπ δεν υπάρχουν!
ΠΑΝΤΑ στην αρχή κάθε script.

Σημαντικό: Αυτά τα 5 λάθη εμφανίζονται πολύ συχνά. Ξαναδιαβάστε τα πριν κάθε εργαστήριο!

Ανακεφαλαίωση (Summary)



Σήμερα μάθαμε:

- 8 θεμελιώδη σήματα: u(t), δ(t), r(t), rect, Λ, sgn, sinc, εκθετικό
- Ημιτονοειδή: A, f, φ. Σχέσεις: T=1/f, ω=2πf. LCM για περιοδικότητα.
- Ενέργεια $E=\int|x|^2dt$, Ισχύς P=E/T, RMS=A/√2, $dB=10\cdot\log_{10}(P_1/P_2)$
- Euler: e^(jθ)=cos(θ)+j·sin(θ), phasors, μιγαδική εκθετική
- Octave: plot/stem, logical indexing, vectorization (.* .^ ./)

Επόμενη Εβδομάδα: Πράξεις σε σήματα: μετατόπιση, κλιμάκωση, ανάκλαση, σχεδιασμός σημάτων.

Cheatsheet: Βασικές Εντολές Octave

Δημιουργία

```
t = linspace(-5,5,1000);
n = -10:10;
dt = t(2)-t(1);
x = sin(2*pi*f*t + phi);
```

Σχεδίαση

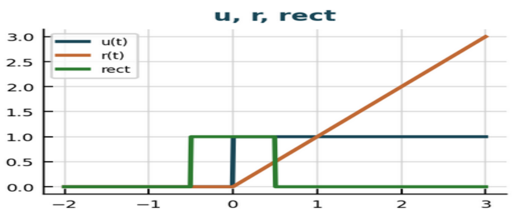
```
plot(t,x,'b','LineWidth',2); %CT
stem(n,x,'filled'); %DT
subplot(2,1,1);
grid on; xlabel('t');
```

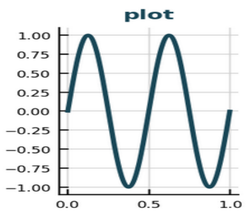
Βασικά Σήματα

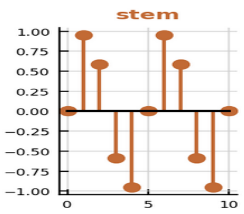
```
u = (t>=0);         %step
r = t.*(t>=0);      %ramp
rect = (abs(t)<=.5);
sgn = sign(t);
```

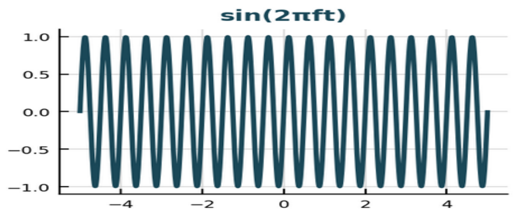
Υπολογισμοί

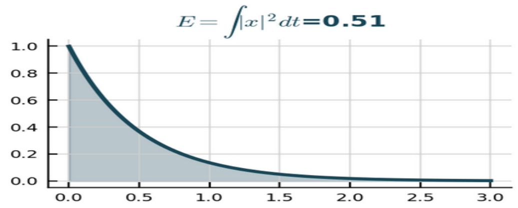
```
E = sum(x.^2)*dt; %energy
y=x; y(y<0)=0;    %logical
dB=20*log10(A1/A2);
RMS=sqrt(mean(x.^2));
```



