# بخش سوم - تحلیل سیستمهای کنترلی به کمک *MATLAB*

یکی از قابلیتهای مهم MATLAB، کاربرد آن در تحلیل و طراحی سیستمهای کنترلی است. رسم نمودارهای مختلف و تحلیل نحوه عملکرد سیستمهای کنترل خطی بدون کمک کامپیوتر بسیار دشوار و بعضاً غیرممکن است و MATLAB در این میان به عنوان یک ابزار ریاضی مهندسی، کمک فراوانی به مهندسین کنترل جهت تحلیل و طراحی سیستم‌ها می‌نماید. در ذیل با اصول اساسی تحلیل سیستمهای کنترلی به کمک MATLAB آشنا می‌شویم.

**تعریف تابع تبدیل در حوزه لاپلاس**

می‌دانید که تابع تبدیل سیستمهای LTI در حوزه لاپلاس گویا و شامل دو چندجمله‌ای برحسب s در صورت و مخرج است:

$$H(s) = \frac{B(s)}{A(s)} = \frac{b_1 s^{n-1} + \cdots + b_{n-1} s + b_n}{a_1 s^{m-1} + \cdots + a_{m-1} s + a_m}$$

این چند جمله‌ای‌ها را می‌توان به صورت حاصل‌ضرب صفرها و قطبها نوشت:

$$H(s) = \frac{Z(s)}{P(s)} = k\frac{(s-z_1)(s-z_2)\cdots(s-z_m)}{(s-p_1)(s-p_2)\cdots(s-p_n)}$$

که k بهره سیستم، $Z_i$ صفرهای سیستم و $P_i$ قطبهای سیستم هستند.

در MATLAB می‌توان یک تابع تبدیل را در هر دو شکل تعریف کرد.

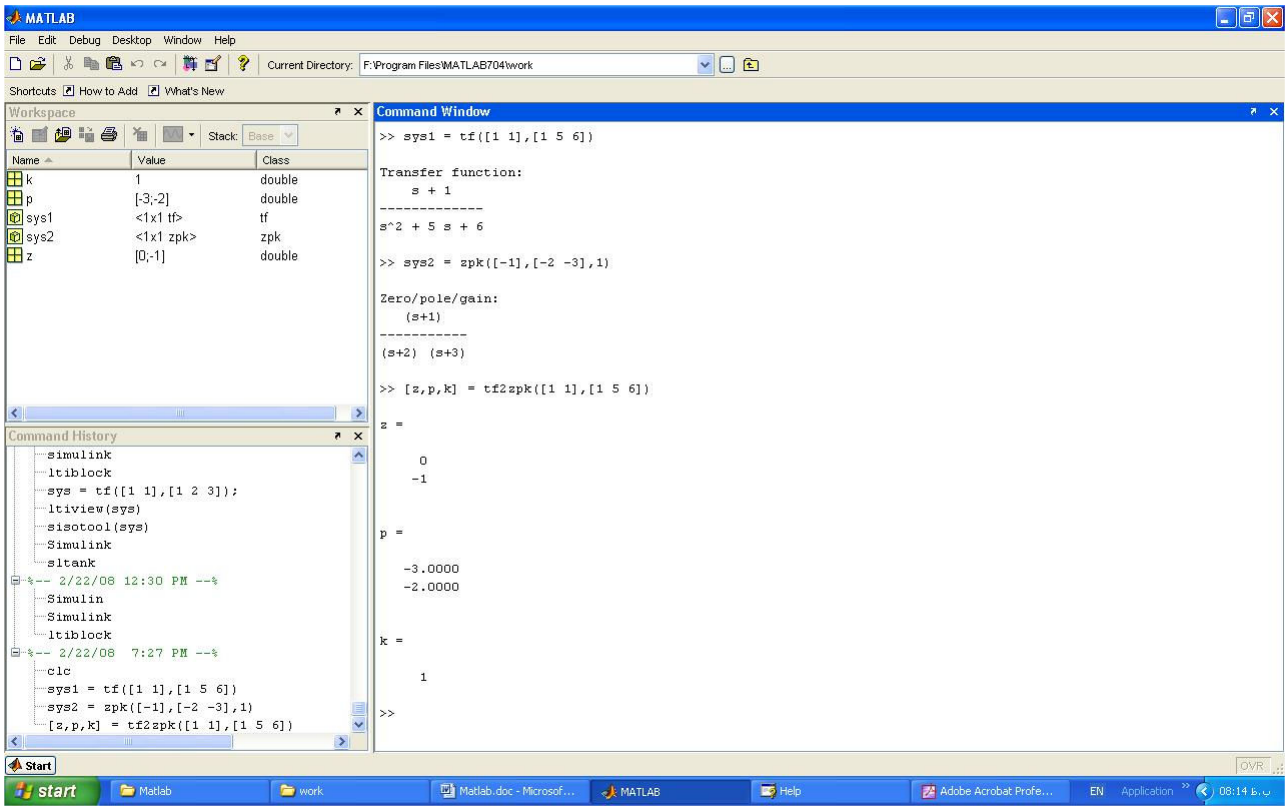
**دستور tf (Transfer Function)**

به کمک دستور *tf ( [ بردار ضرایب چند جمله‌ای مخرج ] , [ بردار ضرایب چندجمله‌ای صورت ] )* می‌توان یک تابع تبدیل کـه فـرم آن مانند شکل اول باشد را تعریف کرد.

**دستور zpk (Zero-Pol-Gain)**

به کمک دستور *zpk ( [ بردار مقادیر صفر ] , [ بردار مقادیر قطب ] , بهره )* می‌توان یک تابع تبدیل به فرم دوم را تعریف نمود.

شکل زیر نحوه به کار بردن توابع فوق و تبدیل آنها به یکدیگر را نشان می‌دهد:




### تجزیه تابع تبدیل و عکس تبدیل لاپلاس

برای تجزیه تابع تبدیل به کسرهای جزیی برای سهولت محاسبه عکس تبدیل لاپلاس، از تابع residue استفاده می‌شود.

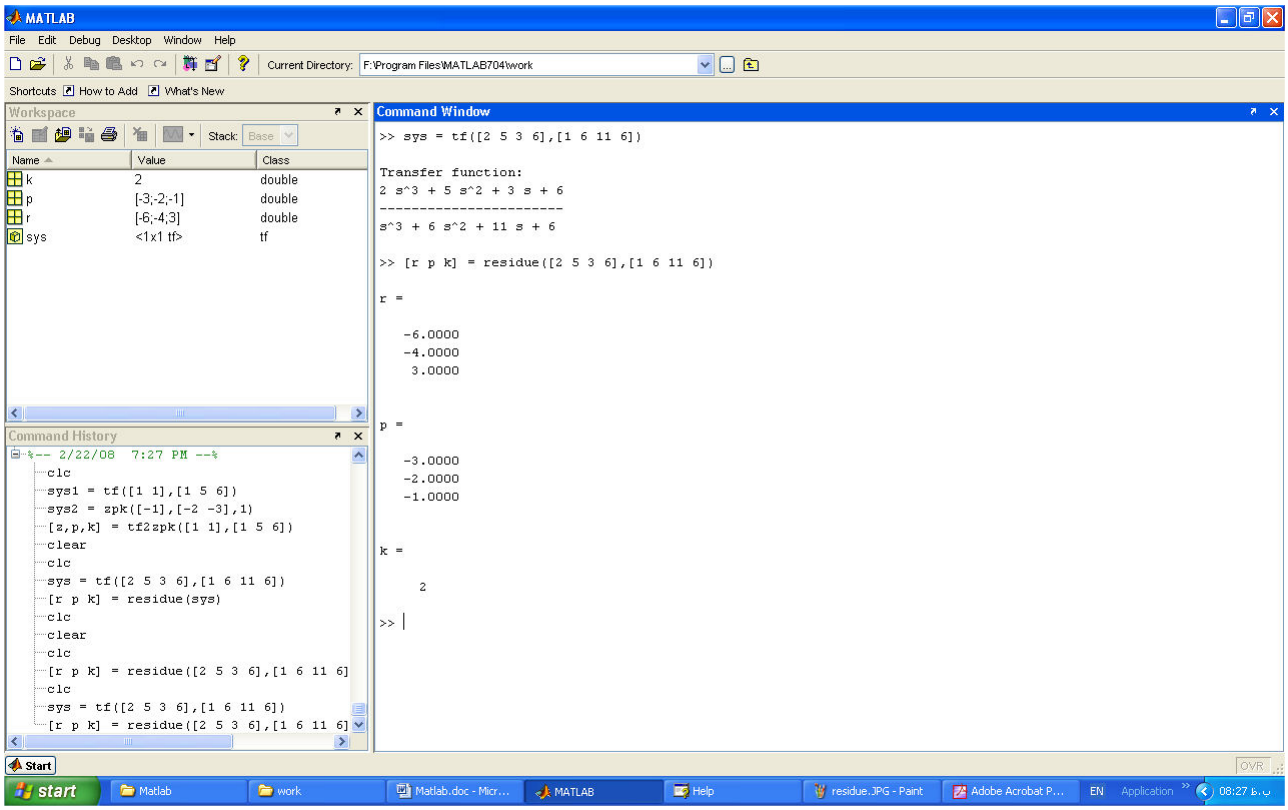
فرض کنید سیستمی به شکل زیر تعریف شده باشد:

$$\frac{b(s)}{a(s)} = \frac{b_1 s^m + b_2 s^{m-1} + b_3 s^{m-2} + \ldots + b_{m+1}}{a_1 s^n + a_2 s^{n-1} + a_3 s^{n-2} + \ldots + a_{n+1}}$$

می‌خواهیم این تابع تبدیل را به کسرهای جزیی به شکل زیر تبدیل کنیم:

$$\frac{b(s)}{a(s)} = \frac{r_1}{s - p_1} + \frac{r_2}{s - p_2} + \ldots + \frac{r_n}{s - p_n} + k(s)$$

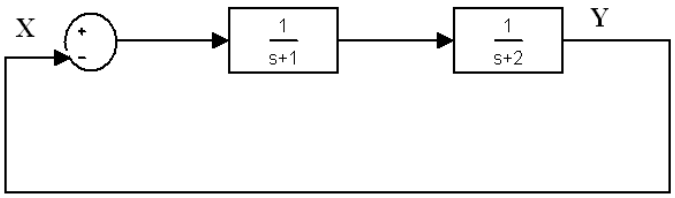
تابع [r p k] = residue(a ,b) ضرایب را محاسبه و در قالب بردارهای r و p و k ارائه می‌کند. a و b ضرایب صورت و مخرج تابع تبدیل هستند. شکل زیر را ببینید:



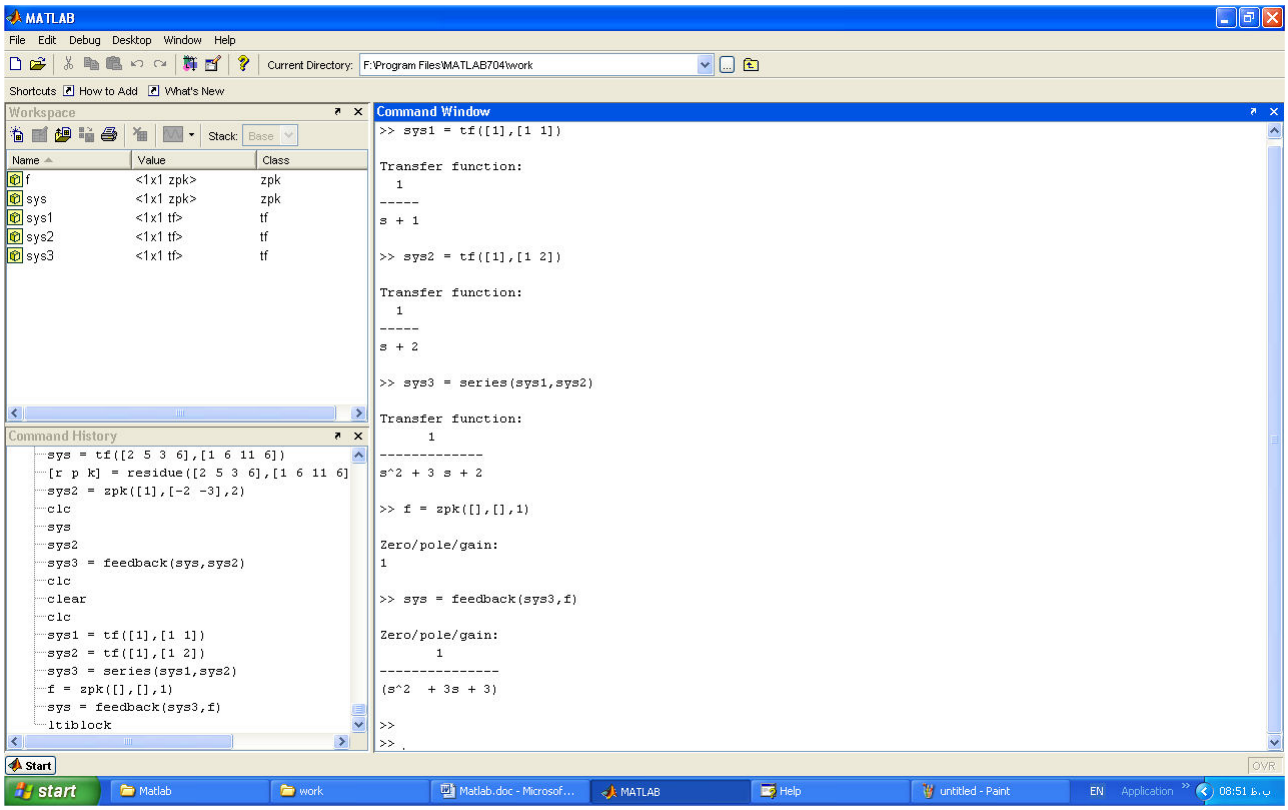

استفاده از تابع tf در اینجا لزومی ندارد و تنها برای نمایش شکل تابع تبدیل به کار رفته است.

### ترکیب سیستمها

به کمک توابع series، parallel و feedback می‌توانید سیستمهای مختلف را با هم ترکیب کرده و سیستم جدید بسازید. فرض کنید می‌خواهیم سیستم زیر را بسازیم:



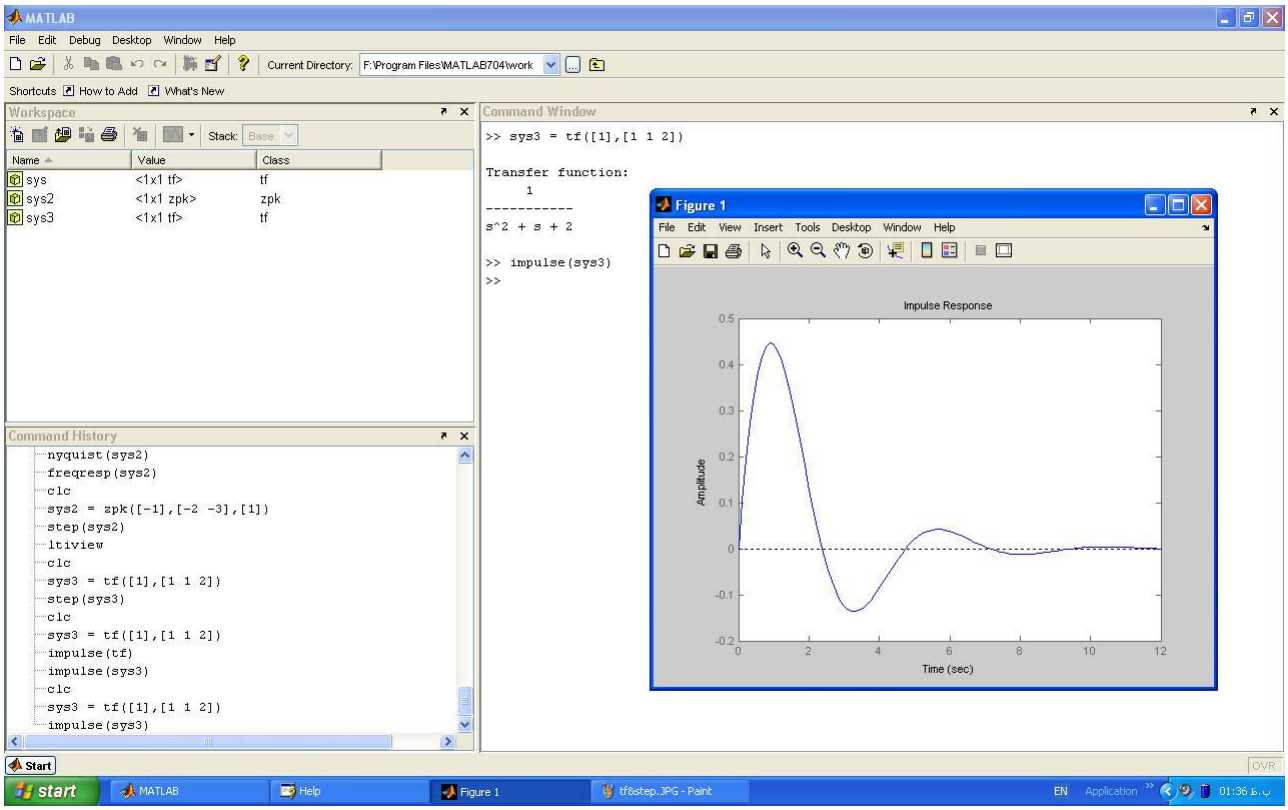

شکل زیر نحوه ساختن سیستم فوق را نشان می‌دهد:



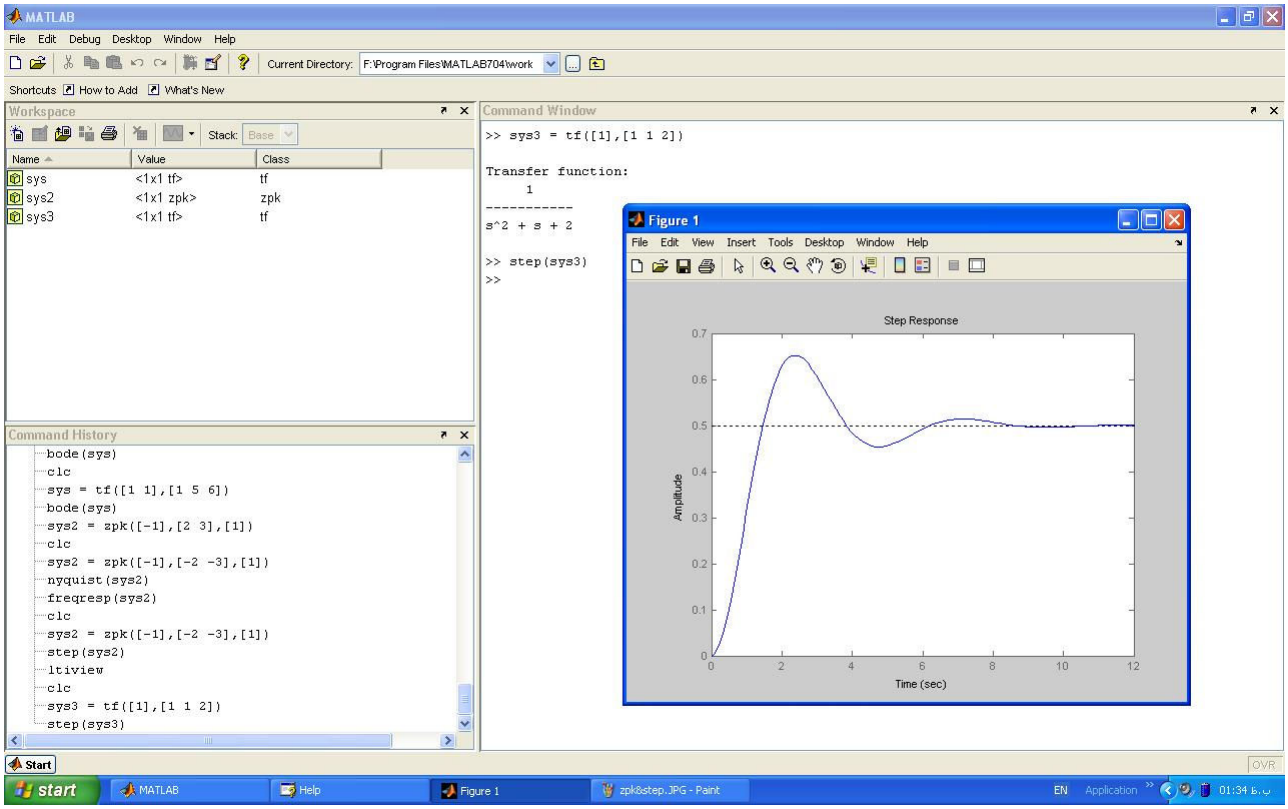

## تحلیل ترسیمی سیستمهای کنترلی

MATLAB توابع بسیار مفیدی برای رسم پاسخها و نمودارهای مرتبط با سیستمهای کنترلی در اختیار کاربر می‌گذارد که در ذیل کاربرد بعضی از آنها نشان داده شده است.

