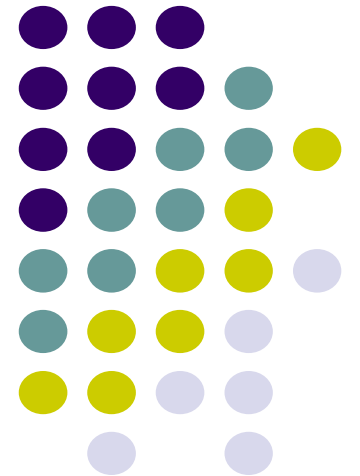
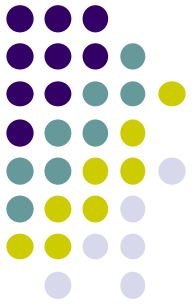


Φροντιστήριο Α

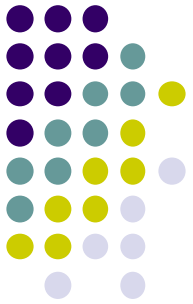
Ανασκόπηση MATLAB





Μετασχηματισμοί Laplace στο Matlab

Symbolic Math Toolbox



Το Symbolic Math Toolbox™ έχει εντολές για τον ορισμό, την επίλυση και δημιουργία γραφημάτων, συμβολικών συναρτήσεων

Δημιουργία συμβολικών μεταβλητών

MATLAB input:

```
syms x y
```

Ορισμός εξίσωσης

MATLAB input:

```
eqn = x^2 + y^2 == 1;
```

Αλγεβρική επίλυση ως προς y:

MATLAB input:

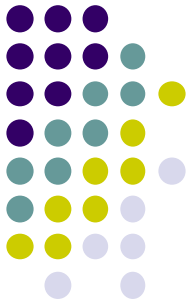
```
solve(eqn,y)
```

MATLAB output:

ans =

$$\begin{pmatrix} -\sqrt{2-x^2} \\ \sqrt{2-x^2} \end{pmatrix}$$

Μετασχηματισμοί Laplace



Μετασχηματισμός Laplace

MATLAB input:

```
syms t  
laplace(sin(t)+cos(t)+5)
```

MATLAB output:

```
ans =  

$$\frac{s}{s^2 + 1} + \frac{1}{s^2 + 1} + \frac{5}{s}$$

```

Αντίστροφος μετασχηματισμός Laplace

MATLAB input:

```
syms s  
ilaplace(1/(s^2+1))
```

MATLAB output:

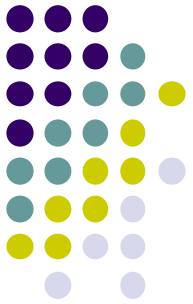
```
ans =  
 $\sin(t)$ 
```

MATLAB input:

```
ilaplace(s)
```

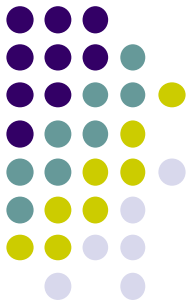
MATLAB output:

```
ans =  
  
dirac(1, t)
```



Σχηματισμός μοντέλων συνεχούς χρόνου

Συναρτήσεις μεταφοράς συνεχούς χρόνου



Δημιουργία συνάρτησης μεταφοράς (πολυωνυμική μορφή):

$$H(s) = \frac{2s + 1}{s^2 + 3s + 2}$$

MATLAB input:

```
num = [2 1];  
den = [1 3 2];
```

```
H = tf(num,den)
```

or

MATLAB input:

```
s = tf('s');
```

```
H = (2*s+1)/(s^2+3*s+2)
```

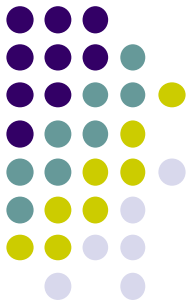
MATLAB output:

H =

$$\frac{2s + 1}{s^2 + 3s + 2}$$

Continuous-time transfer function.

Συναρτήσεις μεταφοράς συνεχούς χρόνου



Εισαγωγή καθυστέρησης (μετατόπιση) σε συνάρτηση μεταφοράς:

$$H(s) = e^{-2s} \frac{2s + 1}{s^2 + 3s + 2}$$

MATLAB input:

```
num = [2 1];  
den = [1 3 2];  
H = tf(num,den,'IODelay',2)
```

or

MATLAB input:

```
s = tf('s');  
  
H = exp(-2*s)*(2*s+1)/(s^2+3*s+2)
```

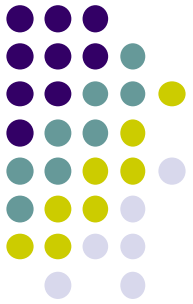
MATLAB output:

H =

$$\exp(-2*s) * \frac{2s + 1}{s^2 + 3s + 2}$$

Continuous-time transfer function.

Συναρτήσεις μεταφοράς συνεχούς χρόνου



Τα Zero-pole-gain μοντέλα είναι μια περιγραφή των συναρτήσεων μεταφοράς σε παραγοντοποιημένη μορφή.

