Σήματα και Συστήματα

Εβδομάδα 3: Πράξεις πάνω σε Σήματα

(με βάση τις παλαιότερες σημειώσεις του μαθήματος)

Ιωάννης Στεφανής

ΕΛΛΗΝΙΚΟ ΜΕΣΟΓΕΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ

Περιεχόμενα

Στόχοι Εβδομάδας 3

1. Πράξεις πλάτους

Κλιμάκωση, πρόσθεση, αφαίρεση, γινόμενο σημάτων

2. Πράξεις χρόνου

Ανάκλαση, μετατόπιση, κλίμακα — CT & DT

3. Σύνθεση

Κατασκευή σημάτων από u(t), r(t), masks

4. Κώδικας

Boolean masks, plot/stem, .m αρχεία Octave

Συνεχίζοντας από W02: Sinusoid (πράξεις πλάτους) + RC exponential (πράξεις χρόνου)

ΕΛΛΗΝΙΚΟ ΜΕΣΟΓΕΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ

Μικρή επανάληψη: CT/DT & Βασικά Εργαλεία

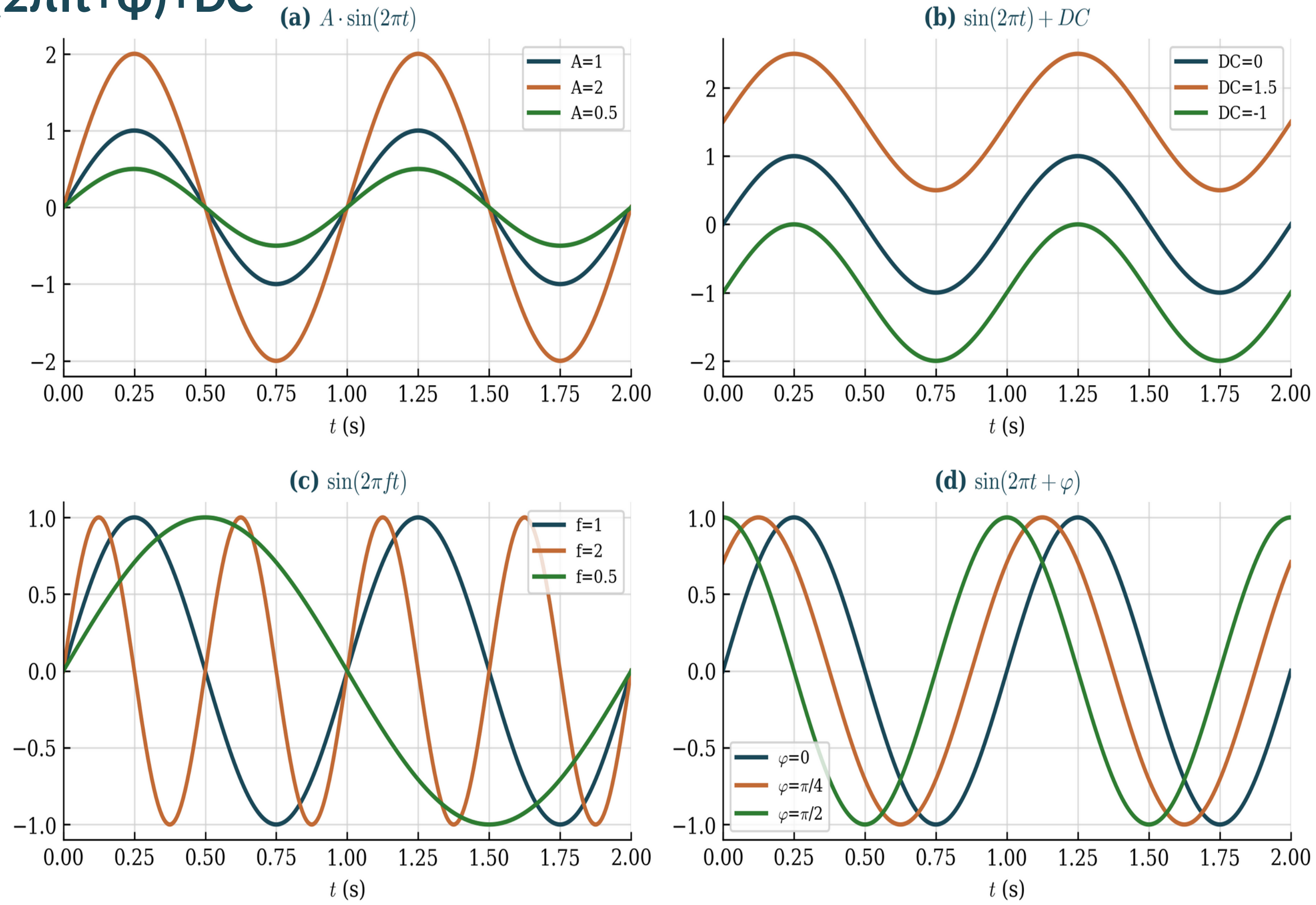
Υπενθύμιση: CT στον υπολογιστή = πυκνή δειγματοληψία. DT = ακέραιοι δείκτες.

Από την Εβδ. 2 θυμόμαστε:

- CT: $t \in \mathbb{R}$, συνεχής χρόνος → plot(t, x)
- DT: $n \in \mathbb{Z}$, διακριτός χρόνος → stem(n, x)

Μικρή επανάληψη: Ημιτονοειδές A·sin(2πft+φ)+DC

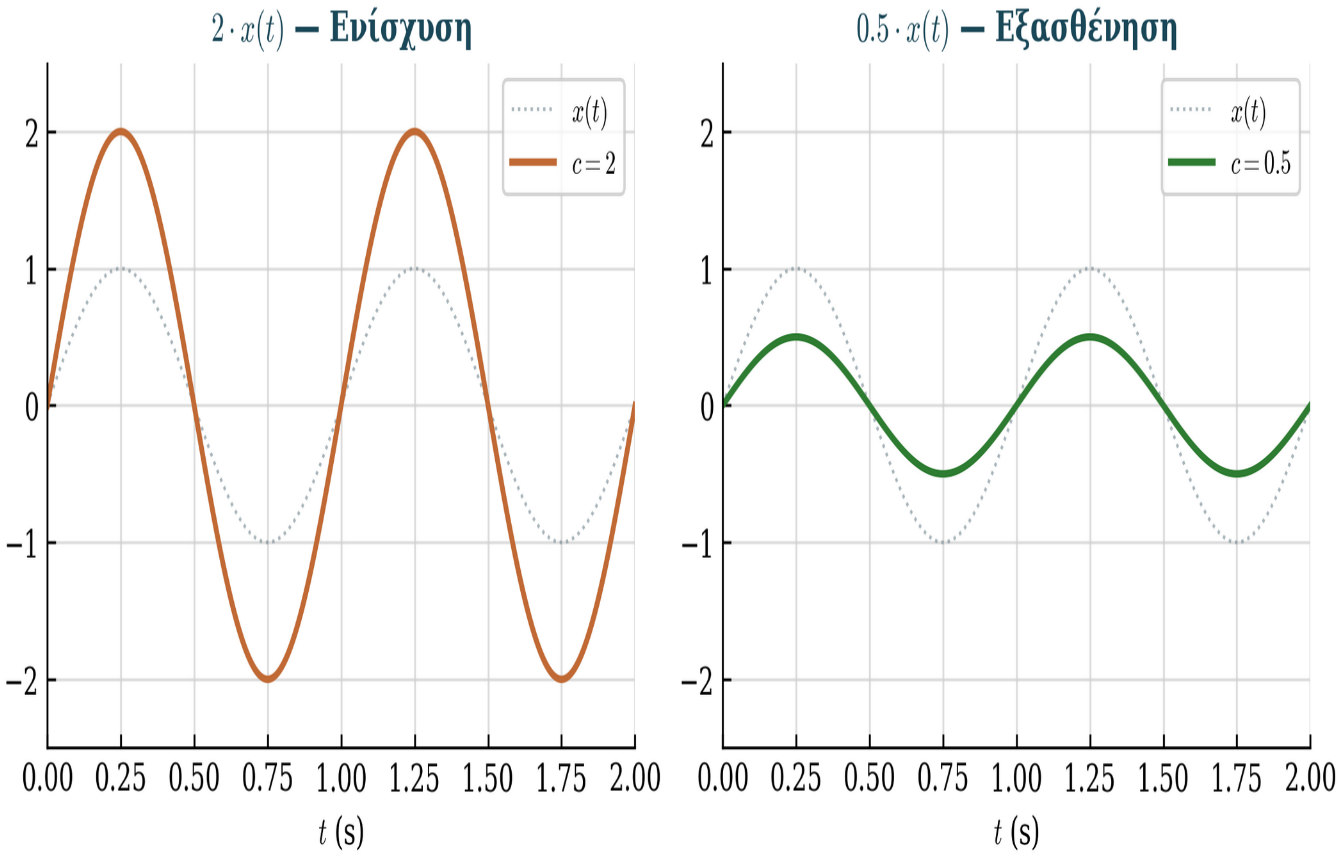
```
% Αλλαγή παραμέτρων ημιτονοειδούς
t = 0:0.001:2;
x1 = 2*sin(2*pi*t);       % A=2
x2 = sin(2*pi*t) + 1.5;   % DC=1.5
x3 = sin(2*pi*2*t);       % f=2 Hz
x4 = sin(2*pi*t + pi/4);  % φ=π/4

% Άσκηση για το σπίτι:
Δημιουργείστε το figure με τα subplots
```



Κλειδί: Η φάση φ είναι στην πραγματικότητα μετατόπιση χρόνου! Θα το δούμε αργότερα.

Κλιμάκωση Πλάτους: y(t) = c·x(t)

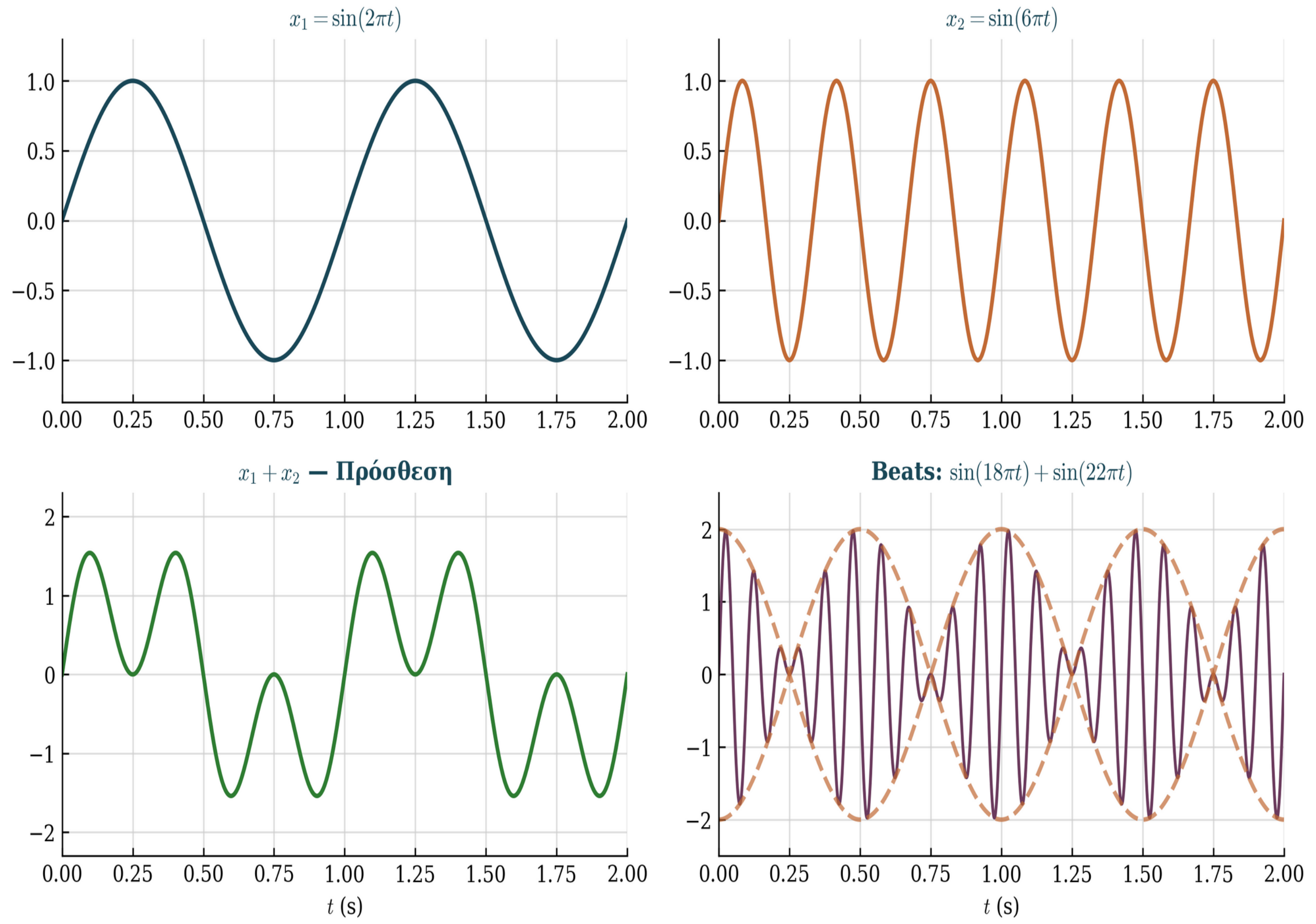
```
% Κλιμάκωση πλάτους ημιτόνου
t = 0:0.001:2;
x = sin(2*pi*t);          % αρχικό
y1 = 2*x;                 % c=2 → ενίσχυση
y2 = 0.5*x;               % c=0.5 → εξασθένηση

subplot(1,2,1); plot(t,x,t,y1); title("2·x(t)");
subplot(1,2,2); plot(t,x,t,y2); title("0.5·x(t)");
```



Κανόνας: c>1 ενίσχυση, 0<c<1 εξασθένηση.

Πρόσθεση Σημάτων & Beats

```
% Πρόσθεση ημιτονοειδών

t = 0:0.001:2;
x1 = sin(2*pi*t);        % 1 Hz
x2 = sin(2*pi*3*t);      % 3 Hz
y = x1 + x2;             % πρόσθεση

% Beats: δύο κοντινές f
b1 = sin(2*pi*9*t);      % 9 Hz
b2 = sin(2*pi*11*t);     % 11 Hz
beats = b1 + b2;         % f_beat=2Hz
subplot(2,1,1); plot(t, y);
subplot(2,1,2); plot(t, beats);
```



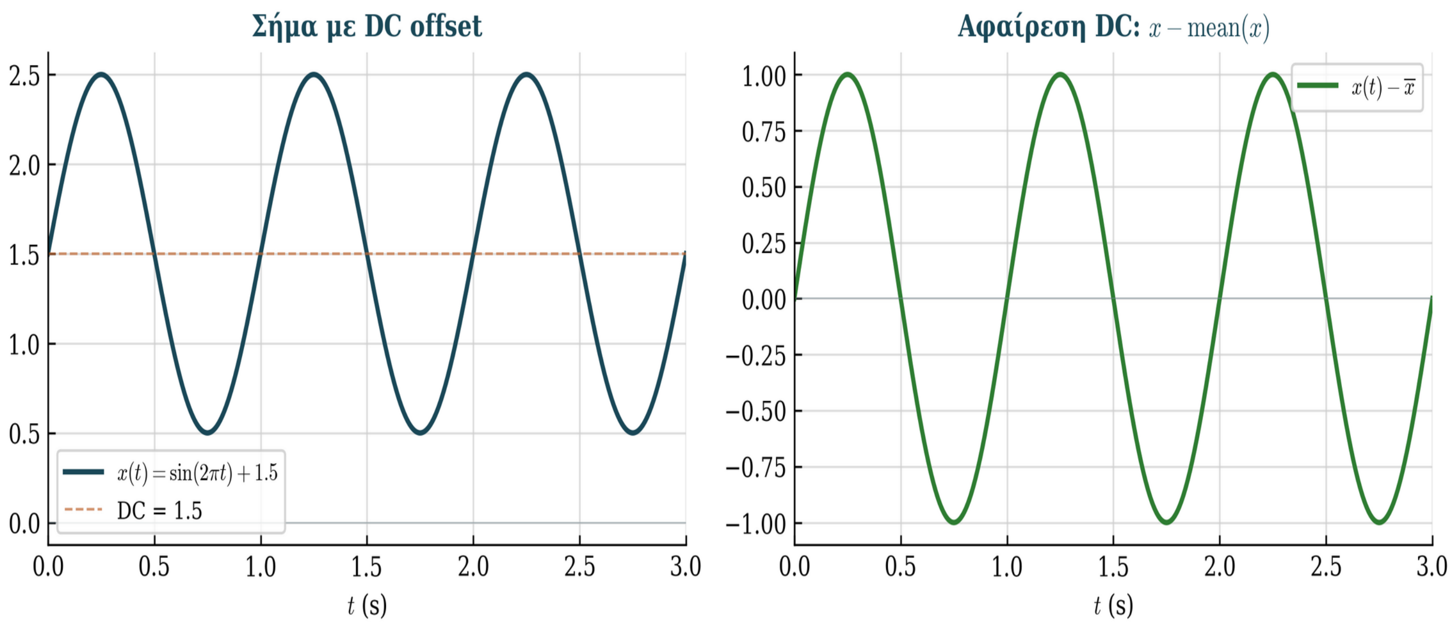
Beats = «χτυπήματα». Όταν 2 ημίτονα κοντινών συχνοτήτων f_1, f_2 προστεθούν, το πλάτος «ανοιγοκλείνει» με ρυθμό $|f_1 - f_2|$ Hz.