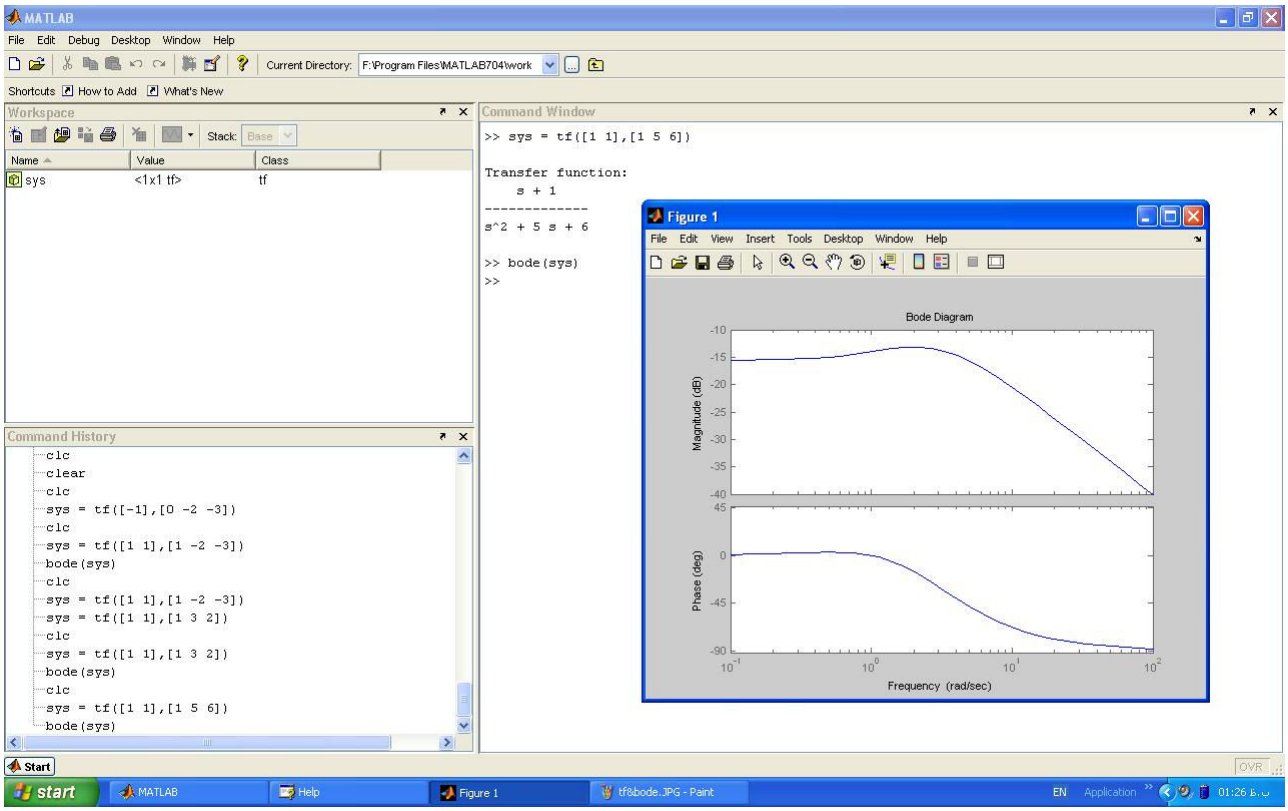
### رسم پاسخ ضربه به کمک تابع impulse



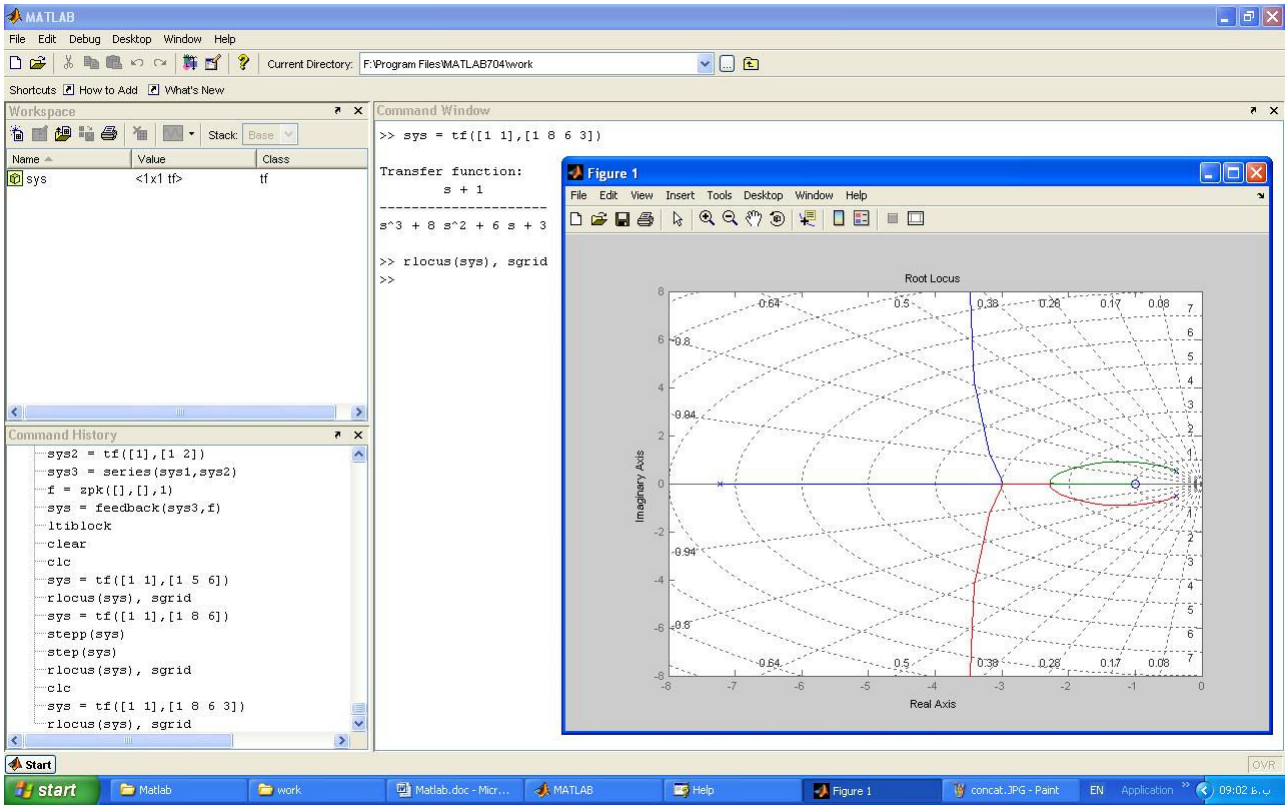

**رسم پاسخ پله به کمک تابع step**



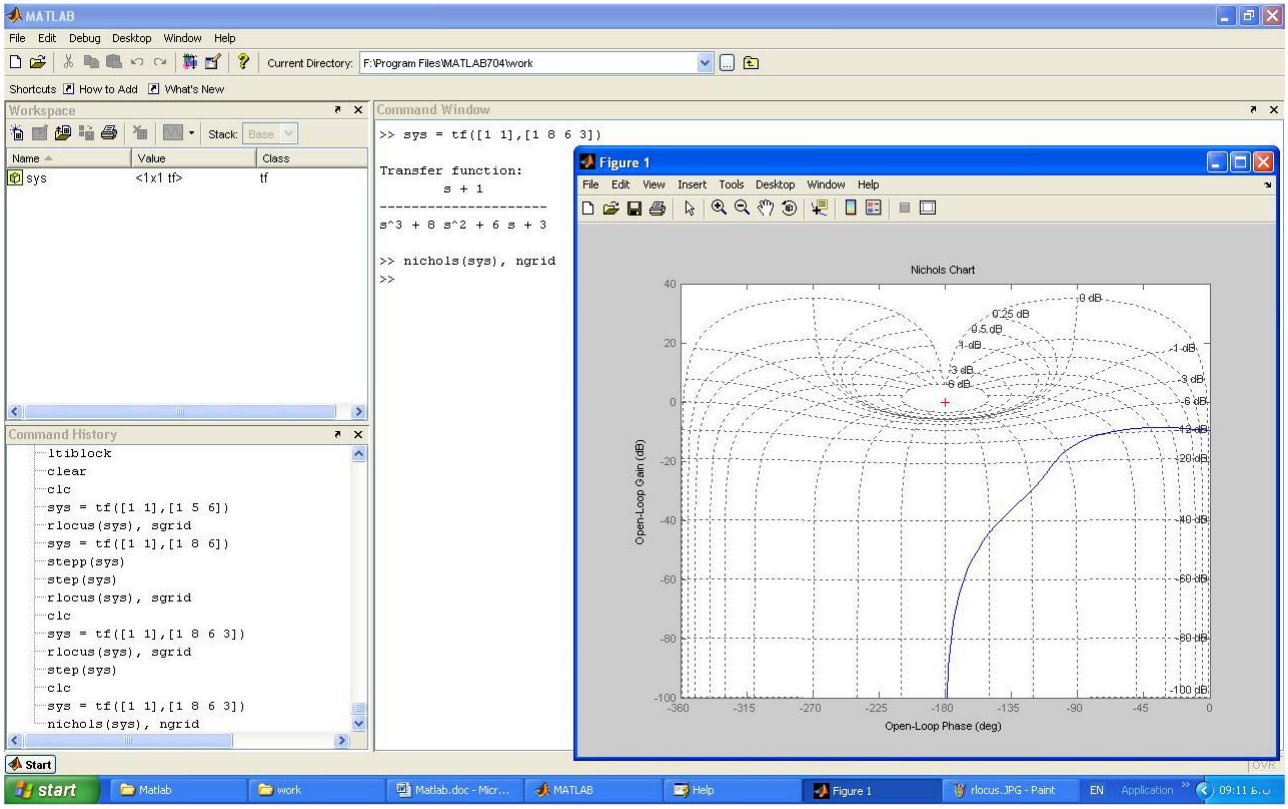
## رسم نمودارهای بوده به کمک تابع bode



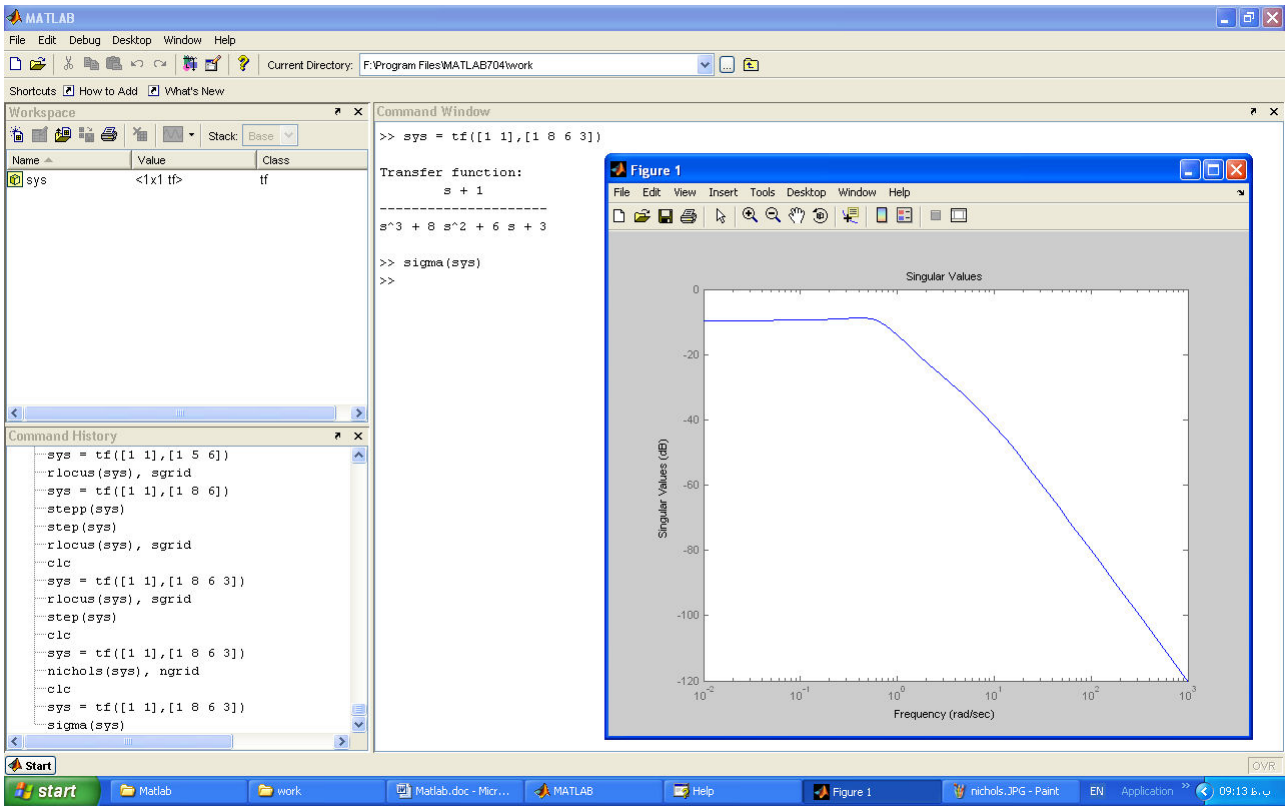