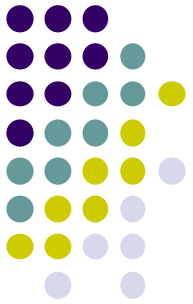
Χρησιμοποιούν την εντολή

```
sys = zpk(zeros,poles,gain)
```

Για παράδειγμα, η παρακάτω SISO ΠΣΜ

$$h(s) = k \frac{(s - z(1))(s - z(2)) \dots (s - z(m))}{(s - p(1))(s - p(2)) \dots (s - p(n))}$$

Συναρτήσεις μεταφοράς συνεχούς χρόνου



Δημιουργία συνάρτησης μεταφοράς σε παραγοντοποιημένη μορφή:

$$h(s) = k \frac{(s - z(1))(s - z(2)) \dots (s - z(m))}{(s - p(1))(s - p(2)) \dots (s - p(n))}$$

Παράδειγμα

$$H(s) = \frac{2s + 1}{s^2 + 3s + 2} = 2 \frac{s + 0.5}{(s + 1)(s + 2)}$$

zero/pole/gain model

MATLAB input:

`z = [-0.5];`

`p = [-1 -2];`

`k=2;`

`H = zpk(z,p,k)`

MATLAB output:

H =

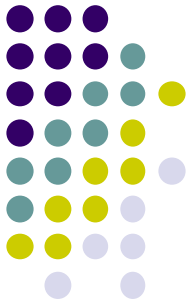
2 (s+0.5)

(s+1) (s+2)

Continuous-time zero/pole/gain model.

Προσοχή: Η στατική ενίσχυση της ΠΣΜ δεν είναι ίση με την ενίσχυση (k)!!

Συναρτήσεις μεταφοράς συνεχούς χρόνου



Μοντέλο χώρου καταστάσεων:

Μαθηματική αναπαράσταση ενός φυσικού συστήματος ως ένα σύνολο από μεταβλητές εισόδου, εξόδου και κατάστασης, οι οποίες συσχετίζονται με διαφορικές εξισώσεις πρώτου βαθμού.

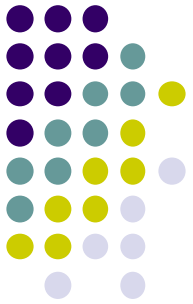
$$\frac{dx}{dt} = Ax + Bu$$
$$y = Cx + Du$$

x : διάνυσμα κατάστασης
 u : διάνυσμα εισόδου
 y : διάνυσμα εξόδου

A : πίνακας κατάστασης
 B : πίνακας εισόδου
 C : πίνακας εξόδου
 D : πίνακας άμεσης μετάδοσης

applied to linear time-invariant (LTI) systems

Συναρτήσεις μεταφοράς συνεχούς χρόνου



Δημιουργία μοντέλων χώρου καταστάσεων:

$$\frac{dx}{dt} = Ax + Bu$$

$$y = Cx + Du$$

Παράδειγμα

$$A = \begin{bmatrix} 0 & 1 \\ -5 & -2 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 3 \end{bmatrix}, \quad C = [0 \quad 1], \quad D = [0].$$

MATLAB input:

```
A = [0 1; -5 -2];
```

```
B = [0;3];
```

```
C = [0 1];
```

```
D = [0];
```

```
sys = ss(A,B,C,D)
```

MATLAB output:

```
sys =
```

```
A =
```

```
    x1 x2  
x1  0  1  
x2 -5 -2
```

```
C =
```

```
    x1 x2  
y1  0  1
```

```
B =
```

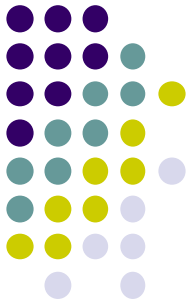
```
    u1  
x1  0  
x2  3
```

```
D =
```

```
    u1  
y1  0
```

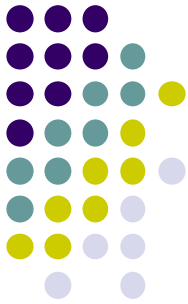
Continuous-time state-space model.

Συναρτήσεις μεταφοράς συνεχούς χρόνου



Μετατροπή από	Μετατροπή σε	Συνάρτηση Matlab
Transfer Function	Zero-pole-gain	$[z,p,k]=tf2zp(num,den)$
Transfer Function	State Space	$[A,B,C,D]=tf2ss(num,den)$
Zero-pole-gain	Transfer Function	$[num,den]=zp2tf(z,p,k)$
Zero-pole-gain	State Space	$[A,B,C,D]=zp2ss(z,p,k)$
State Space	Transfer Function	$[num,den]=ss2tf(A,B,C,D)$
State Space	Zero-pole-gain	$[z,p,k]=ss2zp(A,B,C,D)$

Συνδυασμός μοντέλων



Ένα μοντέλο μπορεί να είναι μία δομή με εισόδους και εξόδους (διάγραμμα βαθμίδων) που περιέχει μία συνάρτηση μεταφοράς ή ένα μοντέλο χώρου καταστάσεων μέσα του.