$\sin(2\pi f_1 t)+\sin(2\pi f_2 t)$ =
$2\cdot\cos(2\pi\cdot\Delta f/2\cdot t)\cdot\sin(2\pi\cdot f\_avg\cdot t)$
↑ envelope (αργό) ↑ carrier (γρήγορο)

Beats: Δύο κοντινές συχνότητες f_1, f_2 → «χτύπημα» στη συχνότητα $f_1 - f_2$.

DC Offset & Αφαίρεση Μέσης Τιμής

```
% DC offset: μετακίνηση κατά σταθερά
t = 0:0.001:3;
x = sin(2*pi*t) + 1.5;    % σήμα με DC=1.5

% Αφαίρεση DC offset
x_centered = x - mean(x); % αφαίρεση μέσης τιμής

subplot(1,2,1); plot(t,x); title("Με DC offset");
subplot(1,2,2); plot(t,x_centered); title("Χωρίς DC");
```



Γιατί δουλεύει η αφαίρεση mean();

Κάθε σήμα γράφεται ως: x(t) = x_AC(t) + DC
όπου DC = σταθερό μέρος, x_AC = μεταβαλλόμενο.

Ο μέσος όρος του x_AC είναι ΜΗΔΕΝ (ταλαντώνεται ίσα πάνω-κάτω). Άρα: mean(x) = mean(x_AC) + DC = 0 + DC = DC

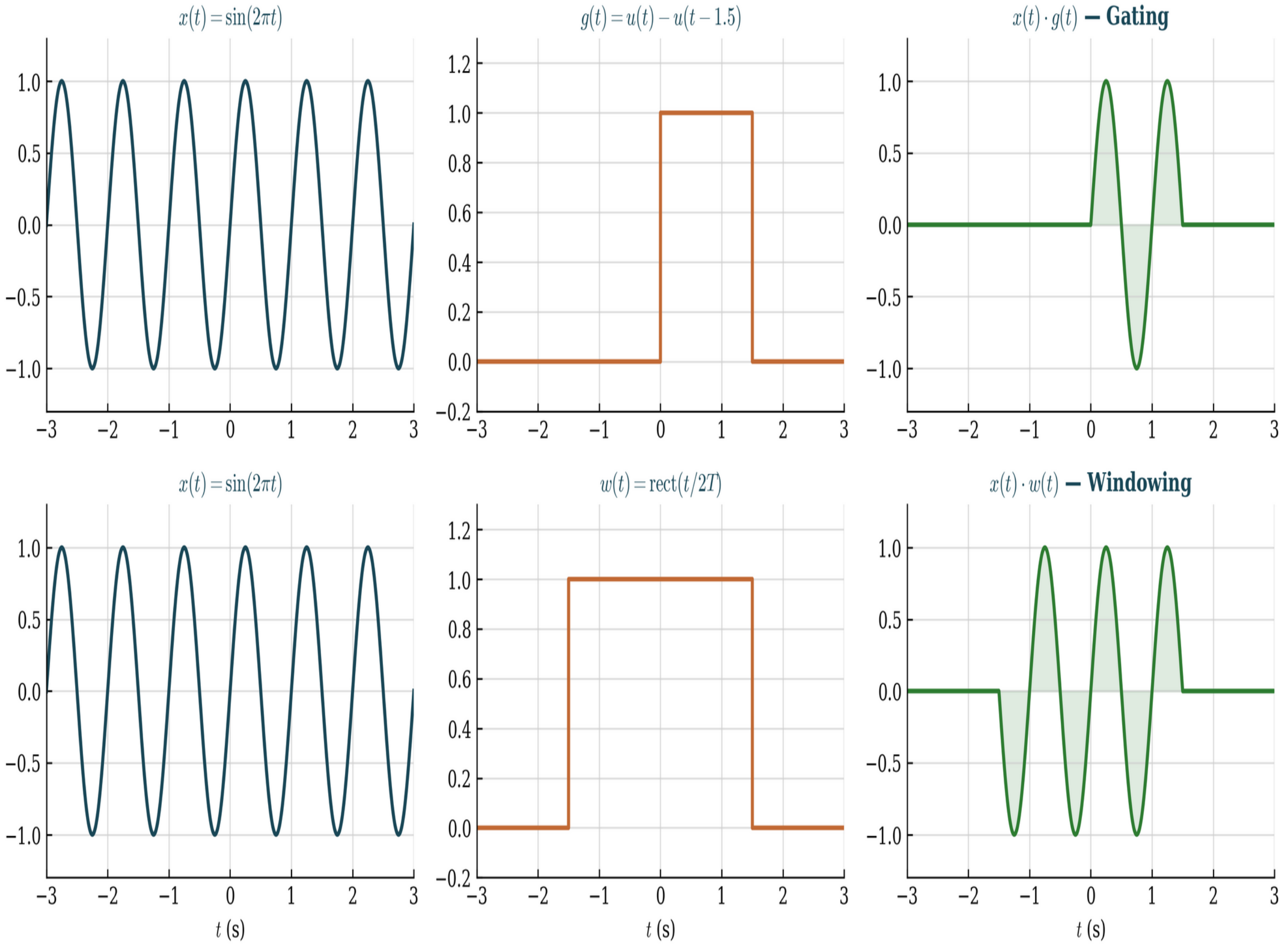
Επομένως: x(t) − mean(x) = x_AC(t) + DC − DC = x_AC(t)

Η αφαίρεση mean() το «κεντράρει» γύρω από το 0.

Πρακτικό: Αισθητήρες που δίνουν σήμα με DC offset.

Γινόμενο Σημάτων - Gating & Windowing

```
% Γινόμενο: gating + windowing
t = -3:0.001:3;
x = sin(2*pi*t);

% Gating: κόβω το σήμα σε [0, 1.5]
gate = (t>=0) & (t<=1.5);  % boolean mask
gated = x .* gate;

% Windowing: κόβω συμμετρικά [-T, T]
T_win = 1.5;
window = (abs(t) <= T_win);
windowed = x .* window;
```



Κλειδί: Gating & windowing = γινόμενο σήματος × mask. Το mask «ανοίγει/κλείνει» το σήμα.

Αναστροφή πλάτους: Προσοχή −x(t) ≠ x(−t) !

−x(t): Αλλάζει η ΤΙΜΗ (↕), ο χρόνος μένει ίδιος
x(−t): Αλλάζει ο ΧΡΟΝΟΣ (↔), η τιμή μένει ίδια

Στο ημίτονο μοιάζουν — στον παλμό φαίνεται η διαφορά!

```
%Πλήρες παράδειγμα: -x(t) vs x(-t)
t = -4:0.01:4;
x = (t >= 0) & (t <= 2);  % παλμός [0, 2]

% (1) Αναστροφή πλάτους: -x(t)
y_amp = -x;               % ΜΟΝΟ αλλαγή τιμής

% (2) Ανάκλαση χρόνου: x(-t)
t_flip = -t;              % ΝΕΟΣ άξονας χρόνου
y_time = x;               % ίδιες τιμές, νέος t

figure;
subplot(1,3,1); plot(t, x, 'LineWidth',2);
  title('x(t)'); ylim([-1.5 1.5]);
subplot(1,3,2); plot(t, y_amp,'r','LineWidth',2);
  title('-x(t): πλάτος'); ylim([-1.5 1.5]);
subplot(1,3,3); plot(t_flip, y_time,'b','LineWidth',2);
  title('x(-t): χρόνος'); ylim([-1.5 1.5]);
```

Πράξεις Πλάτους - Σύνοψη

Πράξη	Τύπος	Octave	Εφαρμογή
Κλιμάκωση	$y=c\cdot x(t)$	y = c*x;	Gain / attenuation
Πρόσθεση	$y=x_1+x_2$	y = x1 + x2;	Σύνθεση, beats
Αφαίρεση	$y=x_1-x_2$	y = x1 - x2;	Noise cancellation
DC Offset	$y=x+DC$	y = x + 1.5;	Sensor offset
Γινόμενο	$y=x_1\cdot x_2$	y = x1 .* x2;	Gating, windowing

Συνέχεια: πράξεις στον χρόνο!

Αλλάζουμε το t, ΟΧΙ το y.

ΕΛΛΗΝΙΚΟ ΜΕΣΟΓΕΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ

Ενότητα: Μετασχηματισμοί Χρόνου

Ανάκλαση → Μετατόπιση → Κλίμακα → Συνδυασμός x(at−b)

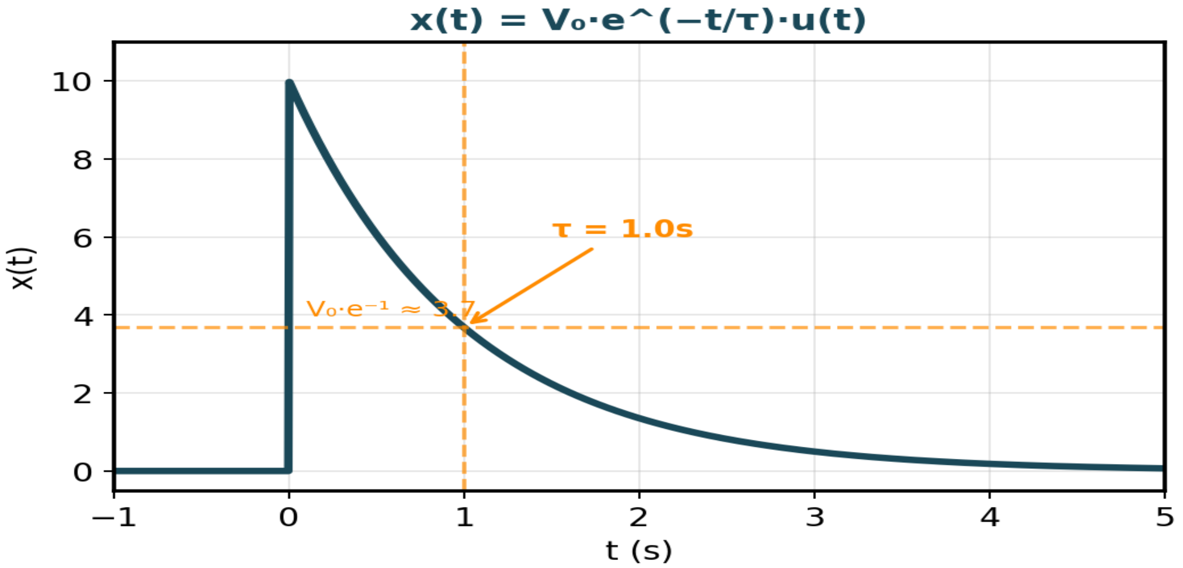
Βασικό σήμα-οδηγός: RC Exponential

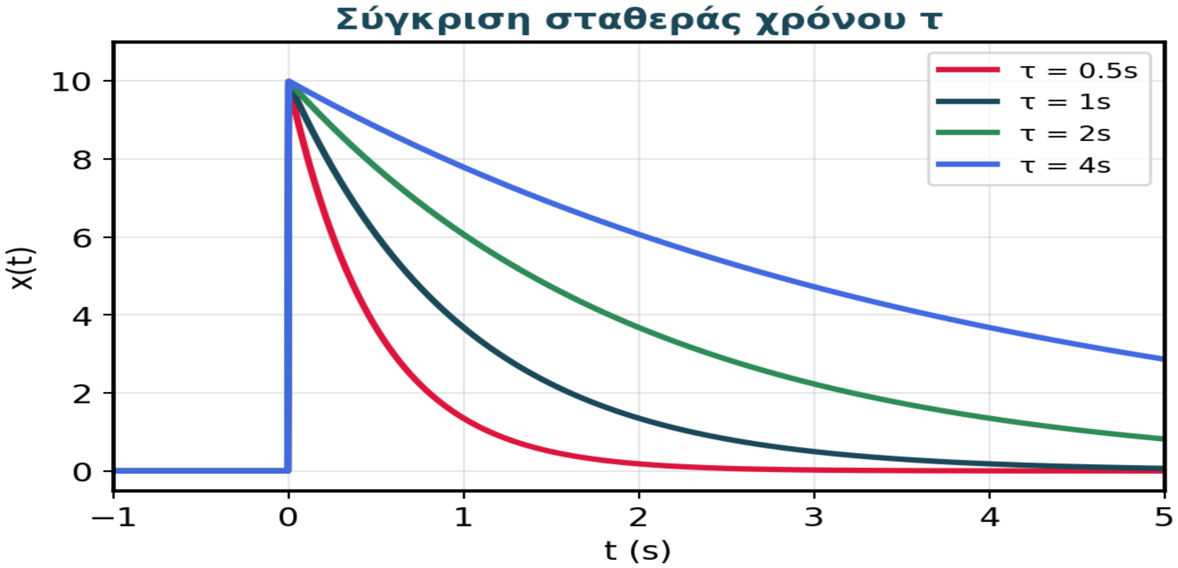
RC κύκλωμα: $x(t) = V_0 \cdot e^{(-t/\tau)} \cdot u(t)$
$\tau = R \cdot C$ (σταθερά χρόνου)

Γιατί 63,2%;
Φόρτιση: $V(t) = V_{max} \cdot (1 - e^{(-t/RC)})$
Στο $t = \tau = RC$:
$V(\tau) = V_{max} \cdot (1 - e^{(-1)})$
$e \approx 2{,}718 \rightarrow e^{(-1)} \approx 0{,}368$
$V(\tau) = V_{max} \cdot (1 - 0{,}368) = 0{,}632 \cdot V_{max}$

Αποφόρτιση: $V(t) = V_{max} \cdot e^{(-t/RC)}$
$V(\tau) = V_{max} \cdot 0{,}368 \rightarrow 36{,}8\%$

Πρακτικός κανόνας:
1τ: 63% | 2τ: 86% | 3τ: 95% | 5τ: ≈99%



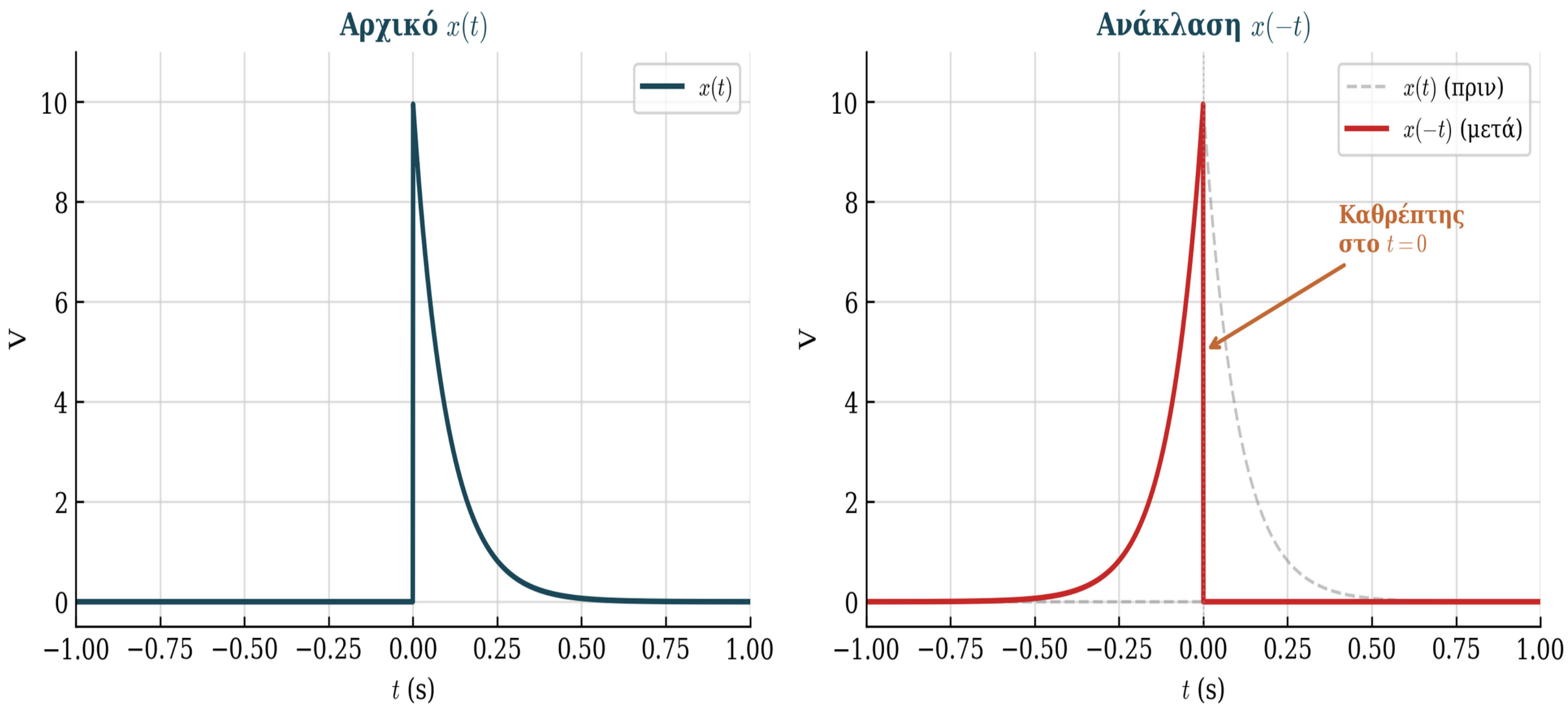


Ανάκλαση (Reflection): y(t) = x(−t)

Αντικαθιστούμε κάθε t με −t.
Γεωμετρικά: αντικατοπτρισμός ως προς t = 0.

