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

## emgr – Empirical Gramian Framework (Version 5.7)

f System Vector Field (Handle) x = f(x,u,p,t) i.e.: f = g(x,u,p,t) x xxxx8xupFxp g Output Functional 1	<b>Mandatory Arguments</b>		Option Flags	
s System Dimensions (Vector) s = [M,N,Q] (Inputs, States, Outputs)  t Time Discretization (Vector) t = [dt,1f1] (Time Step, Time Horizon)  M Gramian Type (Char) Empirical System Gramian Type (Char) Empirical Controllability Gramian (returns W <sub>v</sub> ) (**) Empirical Controllability Gramian (returns W <sub>v</sub> ) (**) Empirical Corsos Gramian (returns W <sub>v</sub> ) (**) Empirical Corsos Gramian (returns W <sub>v</sub> ) (**) Empirical Inlear Cross Gramian (returns W <sub>v</sub> ) (**) Empirical Inlear Cross Gramian (returns W <sub>v</sub> ) (**) Empirical Inlear Cross Gramian (returns W <sub>v</sub> ) (**) Empirical Inlear Cross Gramian (returns W <sub>v</sub> , W <sub>v</sub> ) (**) Empirical Inlear Cross Gramian (returns W <sub>v</sub> , W <sub>v</sub> ) (**) Empirical Inlear Cross Gramian (returns W <sub>v</sub> , W <sub>v</sub> ) (**) Empirical Inlear Cross Gramian (returns W <sub>v</sub> , W <sub>v</sub> ) (**) Empirical Inlear Cross Gramian (returns W <sub>v</sub> , W <sub>v</sub> ) (**) Empirical Inlear Cross Gramian (returns W <sub>v</sub> , W <sub>v</sub> ) (**) Empirical Inlear Cross Gramian (returns W <sub>v</sub> , W <sub>v</sub> ) (**) Empirical Inlear Cross Gramian (returns W <sub>v</sub> , W <sub>v</sub> ) (**) Empirical Inlear Cross Gramian (returns W <sub>v</sub> , W <sub>v</sub> ) (**) Empirical Inlear Cross Gramian (returns W <sub>v</sub> , W <sub>v</sub> ) (**) Empirical Inlear Cross Gramian (returns W <sub>v</sub> , W <sub>v</sub> ) (**) Empirical Inlear Cross Gramian (returns W <sub>v</sub> , W <sub>v</sub> ) (**) Empirical Inlear Cross Gramian (returns W <sub>v</sub> , W <sub>v</sub> ) (**) Empirical Inlear Cross Gramian (returns W <sub>v</sub> , W <sub>v</sub> ) (**) Empirical Inlear Cross Gramian (returns W <sub>v</sub> , W <sub>v</sub> ) (**) Empirical Inlear Cross Gramian (returns W <sub>v</sub> , W <sub>v</sub> ) (**) Empirical Inlear Cross Gramian (returns W <sub>v</sub> , W <sub>v</sub> ) (**) Empirical Inlear Cross Gramian (returns W <sub>v</sub> , W <sub>v</sub> ) (**) Empirical Inlear Cross Gramian (returns W <sub>v</sub> , W <sub>v</sub> ) (**) Empirical Inlear Cross Gramian (returns W <sub>v</sub> , W <sub>v</sub> ) (**) Empirical Inlear Cross Gramian (returns W <sub>v</sub> , W <sub>v</sub> ) (**) Empirical Inlear Cross Gramian (returns W <sub>v</sub> , W <sub>v</sub> , W <sub>v</sub> ) (**) Empirical Inlear Cross Gramian (returns W <sub>v</sub> , W <sub>v</sub> ) (**) Detailed Schur-complement (default) (**) Empirical Inlear Cross Gramian (returns W <sub>v</sub> , W <sub>v</sub> , W <sub>v</sub> , W <sub>v</sub> , W <sub>v</sub> ) (**) Empirical Inlear Cross Gramian (returns	<b>f</b> System Vector Field	(Handle) $x = f(x,u,p,t)$ i.e.: $f = Q(x,u,p,t) A*x+B*u+F*p$	<pre>nf(1) Trajectory centering</pre>	$nf(8)$ Extra input $(\mathbf{W_o}, \mathbf{W_x}, \mathbf{W_s}, \mathbf{W_l}, \mathbf{W_l})$ only)