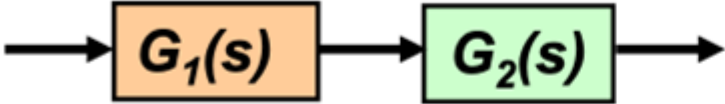
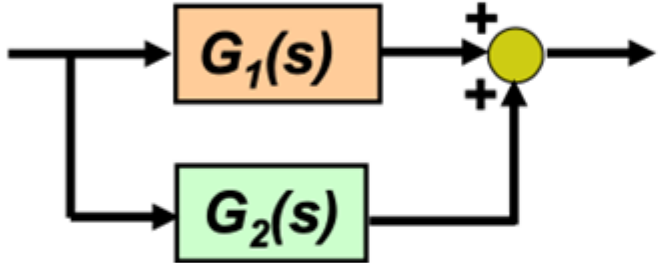
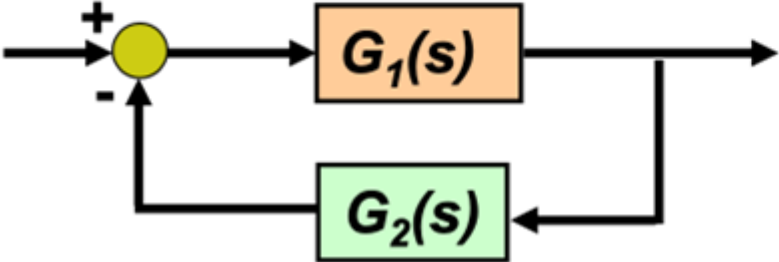


Στοιχεία Διαγράμματος Δομών

Συνδυασμός μοντέλων

Διαχείριση του διαγράμματος βαθμίδων (δουλεύει και με ΠΧΚ):



Combination	Matlab Command
	<code>sys = series (G1 , G2)</code>
	<code>sys = parallel (G1 , G2)</code>
	<code>sys = feedback (G1 , G2)</code>

Προσοχή: Η σειρά με την οποία βάζετε τα G_1 και G_2 επηρεάζει τις ΠΧΚ !!

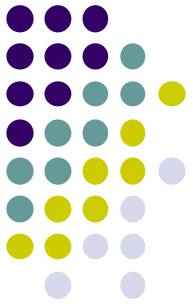
Συνδυασμός μοντέλων

Διαχείριση του διαγράμματος βαθμίδων (δουλεύει και με ΠΧΚ):



Αριθμητική Πράξη	Σύνταξη Matlab
Πρόσθεση	<code>sys = G1+G2;</code>
Πολλαπλασιασμός	<code>sys = G1*G2;</code>
Αντιστροφή	<code>sys = inv(G1);</code>

Προσοχή: Η σειρά με την οποία βάζετε τα G_1 και G_2 επηρεάζει τις ΠΧΚ !!



Ανάλυση μεταβατικής απόκρισης



Βασικές αποκρίσεις:

- Βηματική απόκριση
- Παλμική απόκριση
- Γραμμική (ramp) απόκριση

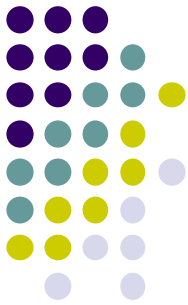
Η μεταβατική απόκριση αναφέρεται στη διεργασία που τελείται για τη μετάβαση από μία **αρχική** σε μία **τελική** κατάσταση

Οι μεταβατικές αποκρίσεις χρησιμοποιούνται για να ερευνηθούν τα χρονικά χαρακτηριστικά των δυναμικών συστημάτων

Συνήθεις Αποκρίσεις: Βηματική Απόκριση, Παλμική Απόκριση, και γραμμική (ramp) απόκριση

Ανάλυση μεταβατικής απόκρισης

Μοναδιαία βηματική απόκριση:



Παράδειγμα

$$H(s) = \frac{25}{s^2 + 4s + 25}$$

MATLAB input:

```
num = [0 0 25];
```

```
den = [1 4 25];
```

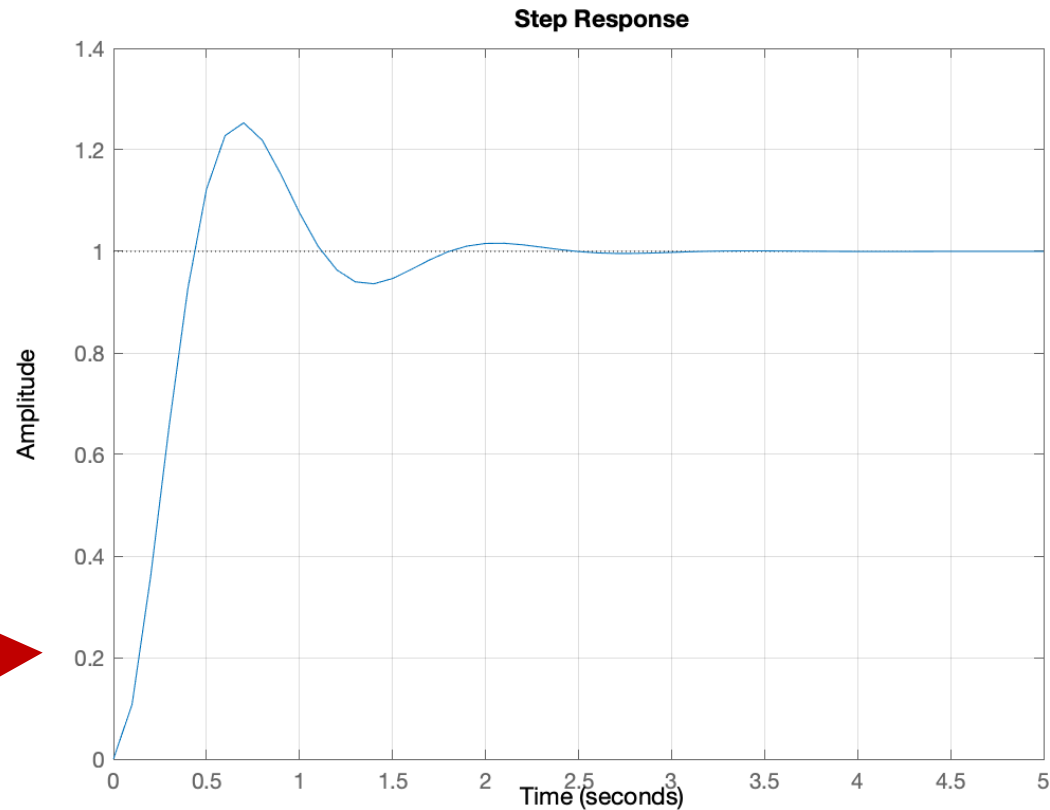
```
H = tf(num,den);
```

```
%specify time points
```

```
t = [0:0.1:5];
```

```
step(H,t)
```

```
grid %add grid to plot
```