## تحلیل روت-لوکاس به کمک تابع rlucas



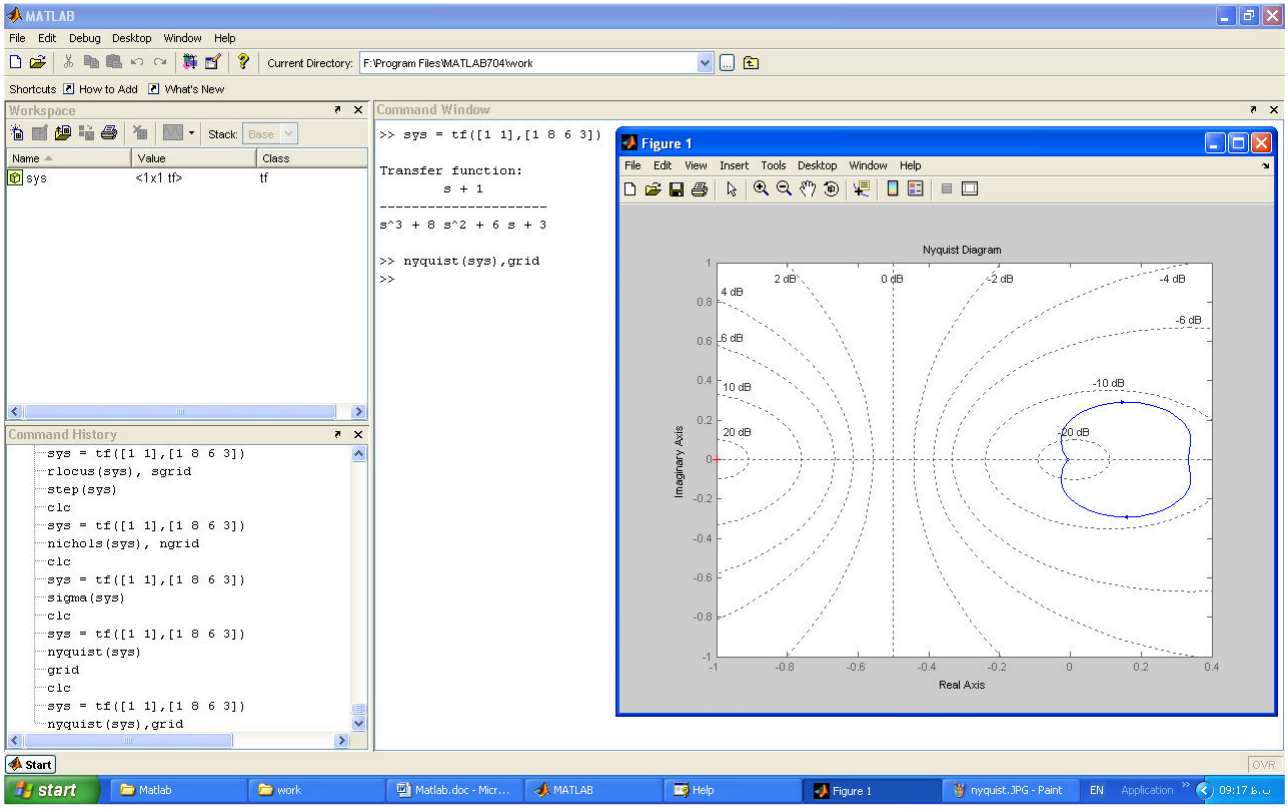
## تحلیل نیکولز و تابع nichols




## مقادیر تکین و تابع sigma

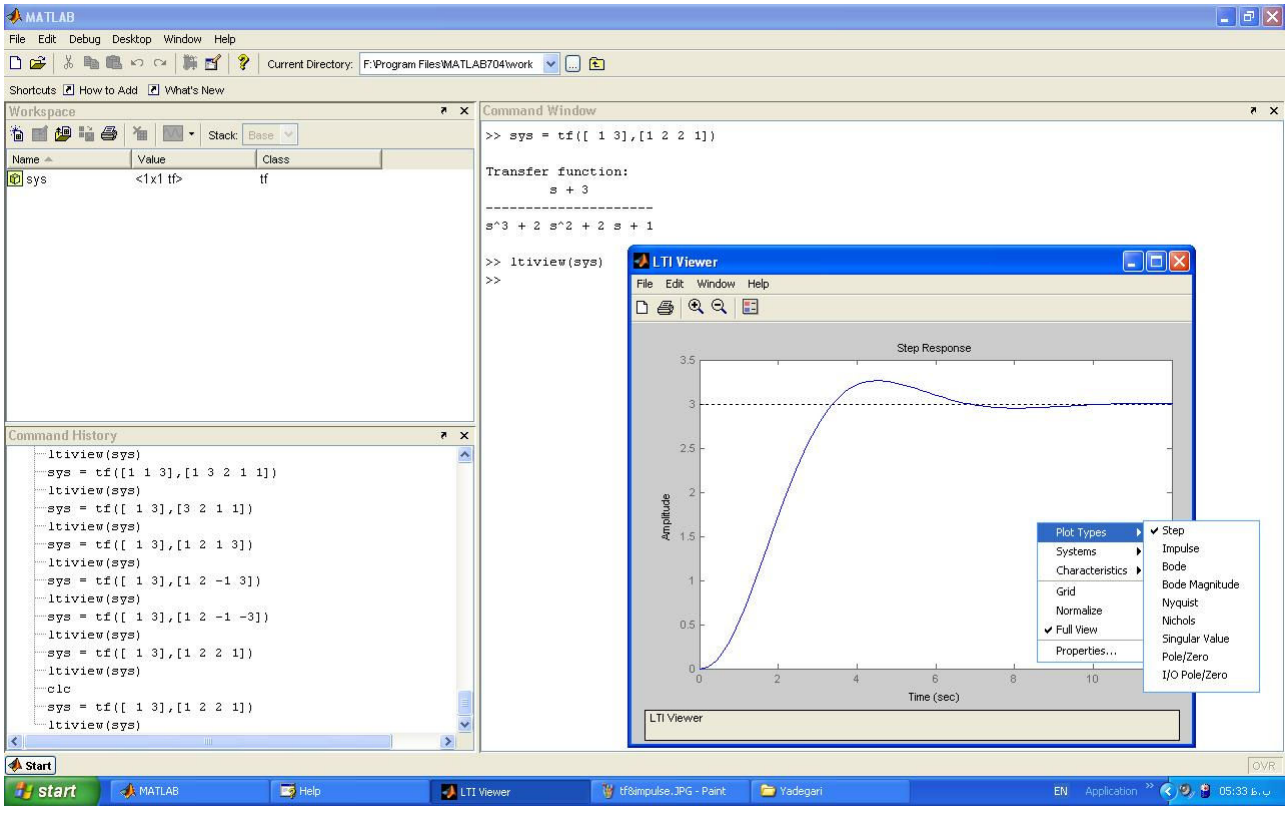


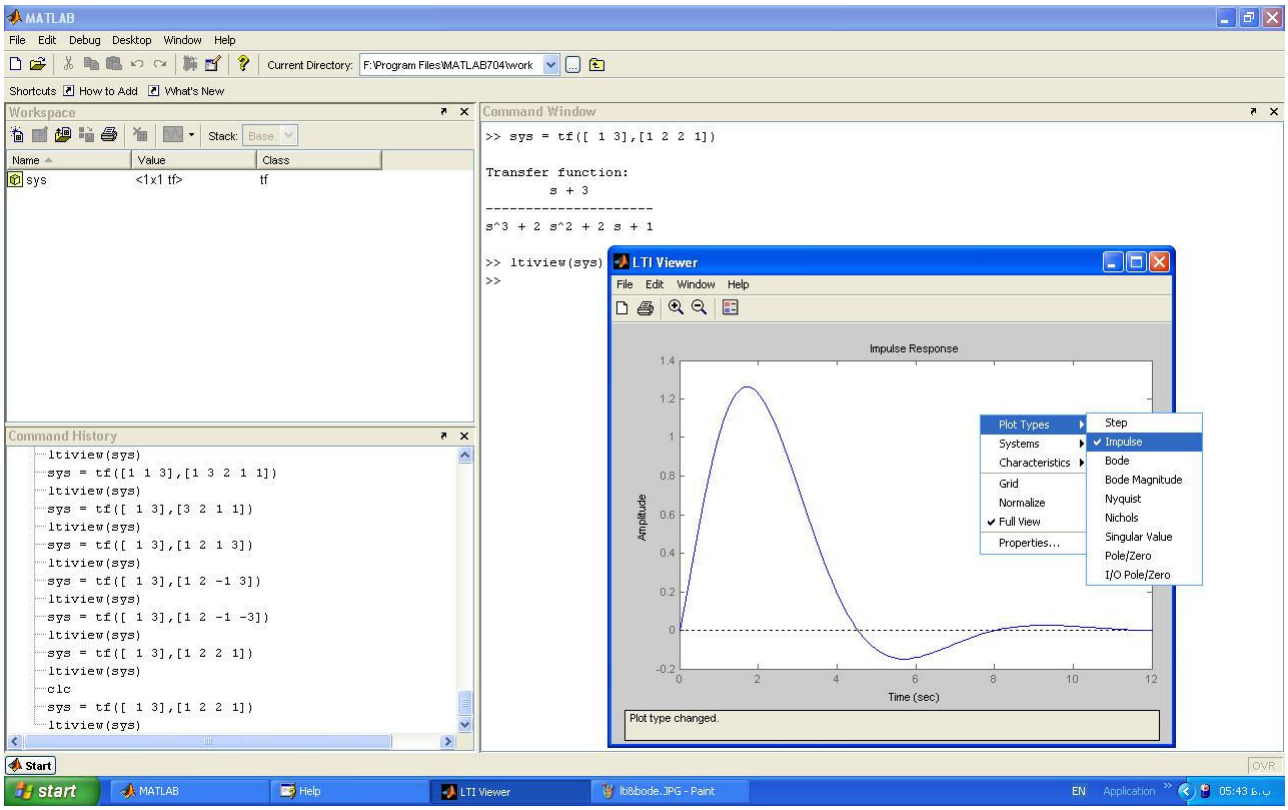