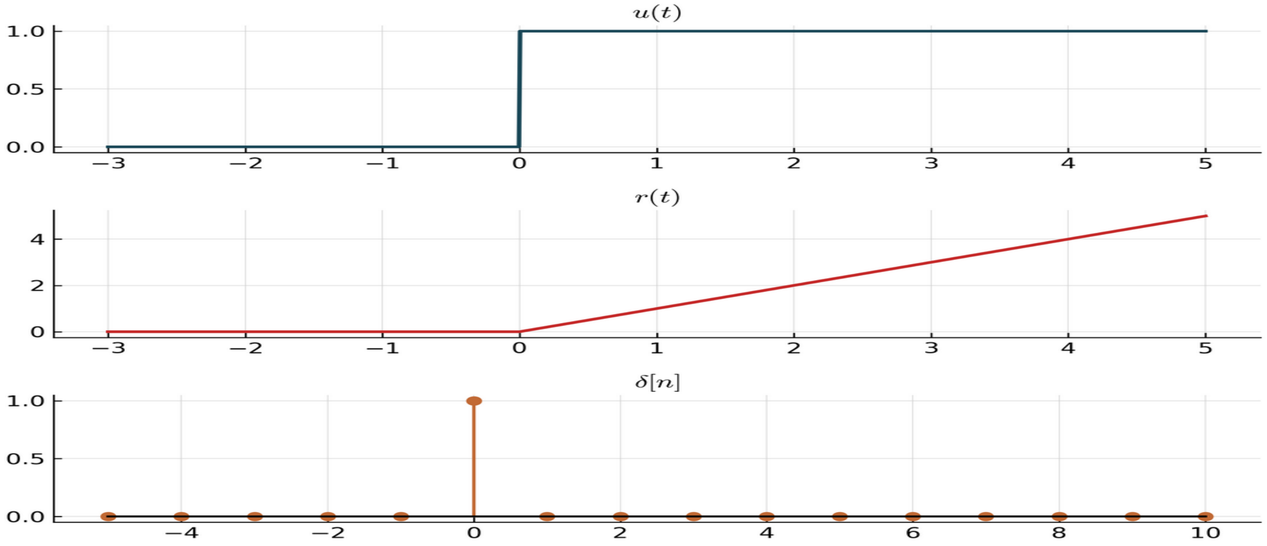








Άσκηση 1: Βασικά Σήματα



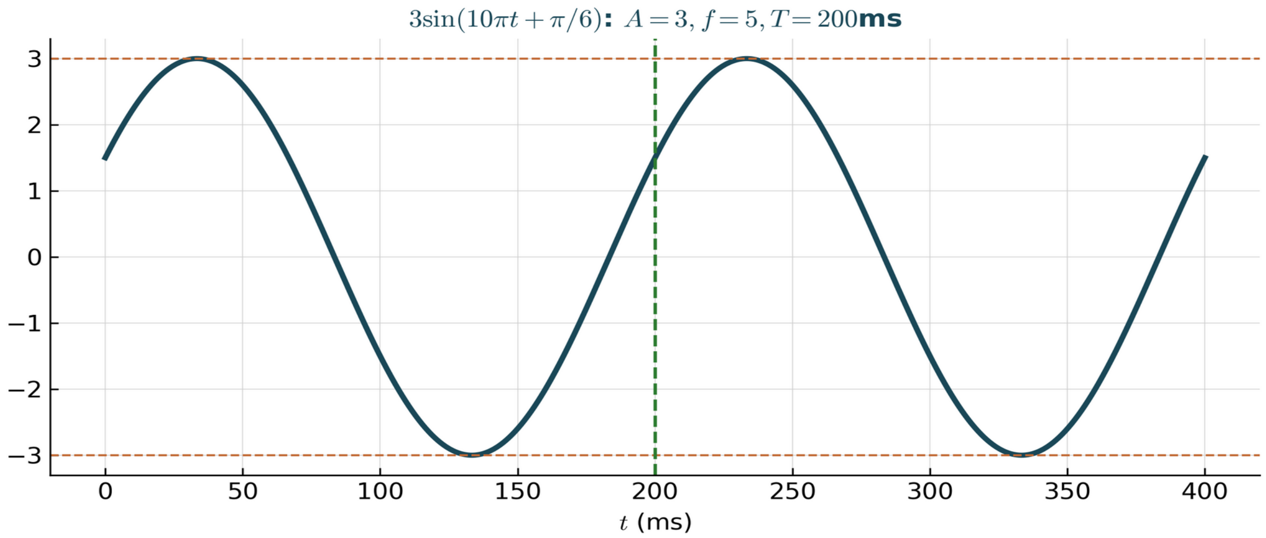

```
t = linspace(-3, 5, 1000);      % χρόνος CT
subplot(3,1,1);
plot(t, (t>=0), 'b', 'LineWidth', 2);  % u(t): step
title('u(t)'); ylabel('u(t)'); grid on
subplot(3,1,2);
plot(t, t.*(t>=0), 'r', 'LineWidth', 2);  % r(t): ramp
title('r(t)'); ylabel('r(t)'); grid on
n = -5:10; subplot(3,1,3);
stem(n, (n==0), 'filled');       % δ[n]: impulse DT
title('delta[n]'); xlabel('n');
```

Λυμένη Άσκηση:

Σχεδιάστε τα u(t), r(t) (CT, plot) και δ[n] (DT, stem) σε 3 subplots.
Σημειώστε τα σημεία κλειδιά: u(0)=1, r(0)=0, δ[0]=1.
Πώς σχετίζονται μεταξύ τους; (ολοκλήρωση / παραγώγιση)

Hint: Ο κώδικας δεξιά είναι η λύση — μελετήστε τον γραμμή-γραμμή. **Εξάσκηση:** Δοκιμάστε να αλλάξετε τις παραμέτρους και παρατηρήστε τι αλλάζει.