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 Controllability Gramian (returns W <sub>c</sub> )  'c' Empirical Controllability Gramian (returns W <sub>c</sub> )  'r' Empirical Cross Gramian (returns W <sub>c</sub> )  'r' Empirical Cross Gramian (returns W <sub>c</sub> )  'r' Empirical Ensentitivity Gramian (returns W <sub>c</sub> )  'r' Empirical Identifiability Gramian (returns W <sub>c</sub> )  'r' Empirical Identifiability Gramian (returns W <sub>c</sub> )  'r' Empirical Identifiability Gramian (returns W <sub>c</sub> , W)  'r' Empirical Identifiability Gramian (returns W <sub>c</sub> , W)  'r' Empirical Identifiability Gramian (returns W <sub>c</sub> , W)  'r' Empirical Identifiability Gramian (returns W <sub>c</sub> , W)  'r' Empirical Identifiability Gramian (returns W <sub>c</sub> , W)  'r' Empirical Identifiability Gramian (returns W <sub>c</sub> , W)  'r' Empirical Identifiability Gramian (returns W <sub>c</sub> , W)  'r' Empirical Identifiability Gramian (returns W <sub>c</sub> , W)  'r' Empirical Identifiability Gramian (returns W <sub>c</sub> , W)  'r' Empirical Identifiability Gramian (returns W <sub>c</sub> , W)  'r' Empirical Identifiability Gramian (returns W <sub>c</sub> , W)  'r' Empirical Identifiability Gramian (returns W <sub>c</sub> , W)  'r' Empirical Identifiability Gramian (returns W <sub>c</sub> , W)  'r' Empirical Identifiability Gramian (returns W <sub>c</sub> , W)  'r' Empirical Identifiability Gramian (returns W <sub>c</sub> , W)  'r' Empirical Identifiability Gramian (returns W <sub>c</sub> , W)  'r' Empirical Identifiability Gramian (returns W <sub>c</sub> , W)  'r' Empirical Identifiability Gramian (returns W <sub>c</sub> , W)  'r' Empirical Identifiability Gramian (returns W <sub>c</sub> , W)  'r' Empirical Identifiability Gramian (returns W <sub>c</sub> , W)  'r' Empirical Identifiability Gramian (returns W <sub>c</sub> , W)  'r' Empirical Identifiability Gramian (returns W <sub>c</sub> , W)  'r' Empirical Identifiability Gramian (returns W <sub>c</sub> , W)  'r' Empirical Identifiability Gramian (returns W <sub>c</sub> , W)  'r' Empirical Identifiability Gramian (returns W <sub>c</sub> , W,	<b>g</b> Output Functional	(Handle) $y = g(x,u,p,t)$ i.e.: $g = Q(x,u,p,t)$ C*x+D*u	0 None (default)	<b>0</b> No (default)
t Time Discretization  w Gramian Type  (Char) Empirical System Gramian Type  (Char) Empirical Controllability Gramian (returns W <sub>c</sub> )  i.e. Empirical Controllability Gramian (returns W <sub>c</sub> )  i.e. Empirical Cross Gramian (returns W <sub>c</sub> )  i.g. Empirical Cross Gramian (returns W <sub>c</sub> )  i.g. Empirical Gramian (returns W <sub>c</sub> , W <sub>c</sub> )  i.g. Empirical Gramian (returns W <sub>c</sub> , W <sub>c</sub> )  i.g. Empirical Gramian (returns W <sub>c</sub> , W <sub>c</sub> )  i.g. Empirical Gramian (returns W <sub>c</sub> , W <sub>c</sub> )  i.g. Empirical Gramian (returns W <sub>c</sub> , W <sub>c</sub> )  i.g. Empirical Gramian (returns W <sub>c</sub> , W <sub>c</sub> )  i.g. Empirical Gramian (returns W <sub>c</sub> , W <sub>c</sub> )  i.g. Empirical Gramian (returns W <sub>c</sub> , W <sub>c</sub> )  i.g. Empirical Gramian (returns W <sub>c</sub> , W <sub>c</sub> )  i.g. Empirical Constantian (returns W <sub>c</sub> , W <sub>c</sub> )  i.g. Empirical