Απλά παραδείγματα:
u(t) → u(−t): βαθμίδα «γυρνά» αριστερά
r(t) → r(−t): ράμπα «γυρνά» αριστερά

RC σήμα:
$x(t) = V_0 \cdot \exp(-t/\tau) \cdot u(t)$ → ζει στο t ≥ 0
$x(-t) = V_0 \cdot \exp(t/\tau) \cdot u(-t)$ → ζει στο t ≤ 0

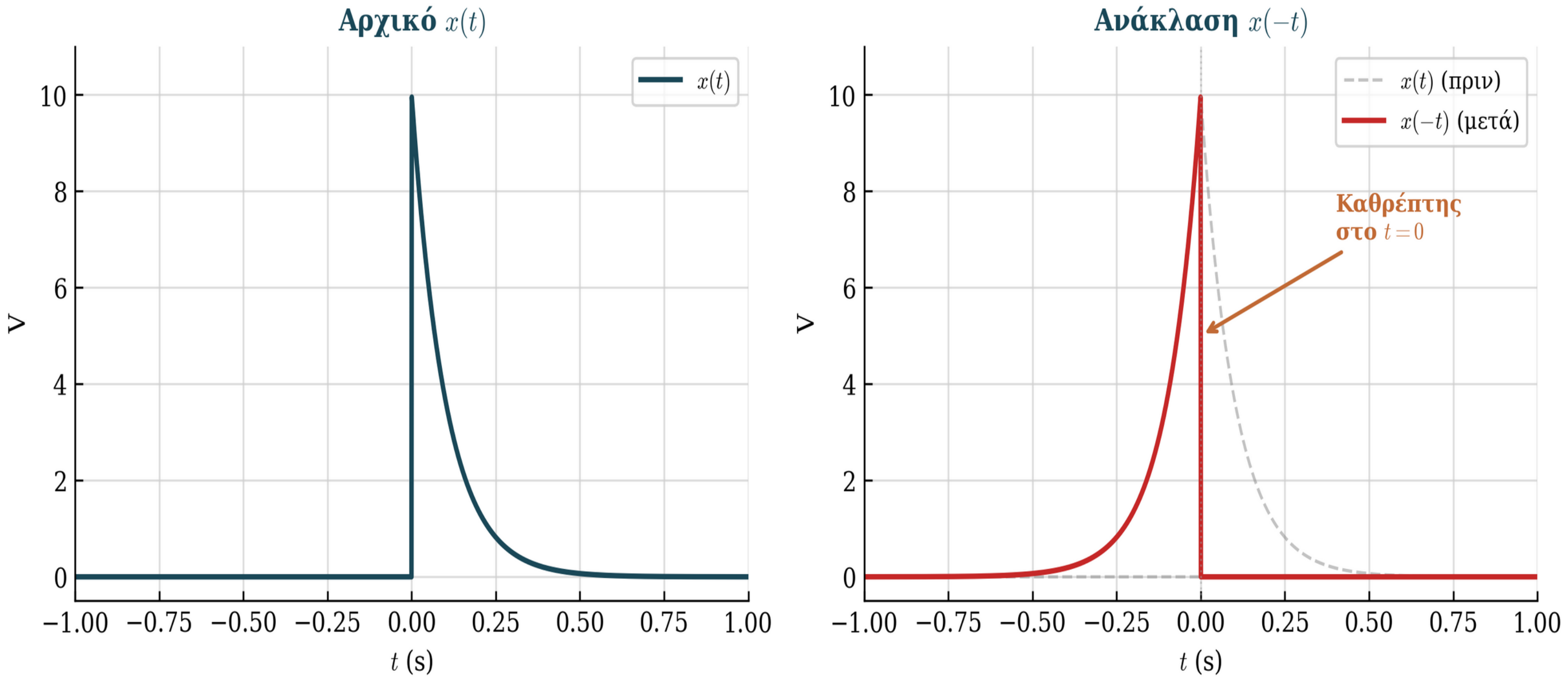


Ανάκλαση: το σήμα «αντικατοπτρίζεται» ως προς τον κατακόρυφο άξονα t = 0. Γκρι = πριν, Κόκκινο = μετά.

Κώδικας:
tx = [t>=0]; → ty = [t<=0];
ή: ty = tx(end:-1:1); (ΟΧΙ fliplr!)

Ανάκλαση CT: Κώδικας Octave

```
% Ανάκλαση RC σήματος
R = 1e3; C = 1e-4; V0 = 10;   % R=1kΩ, C=100μF
tau = R*C;                    % τ = 0.1s
t = -1:0.001:1;
tx = [t>=0];                  % u(t)
ty = [t<=0];                  % u(-t)
x = tx.*V0.*exp(-t./tau);     % x(t)
y = ty.*V0.*exp(t./tau);      % x(-t)

subplot(1,2,1); plot(t,x); title("x(t)");
subplot(1,2,2); plot(t,y); title("x(-t)");
```



Ανάκλαση: το σήμα «αντικατοπτρίζεται» ως προς τον κατακόρυφο άξονα t = 0. Γκρι = πριν, Κόκκινο = μετά.

Γιατί φτιάχνουμε νέο διάνυσμα χρόνου;

Πράξεις πλάτους (σ6): y = c*x → αλλάζει ΜΟΝΟ η τιμή (y-axis). Ο χρόνος t μένει ίδιος.
→ plot(t, y) με το ΙΔΙΟ t.

Πράξεις χρόνου (εδώ): y(t) = x(−t) → αλλάζει ΠΟΥ βρίσκεται η τιμή (x-axis). Χρειαζόμαστε νέο t!
→ t_new = -t; ή t_new = t - t0;

Κανόνας: Πλάτος → ίδιος χρόνος. Χρόνος → νέος χρόνος.

Χρονική Μετατόπιση: $y(t) = x(t - t_0)$

$t_0 > 0 \rightarrow$ δεξιά (καθυστέρηση)
$t_0 < 0 \rightarrow$ αριστερά (πρόωρη εμφάνιση)

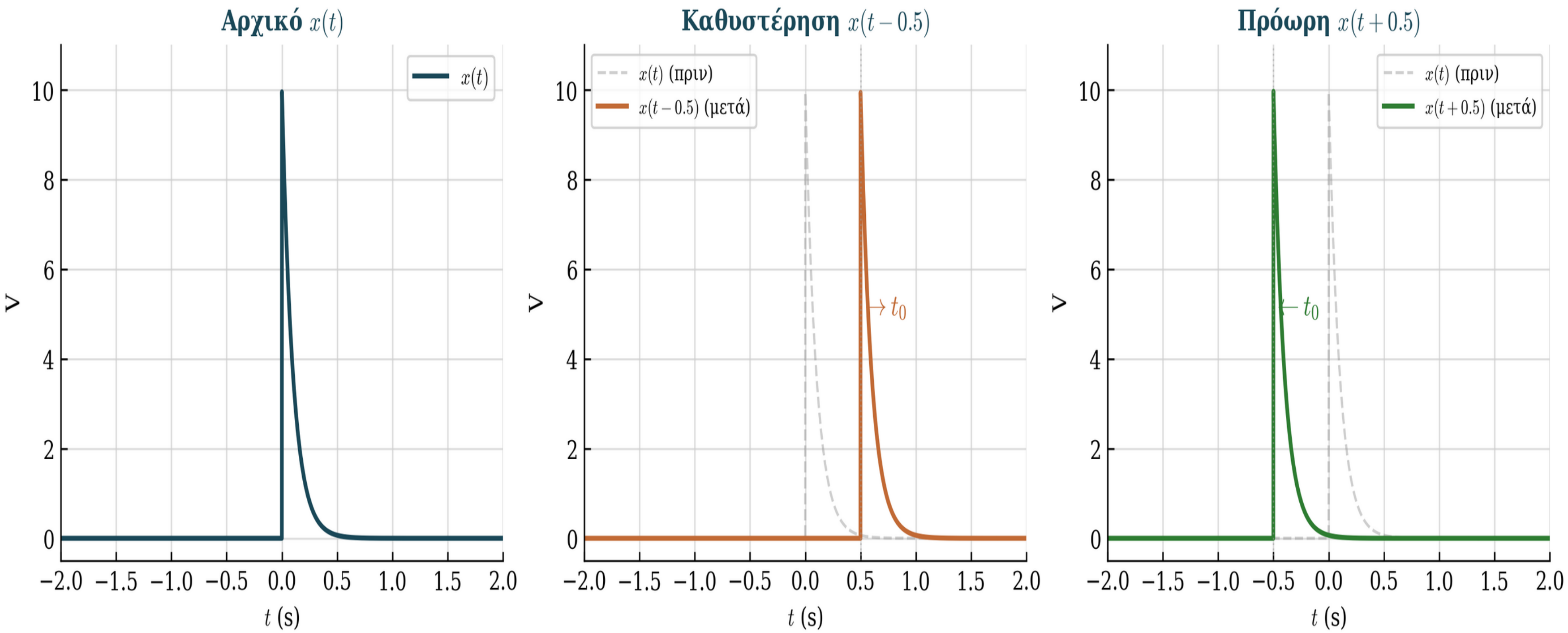
ΠΡΟΣΟΧΗ: $x(t-3)$ = shift ΔΕΞΙΑ κατά 3!
(Αντιδιαισθητικό: μείον = δεξιά!)

Στο DT: $y[n] = x[n - n_0]$ ($n_0 \in \mathbb{Z}$ ακέραιος)

Σύνδεση με ημιτονοειδές:
$\sin(2\pi ft + \varphi) = \sin(2\pi f(t - t_0))$
$\varphi \leftrightarrow$ μετατόπιση χρόνου!

Μετατόπιση CT - Κώδικας

```
% Μετατόπιση RC σήματος

R=1e3; C=1e-4; V0=10; t0=0.5;

ts=0.00001; t=-2:ts:2;

bx=[t>=0]; by=[t-t0>=0]; bg=[t+t0>=0];

x = bx.*V0.*exp(-t./(R.*C));
y = by.*V0.*exp(-(t-t0)./(R.*C));
g = bg.*V0.*exp(-(t+t0)./(R.*C));

subplot(1,3,1); plot(t,x); title("x(t)");
subplot(1,3,2); plot(t,y); title("x(t-0.5)");
subplot(1,3,3); plot(t,g); title("x(t+0.5)");
```



Μετατόπιση: $t_0>0$ → δεξιά (καθυστέρηση), $t_0<0$ → αριστερά (πρόωρη). Γκρι διακεκομμένο = αρχικό σήμα.

Κλειδί: Αλλάζουμε ΜΟΝΟ τα t μέσα στο σήμα: $t \to (t-t_0)$. Η boolean mask ακολουθεί!

Κλίμακα Χρόνου: y(t) = x(at)

a > 1 → συστολή (compression) — πιο γρήγορο
0 < a < 1 → διαστολή (expansion) — πιο αργό
a < 0 → ανάκλαση + κλίμακα

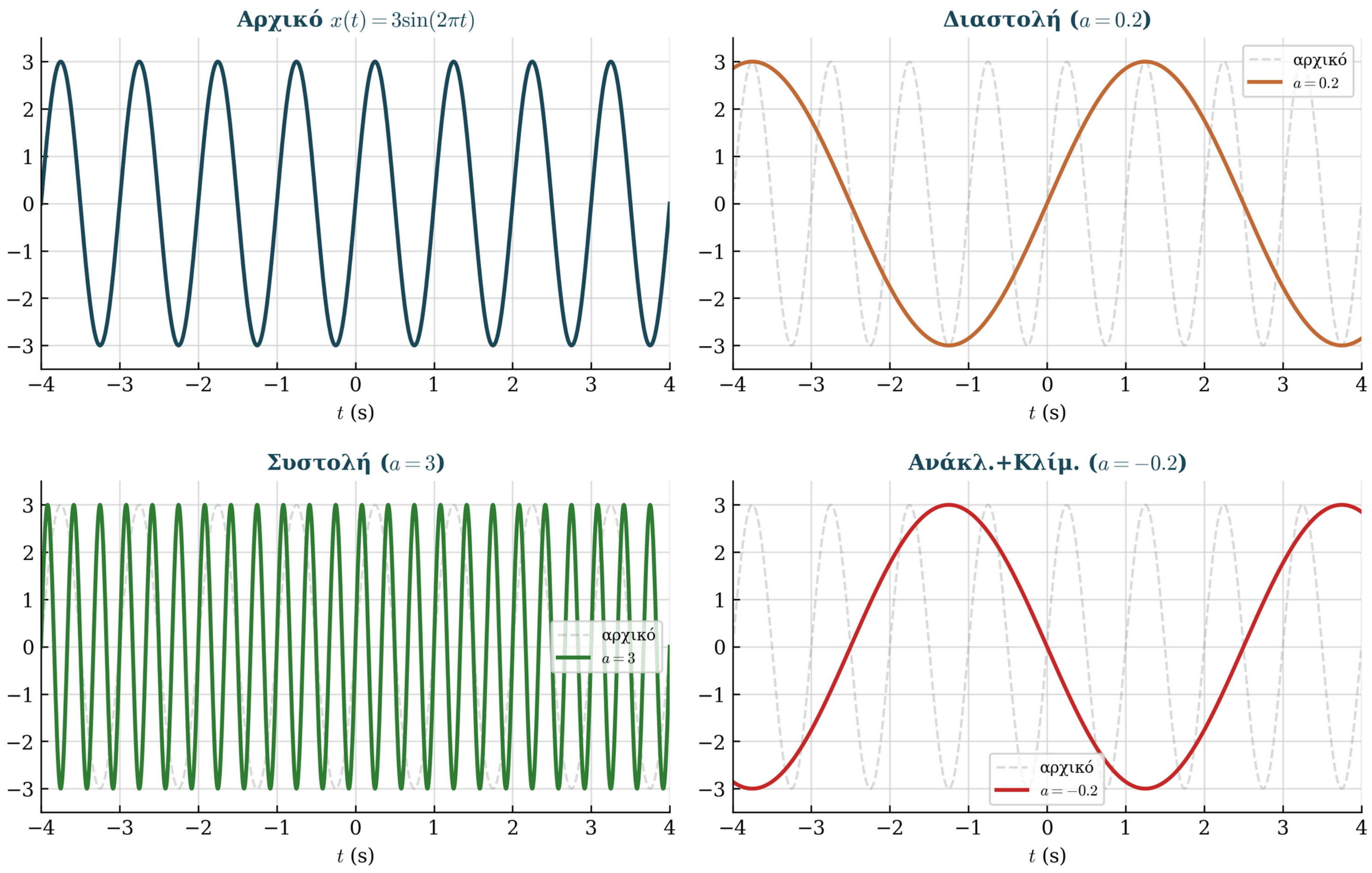
Αντιδιαισθητικό: |a|>1 → σήμα «στενεύει»!

Παράδειγμα: slow motion = a=0.5 (διαστολή)
Fast forward = a=2 (συστολή)

Προσοχή:
Νέα περίοδος = T/|a|
a=0.2 → T'=5T (πενταπλάσια)

Κλίμακα Χρόνου CT: Κώδικας

```
% Κλίμακα χρόνου — ημίτονο
T = 1; t = -4:0.001:4;
x  = 3.*sin(2.*pi.*t./T);       % αρχικό
y1 = 3.*sin(2.*pi.*0.2.*t./T);  % a=0.2 → διαστολή
y2 = 3.*sin(2.*pi.*3.*t./T);    % a=3 → συστολή
y3 = 3.*sin(2.*pi.*(-0.2).*t./T); % a=-0.2

subplot(2,2,1); plot(t,x);  title("Αρχικό");
subplot(2,2,2); plot(t,y1); title("a=0.2");
subplot(2,2,3); plot(t,y2); title("a=3");
subplot(2,2,4); plot(t,y3); title("a=-0.2");
```



Κλίμακα χρόνου: a=0.2 → διαστολή (αργότερο), a=3 → συστολή (γρηγορότερο), a<0 → ανάκλαση+κλίμακα. Γκρι = αρχικό.