## تحلیل نایکوییست و تابع nyquist

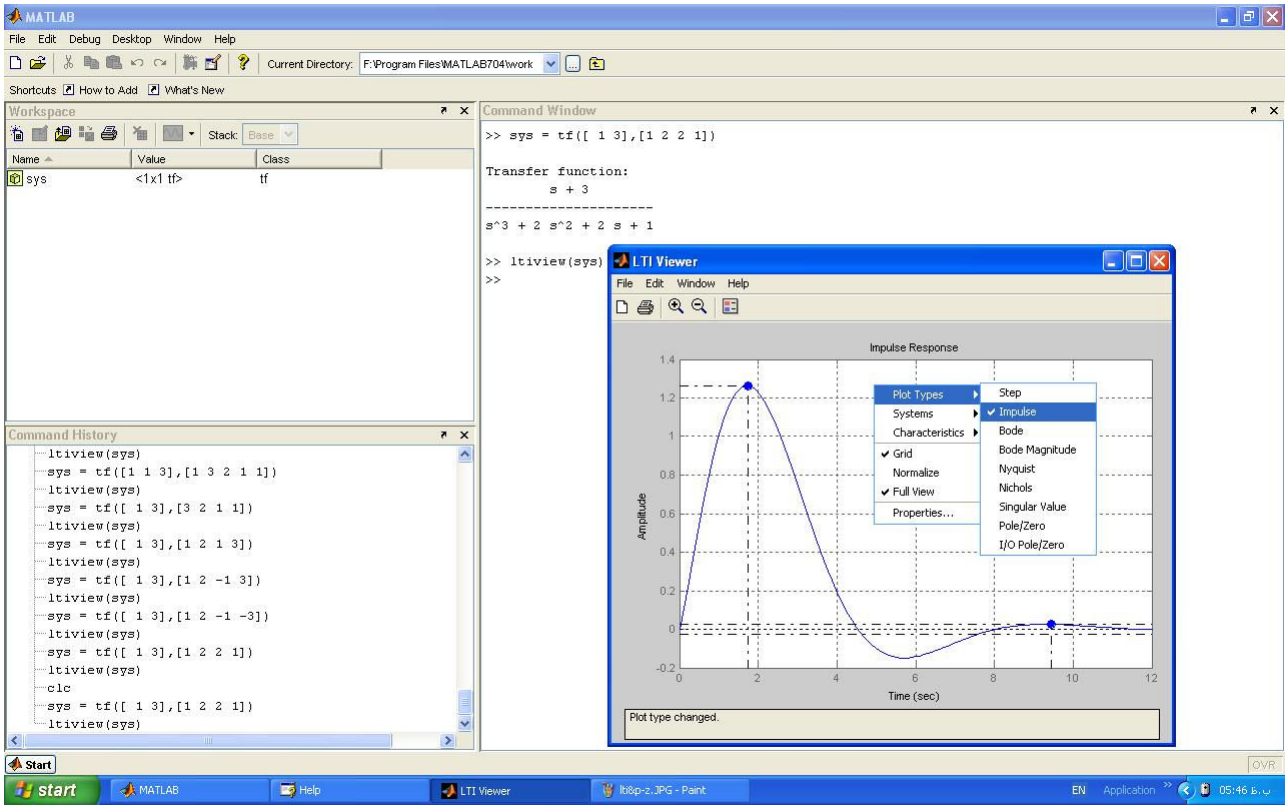


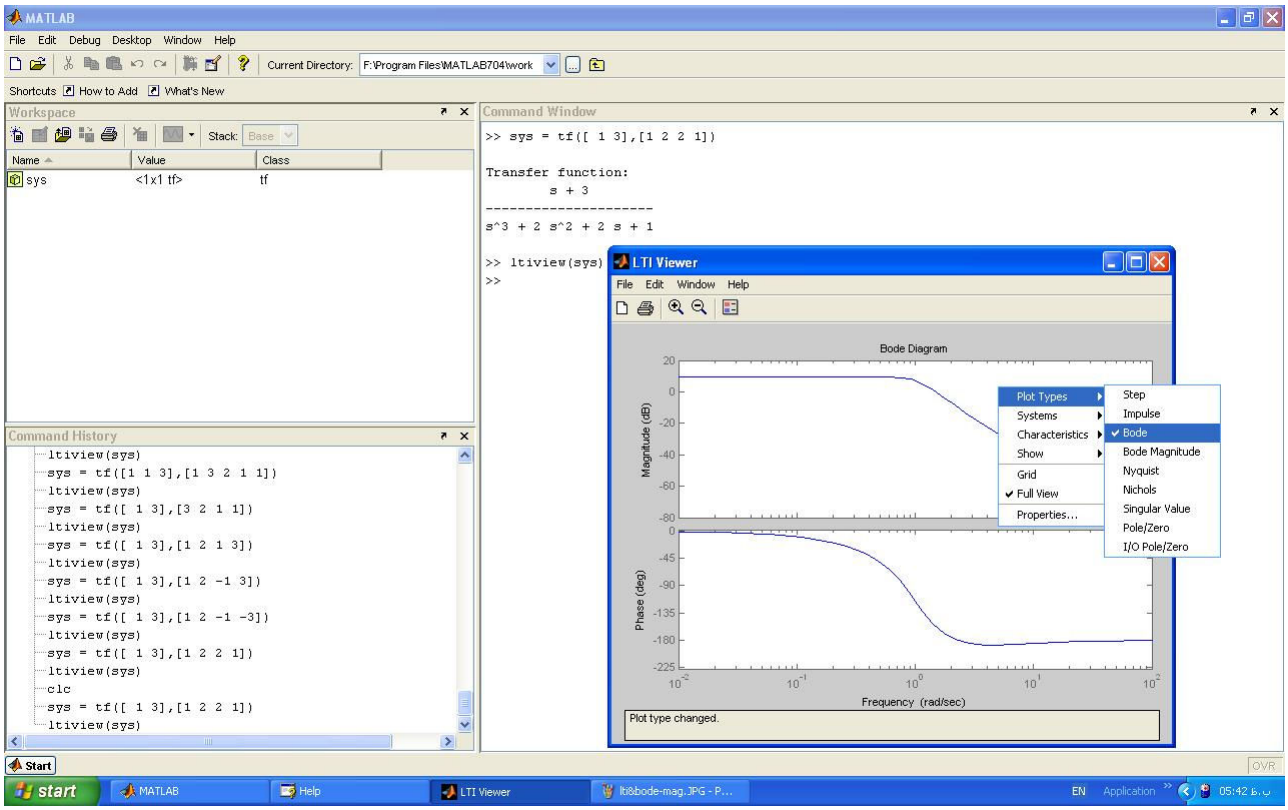

### ابزار ltiview

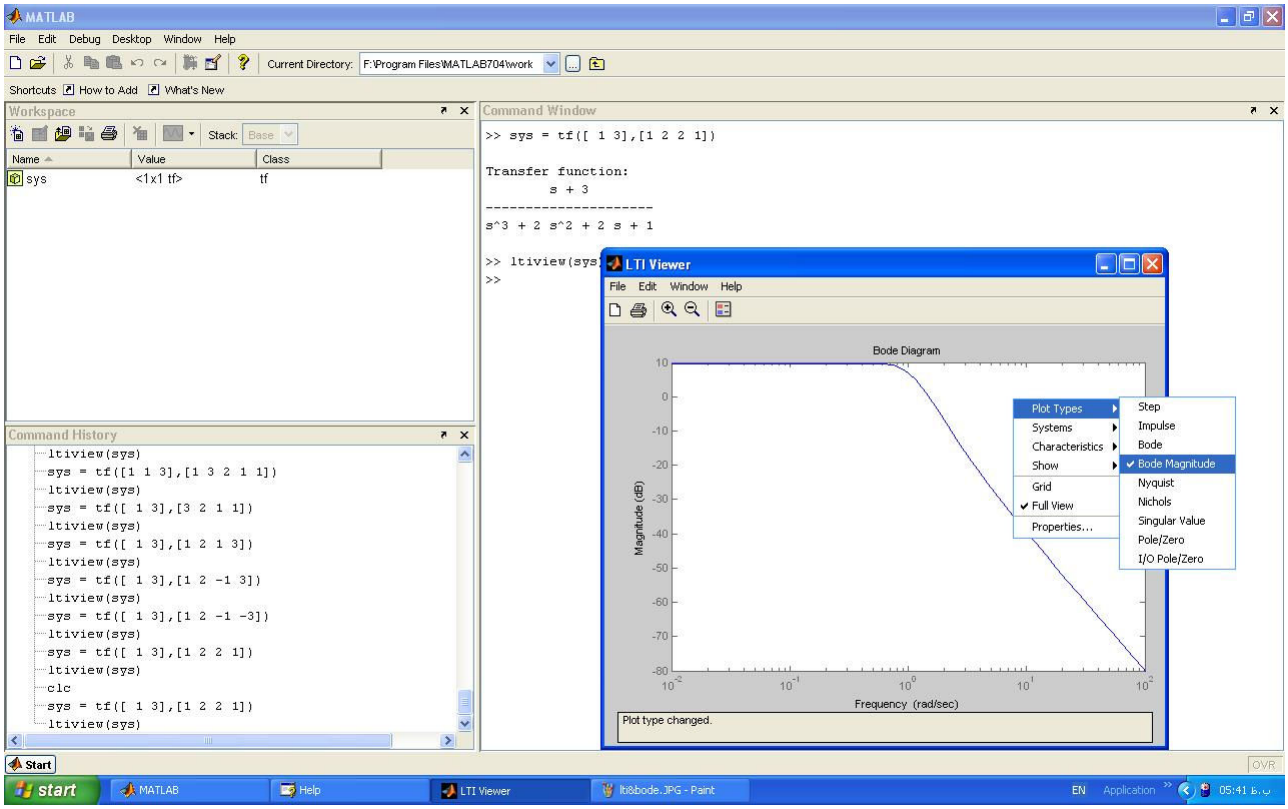
به کمک این ابزار می‌توان همه نمودارهای مرتبط با یک سیستم را بدون نیاز به فراخوانی تک‌تک توابع بررسی کرد:

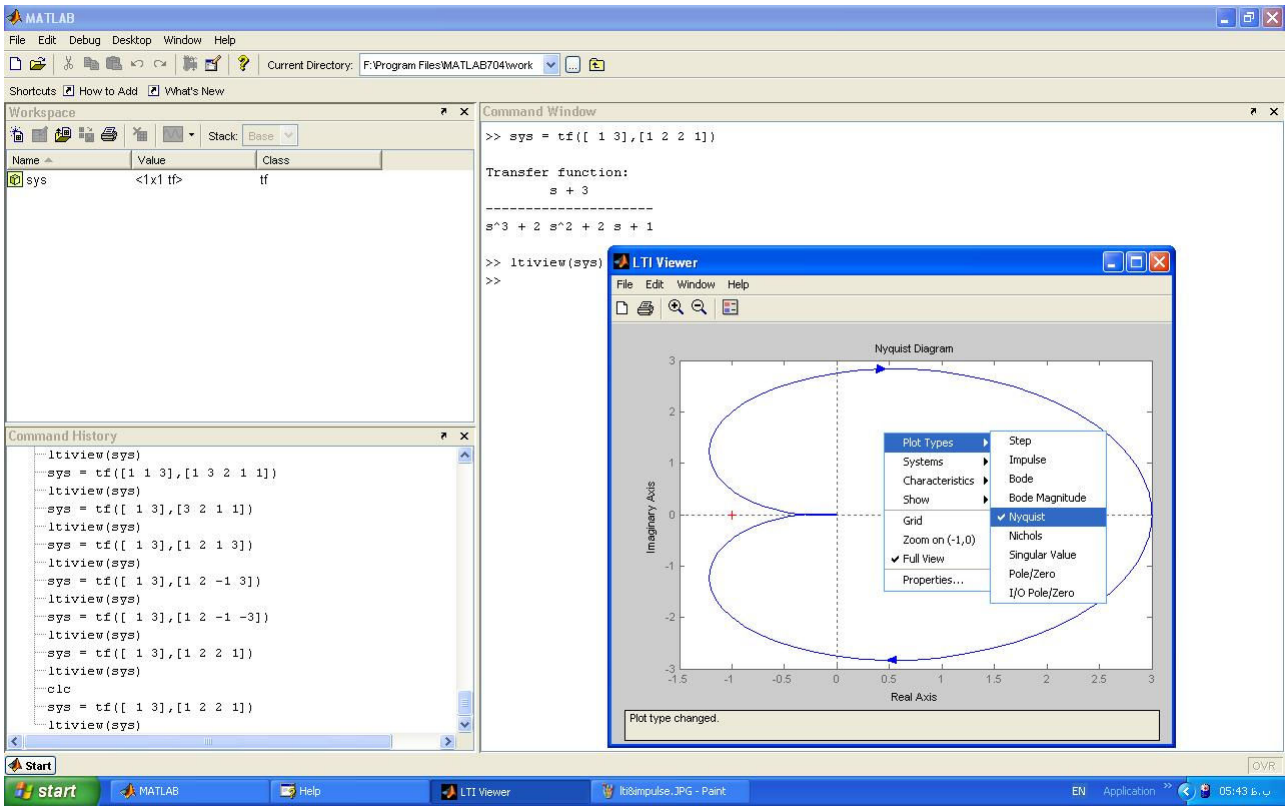


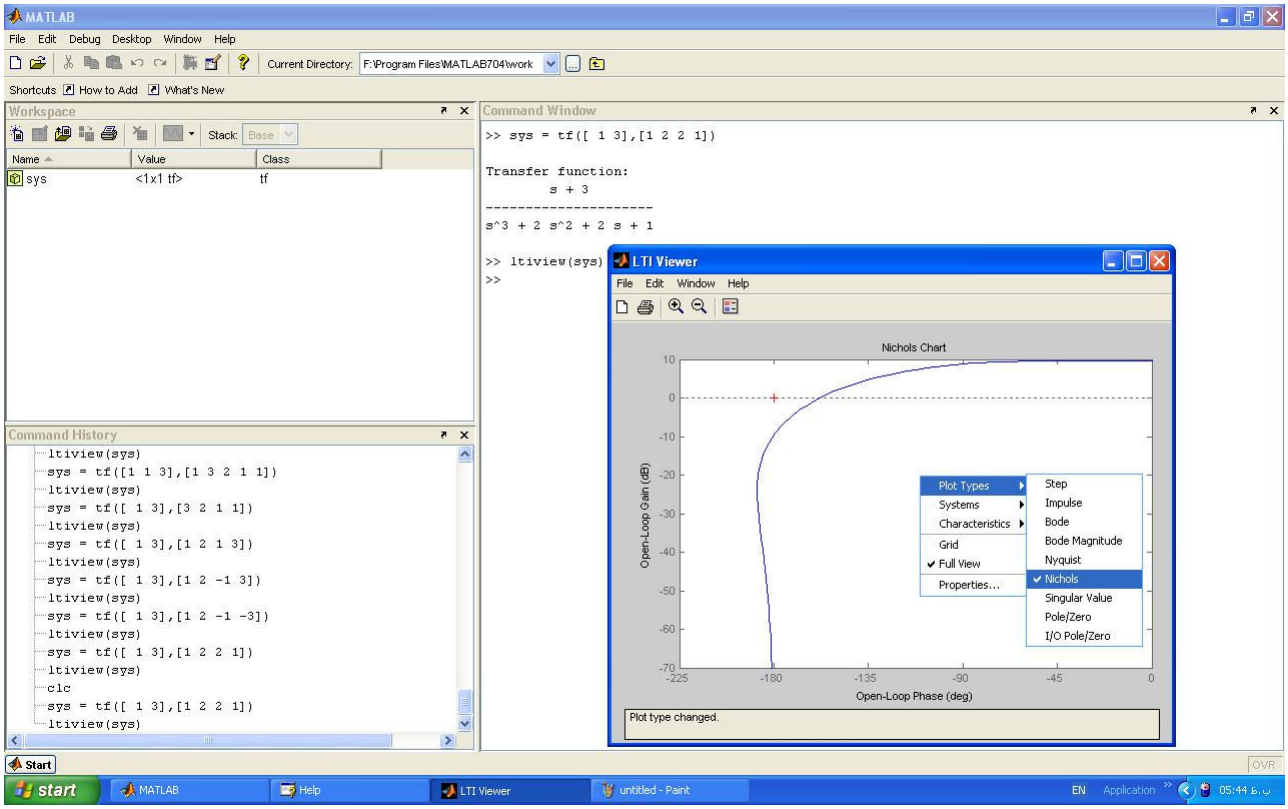

MATLAB
File Edit Debug Desktop Window Help
Current Directory: F:\Program Files\MATLAB704\work
Shortcuts How to Add What's New
Workspace
Stack: Base
Name Value Class
sys <1x1 tf> tf
Command Window
>> sys = tf([ 1 3],[1 2 2 1])
Transfer function:
s + 3
---------------------
s^3 + 2 s^2 + 2 s + 1
>> ltiview(sys)
>>
Command History
ltiview(sys)
sys = tf([1 1 3],[1 3 2 1 1])
ltiview(sys)
sys = tf([ 1 3],[3 2 1 1])
ltiview(sys)
sys = tf([ 1 3],[1 2 1 3])
ltiview(sys)
sys = tf([ 1 3],[1 2 -1 3])
ltiview(sys)
sys = tf([ 1 3],[1 2 -1 -3])
ltiview(sys)
sys = tf([ 1 3],[1 2 2 1])
ltiview(sys)
clc
sys = tf([ 1 3],[1 2 2 1])
ltiview(sys)
LTI Viewer
File Edit Window Help
Impulse Response
Amplitude
Time (sec)
Plot Types
Systems
Characteristics
Grid
Normalize
Full View
Properties...
Step
Impulse
Bode
Bode Magnitude
Nyquist
Nichols
Singular Value
Pole/Zero
I/O Pole/Zero
Plot type changed.
Start
start
MATLAB
Help
LTI Viewer
lti8bode.JPG - Paint
EN
Application
05:43 ب.ظ

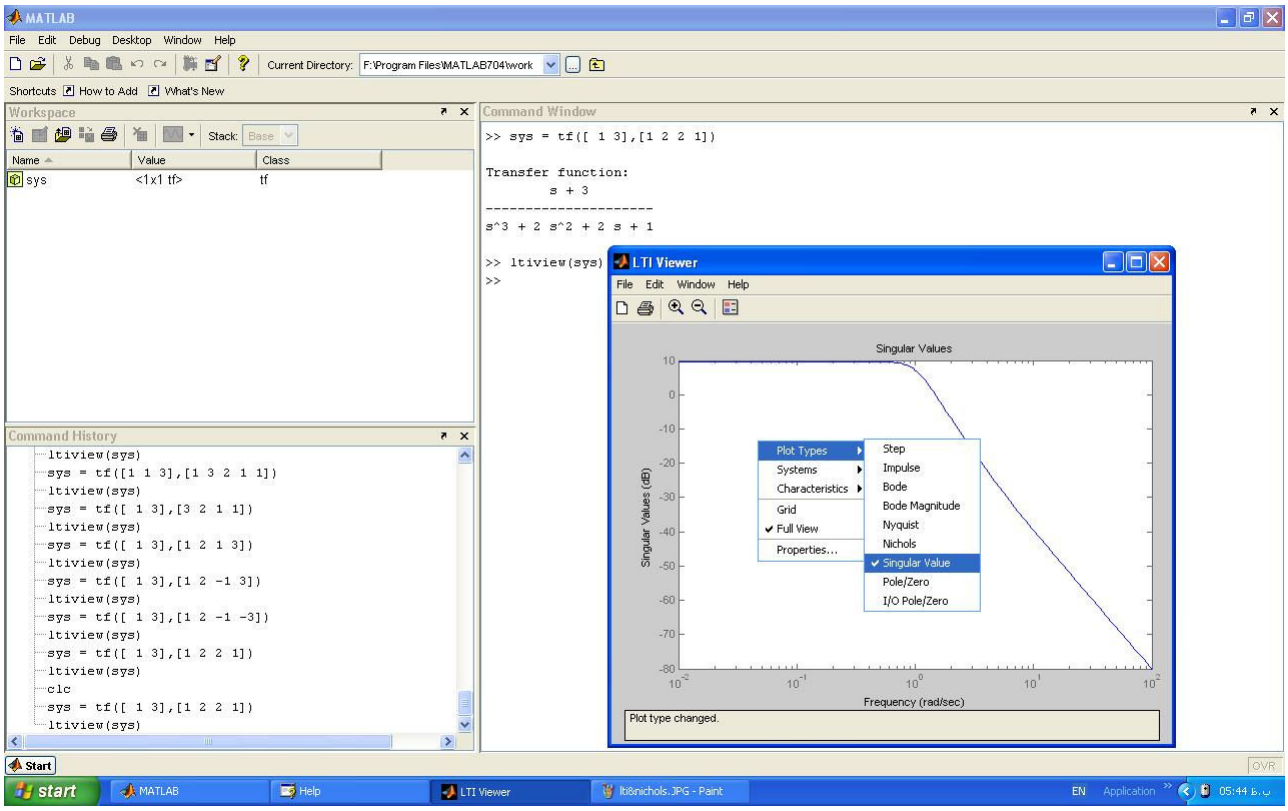

MATLAB
File Edit Debug Desktop Window Help
Current Directory: F:\Program Files\MATLAB704\work
Shortcuts How to Add What's New
Workspace
Stack: Base
Name Value Class
sys <1x1 tf> tf
Command Window
>> sys = tf([ 1 3],[1 2 2 1])
Transfer function:
s + 3
---------------------
s^3 + 2 s^2 + 2 s + 1
>> ltiview(sys)
>>
Command History
ltiview(sys)
sys = tf([1 1 3],[1 3 2 1 1])
ltiview(sys)
sys = tf([ 1 3],[3 2 1 1])
ltiview(sys)
sys = tf([ 1 3],[1 2 1 3])
ltiview(sys)
sys = tf([ 1 3],[1 2 -1 3])
ltiview(sys)
sys = tf([ 1 3],[1 2 -1 -3])
ltiview(sys)
sys = tf([ 1 3],[1 2 2 1])
ltiview(sys)
clc
sys = tf([ 1 3],[1 2 2 1])
ltiview(sys)
LTI Viewer
File Edit Window Help
Impulse Response
Amplitude
Time (sec)
Plot Types
Systems
Characteristics
Grid
Normalize
Full View
Properties...
Step
Impulse
Bode
Bode Magnitude
Nyquist
Nichols
Singular Value
Pole/Zero
I/O Pole/Zero
Plot type changed.
Start
start
MATLAB
Help
LTI Viewer
lti8p-z.JPG - Paint
EN
Application
05:46 ب.ظ

