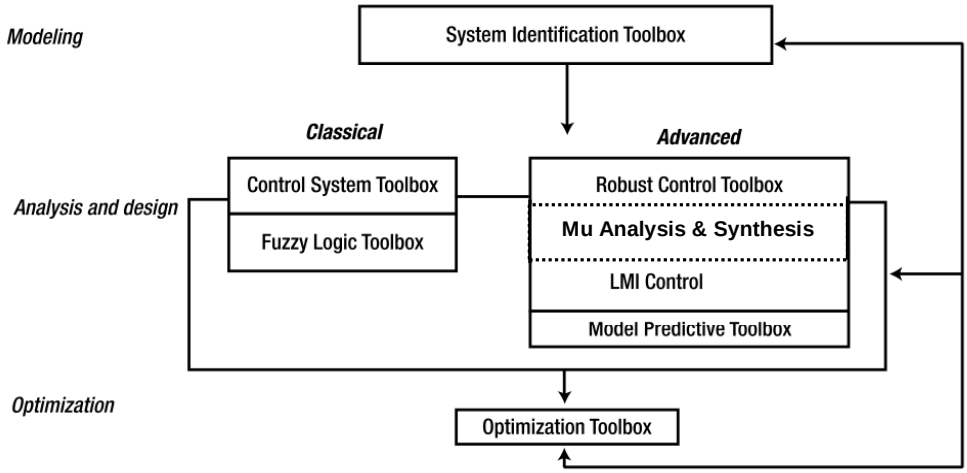
MATLAB toolboxes for control stems analysis and design



The most important MATLAB toolboxes for control systems can be classified into three families:

- modeling – System Identification Toolbox;
- classical analysis and design products – Control System Toolbox, Fuzzy Logic Toolbox;
- advanced analysis and design products – Robust Control Toolbox (including Mu-Analysis & Synthesis and LMI Control), Model Predictive Toolbox;
- optimization products – Optimization Toolbox.

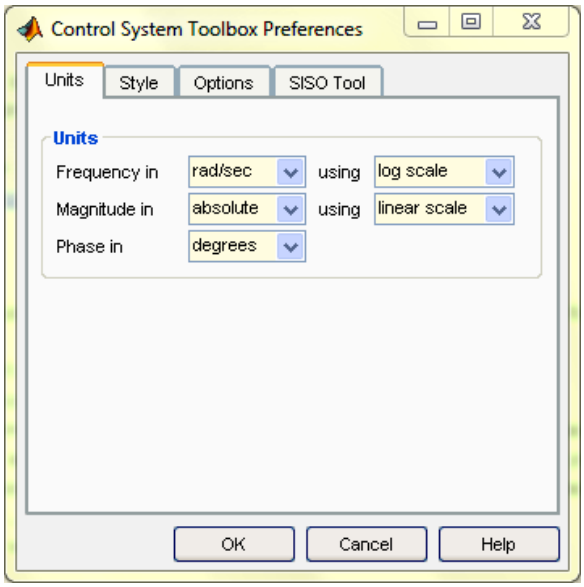
The above diagram illustrates this classification.

The Control System Toolbox Commands

The Control System Toolbox commands can be classified according to their purpose as follows:

General

`Ctrlpref` Opens a GUI which allows you to change the Control System Toolbox preferences.



Creation of linear models

`tf`: Creates a transfer function model
`zpk`: Creates a zero-pole-gain model
`ss`: Creates a state-space model
`dss`: Creates a descriptor state-space model
`frd`: Creates a frequency-response data model
`set`: Locates and modifies properties of LTI models

Data extraction

`tfdata`: Accesses transfer function data (in particular extracts the numerator and denominator of the transfer function)
`zpkdata`: Accesses zero-pole-gain data
`ssdata`: Accesses state-space model data
`get`: Accesses properties of LTI models

