W = emgr(f,g,s,t,w,pr,nf,ut,us,xs,um,xm);

emgr – Empirical Gramian Framework (Version 5.0)

Mandatory Arguments

`f`	System Vector Field	(Handle)	`x = f(x,u,p,t)`	i.e.: **f = @(x,u,p,t) A*x+B*u+F*p;**
`g`	Output Functional	(Handle)	`y = g(x,u,p,t)`	i.e.: **g = @(x,u,p,t) C*x+D*u;**
		1	`y = x`	
`s`	System Dimensions	(Vector)	`s = [M,N,Q]`	(Inputs, States, Outputs)
`t`	Time	(Vector)	`t = [h,T]`	(Step, Stop)
`w`	Gramian Type	(Character)		
		`'c'`	Empirical Controllability Gramian (returns **WC**)	
		`'o'`	Empirical Observability Gramian (returns **WO**)	
		`'x'`	Empirical Cross Gramian (returns **WX**)	
		`'y'`	Empirical Linear Cross Gramian (returns **WY**)	
		`'s'`	Empirical Sensitivity Gramian (returns **{WC,WS}**)	
		`'i'`	Empirical Identifiability Gramian (returns **{WO,WI}**)	
		`'j'`	Empirical Joint Gramian (returns **{WX,WI}**)	

Optional Arguments

`pr`	Parameters	(Vector)	Column Vector of System Parameters (Default: **pr = 0**)
		(Matrix)	Set of Parameter Column Vectors (**'s','i','j'** requires two)
`nf`	Options	(Vector)	Ten Components (Default: **nf = 0**), see Option Flags
`ut`	Input Function	(Handle)	Input function **u = ut(t)**
		1	Delta Impulse Input
		∞	Linear Chirp Function
`us`	Steady-State Input	(Scalar)	Uniform Steady-State Input (Default: **us = 0**)
		(Vector)	Individual Steady-State Input (**J x 1**)
`xs`	Steady State	(Scalar)	Uniform Steady State (**Default: xs = 0**)
		(Vector)	Individual Steady States (**N x 1**)
`um`	Input Scales	(Scalar)	Uniform Maximum Input Scales (Default: **um = 1**)
		(Vector)	Individual Maximum Input Scales (**J x 1**)
		(Matrix)	Custom Input Scales (**J x ***)
`xm`	Steady-State Scales	(Scalar)	Uniform Maximum Steady-State Scales (Default: **xm = 1**)
		(Vector)	Individual Maximum Steady-State Scales (**N x 1**)
		(Matrix)	Custom Steady-State Scales (**N x ***)

Option Flags

`nf(1)`	Trajectory Centering	
	0	None (Default)
	1	Initial State
	2	Final Steady State
	3	Arithmetic Average
	4	Root-Mean-Squared
`nf(2)`	Input Scale Sequence	
	0	Single (Default)
	1	Linear
	2	Geometric
	3	Logarithmic
	4	Sparse
`nf(3)`	State Scale Sequence	
	0	Single (Default)
	1	Linear
	2	Geometric
	3	Logarithmic
	4	Sparse
`nf(4)`	Input Transformation	
	0	Unit (Default)
	1	Inverse
`nf(5)`	State Transformation	
	0	Unit (Default)
	1	Inverse
`nf(6)`	Preconditioning	
	0	None (Default)
	1	Jacobi (Double Run)
	2	Steady-State Scaled
`nf(7)`	Non-Symmetric Cross Gramian **(WX,WY,WJ only)**	
	0	Off (Default)
	1	Non-Sym. Cross-Gramian
`nf(8)`	Extra Input	
	0	No Extra Input (Default)
	1	Paramater Perturbation Only
	2	State Perturbation Only
	3	State and Parameter Perturbation
`nf(9)`	Center Param. Scales **(WS,WI,WJ only)**	
	0	No Centering
	1	Arithmetic Mean Centering
	2	Geometric Mean Centering
`nf(10)`	Schur complement Options **(WI,WJ only)**	
	0	Detailed Schur-Complement (Default)
	1	Approximate Schur-Complement

Custom Solver

Global variable **ODE** to handle with signature:
y = solver(f,g,t,x,u,p); Default: RK – SSP32

Custom Inner Product

Global variable **DOT** to handle with signature:
w = kernel(x,y); Default: identity

Minimal Usage: W = emgr(f,g,s,t,w);

About Info: V = emgr('version');

More info at: http://gramian.de