Constantian (returns W <sub>c</sub> , W <sub>c</sub> )  i.g. Empirical Constantian (returns W <sub>c</sub> , W <sub>c</sub> )  i.g. Empirical Constantian (returns W <sub>c</sub> , W <sub>c</sub> )  i.g. Empirical Constantian (returns W <sub>c</sub> , W <sub>c</sub> )  i.g. Empirical Constantian (returns W <sub>c</sub> , W <sub>c</sub> )  i.g. Empirical Constantian (returns W <sub>c</sub> , W <sub>c</sub> )  i.g. Empirical Constantian (returns W <sub>c</sub> , W <sub>c</sub> )  i.g. Empirical Constantian (returns W <sub>c</sub> , W <sub>c</sub> )  i.g. Empirical Constantian (returns W <sub>c</sub> , W <sub>c</sub> )  i.g. Empirical Constantian (returns W <sub>c</sub> , W <sub>c</sub> )  i.g. Empirical Constantian (returns W <sub>c</sub> , W <sub>c</sub> )  i.g. Empirical Constantian (returns W <sub>c</sub> , W <sub>c</sub> )  i.g. Empirical Constantian (returns W <sub>c</sub> , W <sub>c</sub> )  i.g. Empirical Constantian (returns W <sub>c</sub> , W <sub>c</sub> )  i.g. Empirical Constantian (returns W <sub>c</sub> , W <sub>c</sub> )  i.g. Empirical Constantian (returns W <sub>c</sub> , W <sub>c</sub> )  i.g. Empirical Constantian (returns W <sub>c</sub> , W <sub>c</sub> )  i.g. Empirical Constantian (returns W <sub>c</sub> , W <sub>c</sub> , W <sub>c</sub> i.g. Empirical Constantian (returns W <sub>c</sub> , W <sub>c</sub> )  i.g. Empirical Constantian (returns W <sub>c</sub> , W <sub>c</sub> )  i.g. Empirical Constantian (returns W <sub>c</sub> , W <sub>c</sub> )  i.g. Empirical Constantian (returns W <sub>c</sub> , W <sub>c</sub> )  i.g. Empirical Constantian (returns		1 y = x	1 Initial state	<b>1</b> Yes
Charlian Type   Char   Empirical System Gramian Type   'c'   Empirical Controllability Gramian (returns W <sub>c</sub> )   5 Midrange   5 Midrange   6 Geometric mean   nf(10) Parameter Gramian Type (W <sub>s</sub> , W, W <sub>s</sub> ) only)	<b>s</b> System Dimensions	(Vector) <b>s</b> = [M,N,Q] (Inputs, States, Outputs)	2 Final steady-state	$nf(9)$ Center param. Scales $(W_s, W_l, W_J \text{ only})$
*** C Empirical Controllability Gramian (returns W_o)** 6 Geometric mean of f(10)** Parameter Gramian Type (W_o, W_o, W_only)** of Empirical Cross Gramian (returns W_o)** of Empirical Cross Gramian (returns W_o)** of Empirical Lener Cross Gramian (returns W_o)** of Empirical Lener Cross Gramian (returns W_o)** of Empirical Lener Cross Gramian (returns W_o, W_o, W_o, W_o, W_o, W_o, W_o, W_o,	t Time Discretization	(Vector) <b>t</b> = [dt,Tf] (Time Step, Time Horizon)	<b>3</b> Arithmetic average	0 No centering (default)
Parameter   Para	w Gramian Type	(Char) Empirical System Gramian Type	4 Root-mean-squared	1 Linear mean centering
The properties of the proper		$^{f 'c'}$ Empirical Controllability Gramian (returns ${f W}_c$ )	<b>5</b> Midrange	2 Logarithmic mean centering