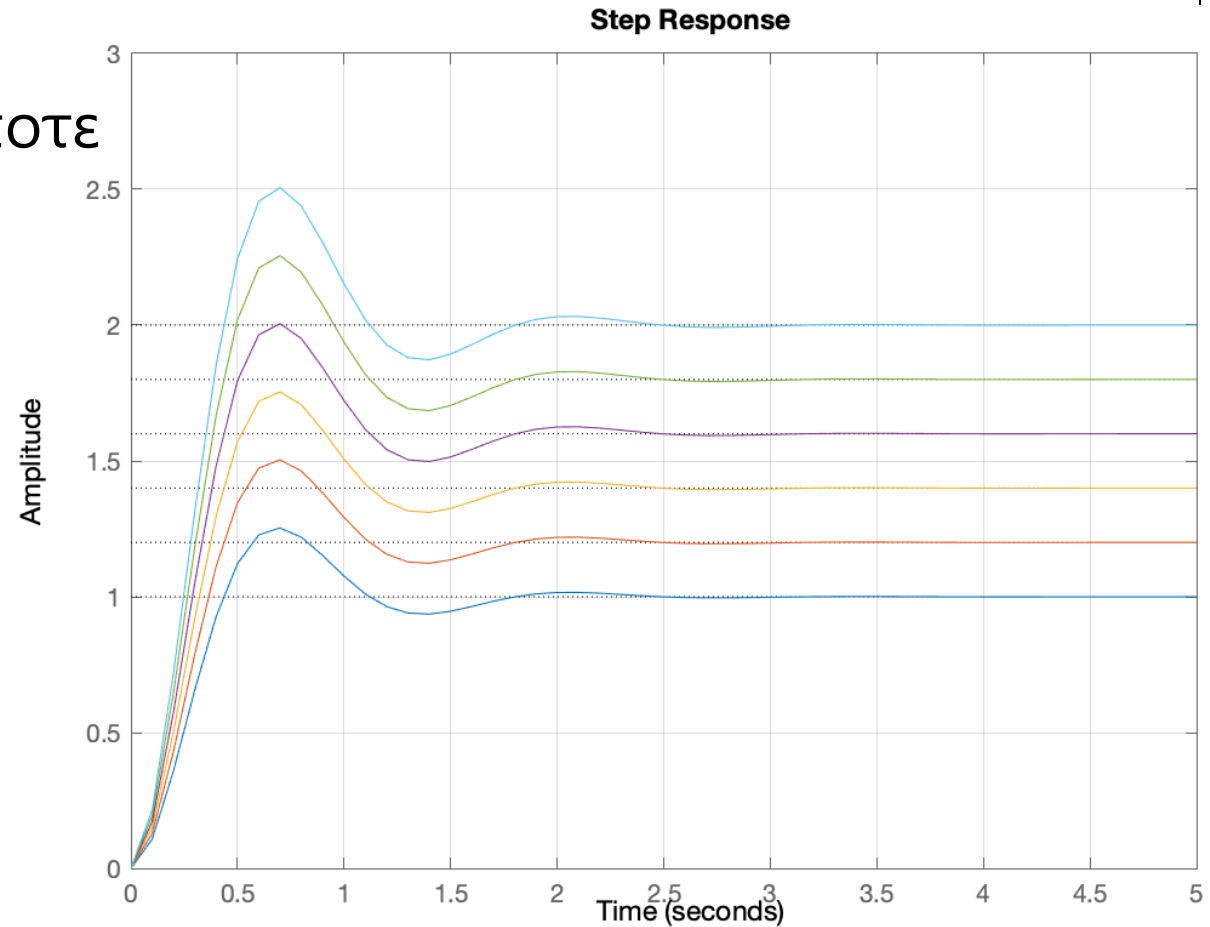
Ανάλυση μεταβατικής απόκρισης



Για **βηματική** είσοδο οποιουδήποτε πλάτους:

MATLAB input:

```
hold on  
step(1.2*H,t);  
step(1.4*H,t);  
step(1.6*H,t);  
step(1.8*H,t);  
step(2.0*H,t);
```



Ανάλυση μεταβατικής απόκρισης

Μοναδιαία παλμική απόκριση:

Παράδειγμα

$$H(s) = \frac{25}{s^2 + 4s + 25}$$

MATLAB input:

```
num = [0 0 25];
```

```
den = [1 4 25];
```

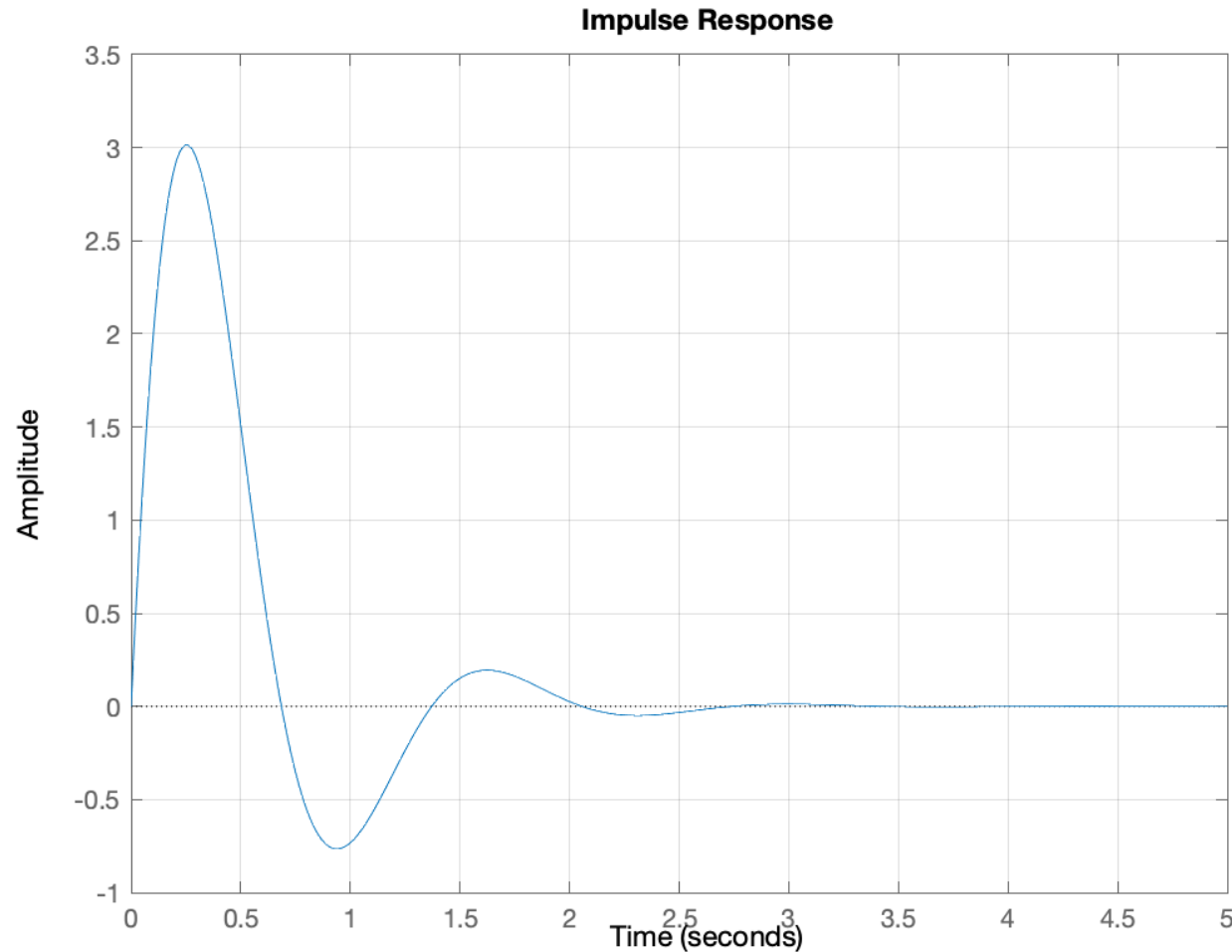
```
H = tf(num,den);
```

```
%specify time points
```

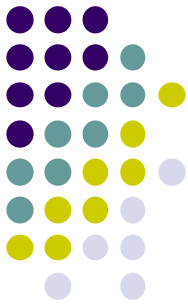
```
t = [0:0.01:5];
```

```
impulse(H,t)
```

```
grid %add grid to plot
```