Σημείωση: a=0.2 → περίοδος πενταπλασιάζεται. a=3 → γίνεται T/3.

DT Scaling - Προσοχή: Decimation vs Interpolation

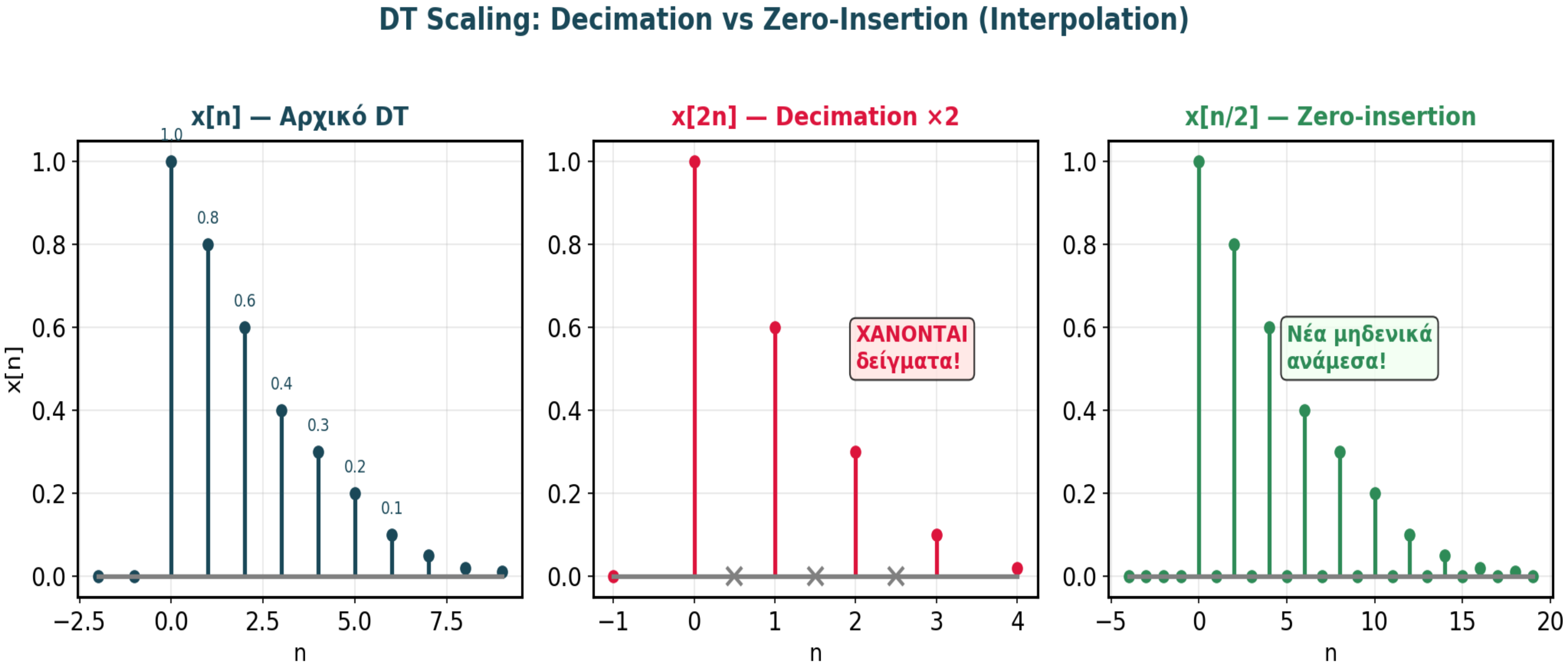
CT κλίμακα x(at): «αθώα» — απλά αλλάζει ρυθμός.
Παράδειγμα: x(2t) σε t=0.7 → x(1.4) ΥΠΑΡΧΕΙ.
Κάθε t δίνει τιμή, δεν χάνεται τίποτα.

DT κλίμακα x[an]: **ΠΡΟΒΛΗΜΑ**!
x[2n] σε n=1 → x[2]
x[2n] σε n=1.5 → x[3]... αλλά n=1.5 ΔΕΝ ΥΠΑΡΧΕΙ!
Τα DT δείγματα είναι **ΜΟΝΟ** σε ακεραίους.

Decimation x[2n]: Κρατάμε κάθε 2ο →
ΧΑΝΟΥΜΕ τα ενδιάμεσα
x = [1 .8 .6 .4 .3] → x[2n] = [1 .6 .3] (χάθηκαν **.8**, **.4**)

Zero-insertion x[n/2]: **Βάζουμε 0 ανάμεσα**
x = [1 .8 .6] → x[n/2] = [1 **0** .8 **0** .6 0]

```
% DT Decimation (x[2n]):
n = 0:9; x =
[1 .8 .6 .4 .3 .2 .1 .05 .02 .01];
n_dec = 0:4;
x_dec = x(1:2:end);  % κάθε 2ο δείγμα
stem(n_dec, x_dec);

% Zero-insertion (x[n/2]):
x_up = zeros(1, 2*length(x));
x_up(1:2:end) = x;  % μηδενικά ανάμεσα
stem(0:length(x_up)-1, x_up);
```



ΚΛΕΙΔΙ: CT κλίμακα = αλλαγή ταχύτητας (αναστρέψιμο).
DT κλίμακα = αλλαγή αριθμού δειγμάτων (ΜΗ ΑΝΑΣΤΡΕΨΙΜΟ)!

Συνδυασμός: y(t) = x(at − b)

ΣΩΣΤΗ σειρά:

1. ΠΡΩΤΑ μετατόπιση κατά b/a
2. ΜΕΤΑ κλίμακα κατά a

y(t) = x(a(t − b/a))

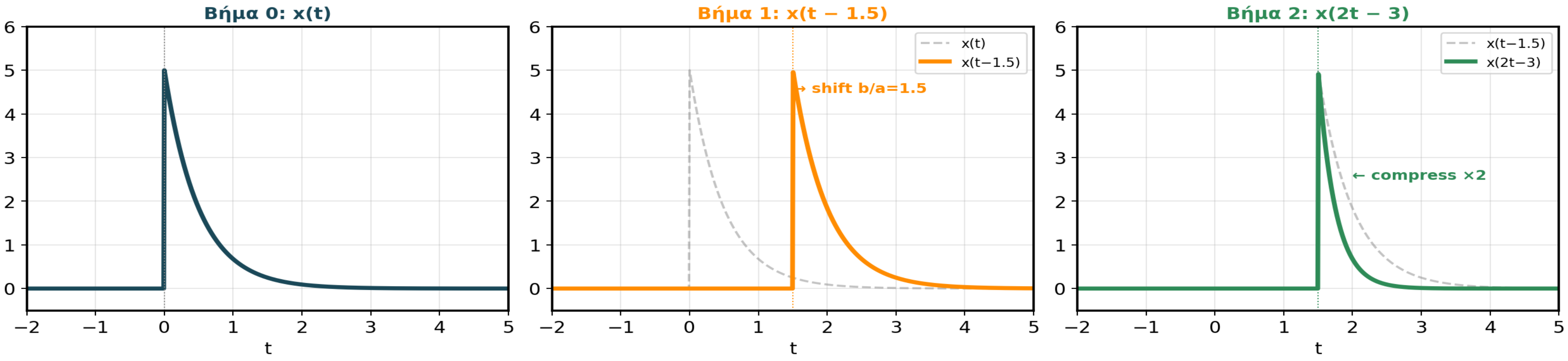
Αν αντιστρέψουμε τη σειρά → ΛΑΘΟΣ!

Παράδειγμα: y(t) = x(2t−3)

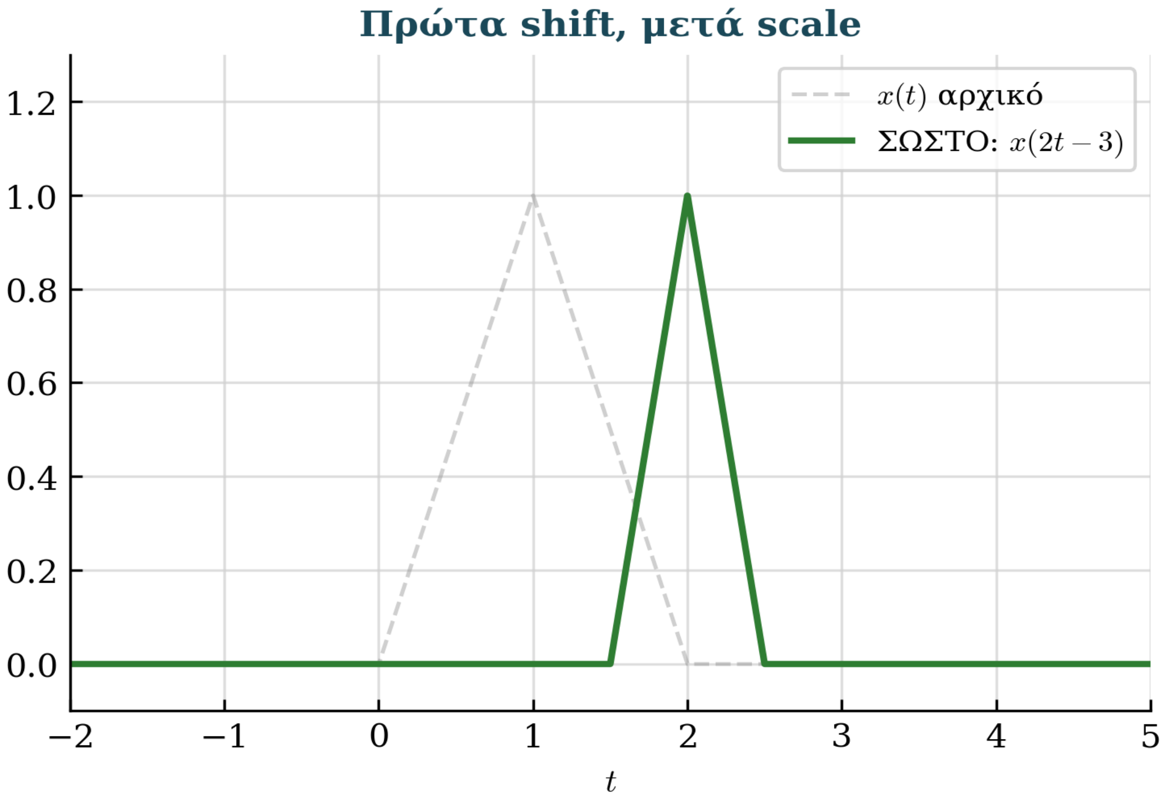
Βήμα 1: shift κατά 3/2=1.5

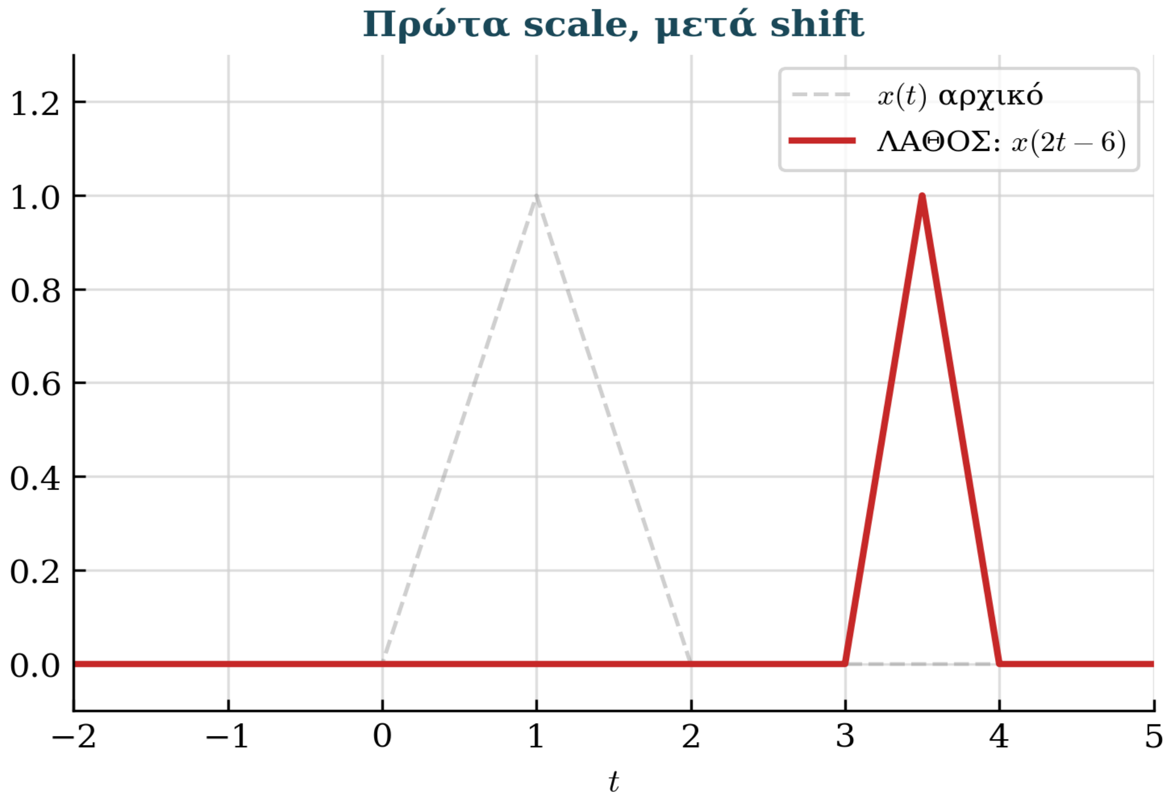
Βήμα 2: scale κατά 2

```
% x(2t-3): shift πρώτα, scale μετά
t=-2:0.001:5; tau=0.5; V0=5;
x = V0*exp(-t/tau).*(t>=0);
t_s = t-1.5; y = V0*exp(-t_s*2/tau).*(t_s*2>=0);
```



Γιατί η σειρά μετρά για σωστό αποτέλεσμα!

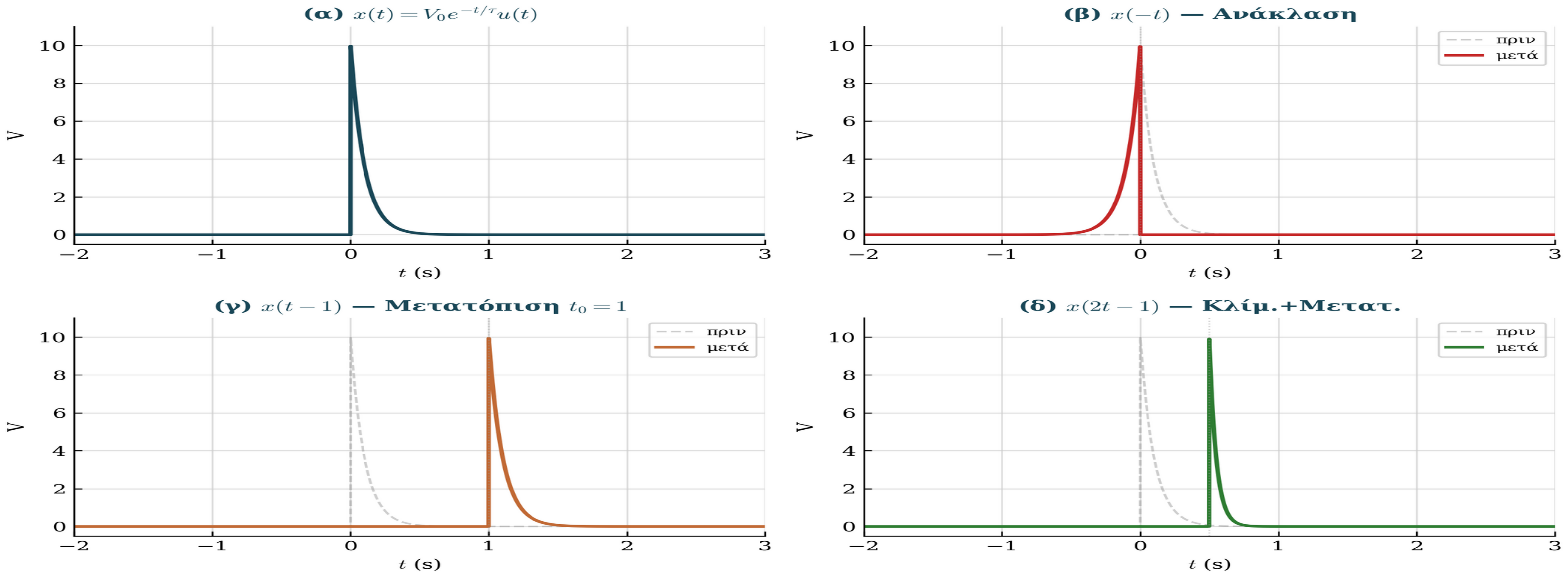




Η σειρά ΜΕΤΡΑΕΙ! Αριστερά = σωστό x(2t−3), Δεξιά = λάθος x(2t−6). Γκρι = αρχικό x(t).