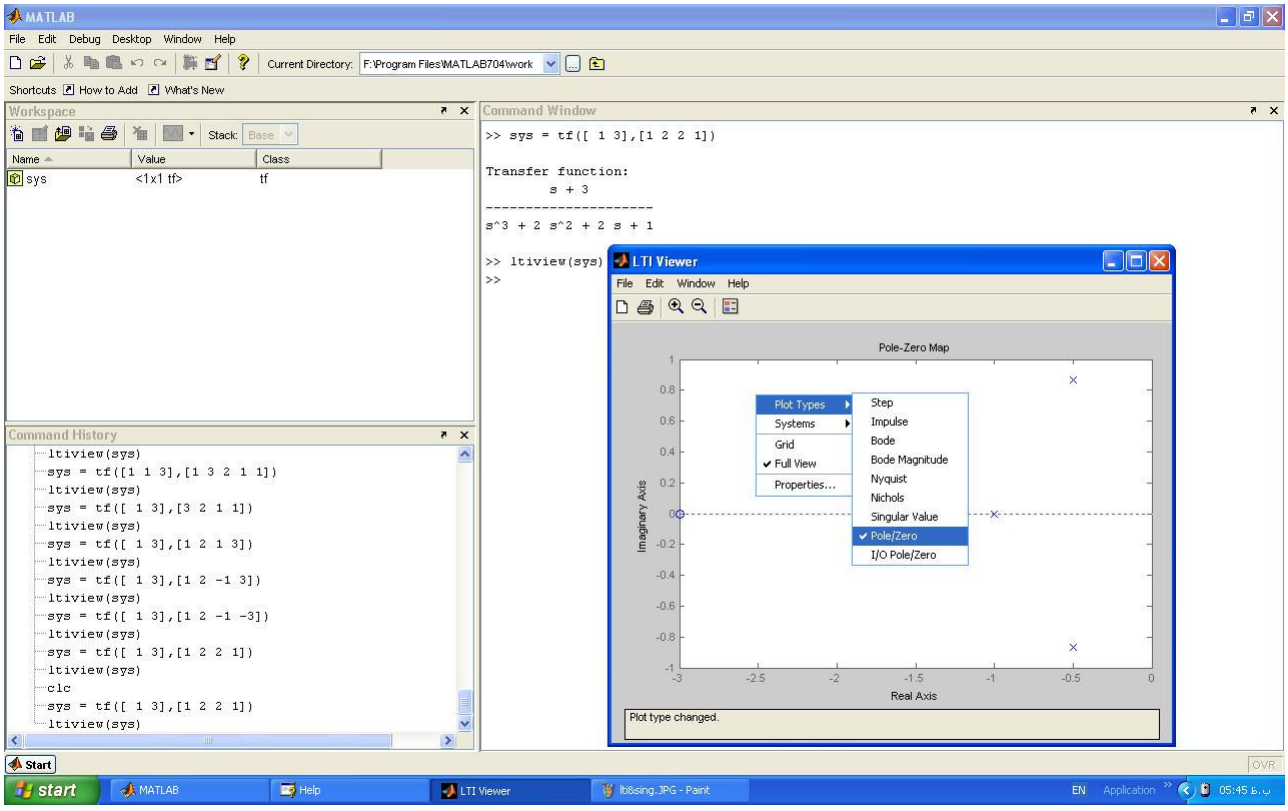

MATLAB
File Edit Debug Desktop Window Help
Current Directory: F:\Program Files\MATLAB704\work
Shortcuts How to Add What's New
Workspace
Stack: Base
Name Value Class
sys <1x1 tf> tf
Command Window
>> sys = tf([ 1 3],[1 2 2 1])
Transfer function:
s + 3
---------------------
s^3 + 2 s^2 + 2 s + 1
>> ltiview(sys)
>>
Command History
ltiview(sys)
sys = tf([1 1 3],[1 3 2 1 1])
ltiview(sys)
sys = tf([ 1 3],[3 2 1 1])
ltiview(sys)
sys = tf([ 1 3],[1 2 1 3])
ltiview(sys)
sys = tf([ 1 3],[1 2 -1 3])
ltiview(sys)
sys = tf([ 1 3],[1 2 -1 -3])
ltiview(sys)
sys = tf([ 1 3],[1 2 2 1])
ltiview(sys)
clc
sys = tf([ 1 3],[1 2 2 1])
ltiview(sys)
LTI Viewer
File Edit Window Help
Bode Diagram
Magnitude (dB)
Phase (deg)
Frequency (rad/sec)
Plot type changed.
Plot Types
Systems
Characteristics
Show
Grid
Full View
Properties...
Step
Impulse
Bode
Bode Magnitude
Nyquist
Nichols
Singular Value
Pole/Zero
I/O Pole/Zero
Start
start
MATLAB
Help
LTI Viewer
lti8bode-mag.JPG - P...
EN
Application
05:42 ب.ظ


MATLAB
File Edit Debug Desktop Window Help
Current Directory: F:\Program Files\MATLAB704\work
Shortcuts How to Add What's New
Workspace
Stack: Base
Name Value Class
sys <1x1 tf> tf
Command Window
>> sys = tf([ 1 3],[1 2 2 1])
Transfer function:
s + 3
---------------------
s^3 + 2 s^2 + 2 s + 1
>> ltiview(sys)
>>
Command History
ltiview(sys)
sys = tf([1 1 3],[1 3 2 1 1])
ltiview(sys)
sys = tf([ 1 3],[3 2 1 1])
ltiview(sys)
sys = tf([ 1 3],[1 2 1 3])
ltiview(sys)
sys = tf([ 1 3],[1 2 -1 3])
ltiview(sys)
sys = tf([ 1 3],[1 2 -1 -3])
ltiview(sys)
sys = tf([ 1 3],[1 2 2 1])
ltiview(sys)
clc
sys = tf([ 1 3],[1 2 2 1])
ltiview(sys)
LTI Viewer
File Edit Window Help
Bode Diagram
Magnitude (dB)
Frequency (rad/sec)
Plot type changed.
Plot Types
Systems
Characteristics
Show
Grid
Full View
Properties...
Step
Impulse
Bode
Bode Magnitude
Nyquist
Nichols
Singular Value
Pole/Zero
I/O Pole/Zero
Start
start
MATLAB
Help
LTI Viewer
lti8bode.JPG - Paint
EN
Application
05:41 ب.ظ


MATLAB
File Edit Debug Desktop Window Help
Current Directory: F:\Program Files\MATLAB704\work
Shortcuts How to Add What's New
Workspace
Stack: Base
Name Value Class
sys <1x1 tf> tf
Command Window
>> sys = tf([ 1 3],[1 2 2 1])
Transfer function:
s + 3
---------------------
s^3 + 2 s^2 + 2 s + 1
>> ltiview(sys)
>>
Command History
ltiview(sys)
sys = tf([1 1 3],[1 3 2 1 1])
ltiview(sys)
sys = tf([ 1 3],[3 2 1 1])
ltiview(sys)
sys = tf([ 1 3],[1 2 1 3])
ltiview(sys)
sys = tf([ 1 3],[1 2 -1 3])
ltiview(sys)
sys = tf([ 1 3],[1 2 -1 -3])
ltiview(sys)
sys = tf([ 1 3],[1 2 2 1])
ltiview(sys)
clc
sys = tf([ 1 3],[1 2 2 1])
ltiview(sys)
LTI Viewer
File Edit Window Help
Nyquist Diagram
Imaginary Axis
Real Axis
Plot Types
Systems
Characteristics
Show
Grid
Zoom on (-1,0)
Full View
Properties...
Step
Impulse
Bode
Bode Magnitude
Nyquist
Nichols
Singular Value
Pole/Zero
I/O Pole/Zero
Plot type changed.
Start
start
MATLAB
Help
LTI Viewer
lti8impulse.JPG - Paint
EN
Application
05:43 ب.ظ

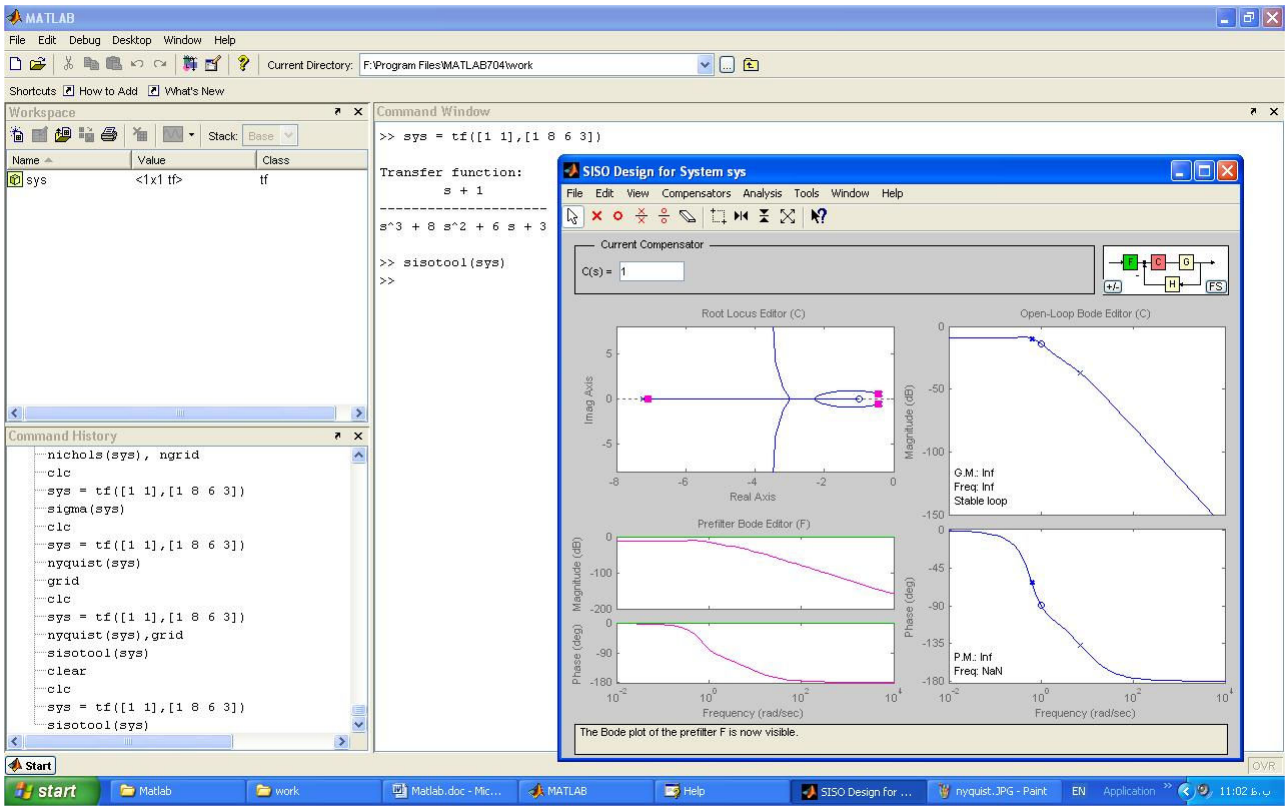

MATLAB
File Edit Debug Desktop Window Help
Current Directory: F:\Program Files\MATLAB704\work
Shortcuts How to Add What's New
Workspace
Stack: Base
Name Value Class
sys <1x1 tf> tf
Command Window
>> sys = tf([ 1 3],[1 2 2 1])
Transfer function:
s + 3
---------------------
s^3 + 2 s^2 + 2 s + 1
>> ltiview(sys)
>>
Command History
ltiview(sys)
sys = tf([1 1 3],[1 3 2 1 1])
ltiview(sys)
sys = tf([ 1 3],[3 2 1 1])
ltiview(sys)
sys = tf([ 1 3],[1 2 1 3])
ltiview(sys)
sys = tf([ 1 3],[1 2 -1 3])
ltiview(sys)
sys = tf([ 1 3],[1 2 -1 -3])
ltiview(sys)
sys = tf([ 1 3],[1 2 2 1])
ltiview(sys)
clc
sys = tf([ 1 3],[1 2 2 1])
ltiview(sys)
LTI Viewer
File Edit Window Help
Nichols Chart
Open-Loop Gain (dB)
Open-Loop Phase (deg)
Plot Types
Systems
Characteristics
Grid
Full View
Properties...
Step
Impulse
Bode
Bode Magnitude
Nyquist
Nichols
Singular Value
Pole/Zero
I/O Pole/Zero
Plot type changed.
Start
start
MATLAB
Help
LTI Viewer
untitled - Paint
EN
Application
05:44 ب.ظ




### ابزار sisotool

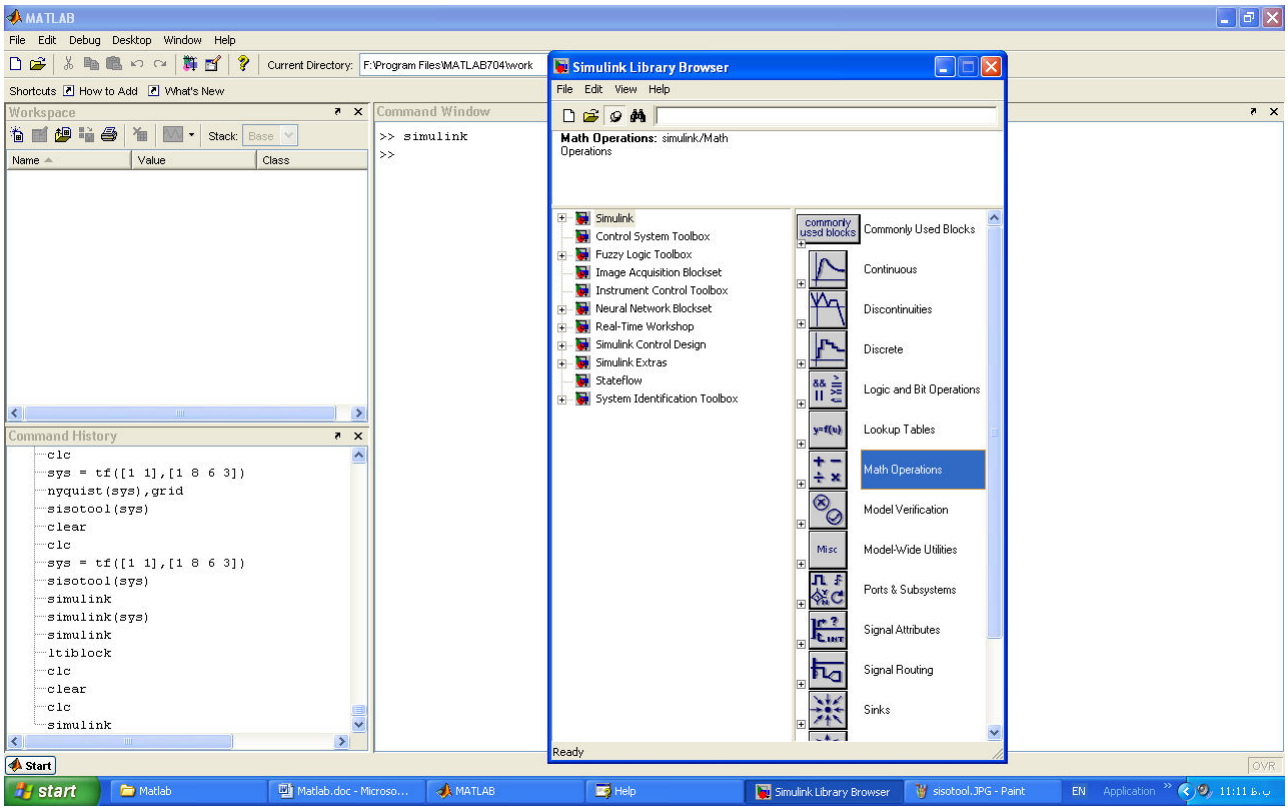
sisotool یک ابزار طراحی و تحلیل سیستمهای یک ورودی – یک خروجی (Single Input Single Output) است.