vy   Empirical Linear Cross Gramian (returns W <sub>v</sub> )   0 Single (default)   1 Linear   W <sub>v</sub> , W <sub>v</sub>   0 Detailed Schur-complement (default)   1 Linear   W <sub>v</sub> , W <sub>v</sub>   1 Approximate Schur-complement (default)   2 Geometric   3 Logarithmic   1 Linear   W <sub>v</sub> , W <sub>v</sub>   1 Approximate Schur-complement (default)   2 Geometric   3 Logarithmic   1 F(11)   Partitioned cross Gramian (W <sub>v</sub> , W <sub>v</sub> only)   3 State scale sequence   0 Single (default)   1 Linear   0 Full cross Gramian (W <sub>v</sub> , W <sub>v</sub> only)   0 Full cross Gramian (W <sub>v</sub> , W <sub>v</sub> only)   0 Full cross Gramian (W <sub>v</sub> , W <sub>v</sub> only)   0 Full cross Gramian (W <sub>v</sub> , W <sub>v</sub> only)   0 Full cross Gramian (W <sub>v</sub> , W <sub>v</sub> only)   0 Full cross Gramian (W <sub>v</sub> , W <sub>v</sub> only)   0 Full cross Gramian (W <sub>v</sub> , W <sub>v</sub> only)   0 Full cross Gramian (W <sub>v</sub> , W <sub>v</sub> only)   0 Full cross Gramian (W <sub>v</sub> , W <sub>v</sub> only)   0 Full cross Gramian (W <sub>v</sub> , W <sub>v</sub> only)   0 Full cross Gramian (W <sub>v</sub> , W <sub>v</sub> only)   0 Full cross Gramian (W <sub>v</sub> , W <sub>v</sub> only)   0 Full cross Gramian (W <sub>v</sub> , W <sub>v</sub> only)   0 Full cross Gramian (W <sub>v</sub> , W <sub>v</sub> only)   0 Full cross Gramian (W <sub>v</sub> , W <sub>v</sub> only)   0 Full cross Gramian (W <sub>v</sub> , W <sub>v</sub> only)   0 Full cross Gramian (W <sub>v</sub> , W <sub>v</sub> only)   0 Full cross Gramian (W <sub>v</sub> , W <sub>v</sub> only)   0 Full cross Gramian (W <sub>v</sub> , W <sub>v</sub> only)   0 Full cross Gramian (W <sub>v</sub> , W <sub>v</sub> only)   0 Full cross Gramian (W <sub>v</sub> , W <sub>v</sub> only)   0 Full cross Gramian (W <sub>v</sub> , W <sub>v</sub> only)   0 Full cross Gramian (W <sub>v</sub> , W <sub>v</sub> only)   0 Full cross Gramian (W <sub>v</sub> , W <sub>v</sub> only)   0 Full cross Gramian (W <sub>v</sub> , W <sub>v</sub> only)   0 Full cross Gramian (W <sub>v</sub> , W <sub>v</sub> only)   0 Full cross Gramian (W <sub>v</sub> , W <sub>v</sub> only)   0 Full cross Gramian (W <sub>v</sub> , W <sub>v</sub> only)   0 Full cross Gramian (W <sub>v</sub> , W <sub>v</sub> only)   0 Full cross Gramian (W <sub>v</sub> , W <sub>v</sub> , W <sub>v</sub> only)   0 Full cross Gramian (W <sub>v</sub> , W <sub>v</sub> , W <sub>v</sub> , W <sub>v</sub> only)   0 Full cross Gramian (W <sub>v</sub> , W <sub>v</sub> ,		$oldsymbol{v_0}oldsymbol{v}$ Empirical Observability Gramian (returns $oldsymbol{W_0}$ )	<b>6</b> Geometric mean	<b>nf(10)</b> Parameter Gramian Type ( $\mathbf{W_s}$ , $\mathbf{W_l}$ , $\mathbf{W_J}$ only)
*** Empirical Sensitivity Gramian (returns $W_c$ , $W_s$ )  *** it Empirical Jdentifiability Gramian (returns $W_c$ , $W_s$ )  *** period Joint Gramian (returns $W_c$ , $W_s$ )  **Optional Arguments  **pr Parameters  **(Vector) Column vector of parameters (default: $pr = 0$ )  **(Matrix) Set of parameter columns ( $W_s$ , $W_s$ , $W_s$ require min / max)  **nf Options Flags  **(Vector) Twelve components (default: $nf = 0$ )  **ut Input Function  **(Handle) Input function $u_s t = ut(t)$ or char (default: $ut = t(t)$ )  **s' Step input  **us Steady-State Input  **(Vector) Individual steady-state input (default: $us = 0$ )  **(Vector) Individual steady-state ( $us = 0$ )  **(Vector) Individual s		$\mathbf{v}_{\mathbf{X}}$ Empirical Cross Gramian (returns $\mathbf{W}_{\mathbf{X}}$ )	nf(2) Input scale sequence	W <sub>s</sub> <b>0</b> Input-state average (default)
Properties   Pro		$\mathbf{v_y}$ Empirical Linear Cross Gramian (returns $\mathbf{W_y}$ )	<b>0</b> Single (default)	W <sub>s</sub> 1 Input-output average