Άσκηση 2: Ημιτονοειδή Σήματα



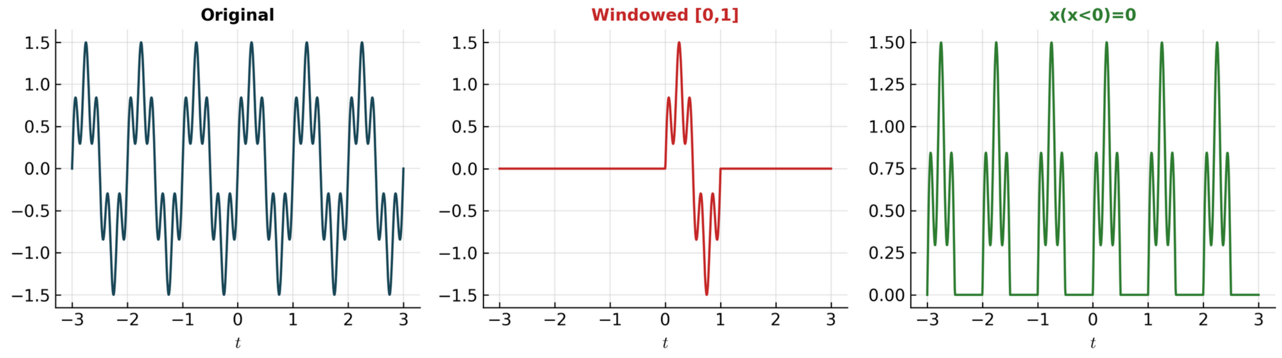

```
% x(t) = 3·sin(2π·5t + π/6)
A=3; f=5; T=1/f; phi=pi/6;
t=linspace(0,2*T,800);              % 2 περίοδοι
x=A*sin(2*pi*f*t+phi);
tms=1000*t;                         % μετατροπή σε ms
plot(tms, x, 'b', 'LineWidth', 3); grid on; hold on
line([min(tms) max(tms)],[A A],'Color','r','LineStyle','--','LineWidth',2)
line([min(tms) max(tms)],[-A -A],'Color','r','LineStyle','--','LineWidth',2)
line([1000*T 1000*T],[-A A],'Color','g','LineStyle','--','LineWidth',2)
xlabel('t (ms)'); title(sprintf('A=%g, f=%gHz, T=%gms',A,f,1000*T))
```

Λυμένη Άσκηση:

Δίνεται x(t)=3sin(10πt+π/6). Βρείτε: (α) A, (β) f, (γ) T, (δ) ω, (ε) φ.
Σχεδιάστε 2 πλήρεις περιόδους και σημειώστε T με κάθετη γραμμή.
Πού μηδενίζεται πρώτη φορά; Υπολογίστε t_0 αναλυτικά.

Hint: Ο κώδικας δεξιά είναι η λύση — μελετήστε τον γραμμή-γραμμή. **Εξάσκηση:** Δοκιμάστε να αλλάξετε τις παραμέτρους και παρατηρήστε τι αλλάζει.

Άσκηση 3: Windowing & Indexing




```
t = linspace(-3, 3, 1000);
x = sin(2*pi*t) + 0.5*sin(2*pi*5*t);  % σύνθετο
win = (t>=0) & (t<=1);               % παράθυρο [0,1]
y = x .* win;                        % windowing!
z = x; z(z<0) = 0;                   % logical indexing
subplot(1,3,1); plot(t,x); title('Αρχικό'); grid on
subplot(1,3,2); plot(t,y,'r'); title('Windowed'); grid on
subplot(1,3,3); plot(t,z,'g'); title('z(z<0)=0'); grid on
```

Λυμένη Άσκηση:

α) Δημιουργήστε x(t)=sin(2πt)+0.5sin(10πt) σε [-3,3].