به کمک این ابزار می‌توانید به راحتی ساختار سیستم خود را تغییر داده و اثر این تغییرات روی نمودارهای سیستم را بررسی کنید. مثلاً با کلیک روی کلید FS (گوشه بالا سمت راست) می‌توانید ساختار فیدبک و با کلیک روی نشانه +/- علامت فیدبک را عوض کنید. از طریق منوی Analysis می‌توانید نمودارهای مهم سیستم را ببینید. به علاوه می‌توانید به کمک منوی Tools، سیستم خود را به simlulink صادر کنید.

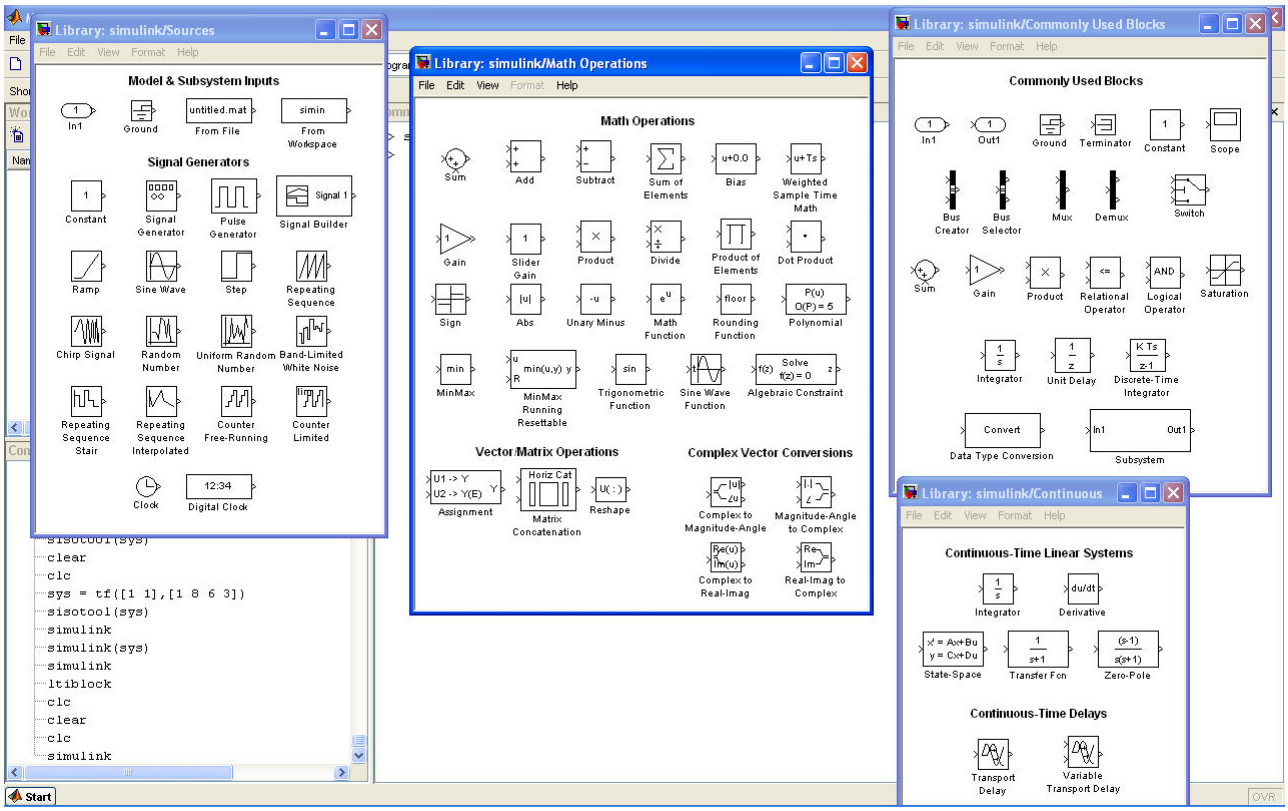
### ابزار simulink

simulink ابزاری قوی برای وارد کردن جزییات سیستمهای مختلف به MATLAB به کمک بلوک دیاگرامهای گرافیکی است. این ابزار می‌تواند در ارتباط با جعبه‌ابزارهای متنوعی استفاده شود. با اجرای دستور simulink کتابخانه این ابزار که شامل بلوکهای متنوع است در دسترس کاربر قرار می‌گیرد:




برای استفاده از simulink در تحلیل سیستمهای کنترلی از فرمان ltiblock استفاده کنید. اجرای این دستور محیطی برای تحلیل سیستمهای کنترلی را در اختیار شما می‌گذارد تا بلوکهای simulink را از داخل کتابخانه به داخل این محیط آورده و پس از انجام اتصالات لازم، عملکرد سیستم را شبیه‌سازی کنید.

در ذیل بعضی از بلوکهای مفید simulink برای تحلیل سیستمهای کنترلی را می‌بینید:



بلوک scope می‌تواند در تحلیل شکلهای خروجی بسیار مفید باشد.

### توابع جعبه‌ابزار سیستمهای کنترلی

## General
ctrlpref Set Control System Toolbox preferences
ltimodels Detailed help on the various types of LTI models
ltiprops Detailed help on available LTI model properties

## Creating Linear Models
filt Specify a digital filter
frd Create a frequency-response data models
lti/set Set/modify properties of LTI models
ss, dss Create state-space models (continuous/discrete)
tf Create transfer function models
zpk Create zero/pole/gain models

## Data Extraction
dssdata Descriptor version of ssdata
frdata Extract frequency-response data
lti/get Access values of LTI model properties
ssdata Extract state-space matrices
tfdata Extract numerators and denominators
zpkdata Extract zero/pole/gain data

## Conversions
c2d Convert from continuous- to discrete-time models
chunits Convert the units property for FRD models
d2c Convert from discrete- to continuous-time models
d2d Test true for continuous-time models
frd Convert to a frequency-response data model
ss Convert to a state-space model
tf Convert to a transfer function model
zpk Convert to a zero/pole/gain model

## System Interconnections
append Group LTI systems by appending inputs and outputs
connect Derive state-space models from block diagram descriptions
feedback Feedback connections of two systems
lft Generalized feedback interconnection (Redheffer star product)
parallel Generalized parallel connection (see also overloaded +)
series Generalized series connection (see also overloaded *)

## System Gain and Dynamics
bandwidth System bandwidth
dcgain D.C. (low-frequency) gain
bandwidth System bandwidth
damp Natural frequency and damping of system poles
dsort Norms of LTI systems
esort Sort continuous poles by real part
iopzmap Input/output pole/zero map
lti/norm Norms of LTI systems
modsep Region-based modal decomposition
pole, eig System poles
pzmap Pole/zero map
stabsep Stable/unstable decomposition

## Time Domain Analysis
covar Covariance of response to white noise

gensig Generate input signal for lsim
impulse Impulse response
initial Response of state-space system with given initial state
lsim Response to arbitrary inputs
ltiview Response analysis GUI (LTI Viewer)
step Step response

## Frequency Domain Analysis

allmargin All crossover frequencies and related gain/phase margins
bode Bode diagrams of the frequency response
bodemag Bode magnitude diagram only
evalfr Evaluate frequency response at given frequency
freqresp Frequency response over a frequency grid
frd/interp Interpolate frequency-esponse data
ltiview Response analysis GUI (LTI Viewer)
margin Gain and phase margins
nichols Nichols plot
nyquist Nyquist plot
sigma Plot the pole/zero map of an LTI model

## Classical Design

rlocus Evans root locus
sisotool SISO design GUI (root locus and loop-shaping techniques)

## Pole Placement

acker SISO pole placement
estim Form estimator given estimator gain
place MIMO pole placement
reg Form regulator given state-feedback and estimator gain

## LQR/LQG Design

augstate Augment output by appending states
lqg Single-step LQG design
lgr, dlqr Linear-quadratic (LQ) state-feedback regulator
lqrd Discrete LQ regulator for continuous plants
lqrreg Form LQG regulator given LQ gain and Kalman estimator
lqgy LQ regulator with output weighting
kalman Kalman estimator
kalmand Discrete Kalman estimator for continuous plants

## State-Space Models

balreal Grammian-based input/output balancing
canon State-space canonical forms
ctrb Controllability matrix
gram Controllability and observability grammians
minreal Minimal realization and pole/zero cancellation
modred Model state reduction
margin Calculate gain and phase margins
ngrid Superimpose grid lines on a Nichols plot
nichols Calculate Nichols plot
nyquist Calculate Nyquist plot
obsv Observability matrix
sminreal Structurally minimal realization
ss2ss State coordinate transformation
ssbal Diagonal balancing of state-space realizations

## Time Delays

delay2z Replace delays by poles at z=0 or FRD phase shift
hasdelay True for models with time delays
pade Pade approximation of time delays

totaldelay Total delay between each input/output pair

## Model Dimensions and Characteristics

class Model type ('tf', 'zpk', 'ss', or 'frd')
isct True for continuous-time models
isdt True for discrete-time models
isproper True for proper models
issiso True for single-input/single-output models
lti/ndims Number of dimensions
lti/isempty True for empty LTI models
reshape Reshape array of linear models size Model sizes and order

## Overloaded and Arithmetic Operators

+ and - Add and subtract systems (parallel connection)
* Multiply systems (series connection)
\ Left divide -- sys1\sys2 means inv(sys1)*sys2
/ Right divide -- sys1/sys2 means sys1*inv(sys2)
^ Powers of a given system
' Pertransposition
.' Transposition of input/output map
[..] Concatenate models along inputs or outputs
stack Stack models/arrays along some array dimension
lti/*inv* Inverse of an LTI system
conj Complex conjugation of model coefficients

## Matrix Equation Solvers

bdschur Block diagonalization of a square matrix
care, dare Solve algebraic Riccati equations
gcare, gdare Solve generalized algebraic Riccati equations
lyap, dlyap Solve Lyapunov equations
lyapchol, dlyapschol Square-root Lyapunov equations

## Command-Line Plot Customization

bodeplot Bode magnitude and phase plus plot handle
getoptions Get the plot options handle
hsvplot Hankel singular value plus plot handle
impulseplot Impulse response plus plot handle
initialplot Initial condition plus plot handle
iopzplot Pole/zero maps for input/output pairs plus plot handle
lsimplot Time response to arbitrary inputs plus plot handle
nicholsplot Nichols plot plus plot handle
nyquistplot Nyquist plus plot handle
pzplot Pole/zero plus plot handle
rlocusplot Root locus plus plot handle
setoptions Set plot options
sigmaplot Singular values of the frequency response plus plot handle
stepplot Step response plus plot handle