ΣΩΣΤΟ:
shift 1.5, scale 2 → x(2t−3)
ΛΑΘΟΣ:
scale 2, shift 3 → x(2t−6) ≠ x(2t−3)

RC Σήμα: Ακολουθία Μετασχηματισμών



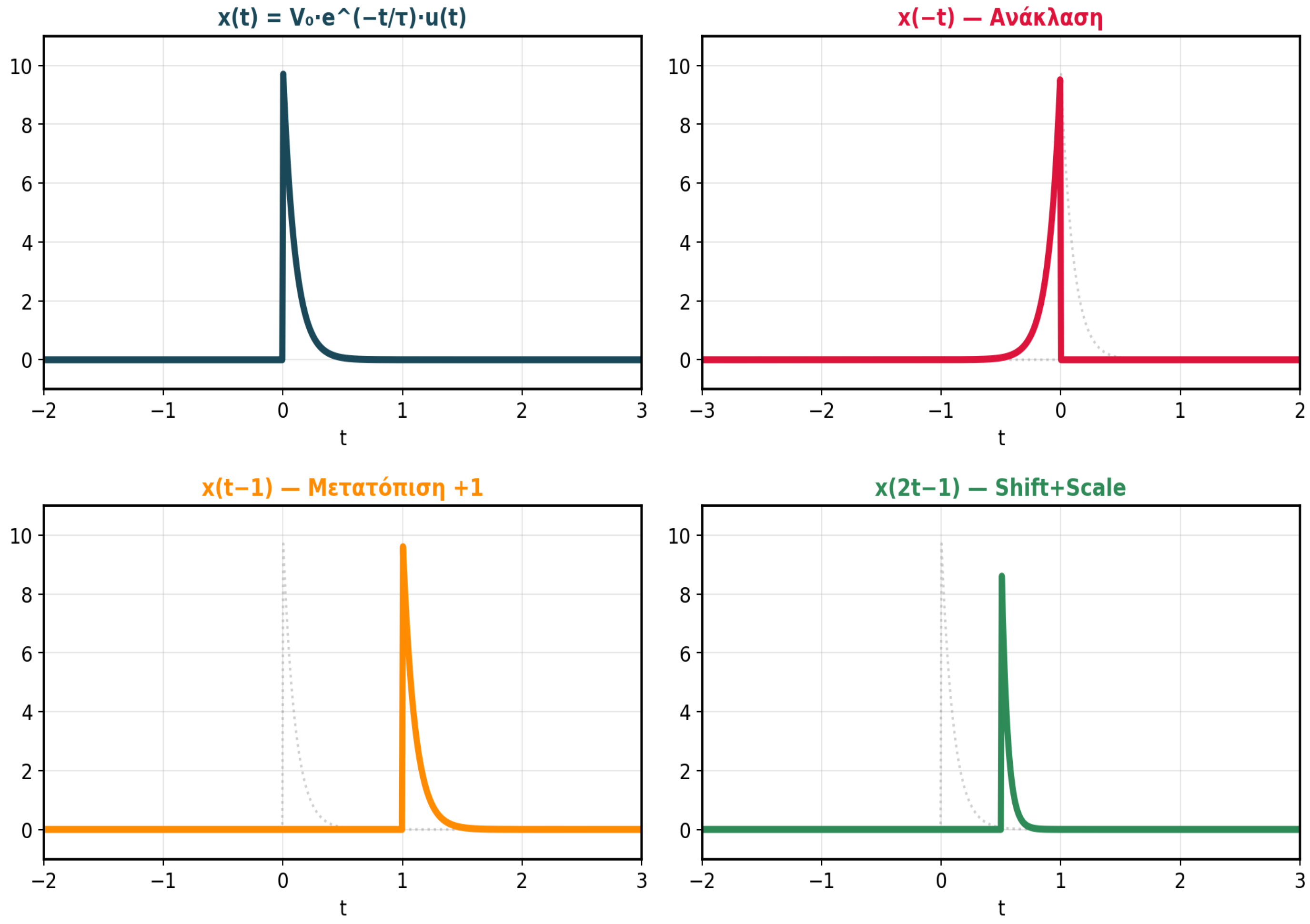
RC: κάθε βήμα δείχνει το αρχικό x(t) γκρι διακεκομμένο. Χρώμα = αποτέλεσμα μετασχηματισμού.

Προσοχή: Το ίδιο RC μετασχηματίζεται: αρχικό → ανάκλαση → μετατόπιση → κλίμακα+μετατόπιση.

RC Μετασχηματισμοί: Κώδικας

```
% Ακολουθία μετασχηματισμών RC
R=1e3; C=1e-4; V0=10; tau=R*C;
t=-2:0.001:3;

x = V0.*exp(-t./tau).*(t>=0);          % αρχικό
x_ref = V0.*exp(t./tau).*(t<=0);       % ανάκλαση
x_shift = V0.*exp(-(t-1)./tau).*(t>=1); % μετατόπιση
x_comb = V0.*exp(-(2*t-1)./tau).*(2*t-1>=0); % κλίμ.
+μετ.

subplot(2,2,1); plot(t,x);       title("x(t)");
subplot(2,2,2); plot(t,x_ref);   title("x(-t)");
subplot(2,2,3); plot(t,x_shift); title("x(t-1)");
subplot(2,2,4); plot(t,x_comb);  title("x(2t-1)");
```

RC Σήμα: Ακολουθία Μετασχηματισμών



Σημαντικό: Στο x(2t−1) αντικαθιστούμε ΠΑΝΤΟΥ: exp ΚΑΙ boolean mask.

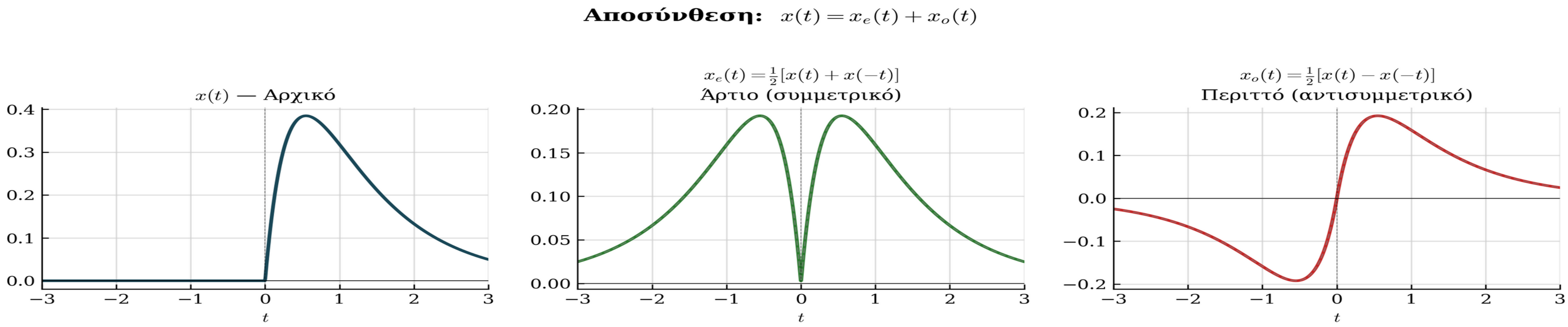
Even/Odd: Αποσύνθεση Σήματος

Κάθε σήμα αποσυντίθεται σε:

Άρτιο: $x_e(t) = ½[x(t) + x(-t)]$ → συμμετρικό
Περιττό: $x_o(t) = ½[x(t) - x(-t)]$ → αντισυμμετρικό

Πάντα: $x(t) = x_e(t) + x_o(t)$

Το γράφημα δείχνει RC σήμα → αποσύνθεση:



Πλήρης αποσύνθεση + κώδικας + plots → Εργαστήριο 4

Even/Odd: Ορισμός & Τύποι

Even/Odd Decomposition — Ορισμός

Άρτιο (Even) μέρος :

$$x_e(t) = \frac{1}{2}[x(t) + x(-t)]$$

Ιδιότητα: $x_e(-t) = x_e(t)$ (συμμετρικό ως προς $t = 0$)

Περιττό (Odd) μέρος :

$$x_o(t) = \frac{1}{2}[x(t) - x(-t)]$$

Ιδιότητα: $x_o(-t) = -x_o(t)$ (αντισυμμετρικό ως προς $t = 0$)

Απόδειξη πληρότητας :

$$x_e(t) + x_o(t) = \frac{1}{2}[x(t) + x(-t)] + \frac{1}{2}[x(t) - x(-t)] = \frac{2\,x(t)}{2} = x(t) \quad \checkmark$$

Ειδικές περιπτώσεις :

Αν $x(t) = x(-t)$ → ήδη άρτιο → $x_o(t) = 0$ (π.χ. $\cos(t),\ t^2,\ e^{-|t|}$)

Αν $x(t) = -x(-t)$ → ήδη περιττό → $x_e(t) = 0$ (π.χ. $\sin(t),\ t^3$)

Octave code :

```
x_flip = x(end:-1:1);          % x(−t)
x_e = 0.5*(x + x_flip);        % Άρτιο
x_o = 0.5*(x - x_flip);        % Περιττό
```

Even/Odd: Μαθηματικός Υπολογισμός

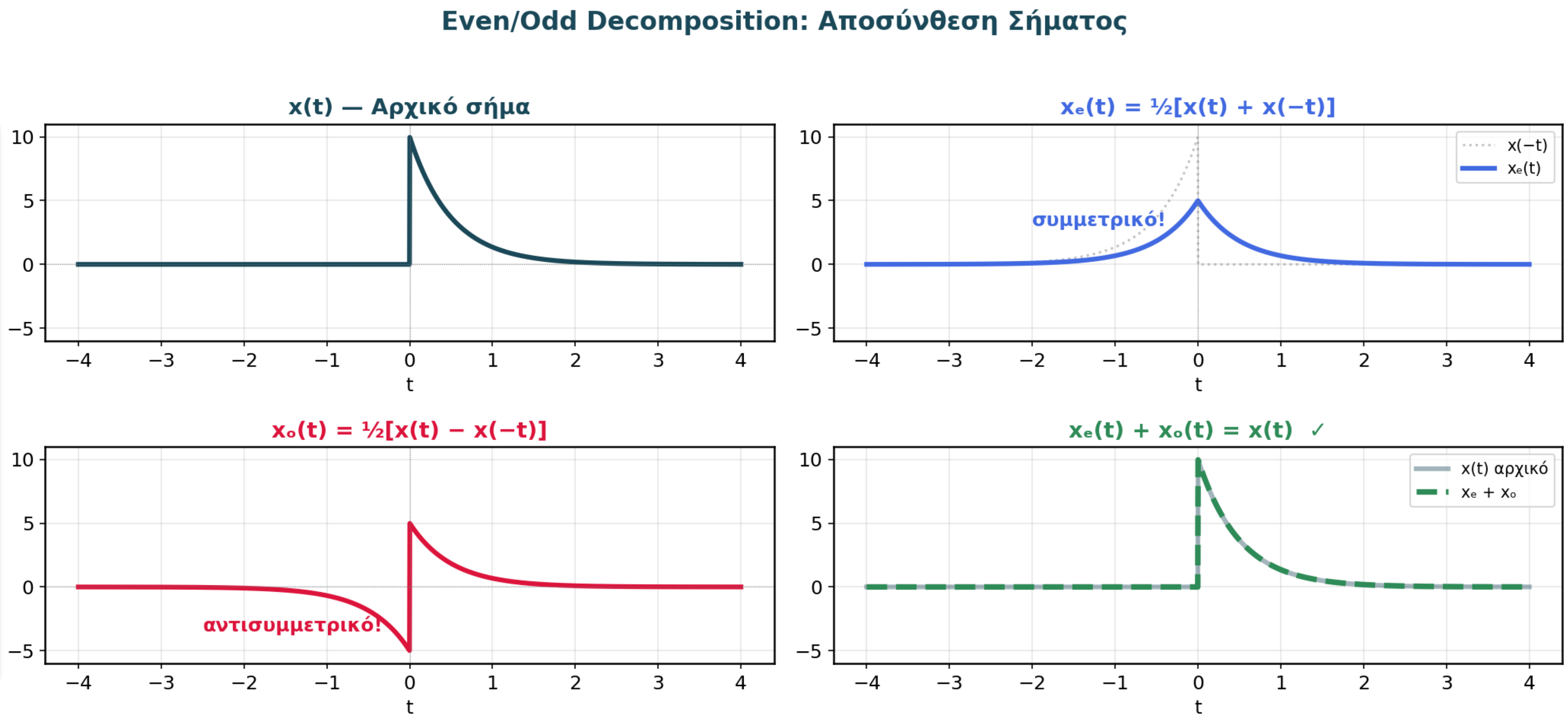
Τύπος αποσύνθεσης:

$x_e(t) = ½ \cdot [x(t) + x(-t)]$ (Άρτιο/Even)
$x_o(t) = ½ \cdot [x(t) - x(-t)]$ (Περιττό/Odd)

Ιδιότητα: $x(t) = x_e(t) + x_o(t)$ ΠΑΝΤΑ!

Octave:

```
 % Even/Odd Decomposition
%(t πρέπει να είναι συμμετρικός!)
x_flip = x(end:-1:1);      % = x(-t)
x_e = 0.5*(x + x_flip);    % Άρτιο
x_o = 0.5*(x - x_flip);    % Περιττό
```



Εισαγωγή: Θεώρημα Δειγματοληψίας

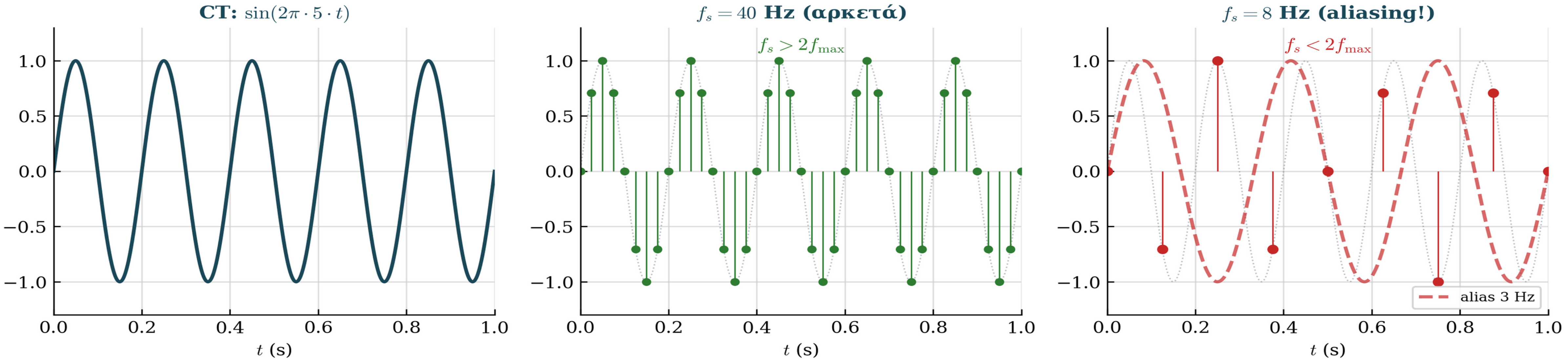
Πώς αναπαριστά ο υπολογιστής ένα CT σήμα;
→ Παίρνει «φωτογραφίες» κάθε T_s δευτερόλεπτα.

$f_s = 1/T_s$ = ρυθμός δειγματοληψίας (samples/s)

Θεώρημα Nyquist–Shannon:
$f_s \geq 2 \cdot f\_max$

Αν $f_s < 2 \cdot f\_max$ → ALIASING
(βλέπουμε «ψεύτικη» χαμηλότερη συχνότητα)

Θεώρημα Nyquist:
$f_s \geq 2 \cdot f\_max$
Αλλιώς: aliasing!