Optional Arguments  pr Parameters (Vector) Column vector of parameters (default: pr = 0)			1 Linear	<b>W</b> <sub>I'</sub> <b>W</b> <sub>J</sub> <b>0</b> Detailed Schur-complement (default)
Optional Arguments  pr Parameters (Vector) Column vector of parameters (default: pr = 0) (Matrix) Set of parameter columns (W <sub>s</sub> , W <sub>p</sub> , W <sub>p</sub> , require min / max)  nf Options Flags (Vector) Twelve components (default: nf = 0) 1 Linear 0 Full cross Gramian partition size 0 Full cross Gramian (W <sub>x</sub> , W <sub>p</sub> only)  ut Input Function (Handle) Input function u_t = ut(t) or char (default: ut = 'i') 2 Geometric 3 Logarithmic  'i' Delta impulse (default) 3 Logarithmic  's' Step input 4 Sparse Custom Solver  'c' Decaying exponential chirp nf(4) Input transformation 0 ± Unit (default) 1 + Unit 1		- '	2 Geometric	<b>W</b> <sub>I'</sub> <b>W</b> <sub>J</sub> <b>1</b> Approximate Schur-complement
pr Parameters (Vector) Column vector of parameters (default: pr = 0) (Matrix) Set of parameter columns (Wg, Wr, W, require min / max)  nf Options Flags (Vector) Twelve components (default: nf = 0)  ut Input Function (Handle) Input function u_t = ut(t) or char (default: ut = 'i')  belta impulse (default)  'i' Delta impulse (default)  's' Step input  'c' Decaying exponential chirp  'r' Pseudo-random binary  us Steady-State Input (Scalar) Uniform steady-state input (M x 1)  xs Steady-State (Scalar) Uniform steady-state (default: us = 0) (Vector) Individual steady-state (default: us = 1) (Vector) Individual max input scales (M x x)  xm Steady-State Scales (Scalar) Uniform max steady-state scales (M x x)  xm Steady-State Scales (Scalar) Uniform max steady-state scales (M x x)  xm Steady-State Scales (Scalar) Uniform max steady-state scales (M x x)  (Matrix) Custom input scales (M x x)  xm Steady-State Scales (Scalar) Uniform max steady-state scales (M x x)  (Matrix) Custom steady-state scales (M x x)  nf(6) Normalizing  None (default)  nf(7) State scale sequence 0 Single (default)  1 Linear 2 Geometric 3 Logarithmic 4 Sparse  Custom Solver Global variable 0DE to a handle with signature: y = 0DE(f,g,t,x0,u,p) Default: RK - SSP32  The input transformation 0 ± Unit (default) 1 + Unit 1 + Unit 1 + Unit 2 + Unit (default) 3 Logarithmic 4 Sparse Custom Solver Global variable 0DE to a handle with signature: y = 0DE(f,g,t,x0,u,p) Default: RK - SSP32  The input transformation 0 ± Unit (default) 1 + Unit 1 + Unit 2 + Unit (default) 3 Logarithmic 4 Sparse Global variable 0DE to a handle with signature: y = 0DE(f,g,t,x0,u,p) Default: RK - SSP32  The input transformation 0 ± Unit (default) 1 + Unit 1 + Unit 2 + Unit (default) 3 + Unit (default) 4 + Unit 4		' <b>j</b> ' Empirical Joint Gramian (returns <b>W</b> <sub>x</sub> , <b>W</b> <sub>J</sub> )	3 Logarithmic	<b>nf(11)</b> Partitioned cross Gramian ( $\mathbf{W}_{x}$ , $\mathbf{W}_{J}$ only)
Matrix   Set of parameter columns (W <sub>s</sub> , W <sub>r</sub> , W <sub>s</sub> require min / max   0 Single (default)   nf (12) Partitioned cross Gramian (W <sub>s</sub> , W <sub>s</sub> only)	<b>Optional Arguments</b>		4 Sparse	• Full cross Gramian