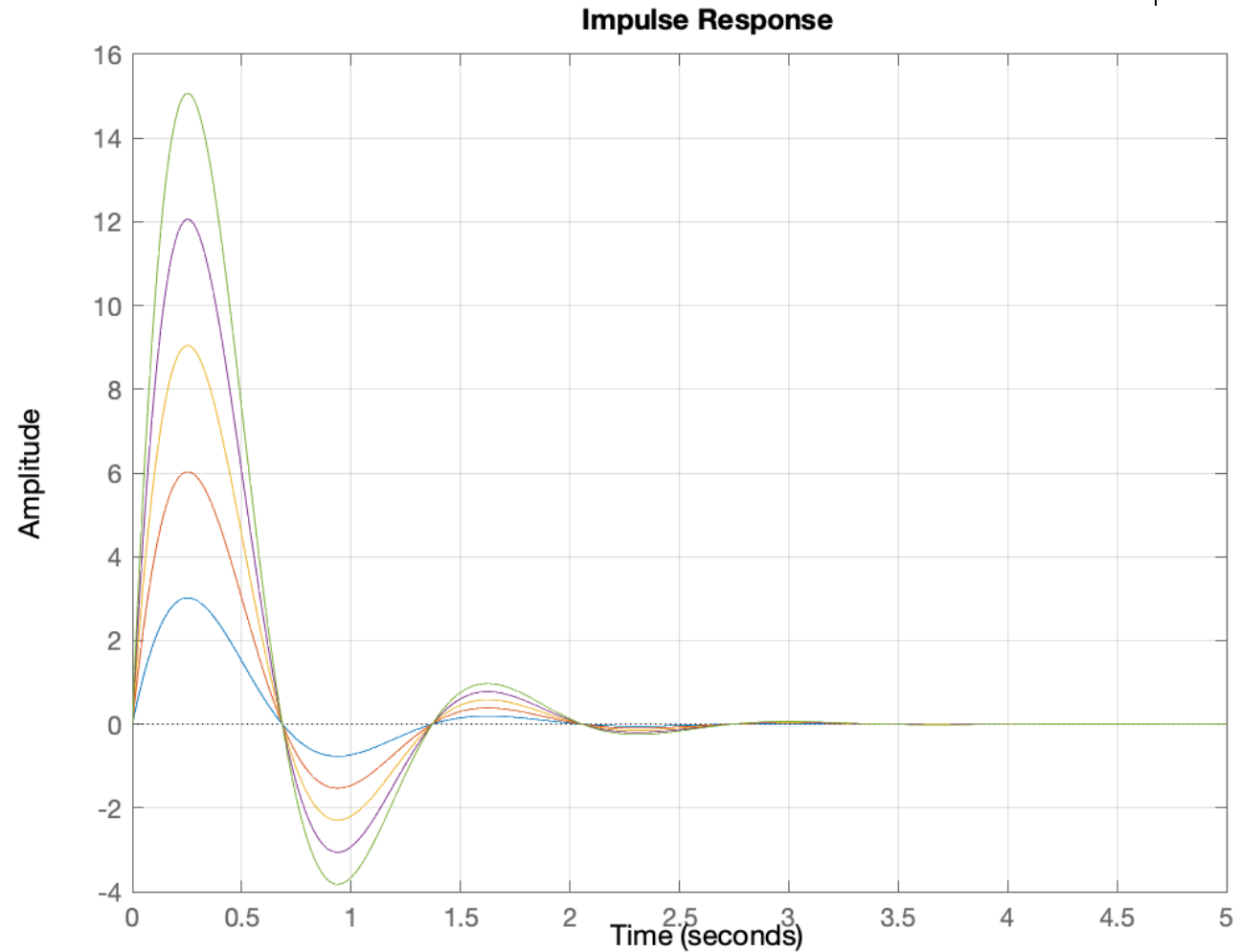
Ανάλυση μεταβατικής απόκρισης



Για παλμική είσοδο
οποιοδήποτε πλάτους:

MATLAB input:

```
hold on  
impulse(2*H, t)  
impulse(3*H, t)  
impulse(4*H, t)  
impulse(5*H, t)  
hold off
```



Ανάλυση μεταβατικής απόκρισης



Γραμμική απόκριση:

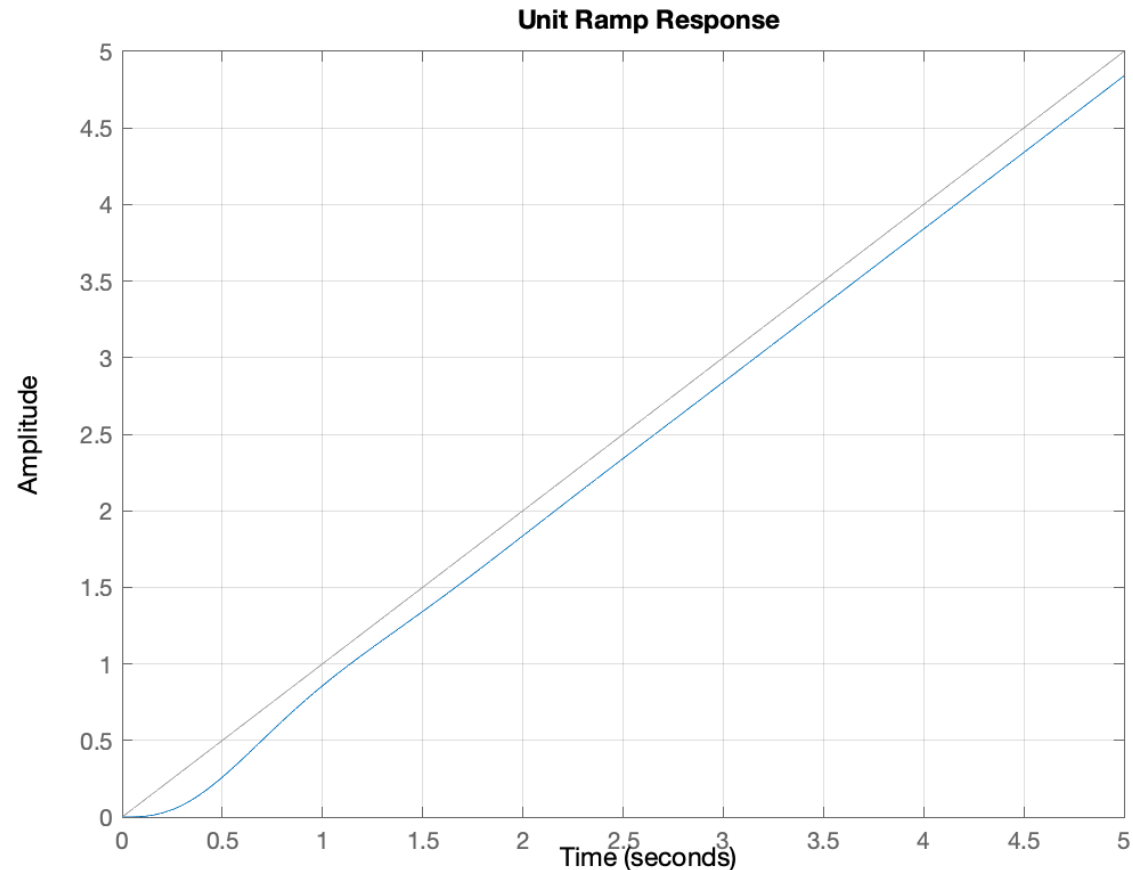
Δεν υπάρχει «γραμμική» συνάρτηση στην MATLAB.