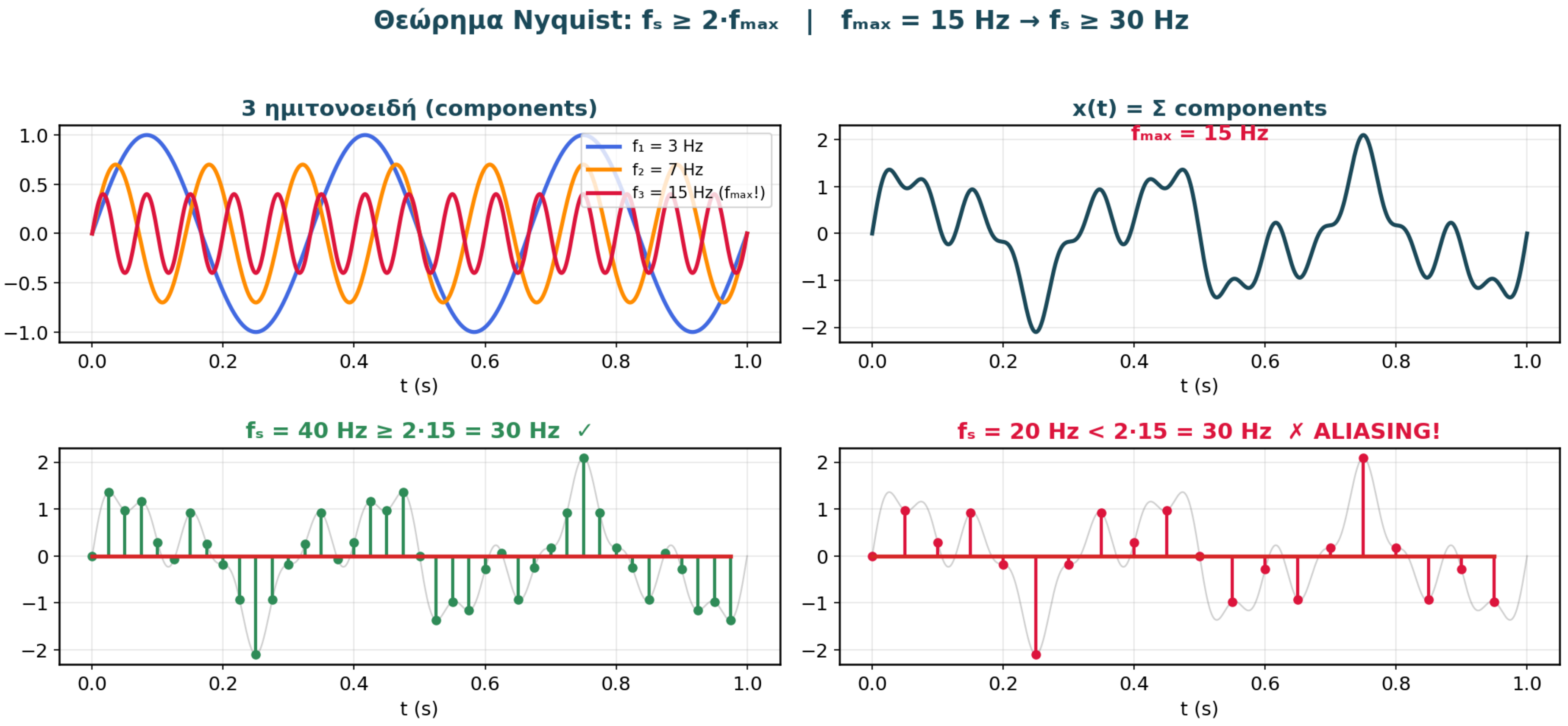
Nyquist: Άθροισμα Ημιτόνων & f_s

Σύνθετο σήμα = άθροισμα ημιτόνων:
$x(t) = \sin(2\pi \cdot 3t) + 0.7 \cdot \sin(2\pi \cdot 7t) + 0.4 \cdot \sin(2\pi \cdot 15t)$

f_{max} = 15 Hz (η ψηλότερη συχνότητα!)

Nyquist: $f_s \geq 2 \cdot f_{max} = 2 \cdot 15 = 30$ Hz

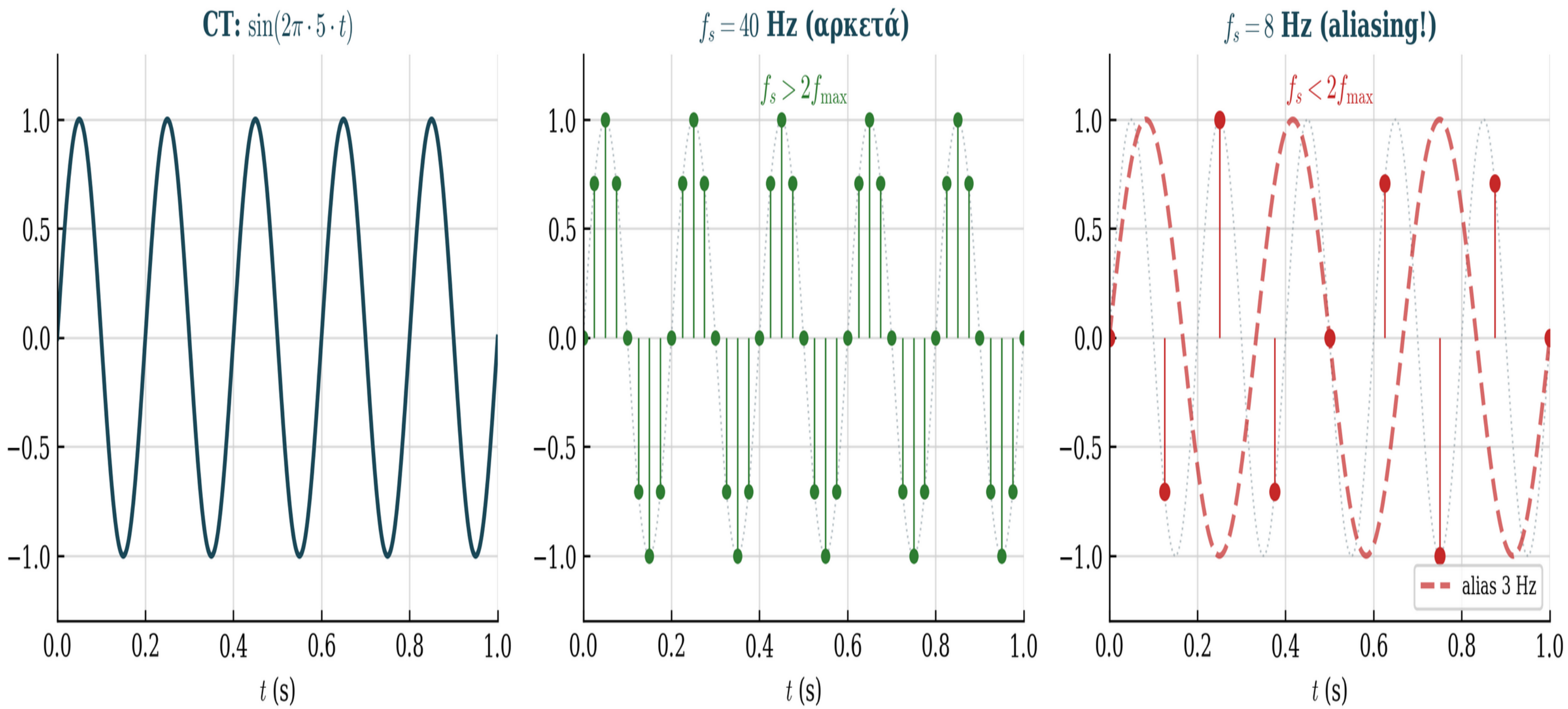
Αν $f_s < 30$ Hz → aliasing στο 15 Hz component!



Στην πραγματικότητα παίρνουμε fs >> 2fmax

Δειγματοληψία: Σωστή vs Λάθος

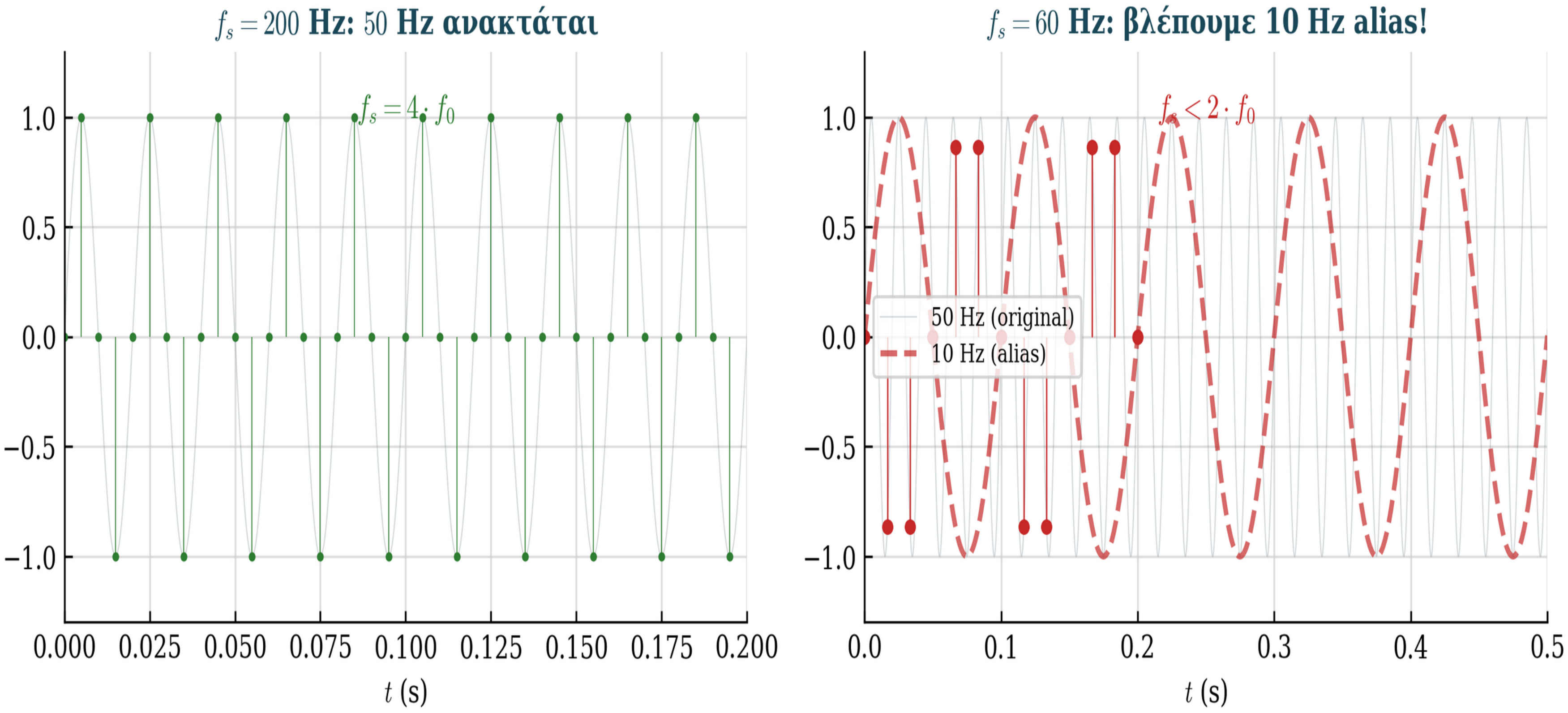
```
% Δειγματοληψία ημιτόνου 5 Hz
f0 = 5;  t = 0:0.0002:1;   % «συνεχές»
x_ct = sin(2*pi*f0*t);

% (α) Αρκετά: fs=40 Hz (40 > 2×5=10)
fs1=40; n1=0:fs1; t1=n1/fs1;
x1 = sin(2*pi*f0*t1);

% (β) Λίγα: fs=8 Hz (8 < 10) → ALIASING!
fs2=8; n2=0:fs2; t2=n2/fs2;
x2 = sin(2*pi*f0*t2);
```



Aliasing: Στα 8 Hz, το σήμα 5 Hz φαίνεται σαν 3 Hz (= 8−5). Ψεύτικη συχνότητα!

Aliasing: Πρακτικό Παράδειγμα

```
% Aliasing: 50 Hz σήμα
f_true = 50;  t = 0:0.00001:0.5;
x = sin(2*pi*f_true*t);

% (α) fs=200: σωστό (200 > 2×50)
fs1=200; t1=(0:fs1*0.2)/fs1;
x1 = sin(2*pi*f_true*t1);

% (β) fs=60: ALIASING (60 < 100)
fs2=60; t2=(0:fs2*0.5)/fs2;
x2 = sin(2*pi*f_true*t2);
% alias = |fs - f0| = 10 Hz!
```



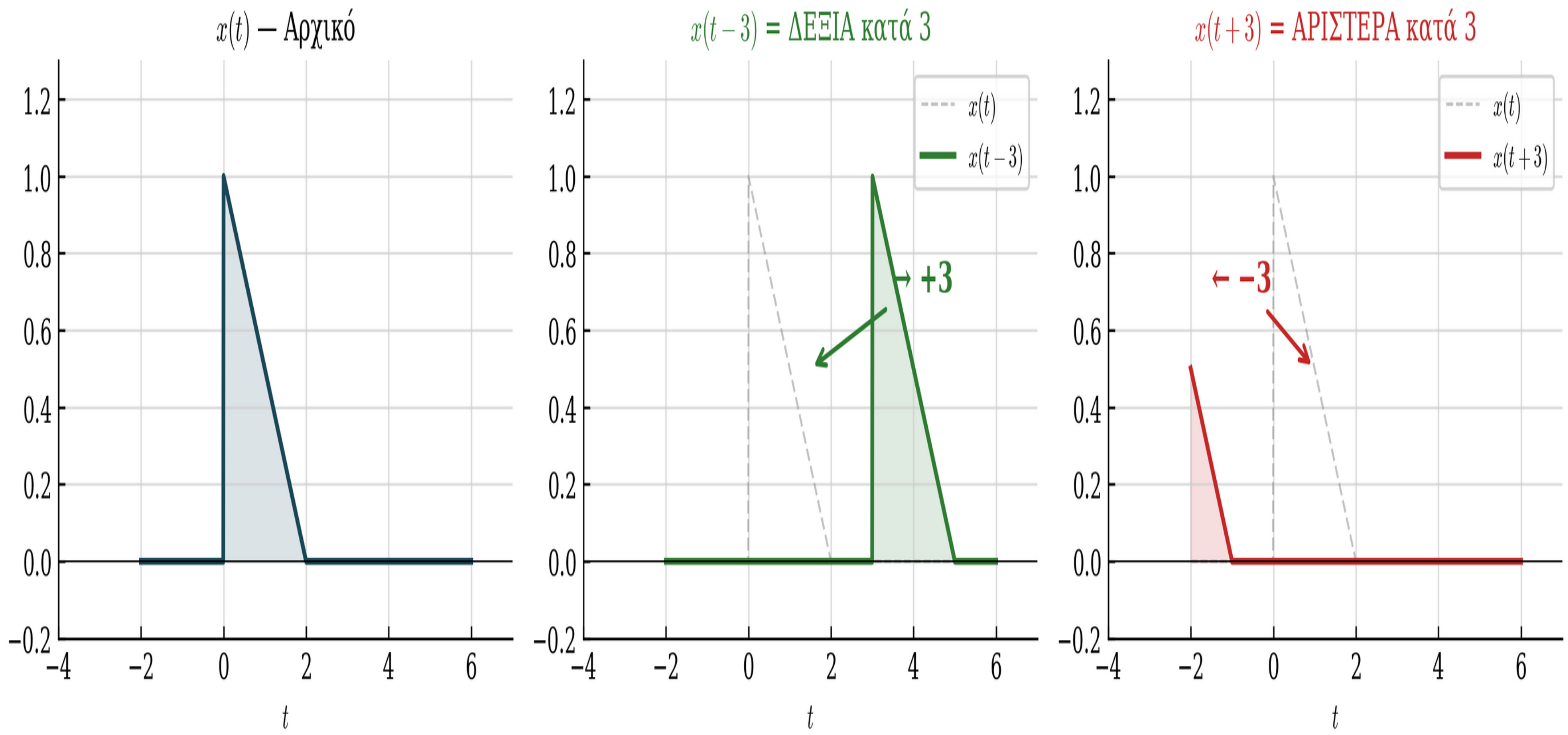
alias = $f_s - f_0$ → 50 Hz σήμα @ 60 Hz = βλέπουμε 10 Hz! Αυτό είναι aliasing.

Συχνά Λάθη #1: Πρόσημο Μετατόπισης

x(t−3) → μετατόπιση ΔΕΞΙΑ κατά 3
x(t+3) → μετατόπιση ΑΡΙΣΤΕΡΑ κατά 3

Μνημονικό: Το πρόσημο είναι ΑΝΤΙΘΕΤΟ από τη φορά κίνησης!

Κοιτάξτε πού μηδενίζεται το (t−a):
x(t−3) = 0 στο t=3 (δεξιά).
x(t+3) = 0 στο t=−3 (αριστερά).