β) Εφαρμόστε ορθογώνιο παράθυρο [0,1] (τι αλλάζει;).

γ) Μηδενίστε τα αρνητικά (logical indexing).

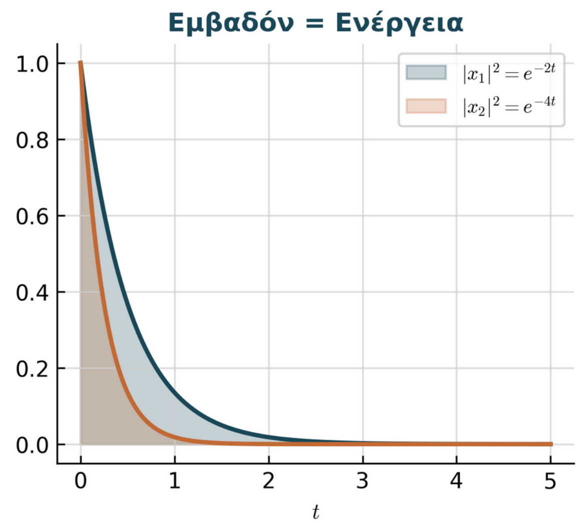
δ) Σχεδιάστε 3 subplots: πριν, μετά windowing, μετά indexing.

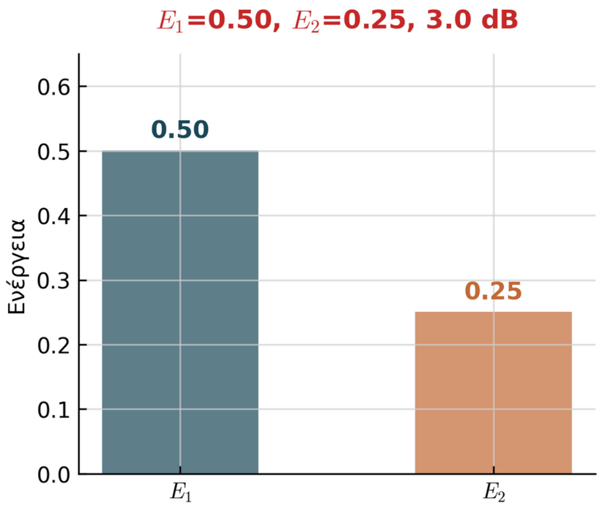
Hint: Ο κώδικας δεξιά είναι η λύση — μελετήστε τον γραμμή-γραμμή. **Εξάσκηση:** Δοκιμάστε να αλλάξετε τις παραμέτρους και παρατηρήστε τι αλλάζει.

ΕΞΑΣΚΗΣΗ

ΕΛΛΗΝΙΚΟ ΜΕΣΟΓΕΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ

Άσκηση 4: Ενέργεια & dB






```
t = linspace(0, 5, 5000); dt = t(2)-t(1);
x1 = exp(-t); x2 = exp(-2*t);
E1 = sum(x1.^2)*dt; E2 = sum(x2.^2)*dt;
dB_diff = 10*log10(E1/E2);
subplot(1,2,1); hold on
area(t, x1.^2, 'FaceColor', [0.1 0.28 0.35], 'FaceAlpha', 0.3);
area(t, x2.^2, 'FaceColor', [0.76 0.42 0.21], 'FaceAlpha', 0.3);
legend('|x1|^2','|x2|^2'); title('Εμβαδόν = Ενέργεια');
subplot(1,2,2);
b = bar([1 2], [E1 E2], 0.5); hold on
% Ξεχωριστά χρώματα:
bar(1, E1, 0.5, 'FaceColor', [0.1 0.28 0.35]);
bar(2, E2, 0.5, 'FaceColor', [0.76 0.42 0.21]);
set(gca,'XTickLabel',{'E1','E2'});
ylabel('Ενέργεια');
title(sprintf('E1=%.2f, E2=%.2f, %.1fdB', E1, E2, dB_diff));
```

Λυμένη Άσκηση:

α) Υπολογίστε αριθμητικά E_1 και E_2 για x_1=e^(-t) και x_2=e^(-2t).