Conversions

`s`: Converts to a state-space model
`zpk`: Converts to a zero-pole-gain model
`tf`: Converts to a transfer function model
`frd`: Converts to a frequency-response data model
`c2d`: Converts a model from continuous to discrete time
`d2c`: Converts a model from discrete to continuous time
`d2d`: Resamples a discrete time model

System interconnection

`append`: Groups models by appending their inputs and outputs
`parallel`: Parallel connection of two models
`series`: Series connection of two models
`feedback`: Connection feedback of two systems
`lft`: Generalized feedback interconnection of two models
`connect`: Block diagram interconnection of dynamic systems

Dynamic models

`iopzmap`: Plots a pole-zero map for input/output pairs of a model
`bandwidth`: Returns the frequency-response bandwidth of the system
`pole`: Computes the poles of a dynamic system
`zero`: Returns the zeros and gain of a SISO dynamic system
`pzmap`: Returns a pole-zero plot of a dynamic system
`damp`: Returns the natural frequency and damping ratio of the poles of a system
`dcgain`: Returns the low frequency (DC) gain of an LTI system
`norm`: Returns the norm of a linear model
`covar`: Returns the covariance of a system driven by white noise

Time-domain analysis

`ltiview`: An LTI viewer for LTI system response analysis
`step`: Produces a step response plot of a dynamic system
`impulse`: Produces an impulse response plot of a dynamic system
`initial`: Produces an initial condition response plot of a state-space model
`lsim`: Simulates the time response of a dynamic system to arbitrary inputs

Frequency-domain analysis

`ltiview`: An LTI viewer for LTI system response analysis
`bode`: Produces a Bode plot of frequency response, magnitude and phase of frequency response
`sigma`: Produces a singular values plot of a dynamic system
`nyquist`: Produces a Nyquist plot of frequency response
`nichols`: Produces a Nichols chart of frequency response
`margin`: Returns gain margin, phase margin, and crossover frequencies
`allmargin`: Returns gain margin, phase margin, delay margin and crossover frequencies
`freqresp`: Returns frequency response over a grid

Classical design

`sisotool`: Interactively design and tune SISO feedback loops (technical root locus and loop shaping)
`rlocus`: Root locus plot of a dynamic system

Pole placement

`place`: MIMO pole placement design
`estim`: Forms a state estimator given estimator gain
`reg`: Forms a regulator given state-feedback and estimator gains

LQR/LQG design

`lqr`: Linear quadratic regulator (LQR) design
`dlqr`: Linear-quadratic (LQ) state-feedback regulator for a discrete-time state-space system
`lqry`: Linear-quadratic (LQ) state-feedback regulator with output weighting
`lqrd`: Discrete linear-quadratic (LQ) regulator for a continuous plant
`kalman`: Kalman estimator
`kalmd`: Discrete Kalman estimator for a continuous plant

State-space models

`rss`: Generates a random continuous test model
`drss`: Generates a random discrete test model
`ss2ss`: State coordinate transformation for state-space models
`ctrb`: Controllability matrix
`obsv`: Observability matrix
`gram`: Control and observability gramians
`minreal`: Minimal realization or pole-zero cancelation
`ssbal`: Balance state-space models using a diagonal similarlity transformation
`balreal`: Gramian-based input/output balancing of state-space realizations
`modred`: Model order reduction

Models with time delays

`totaldelay`: Total combined input/output delay for an LTI model
`delay2z`: Replaces delays of discrete-time TF, SS, or ZPK models by poles at z=0, or replaces delays of FRD models [1]
`pade`: Padé approximation of a model with time delays

Matrix equation solvers

`lyap`: Solves continuous-time Lyapunov equations
`dlyap`: Solves discrete-time Lyapunov equations
`care`: Solves continuous-time algebraic Riccati equations
`dare`: Solves discrete-time algebraic Riccati equations

[1] Note: in more recent versions of MATLAB, `delay2z` has been replaced with `absorbDelay`.