Συχνά Λάθη #2: Σειρά & Mask

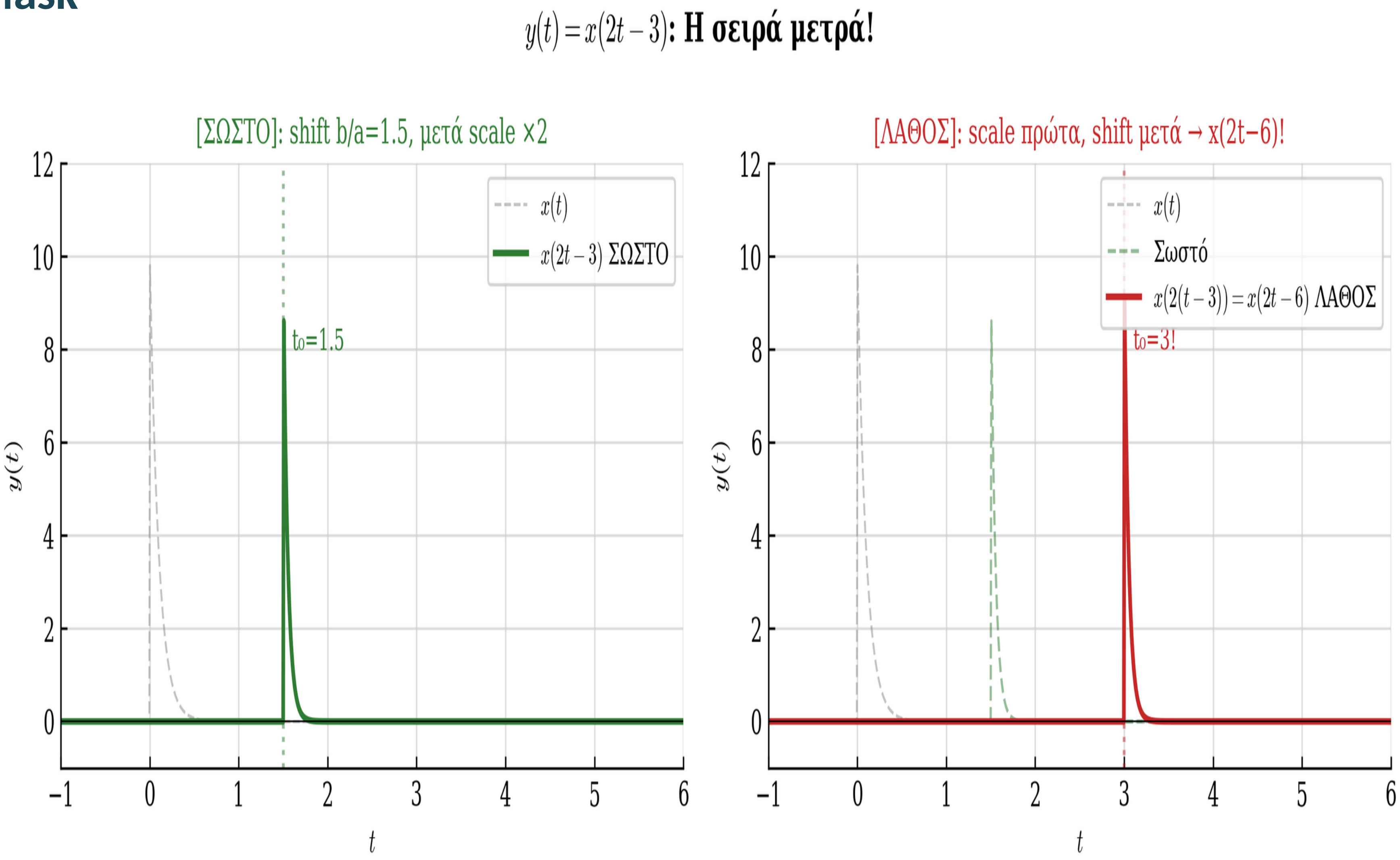
Για y(t) = x(at − b):

ΣΩΣΤΟ:
1. Shift κατά b/a
2. Scale κατά a

ΛΑΘΟΣ:
1. Scale κατά a (πρώτα)
2. Shift κατά b

→ Δίνει x(a(t−b)) = x(at−ab) ≠ x(at−b)!

Παράδειγμα: x(2t−3)
ΣΩΣΤΟ: shift 1.5, scale 2
ΛΑΘΟΣ: scale 2, shift 3 → x(2t−6)!

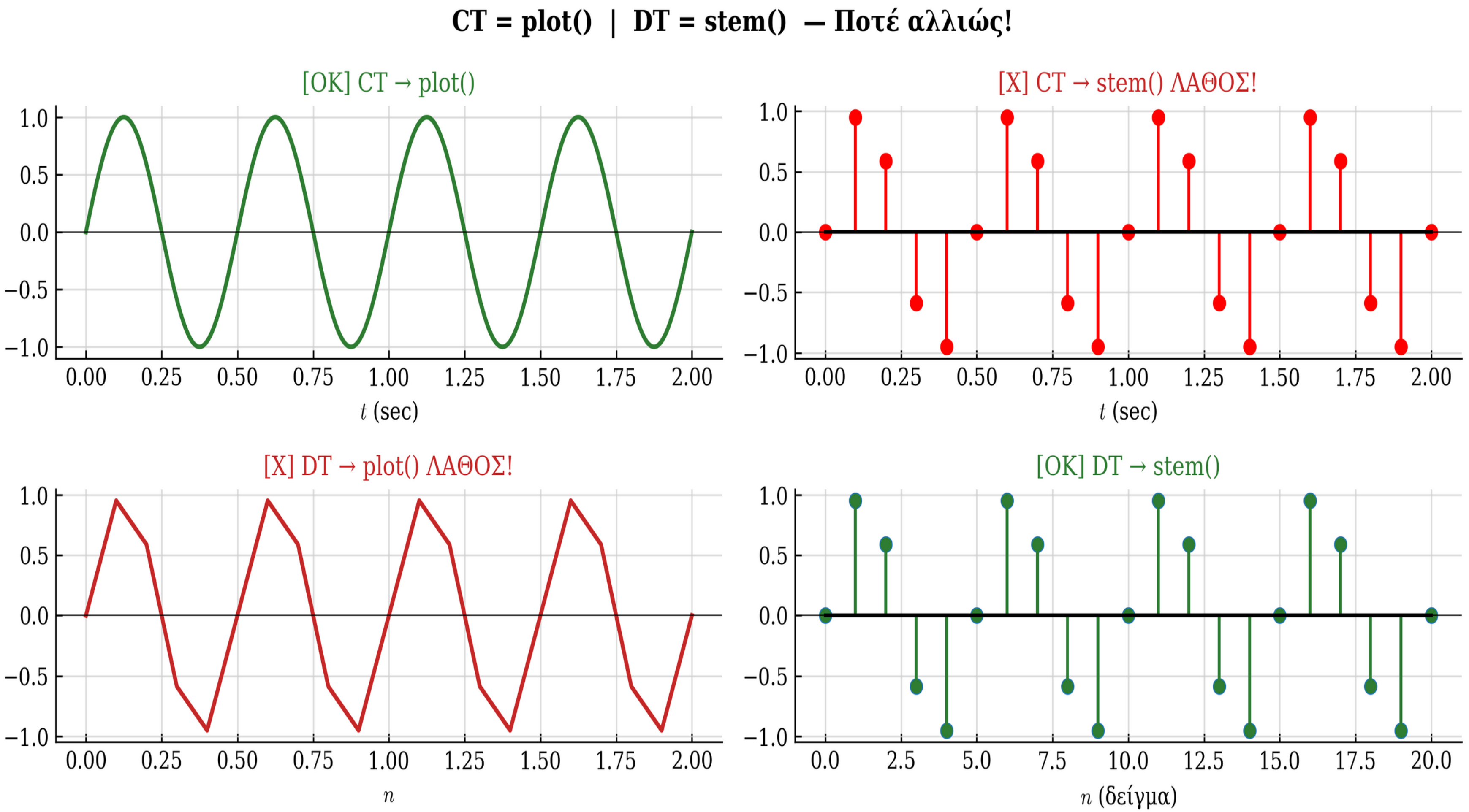


Συχνά Λάθη #3: plot/stem & Δείγματα

ΚΑΝΟΝΑΣ:
CT (συνεχές) → plot()
DT (διακριτό) → stem()

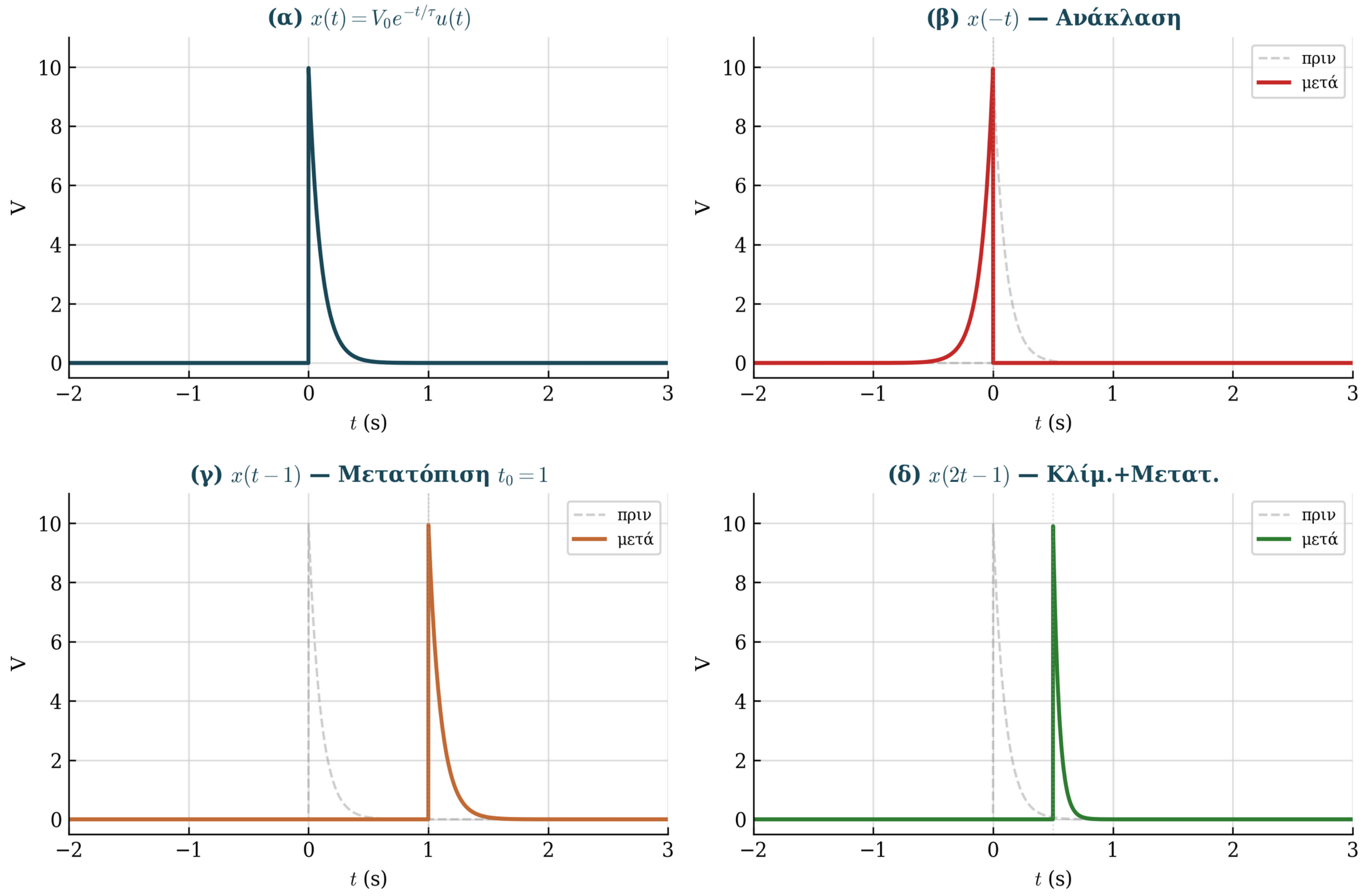
Γιατί:
- plot() συνδέει σημεία = ψεύτικες τιμές μεταξύ δειγμάτων DT
- stem() δείχνει ΜΟΝΟ τα πραγματικά δείγματα — σωστή αναπαράσταση

Στο Octave:
CT: plot(t, x)
DT: stem(n, x)



Λυμένο 1: RC Μετασχηματισμοί

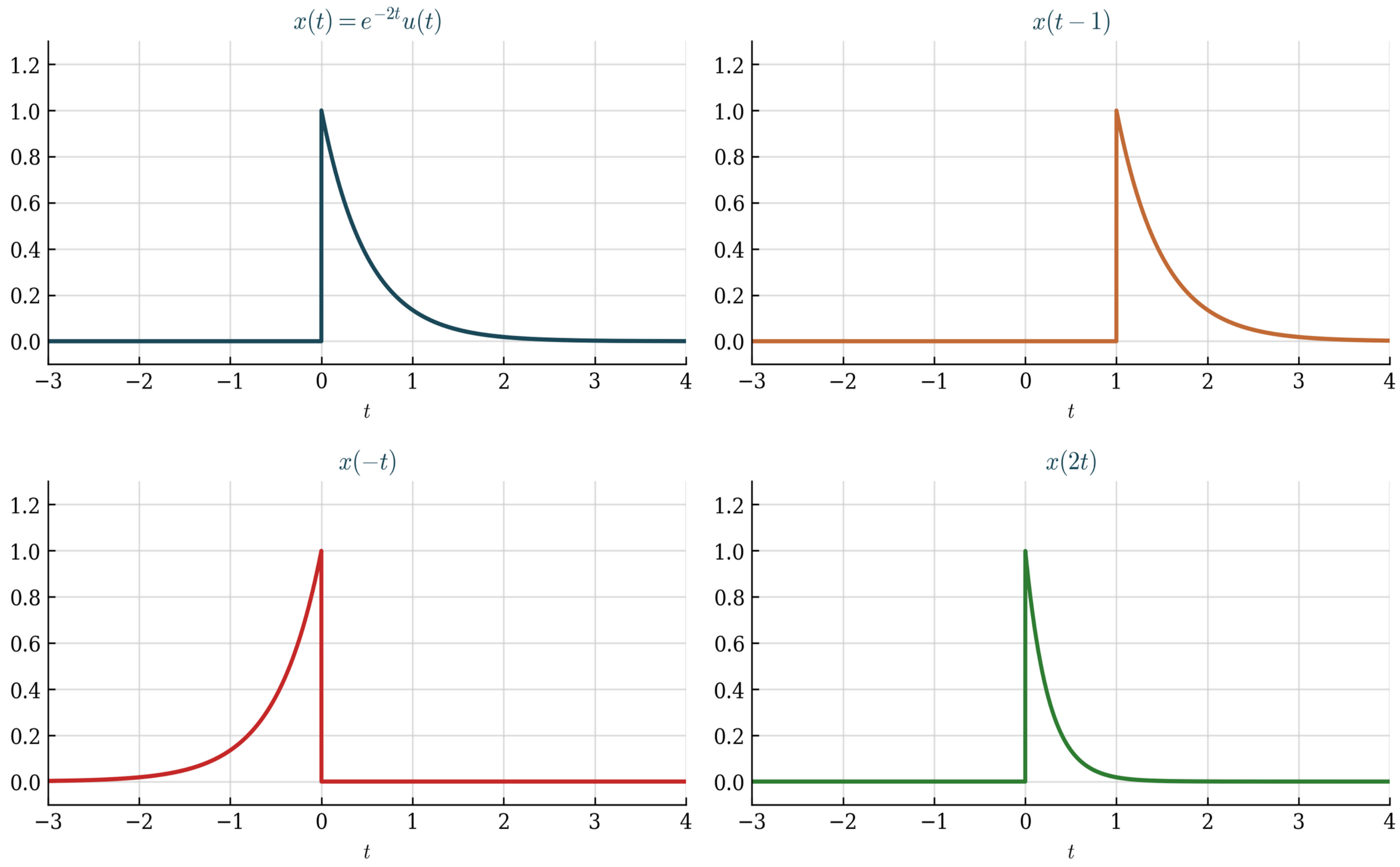
```
% RC: κάνετε γράφημα α) το σήμα x(t)
% β) την ανάκλασή του:  x(-t)
% γ) την μετατόπιση του: x(t-1)
% δ) κλιμάκωση κ μετατόπιση: x(2t-1)



% Λυμένο 1: RC
R=1e3; C=1e-4; V0=10; tau=R*C;
t=-2:0.001:3;
x = V0.*exp(-t./tau).*(t>=0);
y1 = V0.*exp(t./tau).*(t<=0);         % x(-t)
y2 = V0.*exp(-(t-1)./tau).*(t>=1);    % x(t-1)
y3 = V0.*exp(-(2*t-1)./tau).*(2*t-1>=0); % x(2t-1)

subplot(2,2,1);plot(t,x); title("x(t)");
subplot(2,2,2);plot(t,y1);title("x(-t)");
subplot(2,2,3);plot(t,y2);title("x(t-1)");
subplot(2,2,4);plot(t,y3);title("x(2t-1)");
```



RC: κάθε βήμα δείχνει το αρχικό x(t) γκρι διακεκομμένο. Χρώμα = αποτέλεσμα μετασχηματισμού.