nf Options Flags       (Vector)       Twelve components (default: nf = 0)       1 Linear       0 Full cross Gramian         ut Input Function       (Handle)       Input function u_t = ut(t) or char (default: ut = 'i')       2 Geometric       >0 Partition running index         't'       Delta impulse (default)       3 Logarithmic       >0 Partition running index         's'       Step input       4 Sparse       Custom Solver         'r'       Decaying exponential chirp       0 ± Unit (default)       global variable 0DE to a handle with signature:         v Decaying exponential chirp       0 ± Unit (default)       y = 0DE(f,g,t,x0,u,p) Default: RK - SSP32         us Steady-State Input       (Scalar)       Uniform steady-state input (M x 1)       nf(5) State transformation       Minimal Usage: W = emgr(f,g,s,t,w)         xs Steady-State       (Scalar)       Uniform steady-state (default: xs = 0)       0 ± Unit (default)       Minimal Usage: W = emgr(f,g,s,t,w)         um Input Scales       (Scalar)       Uniform steady-state (default: xs = 1)       nf(6) Normalizing       Move info at: https://gramian.de         xm Steady-State Scales       (Matrix)       Custom input scales (M x *)       0 None (default)       More info at: https://gramian.de         xm Steady-State Scales       (N x *)       2 Steady-state       2 Steady-state         xm Steady-State Scales	<b>pr</b> Parameters	·	<pre>nf(3) State scale sequence</pre>	·
ut Input Function(Handle) Input function u_t = ut(t) or char (default: ut = 't')2 Geometric>0 Partition running index'i' Delta impulse (default)3 Logarithmic's' Step input4 SparseCustom Solver'c' Decaying exponential chirpnf(4) Input transformationGlobal variable ODE to a handle with signature:'r' Pseudo-random binary0 ± Unit (default)y = ODE(f,g,t,x0,u,p) Default: RK - SSP32us Steady-State Input(Scalar) Uniform steady-state input (M x 1)nf(5) State transformationMinimal Usage: W = emgr(f,g,s,t,w)xs Steady-State(Scalar) Uniform steady-state (default: xs = 0)0 ± Unit (default)(Vector) Individual steady-states (N x 1)1 + UnitAbout Info: V = emgr('version')um Input Scales(Scalar) Uniform max input scales (M x 1)nf(6) NormalizingMore info at: https://gramian.dexm Steady-State Scales(Scalar) Uniform max steady-state scales (default: xm = 1)2 Steady-state(Vector) Individual max steady-state scales (M x 1)1 Jacobi(Watrix) Custom input scales (M x *)nf(7) State Gramian Type (Wo, Wx, Wy, Wy, Wy, Wy, Wy, Wy, Wy, Wy, Wy, Wy		(Matrix) Set of parameter columns ( $\mathbf{W_s}$ , $\mathbf{W_l}$ , $\mathbf{W_J}$ require min / max)	0 Single (default)	<b>nf(12)</b> Partitioned cross Gramian ( $\mathbf{W}_{x}$ , $\mathbf{W}_{J}$ only)
'i' Delta impulse (default) 's' Step input  's' Step input  'c' Decaying exponential chirp 'r' Pseudo-random binary  us Steady-State Input (Vector) Individual steady-state input (M x 1)  um Input Scales (Scalar) Uniform max input scales (M x 1)  um Input Scales (Scalar) Uniform max input scales (M x 1)  um Steady-State Scales (Scalar) Uniform max input scales (M x 1)  um Input Scales (Scalar	<b>nf</b> Options Flags	(Vector) Twelve components (default: nf = 0)	<b>1</b> Linear	<b>0</b> Full cross Gramian