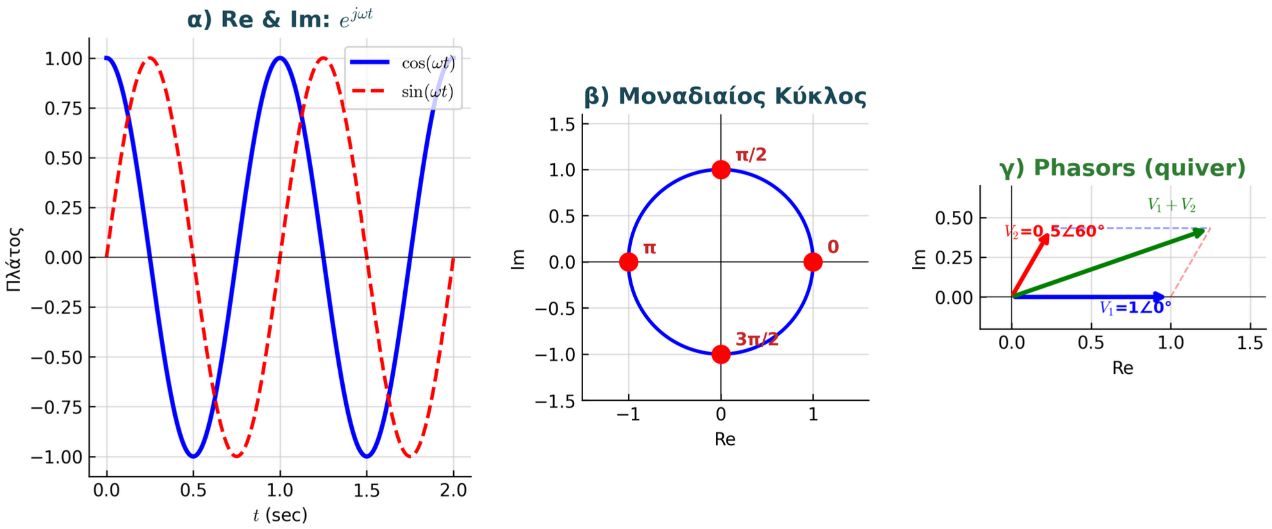
β) Ποια η αναλυτική τιμή; (∫₀^∞ e^(-2at)dt = 1/(2a))

γ) Υπολογίστε τη διαφορά σε dB. Τι σημαίνει πρακτικά;

δ) Σχεδιάστε $|x_1(t)|^2$ και $|x_2(t)|^2$ στο ίδιο plot.

Hint: Ο κώδικας δεξιά είναι η λύση — μελετήστε τον γραμμή-γραμμή. **Εξάσκηση:** Δοκιμάστε να αλλάξετε τις παραμέτρους και παρατηρήστε τι αλλάζει.

Άσκηση 5: Euler & Phasors




```
omega = 2*pi; t = linspace(0, 2, 500);
z = exp(j*omega*t);
% (α) Re & Im vs t
subplot(1,3,1);
plot(t, real(z), 'b', 'LineWidth', 2); hold on
plot(t, imag(z), 'r--', 'LineWidth', 2);
legend('cos','sin'); title('Re & Im');
xlabel('t (sec)'); ylabel('Πλάτος'); grid on
% (β) Κύκλος + 4 σημεία
subplot(1,3,2);
plot(real(z), imag(z), 'b', 'LineWidth', 2); hold on
ang = [0 pi/2 pi 3*pi/2];
plot(cos(ang), sin(ang), 'ro', 'MarkerSize', 10);
% labels στα 4 σημεία του μοναδιαίου κύκλου:
text( 0.05,  1.12, '0 (top)',  'FontSize',8);
text( 1.05,  0.05, '0°',       'FontSize',8);
text( 0.05, -1.18, '3π/2',     'FontSize',8);
text(-1.35,  0.05, 'π',         'FontSize',8);
xlabel('Re'); ylabel('Im');
axis equal; grid on; title('Κύκλος');
% (γ) Phasors ως βέλη (quiver)
subplot(1,3,3);
V1=1; V2=0.5*exp(j*pi/3); Vs=V1+V2;
quiver(0,0,real(V1),imag(V1),0,'b','LineWidth',2); hold on
quiver(0,0,real(V2),imag(V2),0,'r','LineWidth',2);
quiver(0,0,real(Vs),imag(Vs),0,'g','LineWidth',2);
xlabel('Re'); ylabel('Im');
legend('V1','V2','V1+V2'); axis equal; grid on
title('Phasors');
```

Λυμένη Άσκηση:

α) subplot(1,3,1): Re & Im (cos=μπλε, sin=κόκκινο).