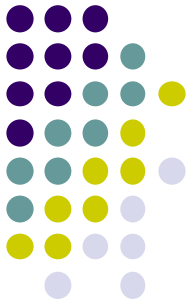
Εντολή Isim : Plot simulated time response of dynamic system to **arbitrary** inputs

$$u(t) = t \quad U(s) = \frac{1}{s^2}$$

MATLAB input:

```
%specify time points  
t = [0:0.01:5];  
  
%linear input u(t) = t  
u = t;  
  
Isim(H,u,t)  
%add grid and title  
grid  
title('Unit Ramp Response');
```

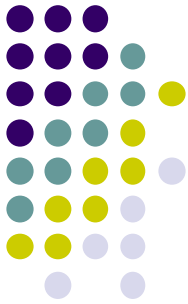




Ανάλυση συχνοτικής απόκρισης

Ανάλυση συχνотικής απόκρισης

Διάγραμμα Bode: συχνотική απόκριση του συστήματος



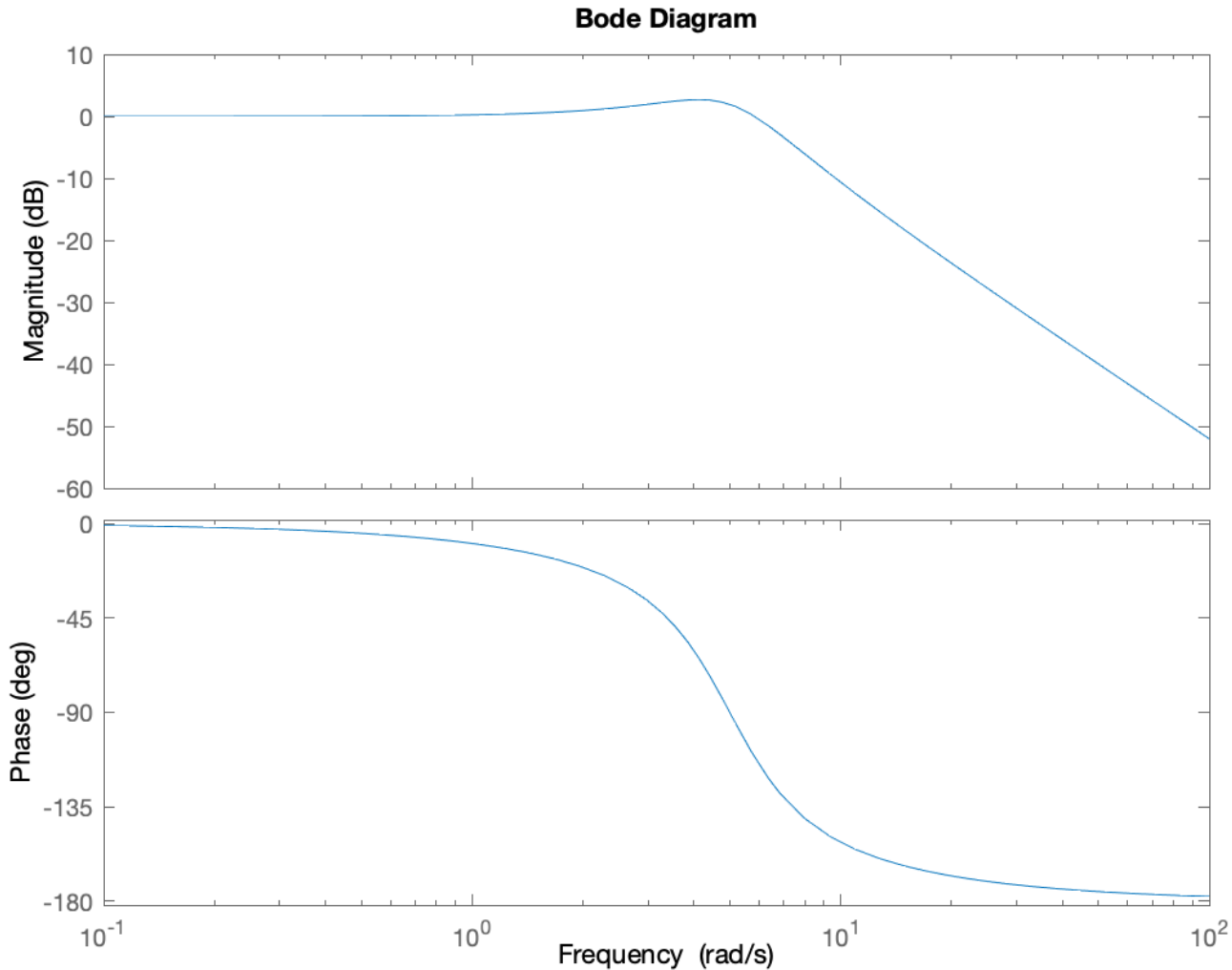
Παράδειγμα

$$H(s) = \frac{25}{s^2 + 4s + 25}$$

MATLAB input:

```
num = [0 0 25];  
den = [1 4 25];  
H = tf(num,den);
```

```
%bode diagrams  
bode(H)
```



Συναρτήσεις Control System Toolbox...



Creation of LTI models

ss - Create a state-space model.
zpk - Create a zero/pole/gain model.
tf - Create a transfer function model.
dss - Specify a descriptor state-space model.
filt - Specify a digital filter.
set - Set/modify properties of LTI models.
ltiprops - Detailed help for available LTI properties.

Data extraction

ssdata - Extract state-space matrices.
zpkdata - Extract zero/pole/gain data.
tfdata - Extract numerator(s) and denominator(s).
dssdata - Descriptor version of SSDATA.
get - Access values of LTI model properties.

Model characteristics

class - Model type ('ss', 'zpk', or 'tf').
size - Input/output dimensions.
isempty - True for empty LTI models.
isct - True for continuous-time models.
isdt - True for discrete-time models.
isproper - True for proper LTI models.
issiso - True for single-input/single-output systems.
isa - Test if LTI model is of given type.

Conversions

ss - Conversion to state space.
zpk - Conversion to zero/pole/gain.
tf - Conversion to transfer function.
c2d - Continuous to discrete conversion.
d2c - Discrete to continuous conversion.
d2d - Resample discrete system or add input delay(s).

Overloaded arithmetic operations

+ and - - Add and subtract LTI systems (parallel connection).
* - Multiplication of LTI systems (series connection).
\ - Left divide -- sys1\sys2 means inv(sys1)*sys2.
/ - Right divide -- sys1/sys2 means sys1*inv(sys2).
' - Pertransposition.
' - Transposition of input/output map.
[.] - Horizontal/vertical concatenation of LTI systems.
inv - Inverse of an LTI system.

Model dynamics

pole, eig - System poles.
tzero - System transmission zeros.
pzmap - Pole-zero map.
dcgain - D.C. (low frequency) gain.
norm - Norms of LTI systems.
covar - Covariance of response to white noise.