's' Step input 'c' Decaying exponential chirp 'r' Pseudo-random binary  us Steady-State Input (Vector) Individual steady-state input (default: us = 0) (Vector) Individual steady-state (default: us = 0) (Vector) Individual steady-state (M x 1)  um Input Scales (Scalar) Uniform max input scales (M x 1)  um Input Scales (Scalar) Uniform max input scales (M x 1)  um Steady-State Scales (Scalar) Uniform max input scales (M x 1)  um Steady-State Scales (Matrix) Custom input scales (M x 1)  um Steady-State Scales (Vector) Individual max steady-state scales (default: xm = 1) (Vector) Individual max steady-state scales (M x 1)  um Steady-State Scales (Vector) Individual max steady-state scales (default: xm = 1) (Vector) Individual max steady-state scales (M x 1)  um Steady-State Scales (Vector) Individual max steady-state scales (M x 1) (Vector) Individual max steady-state scales (M x 1) (Vector) Individual max steady-state scales (N x 1) (Vector) I	<b>ut</b> Input Function	(Handle) Input function $\mathbf{u_t} = \mathbf{ut(t)}$ or char (default: $\mathbf{ut} = 'i'$ )	2 Geometric	>0 Partition running index
'c' Decaying exponential chirp 'r' Pseudo-random binary  us Steady-State Input (Scalar) Uniform steady-state input (default: us = 0) (Vector) Individual steady-state (default: xs = 0) (Vector) Individual steady-state (default: um = 1) (Vector) Individual max input scales (M x 1)  um Input Scales (Scalar) Uniform max input scales (M x 1)  xm Steady-State Scales (Scalar) Uniform max steady-state scales (M x 1)  xm Steady-State Scales (Scalar) Uniform max steady-state scales (M x 1) (Matrix) Custom input scales (N x 1)  dp Dot Product (Handle) Handle to custom inner product xy = dp(x,y)  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 2 Steady-State Nf(7) State Gramian Type (W <sub>o</sub> , W <sub>x</sub> , W <sub>y</sub> , W <sub></sub>		'i' Delta impulse (default)	<b>3</b> Logarithmic	
'r' Pseudo-random binary  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-state (default: xs = 0) (Vector) Individual steady-state (N x 1)  um Input Scales (Scalar) Uniform max input scales (M x 1) (Vector) Individual max input scales (M x 1)  (Matrix) Custom input scales (M x 1)  xm Steady-State Scales (Scalar) Uniform max steady-state scales (N x 1)  (Vector) Individual max steady-state sca		's' Step input	4 Sparse	Custom Solver
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)  xm Steady-State Scales (Scalar) Uniform max steady-state scales (default: xm = 1) (Vector) Individual max steady-state scales (N x 1)  xm Steady-State Scales (Scalar) Uniform max steady-state scales (default: xm = 1) (Vector) Individual max steady-state scales (N x 1) (Vector) Individual max stea		'c' Decaying exponential chirp	<b>nf(4)</b> Input transformation	Global variable <b>ODE</b> to a handle with signature:
(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 steady-state scales (N x 1) (Vector) Individual max steady-state scales (N x 1) (Matrix) Custom steady-state scales (N x 1) (More info at: https://gramian.de  2 Steady-state  2 Steady-state  3 Regular (default)  4 Non-symmetric Cross Gramian		'r' Pseudo-random binary	0 ± Unit (default)	y = ODE(f,g,t,x0,u,p) Default: RK - SSP32
