MAX PLANCK INSTITUTE FOR DYNAMICS OF COMPLEX TECHNICAL SYSTEMS MAGDEBURG

CSC COMPUTATIONAL METHODS IN SYSTEMS AND CONTROL THEORY

Math Energy

Federal Ministry for Economic Affairs and Energy

emgr - EMpirical GRamian Framework (Version 5.7)

Christian Himpe (0000-0003-2194-6754)[1]

[1]Max Planck Institute for Dynamics of Complex Technical Systems Magdeburg, Computational Methods in Systems and Control Theory

Input-Output Systems:

Dynamical System → $\dot{x}(t) = f(x(t), u(t), \theta)$ ← Vector Field, State, Input

Output → $y(t) = g(x(t), u(t), \theta)$ ← Output Functional, Parameter

Applicability:

- ► Linear Time-Invariant Systems
- ► Time-Varying Systems
- ► Parameteric Systems
- ► Control-Affine Systems
- ► General Nonlinear Systems
- ► Spatially discretized PDEs
- x Descriptor Systems
- x Switched Systems

Applications:

- **Model Reduction**
- Parameter Identification
- Combined Reduction
- Decentralized Control
- Optimal Placement
- Sensitivity Analysis
- Nonlinearity Quantification
- Uncertainty Quantification

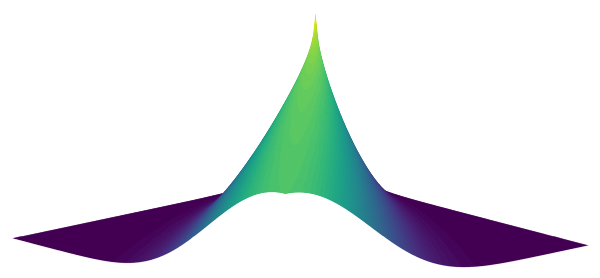
Mathematical System Theory:

- Stabilizability
- Accessibility
- Reachability / Controllability
- Detectability
- Observability / Constructibility
- Minimality
- Identifiability / Sensitivity
- Loadability (Covariance)

The Seven Empirical System Gramians:

- Empirical Controllability Gramian
- Empirical Observability Gramian
- Empirical Linear Cross Gramian
- Empirical Cross Gramian
- Empirical Sensitivity Gramian (Parameter Controllability)
- Empirical Identifiability Gramian (Parameter Observability)
- Empirical Joint Gramian (Parameter Observability)

Linear Advection Cross Gramian



Why Empirical Gramians:

- Data-Driven Computation
- Time-Domain Focus
- Time-Limited Gramians
- Unstable Systems
- Operating-Region-Specific

How to Compute:

1. Design Operating Region
2. Perturb Input/State/Parameter
3. Compute Trajectories
4. Assemble Gramians
5. Average over Gramians

On `https://gramian.de`

- Download
- Demos
- Documentation

Statistics:

- 511 Lines-of-Code
- 1st Release: 2012 (MoRePaS2)
- Oberwolfach References: 2013

Interfaces:

- Solver, included default:
  - Low-Storage
  - Enhanced Stability
  - Strong Stability Preserving
  - Explicit
  - 2nd-Order Runge-Kutta
- Kernel:
  - Custom Inner Product
  - Nonlinear Kernels
  - Trace-Only
  - Diagonal-Only
  - Sub-System-Only
- Low-Rank:
  - Cross Gramian
  - Joint Gramian

Roadmap:

- `morgen` backend (2019)
- `pyMOR` backend (2020)
- **expert system** (2021)

QR-Code:

Features:

- Configurable
- Vectorized
- Parallelizable (`parfor` hints)
- Open-Source
- Solver Interfaces
- Kernel Interface
- Low-Rank Interface
- Combined State and Parameter Gramians
- Non-symmetric Cross Gramians
- Unique Cross Gramian Focus
- Only Basic Linear Algebra
- Functional Design

Code Example:

```
emgr(@(x,u,p,t) A*x + B*u, ...
     @(x,u,p,t) C*x, ...
     [M,N,Q], [dt,Tf], 'x')
```

Compatibility:

- GNU Octave $\geq$ 4.2
- Mathworks MATLAB $\geq$ 2016b
- Python NumPy ≥ 2.7, ≥ 3.7

README

- C. Himpe. `emgr` - **EMpirical GRamian Framework** (Version 5.7), 2019. `http://gramian.de` , `doi:10.5281/zenodo.2577980`
- S. Grundel, C. Himpe, J. Saak. **On Empirical System Gramians**. Proceedings of Applied Mathematics and Mechanics 19: Accepted, 2019. `doi:10.1002/PAMM.201900006`
- P. Benner, C. Himpe. **Cross-Gramian-Based Dominant Subspaces**. arXiv, math.OC: 1809.08066, 2019.
- C. Himpe. **emgr - The Empirical Gramian Framework**. Algorithms 11(7): 91, 2018. `doi:10.3390/a11070091`
- C. Himpe, M. Ohlberger. **A Unified Software Framework for Empirical Gramians**. Journal of Mathematics, 2013: 365909, 2013. `doi:10.1155/2013/365909`

Acknowledgement: Supported by the German Federal Ministry for Economic Affairs and Energy, in the joint project: "**MathEnergy** – Mathematical Key Technologies for Evolving Energy Grids", sub-project: Model Order Reduction (Grant number: 0324019**B**).