β) subplot(1,3,2): Κύκλος + 4 κόκκινα σημεία (0, π/2, π, 3π/2).

γ) subplot(1,3,3): quiver: V_1=μπλε, V_2=κόκκινο, V_1+V_2=πράσινο.

δ) Υπολογίστε $|V_1+V_2|$ και $\angle(V_1+V_2)$ σε μοίρες.

Hint: Ο κώδικας δεξιά είναι η λύση — μελετήστε τον γραμμή-γραμμή. **Εξάσκηση:** Δοκιμάστε να αλλάξετε τις παραμέτρους και παρατηρήστε τι αλλάζει.

Εξάσκηση Α: Σύνθεση Σημάτων

Εκφώνηση — Λύστε μόνοι σας:

Κατασκευάστε το σήμα:

$x(t) = 2 \cdot u(t) - 3 \cdot u(t-1) + u(t-3)$

Σχεδιάστε σε [-1, 5]. Τι σχήμα βλέπετε;

Ποια η ενέργεια E; (αριθμητικά: sum(x.^2)*dt)

Bonus: Εκφράστε με rect() αντί u().

Χωρίς λύση: Δοκιμάστε μόνοι σας πρώτα! **Σε .m αρχείο:** Αποθηκεύστε σε askX_name.m, τρέξτε στο Octave.

Εξάσκηση Β: Περιοδικότητα

Εκφώνηση — Λύστε μόνοι σας:

Δίνονται: $x_1(t)=\cos(6\pi t)$, $x_2(t)=\cos(8\pi t)$, $x_3(t)=\cos(\sqrt{2}\cdot\pi t)$.

α) Βρείτε T_1, T_2, T_3 και f_1, f_2, f_3.

β) Είναι x_1+x_2 περιοδικό; Βρείτε $T_0=\mathrm{LCM}(T_1,T_2)$.

γ) Είναι x_1+x_3 περιοδικό; Γιατί;

δ) Σχεδιάστε x_1+x_2 σε $[0, 2\cdot T_0]$ και επαληθεύστε.

Χωρίς λύση: Δοκιμάστε μόνοι σας πρώτα! **Σε .m αρχείο:** Αποθηκεύστε σε askX_name.m, τρέξτε στο Octave.

Εξάσκηση Γ: Μιγαδικά Εκθετικά

Εκφώνηση — Λύστε μόνοι σας:

Δίνεται z(t) = e^((-1+j4π)t).

α) Ποιο το σ, ω, f;

β) Σχεδιάστε Re{z(t)} και Im{z(t)} σε [0, 3].

γ) Σχεδιάστε |z(t)| (περιβάλλουσα).

δ) Αλλάξτε σ=+1. Τι συμβαίνει; (ΠΡΟΣΟΧΗ: ylim!)

Χωρίς λύση: Δοκιμάστε μόνοι σας πρώτα! **Σε .m αρχείο:** Αποθηκεύστε σε askX_name.m, τρέξτε στο Octave.

Αναφορές & Πηγές (References)

Oppenheim & Willsky
Signals and Systems, 2nd Ed. (Prentice Hall)
Κεφ. 1-2: Βασικά σήματα

Μ. Παρασκευάς
Σήματα & Συστήματα με MATLAB/Octave, 3η Έκδ. (Τζιόλα, 2022)
Εύδοξος: 68402690 — Κεφ. 1: Σήματα CT

Ν. Ασημάκης & Μ. Αδάμ
Σήματα & Συστήματα (Κάλλιπος, 2015) — ΔΩΡΕΑΝ e-book
openbook.gr/simata-systimata — CC BY-NC-ND

Σημειώσεις Μαθήματος
4.004 Σήματα και Συστήματα — ΕΛΜΕΠΑ ΗΜΜΥ
eclass + σημειώσεις παλαιότερων ετών

Σημείωση: Οι αναφορές αφορούν κεφάλαια σχετικά με βασικά σήματα. **Octave:** octave.org/doc → πλήρες εγχειρίδιο αναφοράς.

Τέλος Εβδομάδας 2

Ερωτήσεις;

Επόμενη Εβδομάδα: Πράξεις σε Σήματα (Signal Operations)

ΕΛΛΗΝΙΚΟ ΜΕΣΟΓΕΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