damp - Natural frequency and damping of system poles.
esort - Sort continuous poles by real part.
dsort - Sort discrete poles by magnitude.
pade - Pade approximation of time delays.

State-space models

rss, drss - Random stable state-space models.
ss2ss - State coordinate transformation.
canon - State-space canonical forms.
ctrb, obsv - Controllability and observability matrices.
gram - Controllability and observability gramians.
ssbal - Diagonal balancing of state-space realizations.
balreal - Gramian-based input/output balancing.
modred - Model state reduction.
minreal - Minimal realization and pole/zero cancellation.
augstate - Augment output by appending states.

Time response

step - Step response.
impulse - Impulse response.
initial - Response of state-space system with given initial state.
lsim - Response to arbitrary inputs.
ltiview - Response analysis GUI.
gensig - Generate input signal for LSIM.
stepfun - Generate unit-step input.

Frequency response

bode - Bode plot of the frequency response.
sigma - Singular value frequency plot.
nyquist - Nyquist plot.
nichols - Nichols chart.
ltiview - Response analysis GUI.
evalfr - Evaluate frequency response at given frequency.
freqresp - Frequency response over a frequency grid.
margin - Gain and phase margins

System interconnections

append - Group LTI systems by appending inputs and outputs.
parallel - Generalized parallel connection (see also overloaded +).
series - Generalized series connection (see also overloaded *).
feedback - Feedback connection of two systems.
star - Redheffer star product (LFT interconnections).
connect - Derive state-space model from block diagram description.

Classical design tools

rlocus - Evans root locus.
rlocfind - Interactive root locus gain determination.
acker - SISO pole placement.
place - MIMO pole placement.
estim - Form estimator given estimator gain.
reg - Form regulator given state-feedback and estimator gains.

LQG design tools

lqr, dlqr - Linear-quadratic (LQ) state-feedback regulator.
lqry - LQ regulator with output weighting.
lqrd - Discrete LQ regulator for continuous plant.
kalman - Kalman estimator.
kalmd - Discrete Kalman estimator for continuous plant.
lqgreg - Form LQG regulator given LQ gain and Kalman estimator.

Matrix equation solvers

lyap - Solve continuous Lyapunov equations.
dlyap - Solve discrete Lyapunov equations.
care - Solve continuous algebraic Riccati equations.
dare - Solve discrete algebraic Riccati equations.