Άσκηση 1: Πράξεις σε σήμα RC

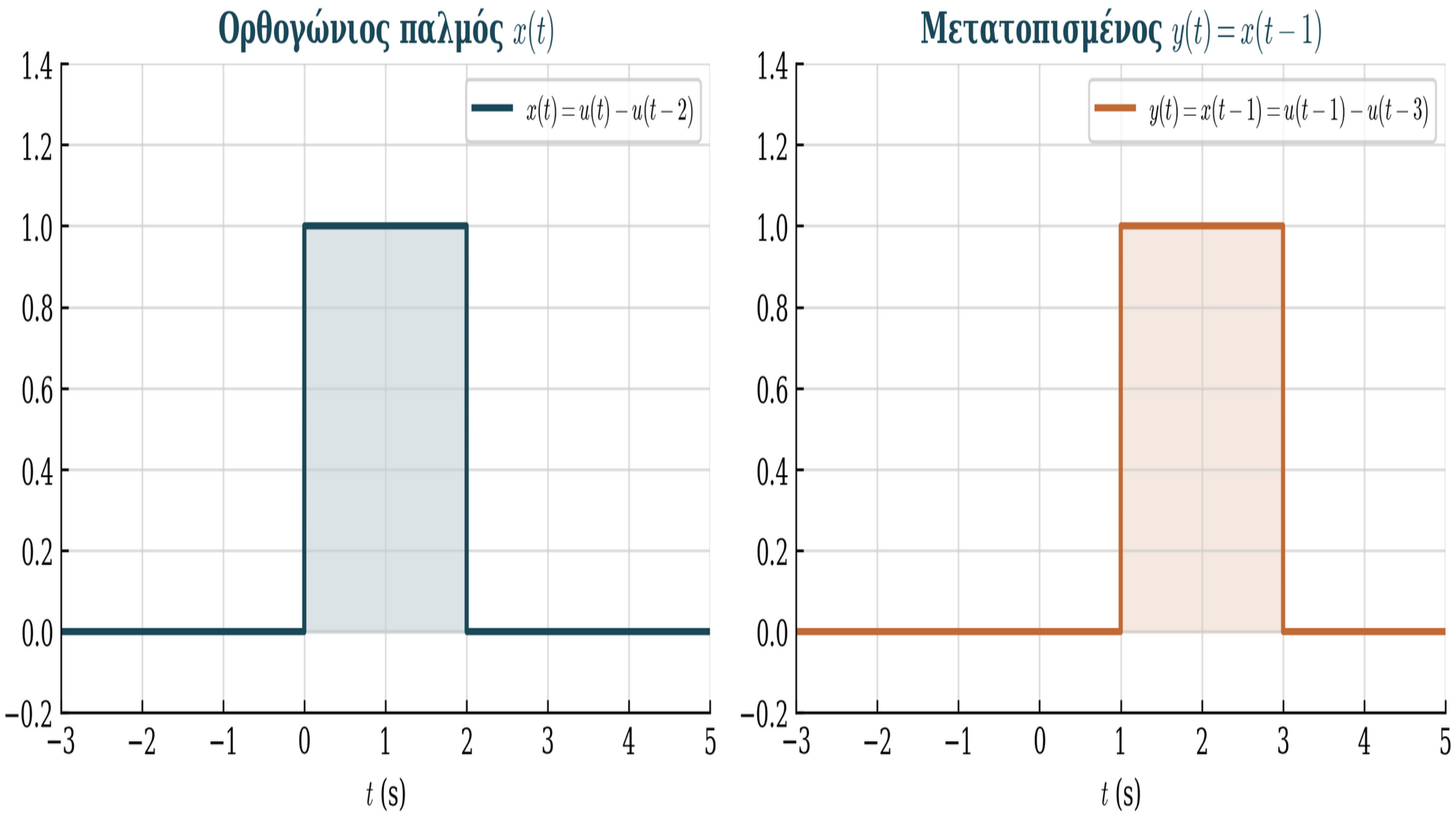
Δίνεται x(t)=exp(−2t)·u(t).
Σχεδιάστε: x(t−1), x(−t), x(2t)
Γράψτε κώδικα Octave.



Υπόδειξη: Σχεδιάστε ΠΡΩΤΑ στο χαρτί, μετά γράψτε κώδικα.

Άσκηση 2: Ορθογώνιος Παλμός

Δίνεται x(t)=u(t)−u(t−2).
Σχεδιάστε τον y(t)=x(t−1).
Ποια η διάρκεια; Πού ξεκινά;



Υπόδειξη: Σχεδιάστε ΠΡΩΤΑ στο χαρτί, μετά γράψτε κώδικα.

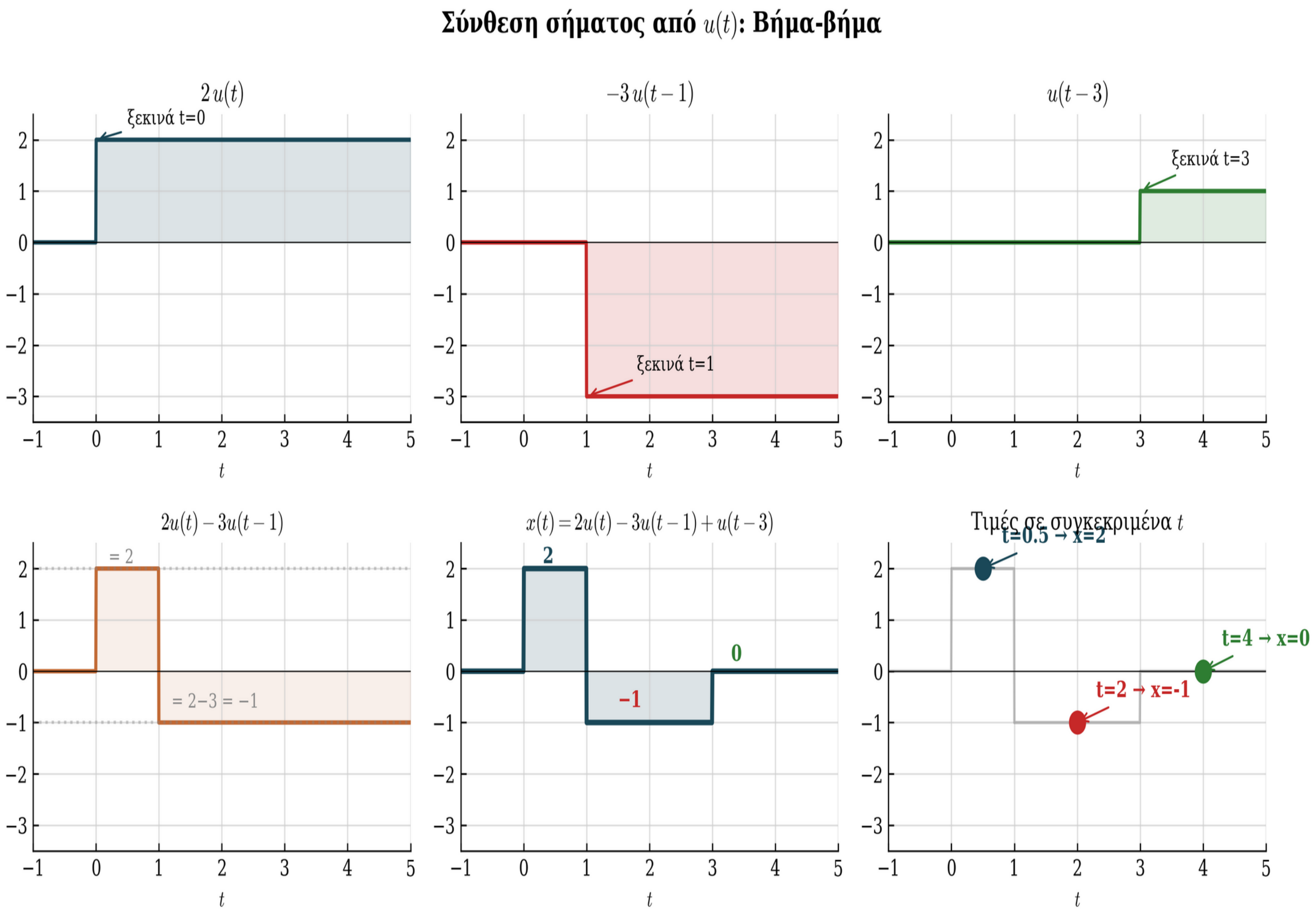
Άσκηση 3: Σύνθεση Βηματικών

x(t) = 2u(t) − 3u(t−1) + u(t−3)

Ανάλυση (δείτε γράφημα):

- **t < 0: όλα u(t)=0, x=0**
- **0 ≤ t < 1: μόνο u(t)=1, x=2**
- **1 ≤ t < 3: u(t)=1, u(t−1)=1, x = 2−3 = −1**
- **t ≥ 3: όλα 1, x = 2−3+1 = 0**

Τιμές: x(0.5)=2, x(2)=−1, x(4)=0



Μέθοδος: Βρείτε τα σημεία-κλειδιά (t=0, 1, 3) — σε κάθε διάστημα, προσθέστε τις ενεργές u(t−a) με τους συντελεστές τους.

ΕΛΛΗΝΙΚΟ ΜΕΣΟΓΕΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ

Άσκηση 4: Διαμόρφωση Πλατους

$x_1=\sin(2\pi t)$, $x_2=\cos(4\pi t)$.
Σχεδιάστε: $2\cdot x_1$, x_1+x_2, $x_1\cdot x_2$.
Ποιο μοιάζει με AM;

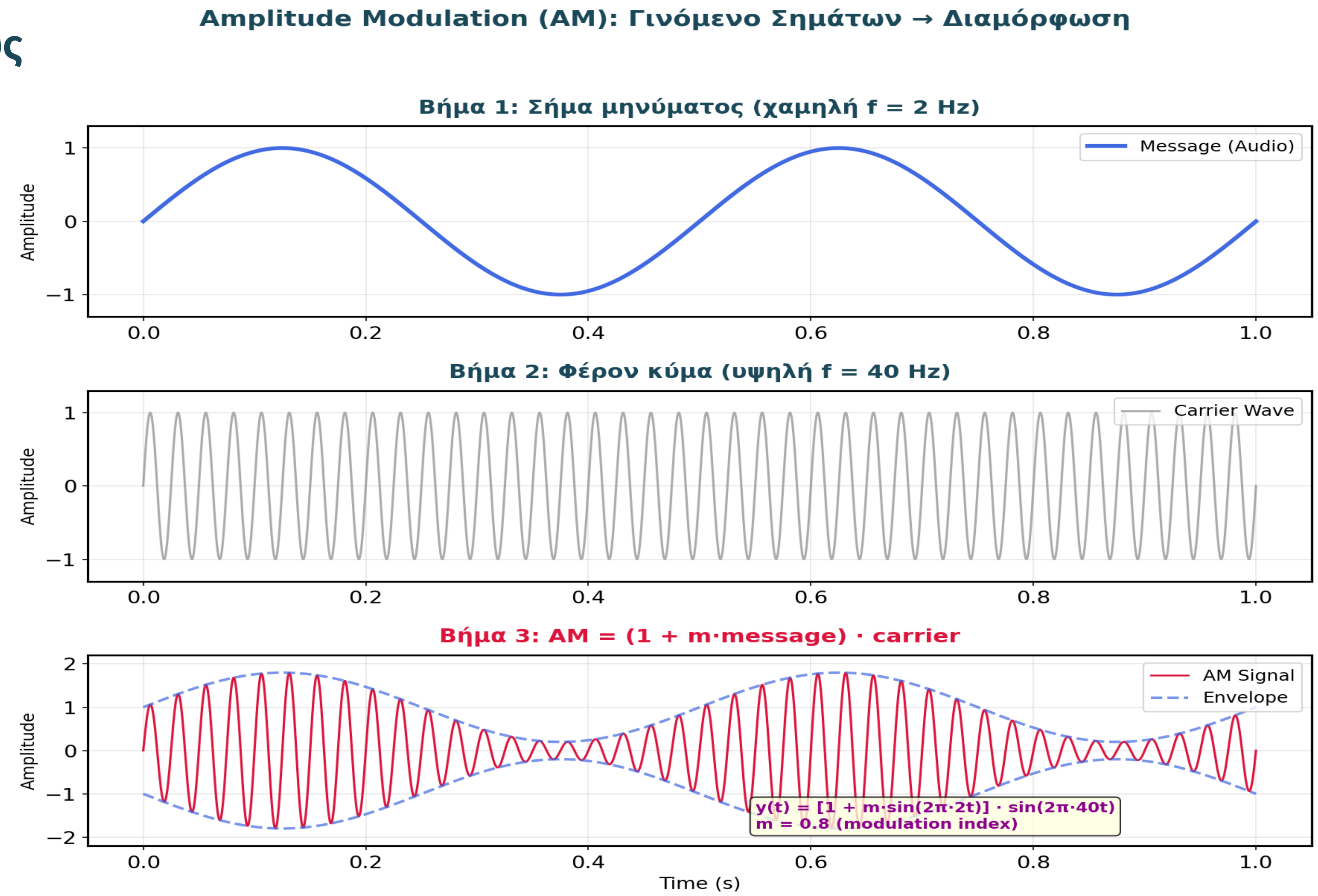
Amplitude Modulation (AM):
Γινόμενο σημάτων → Διαμόρφωση πλάτους!

$y(t) = [1 + m\cdot x_{msg}(t)] \cdot x_{carrier}(t)$

- xmsg: χαμηλή f (π.χ. ήχος)
- xcarrier: υψηλή f (φέρον κύμα)
- m: δείκτης διαμόρφωσης ($0 < m \leq 1$)

Στην Άσκηση 4:
$x_1\cdot x_2 = \sin(2\pi t)\cdot\cos(4\pi t)$
→ Το x_1 (χαμηλή f) γίνεται envelope
→ Το x_2 (υψηλή f) γίνεται carrier
→ Αυτό ΕΙΝΑΙ AM!

Εφαρμογές: Ραδιόφωνο AM, τηλεπικοινωνίες, sonar



Υπόδειξη: Σχεδιάστε ΠΡΩΤΑ στο χαρτί, μετά γράψτε κώδικα.

Ανακεφαλαίωση Εβδομάδας 3

Πράξεις πλάτους

Κλιμάκωση c·x(t), πρόσθεση, γινόμενο, DC offset, gating

Πράξεις χρόνου

Ανάκλαση x(−t), μετατόπιση $x(t-t_0)$, κλίμακα x(at)

Σύνθεση

Piecewise: r(t), u(t), boolean masks

Sampling

$f_s \geq 2 \cdot f\_max$ → Nyquist, αλλιώς aliasing

Κόκκινο νήμα

Sinusoid (amplitude) + RC (time) — δυαδικό

Εβδομάδα 4 → Even/Odd πλήρες, Modulation, Περιοδικότητα, LTI bridge

Αναφορές & Χρήσιμα Links

Oppenheim & Willsky
Signals and Systems, 2nd Ed. (Prentice Hall)
Κεφ. 1-2: Σήματα, Κεφ. 3: Μετασχηματισμοί

Μ. Παρασκευάς
Σήματα & Συστήματα με MATLAB/Octave, 3η Έκδ. (Τζιόλα, 2022)
Εύδοξος: 68402690 — Κεφ. 2: Πράξεις σε σήματα

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

GNU Octave
octave.org/doc — Πλήρες εγχειρίδιο αναφοράς
plot, stem, subplot, logical indexing, element-wise ops

Online: δωρεάν **Octave:** octave.org/doc

Τι Ερωτήσεις Μπορώ να Απαντήσω μετά το Εργ. 3;

Πράξεις Πλάτους

→ Πώς αλλάζει ένα σήμα αν πολλαπλασιαστεί με c;

→ Τι συμβαίνει όταν προσθέτω δύο ημίτονα κοντινών συχνοτήτων (beats);

→ Πώς αφαιρώ DC offset από σήμα;

→ Τι κάνει το gating/windowing σε σήμα;

Μετασχηματισμοί Χρόνου

→ Πώς σχεδιάζω x(−t), x(t−2), x(3t);

→ Ποια η σωστή σειρά για x(at−b);

→ Γιατί η σειρά shift/scale μετράει;

→ Τι πρόβλημα έχει η κλίμακα στο DT;

Σύνθεση & Ανάλυση

→ Πώς υπολογίζω τιμές σε $x(t) = \Sigma\, c_i\, u(t-a_i)$;

→ Τι είναι aliasing και πότε εμφανίζεται;

→ Πώς εφαρμόζω $f_s \geq 2f\_max$ στην πράξη;

Τέλος Εβδομάδας 3

Τι θα δούμε στο Εργ. 4:

→ Αποσύνθεση σε Άρτιο / Περιττό μέρος — πλήρης κώδικας & plots

→ Έλεγχος Περιοδικότητας — CT & DT, αριθμητικός τρόπος

→ Διαμόρφωση Πλάτους (AM) — modulation ως γινόμενο σημάτων

→ Σύνθετα Piecewise σήματα — πολλαπλά masks & ramps

→ Ολοκληρωμένα παραδείγματα — transform chain + ερμηνεία

→ Γέφυρα προς LTI συστήματα & Fourier

Ερωτήσεις; → email / eclass forum