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

emgr – Empirical Gramian Framework (Version 5.6)

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]	(Time Step, Time Horizon)
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,Wj}**)	

Optional Arguments

pr	Parameters	(Vector)	Column vector of parameters (default: **pr = 0**)
		(Matrix)	Set of parameter columns (**Ws,Wi,Wj** require min and max)
nf	Options Flags	(Vector)	Twelve components (default: **nf = 0**), see option flags
ut	Input Function	(Handle)	Input function **u_t = ut(t)**
			Pseudo-random binary
		1	Delta impulse (default)
		∞	Decaying exponential chirp
us	Steady-State Input	(Scalar)	Uniform steady-state input (default: **us = 0**)
		(Vector)	Individual steady-state input (**M x 1**)
xs	Steady-State	(Scalar)	Uniform steady-state (default: **xs = 0**)
		(Vector)	Individual steady-states (**N x 1**)
um	Input Scales	(Scalar)	Uniform max input scales (default: **um = 1**)
		(Vector)	Individual max input scales (**M x 1**)
		(Matrix)	Custom input scales (**M x ***)
xm	Steady-State Scales	(Scalar)	Uniform max steady-state scales (default: **xm = 1**)
		(Vector)	Individual max steady-state scales (**N x 1**)
		(Matrix)	Custom steady-state scales (**N x ***)
dp	Dot Product	(Handle)	Handle to custom inner product **z = dp(x,y)**
		[]	Default matrix product

Option Flags

nf(1)	Trajectory centering	
	0	None (default)
	1	Initial state
	2	Final steady-state
	3	Arithmetic average
	4	Root-mean-squared
	5	Midrange
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	+ Unit
nf(5)	State transformation	
	0	± Unit (default)
	1	+ Unit
nf(6)	Normalizing	
	0	None (default)
	1	Jacobi
	2	Steady-state
nf(7)	Non-symmetric cross Gramian (**Wx,Wy,Wj** only)	
	0	Regular (default)
	1	Non-symmetric
nf(8)	Extra input (**Wo,Wx,Ws,Wi,Wj** only)	
	0	No (default)
	1	Yes
nf(9)	Center param. Scales (**Ws,Wi,Wj** only)	
	0	No centering (default)
	1	Linear mean centering
	2	Logarithmic mean centering
nf(10)	Gramian variant (**Ws,Wi,Wj** only)	
Ws	0	Input-state average (default)
Ws	1	Input-output average
Wi, Wj	0	Detailed Schur-complement (default)
Wi, Wj	1	Approximate Schur-complement
nf(11)	Partitioned cross Gramian (**Wx,Wj** only)	
	0	Full cross Gramian
	<N	Cross Gramian partition size
nf(12)	Partitioned cross Gramian (**Wx,Wj** only)	
	0	Full cross Gramian
	>0	Partition running index

Custom Solver

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

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

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

More info at: https://gramian.de