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

emgr – EMpirical GRamian Framework (Version 5.99)

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 Discretization	(Vector)	**t = [dt,Tf]**	(Time Step, Time Horizon)
w	Gramian Type	(Char)	Empirical System Gramian Type	
		'c'	Empirical Controllability Gramian (returns $\mathbf{W_c}$)	
		'o'	Empirical Observability Gramian (returns $\mathbf{W_o}$)	
		'x'	Empirical Cross Gramian (returns $\mathbf{W_x}$)	
		'y'	Empirical Linear Cross Gramian (returns $\mathbf{W_y}$)	
		's'	Empirical Sensitivity Gramian (returns $\mathbf{\{W_c,W_s\}}$)	
		'i'	Empirical Identifiability Gramian (returns $\mathbf{\{W_o,W_i\}}$)	
		'j'	Empirical Joint Gramian (returns $\mathbf{\{W_x,W_j\}}$)	

Optional Arguments

pr	Parameters	(Vector)	Column vector of parameters (default: **pr = 0**)
		(Matrix)	Set of parameter columns ($\mathbf{W_s}$, $\mathbf{W_i}$, $\mathbf{W_j}$ require min & max)
nf	Options Flags	(Vector)	Thirteen components (default: **nf = 0**)
ut	Input Function	(Handle)	Input function **u_t = ut(t)** or char (default: **ut = 'i'**)
		'i'	Delta impulse input (default)
		's'	Step input
		'h'	Haversine decaying exponential chirp input
		'a'	Sine cardinal input
		'r'	Pseudo-random binary input
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)	Custom Inner product/kernel **xy = dp(x,y)**, (default: **dp = []**)

Option Flags

nf(1)	Trajectory Centering	
0	None (default)	
1	Steady state	
2	Final state	
3	Arithmetic average	
4	Root-mean-squared	
5	Mid-range	
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	Steady state	
2	Jacobi	
nf(7)	State Gramian Type	
0	Regular (default)	
$\mathbf{W_c,W_s}$	**1**	Output controllability Gramian
$\mathbf{W_o,W_i}$	**1**	Averaged observability Gramian
$\mathbf{W_x,W_y,W_j}$	**1**	Non-symmetric cross Gramian
nf(8)	Extra Input ($\mathbf{W_o}$, $\mathbf{W_x}$, $\mathbf{W_s}$, $\mathbf{W_i}$, $\mathbf{W_j}$ only)	
0	No (default)	
1	Yes	
nf(9)	Center Parameter Scales ($\mathbf{W_s}$, $\mathbf{W_i}$, $\mathbf{W_j}$ only)	
0	None (default)	
1	Linear mean centering	
2	Logarithmic mean centering	
3	Nominal centering	
nf(10)	Parameter Gramian Type ($\mathbf{W_s}$, $\mathbf{W_i}$, $\mathbf{W_j}$ only)	
$\mathbf{W_s}$	**0**	Input-state average (default)
$\mathbf{W_s}$	**1**	Input-output average
$\mathbf{W_i,W_j}$	**0**	Approx. Schur-complement (default)
$\mathbf{W_i,W_j}$	**1**	Coarse Schur-complement
$\mathbf{W_i,W_j}$	**2**	Exact Schur-complement
nf(11)	Partitioned Cross Gramian ($\mathbf{W_x}$, $\mathbf{W_j}$ only)	
0	Full cross Gramian (default)	
<N	Cross Gramian partition size	
nf(12)	Partitioned Cross Gramian ($\mathbf{W_x}$, $\mathbf{W_j}$ only)	
0	Full cross Gramian (default)	
>0	Partition running index	
nf(13)	Trajectory Weighting	
0	None (default)	
1	Linear time-weighting	
2	Quadratic time-weighting	
3	State weighting	
4	Scale weighting	
5	Reciprocal square-root time-weighting	

Custom Solver

Global variable **ODE** is a handle with signature:
y = ODE(f,g,t,x0,u,p) (default: RK – SSP32)

Version info: **V = emgr('version')**
More info at: https://gramian.de

Authored by *Christian Himpe*, licensed under CC-BY