xs Steady-State (Scalar) Uniform steady-state (default: xs = 0)	us Steady-State Input	(Scalar) Uniform steady-state input (default: us = 0)	1 + Unit	
(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) (Vector) Individual max steady-state scales (N x 1) (Matrix) Custom steady-state scales (N x *)  dp Dot Product  (Vector) Individual max steady-state scales (N x *)  (Matrix) Custom steady-state scales (N x *)		(Vector) Individual steady-state input (M x 1)	<b>nf(5)</b> State transformation	<pre>Minimal Usage: W = emgr(f,g,s,t,w)</pre>
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 (M x 1) (Vector) Individual max steady-state scales (N x 1) (Matrix) Custom steady-state scales (N x *)  dp Dot Product  (Scalar) Uniform max input scales (M x 1) (Matrix) Custom input scales (M x *)  1	xs Steady-State	(Scalar) Uniform steady-state (default: xs = 0)	<pre>0 ± Unit (default)</pre>	
(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  (Vector) Individual max input scales (M x 1) (Scalar) Uniform max steady-state scales (default: xm = 1) (Vector) Individual max steady-state scales (N x 1) (Natrix) Custom steady-state scales (N x *)  (Matrix) Custom steady-state scales (N x *)  (Handle) Handle to custom inner product xy = dp(x,y)  (Vector) Individual max input scales (M x 1)  1		(Vector) Individual steady-states (N x 1)	1 + Unit	About Info: V = emgr('version')
(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  (Matrix) Custom input scales (M x *)  1 Jacobi  2 Steady-state  nf(7) State Gramian Type (W <sub>o</sub> , W <sub>x</sub> , W <sub>y</sub> , W <sub>y</sub> , W <sub>y</sub> , only)  0 Regular (default)  W <sub>x</sub> , W <sub>y</sub> , W <sub>y</sub> , V <sub>y</sub>	um Input Scales	(Scalar) Uniform max input scales (default: um = 1)	nf(6) Normalizing	
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 *) (Matrix) Custom steady-state scales (N x *) (Matrix) Custom inner product xy = dp(x,y) (Matrix) Non-symmetric Cross Gramian		(Vector) Individual max input scales (M x 1)	<b>0</b> None (default)	More info at: https://gramian.de
(Vector) Individual max steady-state scales (N x 1)  (Matrix) Custom steady-state scales (N x *)  dp Dot Product  (Vector) Individual max steady-state scales (N x 1)  (Matrix) Custom steady-state scales (N x *)  0 Regular (default)  W <sub>x</sub> , W <sub>y</sub> , W <sub>y</sub> , W <sub>y</sub> , V		(Matrix) Custom input scales (M x *)	<b>1</b> Jacobi	
(Matrix) Custom steady-state scales (N x *)  dp Dot Product (Handle) Handle to custom inner product $xy = dp(x,y)$ $v_{x'}v_{y'}v_{y'}$ 1 Non-symmetric Cross Gramian	xm Steady-State Scales	(Scalar) Uniform max steady-state scales (default: xm = 1)	•	
<b>dp</b> Dot Product (Handle) Handle to custom inner product $xy = dp(x,y)$ $W_{x'}W_{y'}W_{y}$ 1 Non-symmetric Cross Gramian			$nf(7)$ State Gramian Type ( $W_o$ , V	<b>W<sub>x</sub>, W<sub>y</sub>, W<sub>I</sub>, W<sub>J</sub></b> only)
			_	
[] Default matrix product W <sub>o</sub> , W <sub>i</sub> 1 Averaged Observability Gramian licensed under CC-BY	<b>dp</b> Dot Product		·	
		[] Default matrix product	<b>w<sub>o</sub>,w<sub>i</sub> 1</b> Averaged Observability	/ Gramian licensed under CC-BY