LinearSys

1) Basic Parameter setting
Var: params -- struct
[1] P48
- .tStart initial time t0 (default value 0)
- .tFinal final time tf
- .R0 initial set X0 specified by one of the set representations
- .U input set U specified as an object of class zonotope
- etc.

2) Reachability Settings
Var: options -- struct
[1] P56
only for linearSys
- .linAlg string specifying the reachability algorithm that is used — **** important**
- .timeStep time step size required for all algorithms
- .taylorTerms number of Taylor terms for the computation of the exponential matrix e^(A*delta_t) — **** determine the accuracy**
- .zonotopeOrder upper bound for the zonotope order ρ
- .error upper bound for the error containing over-approximative terms — only for adaptive mode

3) Define systems
[1] P56
- linearSys -- inherit from contDynamics

4) Reachability analysis
- (1) Preprocessing options — autofill the missing elements in options
- (2) Do reachability analysis in different modes
  - standard
  - wrapping free
  - adaptive
  - etc.

Var: uTrans|uTransVec -- input set U and input vector

Ref:
[1] CORA document
[2] PhD thesis from Prof. Althoff
[3] Reachable Set Computation for Uncertain Time-Varying Linear Systems

Standard RA

Important value:
Rfirst.tp -- time point
Rfirst.ti -- time interval

(1) Func: initReach_Euclidean
get the initial reachable set

Func: inputSolution
[2] P35
- compute reachable set due to input
- Asum and Vsum all come from [2] P36 (3.7)
- overall solution [2] P36 — compute the constant solution (Asum) and solution with 0-contained set (Vsum)
- inputTie — compute the input time interval error

Important value:
RV (inputSolV): overall solution
eAtInt: solution due to constant input
inputF: F of input
inputCorr: input correction
Rtrans (inputSolVtrans): solution due to constant input

Func: exponential
[3] P4 — compute e^(A*delta_t) with the error term W — Important value: obj.taylor.power -- value; obj.taylor.error -- error

Func: tie
[2] P34 — compute the correction matrix; to make sure that all the trajectory from the initial set can be detected. — Important value: F

Func: enclose
[2]

Func: reduce
[2]

(2) Func: initOutputEquation
initialize parameter for the output equation

(3) Loop: — using the previous step to compute the next solution, the results are stored into Rout

wrapping-free
- warpping effect [2] P37 — The over-approximation error is propagated through the computations at later time steps
- (1) Func: initReach_Euclidean get the initial reachable set — same
- (2) using algorithm 2 in [2] P37 to compute reachable set